
# Tensors in Image Processing and Computer Vision

Santiago Aja-Fernández
Rodrigo de Luis García
Dacheng Tao
Xuelong Li
(Eds.)

Springer

Advances in Pattern Recognition

For futher volumes:
http://www.springer.com/series/4205

Santiago Aja-Fernández · Rodrigo de Luis García ·
Dacheng Tao · Xuelong Li
Editors

# Tensors in Image Processing and Computer Vision

Springer

*Editors*
Asst. Prof. Santiago Aja-Fernández
Universidad de Valladolid
Escuela Técnica Superior de
Ingenieros de
Telecomunicación
Campus Miguel Delibes
47011 Valladolid
Spain
sanaja@tel.uva.es

Asst. Prof. Rodrigo de Luis García
Universidad de Valladolid
Escuela Técnica Superior de
Ingenieros de
Telecomunicación
Campus Miguel Delibes
47011 Valladolid
Spain
rodlui@tel.uva.es

Dr. Dacheng Tao
Blk N4, 02b-67
School of Computer Engineering
Nanyang Technological University
Singapore 639798
dctao@ntu.edu.sg

Dr. Xuelong Li
University of London
Birkbeck College
School of Computer Science &
Information Systems
Malet Street
London WC1E 7HX
United Kingdom
xuelong_li@ieee.org

*Series editor*
Professor Sameer Singh, PhD
Research School of Informatics
Loughborough University
Loughborough, UK

ISSN 1617-7916
ISBN 978-1-84882-298-6 e-ISBN 978-1-84882-299-3
DOI 10.1007/978-1-84882-299-3
Springer Dordrecht Heidelberg London New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2009926807



Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

Over the past few years, tensor processing tools have become more and more popular in the fields of computer vision and image processing, and tensor-valued image modalities have also been more commonly employed, with the remarkable example of Diffusion Tensor Magnetic Resonance Imaging.

However, tensor applications and tensor processing tools arise from very different areas. This can prevent important advances from rapidly spreading over the scientific community. Even though novel discoveries can greatly benefit many heterogeneous fields such as medical image processing or multilinear analysis, these advances are too often kept within the areas of knowledge where they were first employed.

Given this fact, a compilation of some of the most recent advances in tensor processing can be a valuable tool for the scientific community, thus providing a useful reference state of the artfor tensor processing applications as well as a general insight on the areas where tensor processing is being successfully applied.

The idea behind this book started after the Tensor Workshop held at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition held in Anchorage, Alaska, in June 2008. This workshop gathered experts from different fields working on tensor processing, and some of the ideas presented there were further developed toghether with the results of the Tensor Workshop held earlier in Gran Canaria (Spain) in November 2006, which was sponsored by the SIMILAR Network of Excellence of the European 6th Framework Programme.

As this book comprises theoretical advances and applications regarding very heterogeneous areas of image processing and computer vision, it has been organized into five parts, which are nevertheless preceeded by an introductory chapter about the different applications of the use of tensors in signal processing. Part I is devoted to the use of tensors and tensor field processing in general. The processing tools described in these chapters can be applied in a number of different applications. In Part II, two tensor techniques for image processing are presented. Later, Part III focuses on the use of tensors in computer vision applications, such as camera models or multilinear applications. As medical imaging is one of the areas that has taken more advantage of the advances of tensor processing, Part IV is dedicated to this

issue, collecting applications from Diffusion Tensor Magnetic Resonance Imaging to strain tensor estimation in cardiac analysis or elastography imaging. Finally, Part V is devoted to storage, visualization and interfaces with tensors, an issue of considerable importance since this new data modality presents particularities that require new approaches to these otherwise traditional problems.

The preparation of this book has been an arduous and difficult task. We would like to thank all the authors for their great effort and dedication in preparing their contributions. Also, all the reviewers, and specially the members of the editorial boards of the Tensor Workshops at CVPR'08 and in Gran Canaria in 2006, deserve our utmost gratitude. Thanks a lot to all of them.

Boston, MA
Fall 2008

*Rodrigo de Luis García*
*Santiago Aja Fernández*

# Contents

# A Review of Tensors and Tensor Signal Processing

L. Cammoun[1], C. A. Castaño-Moraga[2], E. Muñoz-Moreno[4], D. Sosa-Cabrera[2], B. Acar[5], M. A. Rodriguez-Florido[2,3], A. Brun[6], H. Knutsson[6], J. P. Thiran[1]

**Abstract** Tensors have been broadly used in mathematics and physics, since they are a generalization of scalars or vectors and allow to represent more complex properties. In this chapter we present an overview of some tensor applications, especially those focused on the image processing field. From a mathematical point of view, a lot of work has been developed about tensor calculus, which obviously is more complex than scalar or vectorial calculus. Moreover, tensors can represent the metric of a vector space, which is very useful in the field of differential geometry. In physics, tensors have been used to describe several magnitudes, such as the strain or stress of materials. In solid mechanics, tensors are used to define the generalized Hooke's law, where a fourth order tensor relates the strain and stress tensors. In fluid dynamics, the velocity gradient tensor provides information about the vorticity and the strain of the fluids. Also an electromagnetic tensor is defined, that simplifies the notation of the Maxwell equations. But tensors are not constrained to physics and mathematics. They have been used, for instance, in medical imaging, where we can highlight two applications: the diffusion tensor image, which represents how molecules diffuse inside the tissues and is broadly used for brain imaging; and the tensorial elastography, which computes the strain and vorticity tensor to analyze the tissues properties. Tensors have also been used in computer vision to provide information about the local structure or to define anisotropic image filters.

[1]Signal Processing Institute Ecole Polytechnique Fédérale de Lausanne Switzerland
Email: {leila.cammoun,JP.Thiran}@epfl.ch
[2]Center for Technology in Medicine. Department of Signals and Communciations
University of Las Palmas de Gran Canaria.
[3] Canary Islands Institute of Technology
Email: {ccasmor, dario, marf}@ctm.ulpgc.es
[4] Laboratory of Image Processing. Univ. de Valladolid (Spain)
Email: emunmor@lpi.tel.uva.es
[5]Electrical-Electronics Eng. Dept. Bogazici University Istanbul Turkey
Email: acarbu@boun.edu.tr
[6] Biomedical Engineering Dept. CMIV Linköpings Universitet Sweden
Email: {andbr,knutte}@imt.liu.se

## 1 Introduction

It is obvious that processing of higher-dimensional data sets puts high demands on computer power and storage capacity. Perhaps less obvious is that increasing the dimensionality of the data also has profound implications for the analysis of the data. It turns out that the scalar and vector fields are no longer sufficient for the analysis of data on curved surfaces and representation of high order differential operators such as the Hessian. Tensor analysis provides the adequate tool for such cases.

Tensor analysis is a generalization of the notions from vector analysis. The motivation for the theory is the fact that there are many physical quantities of complicated nature that cannot naturally be described or represented by scalars or vectors. We could find a lot of examples in various fields: in mathematics as in Riemanian geometry; in physics or engineering as the stress at a point in a solid body due to internal forces, the deformation of an arbitrary infinitesimal element of volume of an elastic body, or the moments of inertia, electrical and thermal conductivity in anisotropic materials or diffusion, etc. These quantities can be described and represented adequately only by the more sophisticated mathematical entities called *tensors*. It is appropriate to visualize tensors as generalizations of the more commonly known entities such as scalars and vectors. The order of a tensor can be thought of as the complexity of the entity it represents. While algebra and linear algebra is applicable to scalars and vectors, tensors of higher order are the building blocks of multilinear algebra.

## 2 Tensor Definition

The word *tensor* was first introduced by William Rowan Hamilton in 1846 in algebraic system domain, but the *tensor* as used in its current meaning was introduced by Woldemar Voigt in 1899. The notation was developed around 1890 by Gregorio Ricci-Curbastro under the title absolute differential calculus, and was made accessible to many mathematicians by the publication of Tullio Levi-Civita in 1900. In the 20th century, the subject came to be known as tensor analysis and around the year 1915 the introduction of Einstein's theory of general relativity achieved border acceptance. In fact, general relativity is formulated completely in the language of tensors.

Before any further development, it is an interesting task to understand the mathematical concept of tensors in order to better exploit their properties for the development of specific signal processing algorithms.

In an informal sense, a tensor is a generalized linear quantity or geometrical entity that can be expressed as a multi-dimensional array relative to a choice of basis; however, as an object itself, a tensor is independent of any chosen frame of reference. The rank of a particular tensor is the number of array indices required to describe such a quantity. For example, mass, temperature and other scalar quantities

are tensors of rank 0; force, displacement and other vector-like quantities are tensors of rank 1; diffusion, electrical impedance and other anisotropic measurements are tensors of rank 2. In a more strict sense, a $n$th-rank tensor in an $m$-dimensional space is a mathematical object that has $n$ indices and $m \times n$ components and obeys certain transformation rules. Each index of a tensor ranges over the number of dimensions of the space. However, the dimension of the space is largely irrelevant in most tensor equations (with the notable exception of the contracted Kronecker delta). Tensors are generalizations of scalars (that have no indices), vectors (that have exactly one index) and matrices (that have exactly two indices) to an arbitrary number of indices. Tensors provide in this way a natural and concise mathematical framework for formulating and solving problems in areas of physics such as elasticity, fluid mechanics and general relativity. In tensor notation, a vector $\mathbf{v}$ would be written $v_i$, where $i = 1, ..., m$ and a matrix is a tensor of type (1, 1), which would be written $a_{ij}$. A more general tensor may have an arbitrary number of indices $r+s$ which may be of mixed type, consisting of $r$ so-called contravariant (upper indices) and $s$ covariant (lower indices). While the distinction between covariant and contravariant indices must be made for general tensors, the two can be considered equivalent for tensors in a rectilinear coordinate system in Euclidean space, in particular when the coordinate basis has unit lenght and the metric tensor is the Kronecker delta, $\delta_{ij}$. Such tensors, known as Cartesian tensors, are widely used in fluid mechanics and elasticity to simplify calculations. In the notation for Cartesian tensors, all indices are lowered, $a_{ijk}$, since there is no need to discriminate between covariant and contravariant indices. For tensors in general however, upper indices denote contravariance and lower indices denote covariance and $a_{ij}{}^{k}$ is distinct from $a_{i}{}^{jk}$. The ordering of the indices, from left to right, is also important while the exact letters denoting the indices are not, for instance $a_{i}{}^{j}{}_{k}$ and $a_{q}{}^{p}{}_{r}$ denote the same tensor.

Tensors are used everywhere in natural science, engineering and mathematics, and to cover every field is a huge task. In the following sections we will describe the applications of tensors in some important fields, with an overall goal to give an understanding of how these relate to signal- and image processing in particular.

## 3 Example of Tensors in Mathematics

Due to the multi-linear nature of tensors, an often used convention in tensor notation is the so called Einstein summation convention. It means that identically named indices in tensor expressions are converted to sums over these indices. For instance, $\sum_i a^i b_i$ is equivalent to $a^i b_i$ and $\sum_i \sum_j x^i g_{ij} y^j$ is equivalent to $x^i g_{ij} y^j$. Summation is always from indices $1 \ldots m$, where $m$ is the dimension of the vector space.

If the vector space has a scalar product, i.e. the metric tensor $g_{ij}$ is known which is the case in Riemannian manifolds, one may identify covariant and contravariant indices. A contravariant vector $x^i$ may be "lowered" to a covariant vector using the metric, $x_j = \sum_i g_{ij} x^i = g_{ij} x^i$ and in general any contravariant index in a higher order tensor may analogously. By defining the inverse of the metric, $g^{ij}$ by, using the

Einstein convention, $g_{ik}g^{kj} = \delta_i^{\ k}$, one may also "rise" an index, $x^i = g^{ij}x_j$. Manipulating the tensor index notation is sometimes referred to as "index gymnastics". It is a powerful tool for manipulation of tensors, which nevertheless should be used with care when presenting tensor calculus to people who are not used to tensor notation.

Tensors are like vectors and scalars defined in vector spaces. With the additional structure of a Riemannian manifold, where tensor fields are defined in the tangent space of each point $p$ on the manifold, it is possible to also differentiate tensor fields. The thing that makes this interesting, and in some cases slightly involved, is the fact that tensors in different locations on the manifold are not defined in the same tangent vector space. To compare tensors in different locations on the manifold, and to differentiate tensor fields which involves comparing tensors at very nearby locations, one defines the exact meaning of this comparison by the notion of affine connections. In Riemannian geometry there is a special connection called the Levi-Civita connection which is particularly natural to use. It is the only torsion-free Riemannian connection preserving the Riemannian metric. The latter means that the derivative of the metric tensor is zero, i.e. two metric tensors in different locations on the manifold must be equal.

Differentiation of a scalar field on a manifold is facilitated by simply taking the partial derivative of the field in the current coordinate chart being used. Given a scalar field $a$, the gradient or partial derivative is denoted $a_{,i}$ and it is referred to as "comma derivative". The differentiation of a vector field in a manifold is slightly more involved. Like in differentiation of vector fields defined for curvilinear coordinates in vector analysis, both the derivative of the basis vectors and the coordinates must be considered. In differential geometry, this is called semicolon derivative or covariant derivative. For a contravariant vector field $v^i$, this mixed tensor field is defined

$$v^i_{\ ;j} = v^i_{\ ,j} + v^k \Gamma^i_{\ kj}, \tag{1}$$

where $\Gamma^i_{\ kj}$ is the Christoffel symbol defined by partial derivatives of the metric,

$$\Gamma^i_{\ kj} = \frac{1}{2} g^{im}(g_{mk,j} + g_{mj,k} - g_{kj,m}). \tag{2}$$

One example of an important equation which can now be described for manifolds is the geodesic equation, the ordinary differential equation describing particles moving in the manifold without the influence of any force. For a flat manifold in Cartesian coordinates this equation is $\ddot{x} = 0$, a path where the acceleration is zero. In the current setting for a manifold this translates to conditions on the momentary velocity, which is now interpreted as a velocity field $v^i = \dot{x}$, and thus can be handled by the covariant derivative. The condition $\ddot{x} = 0$ now translates to the directional derivative of the velocity field, in the direction of the velocity, which is exactly $\ddot{x}$.

$$\ddot{x} = 0 \Leftrightarrow v^i_{\ ;j} v^j = 0 \Leftrightarrow v^i_{\ ,j} v^j + v^k \Gamma^i_{\ jk} v^j = 0 \Leftrightarrow \frac{\partial x^i}{\partial t^2} + \Gamma^i_{\ jk} \frac{\partial x^j}{\partial t} \frac{\partial x^k}{\partial t}. \tag{3}$$

In the last step above, the velocity field is again transformed back to time-derivatives of $x^i$. This equation is an ordinary differential equation (ODE) which can be solved numerically and sometimes also analytically.

Three examples of manifolds which can be completely described by a single coordinate chart are the half-sphere, the cone and the hyperbolic plane. In all of these examples the chart is defined in the unit disk in $\mathbf{R}^2$ and by choosing an appropriate metric tensor field, the characteristics of the three manifolds can be described.

In figure 1 (a) the upper half of a sphere is visualized in 3-D and in (b) a chart defined in the unit disk representing the same sphere is visualized. The metric of the upper half of the unit sphere is

$$g_{ij}(u^1,u^2) = \begin{bmatrix} u^1u^1 - 1 & -u^1u^2 \\ -u^1u^2 & u^1u^1 - 1 \end{bmatrix}_{ij}. \tag{4}$$

It is a symmetric positive definite (PD) matrix which determines the scalar product at a specific point in the 2-D chart. This PD matrix may be visualized by a metric ellipsoid, i.e. a set of points $a^i$ such that the norm is equal to a fixed radius, $\sqrt{a^i g_{ij} a^j} = r$. In the circles have been calculated as geodesic circles. These are almost the same but the distance in the manifold is integrated along geodesics instead of using the scalar product in a particular point. Geodesic circles give very similar results to metric ellipsoids but yields more deformed shapes in areas where the curvature is large as seen in later examples. For this visualization they may be considered equal.

Another example is the cone, seen in figure 2. The metric of a cone with slope $k$ is

$$g_{ij}(u^1,u^2) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{ij} + \frac{k^2}{u^1u^1 + u^2u^2} \begin{bmatrix} u^1u^1 & u^1u^2 \\ u^1u^2 & u^1u^1 \end{bmatrix}_{ij} \tag{5}$$

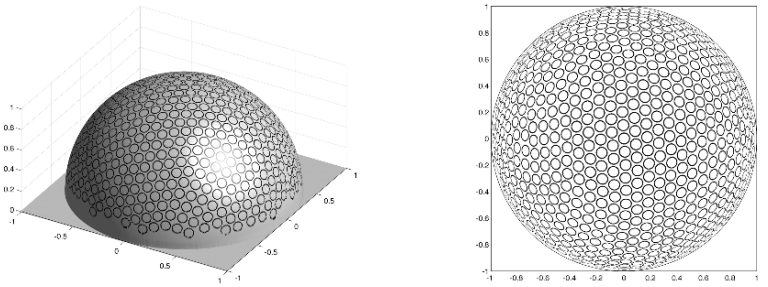
In this example it is possible to see that some geodesic circles are not ellipsoidal close to the center of the chart.

Finally a more exotic manifold, hyperbolic plane in figure 3. This manifold is not possible to embed in 3-D, but thanks to the Poincaré disk model seen in figure 3 (b) it is possible to use differential geometry. The metric of the Poincaré disk model, defined in the unit disk, is

$$g_{ij}(u^1,u^2) = \frac{1}{(1 - u^1u^1 - u^2u^2)^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{ij}. \tag{6}$$

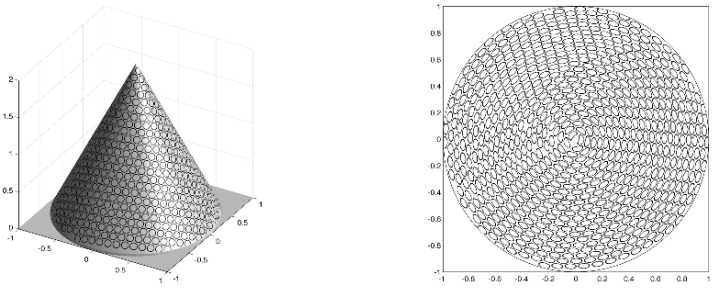
## *3.1 Differential Geometry*

Tensors naturally arise in the field of mathematics. In this section we briefly review some important concepts that show the importance of tensors in the field of differential geometry. The basis of this study relies on the interpretation of the topological space as a differentiable manifold. Basically, a differentiable manifold is an



(a) (b)

**Fig. 1** a) The upper half of the unit sphere in Cartesian coordinates $(x^1,x^2,x^3) \in \mathbf{R}^3$ painted with geodesic circles. b) The chart $(u^1,u^2) = (x^1,x^2)$ with the same geodesic circles drawn in the plane. The ellipsoid-shaped circles in the plane can be regarded as a representation of the metric tensor field $g_{ij}$, defined in the tangent plane to each point in the chart. From the metric it is possible to derive all the relevant geometrical properties of the half-sphere, i.e. to measure area, length and curvature directly in the chart without considering the embedding in $\mathbf{R}^3$.

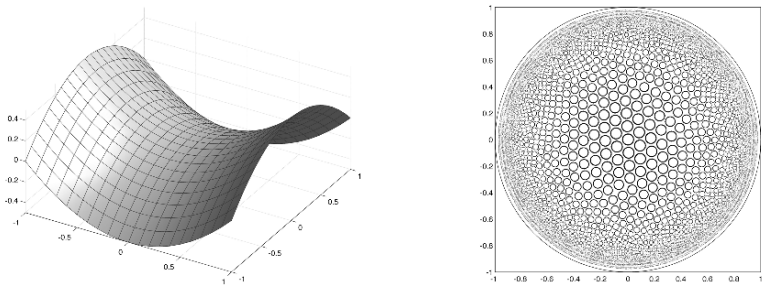


(a) (b)

**Fig. 2** a) A cone in Cartesian coordinates $(x^1,x^2,x^3) \in \mathbf{R}^3$ painted with geodesic circles. b) The chart $(u^1,u^2) = (x^1,x^2)$. Again the ellipsoid-shaped circles can be considered as a representation of the metric $g_{ij}$, which is defined for all points in the unit disc except $(0,0)$.

algebraical structure with the special property that any local neighborhood resembles the Euclidean space, although the global structure might be much more complicated, with a globally defined differentiable structure. In other words, for every point, its neighborhood is similar enough to Euclidean space to allow one to define differentiable equations.

For every point $P$ in a manifold $\mathscr{M}$, we can define a vector space with the same dimension as the manifold which consists of all the possible directional derivatives at that point. This vector field is called the tangent space at point $P$, $T_P\mathscr{M}$. Its importance resides in the fact that it allows to define generalized differential equations on the manifold. In the particular case that the tangent space at every point is equipped



(a) (b)

**Fig. 3** a) A surface with negative curvature embedded in $\mathbf{R}^3$. b) The Poincaré disk model of the hyperbolic plane, drawn with geodesic circles of equal radius. While it is not possible to embed this infinitely large and curved manifold in $\mathbf{R}^3$, differential geometry allow for a complete analysis in $\mathbf{R}^2$ by defining the proper metric tensor in the unit disk. Regular tilings of the Poincaré disk model are known to a broad audience from the work of the artist M. C. Escher.

with an inner product and it varies smoothly from point to point, the manifold is called *Riemannian manifold*. The continuous collection of scalar products on the tangent space at each point of the manifold is called *Riemannian metric* and it is denoted as $g$. The Riemannian metric, then, is nothing different from a tensor that is symmetric and positive definite.

Riemannian manifolds are quite interesting structures since it is possible to define concepts such as lengths of curves, areas, volumes, gradient of functions and divergence of vector fields by means of the Riemannian metric tensor. Thus, if we consider a continuously differentiable curve $\gamma(t) : [a,b] \mapsto \mathscr{M}$ on the manifold $\mathscr{M}$, we can compute at each point its instantaneous speed vector $\gamma'(t_0)$ in the tangent space $T\mathscr{M}(t_0)$ at any point $t_0 \in [a,b]$ and the instantaneous speed $\| \gamma'(t_0) \|_g$, which is the norm of the speed vector induced by the inner product with metric tensor $g$. To compute the length of the curve, we can proceed as usual by integrating this values along the whole curve:

$$L(\gamma) = \int_a^b \| \gamma'(t) \|_g \, dt \tag{7}$$

The distance between two points of a connected Riemannian manifold is the minimum length among the curves joining these points, as shown in Eq. (8). The curves realizing this minimum for any two points of the manifold are called geodesics.

$$\mathscr{D}(a,b) = \inf\{L(\gamma)\} \tag{8}$$

# 4 Example of Tensors in Physics and Engineering

In the next sections we identify several physics or engineering domains where tensors arise. In general, the use of second order tensors suits whenever the feature to be measured is intrinsically anisotropic, like electrical conductivity, diffusion of particles, etc.

In continuum mechanics there are two tensor magnitudes that are of great importance. The first one is the stress tensor that measures the internal distribution of force per unit area. The second one is the strain tensor that measures the deformation per unit area caused by the action of stress on a physical body. In solid mechanics, these tensor fields are linked by the generalized Hooke's Law, where as in fluid dynamics they are analogically linked by the Navier-Stokes equations.

## *4.1 Tensors in Solid Mechanics*

Actually, Hooke's law of elasticity is an approximation that states that the amount by which a material body is deformed (the strain) is linearly related to the force causing the deformation (the stress), its formula is the well known expression in the spring case for example:

$$F = -kx \tag{9}$$

where $x$ is the elongation distance, $F$ is the restoration force exerted by the spring, and $k$ is the force constant of the spring. But this approximation is only possible for small deformation where a linear approximation of the reality is possible (first order Taylor series). When working with larger deformation this approximation is not more possible, and the elasticity becomes non linear. The Hooke's Law is therefore replaced by its 'tensor form' which is the generalized Hooke's Law that links the stress tensors to the strain tensors.

*Stress* is a measure of the internal distribution of force per unit area that balances and reacts to the external loads or boundary conditions applied to a body. It is an example of a second-order tensor. In three dimensions, a second-order tensor can be represented by a $3 \times 3$ square matrix (containing nine components). However, in the absence of body moments, the stress tensor is symmetric and can be fully specified by six components.

In $N$ dimensional space, the stress tensor $\sigma_{ij}$ is defined as follow

$$dF_j = \sum_{i=1}^{N} \sigma_{ij} dA_i \tag{10}$$

where $dF_j$ are the components of the resultant force vector acting on a small area $dA$. which can be represented by a vector $dA_i$ perpendicular to the area element, facing outwards and with length equal to the area of the element. In three dimensions,

the internal force acting on a small area $dA$ of a plane that passes through a point $P$ can be resolved into three components: one normal to the plane and two parallel to the plane. The normal component divided by $dA$ gives the normal stress, and the parallel components divided by the area $dA$ give shear stresses.

*Strain* is the geometrical expression of deformation caused by the action of stress on a physical body, in the case of the string deformation where the deformation is small and linear, this deformation is the elongation distance $x$ in Eq. (9). However, for a small 3-D deformation, the strain tensor, is a symmetric tensor. Its $\varepsilon_{ii}$ diagonal coefficients are the relative change in length in the $i$ direction where the other terms $\varepsilon_{ij}$ are the shear strains.

The deformation of an object is therefore defined by a tensor field, i.e. this strain tensor is defined for every point of the object. This field is linked to the field of the stress tensor by a fourth order tensor $C_{ijkl}$ representing the stiffness in the generalized Hooke's Law as follow:

$$\sigma_{ij} = \sum_{kl} C_{ijkl}.\varepsilon_{kl} \tag{11}$$

These three tensors completely describe the mechanical properties of the material under study.

If the particular case of isotropic materials, characterized by properties which are independent of direction in space, the strain tensor expression could be decomposed as the sum of a constant tensor, known also as the pressure and a traceless symmetric tensor known as shear tensor:

$$\varepsilon_{ij} = (\frac{1}{3}\varepsilon_{kk}1_{ij}) + (\varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}1_{ij}) \tag{12}$$

where $1_{ij}$ is the unit tensor.

The general form of Hooke's Law for isotropic material could be written as linear combination of the stress and the strain tensor:

$$\sigma_{ij} = 3K(\frac{1}{3}\varepsilon_{kk}1_{ij}) + 2G(\varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}1_{ij}) \tag{13}$$

Where $K$ and $G$ are respectively known as bulk modulus and shear modulus.

When the deformation of a body is sufficiently large to invalidate the assumptions inherent in small strain theory, finite deformation tensors are used. This is commonly the case with elastomers, plastically deforming materials and other fluids and biological soft tissue.

## *4.2 Tensors in Fluid Dynamics*

In fluid dynamics, the Navier-Stokes equations are a set of equations that describe the motion of fluids. These differential equations state that changes in fluid particles momentum are simply the product of changes in pressure and dissipative viscous

forces acting inside the fluid. The general form of the Navier-Stokes equations for the momentum conservation involve the use of tensors as shown in the following formula:

$$\rho \frac{d\mathbf{v}}{dt} = \nabla.\mathbb{P} + \rho f \tag{14}$$

where $\rho$ is the fluid density, $\mathbf{v}$ is the velocity vector, $f$ is the body force vector and $\mathbb{P}$ is the stress tensor that represents the surface force applied on the fluid and has the following form:

$$\mathbb{P} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{pmatrix} \tag{15}$$

where $\sigma$ are normal stresses and $\tau$ shear stresses.

Another important source of information of tensor nature is the so called *velocity gradient tensor* described in Eq. (16). This tensor is given by the spatial gradient of the velocity vector field denoted by $\mathbf{v}(\mathbf{x}) = [u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x})]^T$ and contains the information on how the velocity is changing in space, which helps to detect important fluid flow features such as vortices, separation and shocks.

$$\mathbf{V}(\mathbf{x}) = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix} \tag{16}$$

A common practice to identify the relevant features of the fluid flow is to decompose $\mathbf{V}$ into its symmetric and antisymmetric parts, which yields to the definition of the symmetric tensor $\mathbf{V}_S = \frac{1}{2}(\mathbf{V} + \mathbf{V}^T)$ and the antisymmetric tensor $\mathbf{V}_A = \frac{1}{2}(\mathbf{V} - \mathbf{V}^T)$ [1]. The symmetric part represents the *strain tensor* with the elongational strains on the diagonal and the shearing strains on the off-diagonal, whereas the non-null elements of the antisymmetric part contain the components of the *vorticity* vector, which measures the circulation per unit area of the fluid flow and allow us to detect and measure the strength of vortices [2].

An eigenanalysis of the velocity gradient tensor is used to classify the local features of the flow pattern [3]. In general, we can distinguish two general types of flow types characterized by the fact that the 3 eigenvalues of $\mathbf{V}$ are real numbers or only 1 is real and 2 complex conjugate numbers.

The sign of the real eigenvalues indicates whether the flow is accelerating, if it is positive, or decelerating, if it is negative, along the direction determined by the associated eigenvector. The magnitude of the eigenvalues determines the strength. When there is a complex conjugate pair of eigenvalues it indicates that there is a spiral flow. The magnitude of the imaginary part indicates the strength of the spiraling flow. If the value is small the flow is hardly swirling, whereas a big magnitude means that the flow is rotating rapidly about the point. The sign of the real part indicates whether the flow is converging, if it is negative, or diverging, if it is positive, with the magnitude of the real part reflecting the strength of the attraction or repulsion.

The special case where the real part is zero, the flow produces concentric periodic paths [3].

### *4.3 Tensors in Electromagnetism*

Electromagnetism is a field of science where tensors also arise. The *electromagnetic tensor* is an antisymmetric $4 \times 4$ second order tensor that describes the electric and magnetic fields, yielding to a simplification in notation of the Maxwell equations. It is usually written as:

$$\mathbf{F} = \begin{pmatrix} 0 & \frac{1}{c}e_x & \frac{1}{c}e_y & \frac{1}{c}e_z \\ -\frac{1}{c}e_x & 0 & -b_z & b_y \\ -\frac{1}{c}e_y & b_z & 0 & b_x \\ -\frac{1}{c}e_z & b_y & b_x & 0 \end{pmatrix} \tag{17}$$

where $[e_x, e_y, e_z]^T$ are the components of the electric field $\mathbf{e}$, $[b_x, b_y, b_z]^T$ are the components of the magnetic field $\mathbf{b}$ and $c$ is the speed of light. The tensor representation allows us a compact representation of the Maxwell's equations in terms of this tensor and its dual $\mathbf{G}$, which is obtained by replacing the electric components by the magnetic components and vice versa:

$$\frac{\partial f^{\alpha\beta}}{\partial x^{\beta}} = \mu_0 j^{\alpha} \tag{18}$$

$$\frac{\partial g^{\alpha\beta}}{\partial x^{\beta}} = 0 \tag{19}$$

where $\alpha, \beta$ denote the indices of the tensor and $j^{\alpha}$ is a 4-D vector which denotes the current density. Hence, the electromagnetic field tensor not only is a mathematical object to simplify notation, but it also completely describes the inherent electromagnetic field.

## 5 Tensors Usage in Medical Imaging

### *5.1 Diffusion Tensor Imaging*

One of the most interesting applications of tensors in medical imaging is the DT-MRI (diffusion tensor magnetic resonance imaging) [4]. This is a special kind of magnetic resonance that shows the motion of water molecules in tissues. Its application in brain imaging is of paramount importance, since it allows to identify the nerve fiber bundles, what is very useful to understand the brain development, or to diagnose several pathologies like brain ischemia, axon damages or psychiatrics

problems [5]. In this section we explain how diffusion tensor (DT) is computed as well as its usefulness in medical issues. Besides, it is described a Riemmanian framework for DT and how the fiber tracts can be identified and visualized.

### 5.1.1 Acquisition and Computation

Water molecules have a Brownian motion, that means that they constantly randomly collide with other molecules. If there are not obstacles in the environment, this effect causes an isotropic diffusion, which can be seen like a Gaussian distribution of water molecules released from one point, expanding dynamically [6]. In case of obstacles constraining motion in some given direction, diffusion becomes anisotropic. This is what occurs within fibers. Nerves are made up of neurons axons, whose myelin covering does not allow water diffusion across it. Because of it, fibers can be identified as areas where diffusion has a clearly main direction.

Anisotropic diffusion has been described by ellipsoids, whose radii represent diffusion amount in each of the main directions. On the other hand, a symmetric second order $3 \times 3$ tensor can be visualized like an ellipsoid, whose axes direction and length are given, respectively, by the eigenvectors and eigenvalues of the matrix representing the tensor. For this reason, tensors are used to characterize diffusion in tissues [4], and, in such a way, to describe their structure.

Magnetic resonance images (MRI) of the brain are widely used in neurology to identify regions like white matter, gray matter or ventricles. MRI consists in the excitation of the tissue with an RF-pulse. When molecules return to their relaxed state, they release energy, which is measured and translated into pixel intensity in image. Since different tissues have different energetic behavior, they can be distinguished in the MRI. However, white matter appears like an homogeneous region, where fiber structure cannot be notice. Therefore, in order to know about the structural anatomy of white matter, diffusion measurement becomes necessary. To assess this magnitude, the MRI is carried out by applying a pulse sequence, known as Stejskal-Tanner imaging sequence [7]. It consists in two symmetrically positioned gradient pulses, with a $180^\circ$ refocusing pulse between them. In such a way, diffusion in a given direction can be measured.

The aim of the two symmetric pulses is to cancel the phase shift for static spins. Spins whose location has changed between both gradients do not recover their original phase causing energy loss. In order to eliminate the signal not related with diffusion, two images must be captured, one of them without diffusion weighting, called $S_0$, and the other one with diffusion weigthing, denoted as $S$. Both images are related by the diffusion $D$ through the next equation:

$$S = S_0 e^{-bD} \tag{20}$$

The $b$-factor is the diffusion weighting factor defined by LeBihan [8], according to the following equation:

$$b = \gamma^2 \delta^2 (\Delta - \frac{\delta}{3}) |\mathbf{g}|^2 \tag{21}$$

In this equation, $\gamma$ is the proton gyromagnetic ratio, $|\mathbf{g}|$ and $\delta$ are the strength and duration, respectively, of the gradient pulses used to obtained the image, and $\Delta$ the time between the two gradient pulses.

The value of diffusion in Eq. (20) is known as ADC (apparent diffusion component) and depend on the gradient pulse and the parameters of the pulse sequence. Because of this, it does not reflect how anisotropically molecules are diffused. In order to obtain this information, we need to apply gradient pulses in several directions, and compute from such measures the diffusion tensor $\mathbf{D}$ [4], from which diffusion in a given direction $\mathbf{x}$ can be computed according to next equation:

$$d(\mathbf{x}) = \mathbf{x}^T \mathbf{D} \mathbf{x} \tag{22}$$

In this tensorial case, Eq. (20) can be rewritten including the normalized gradient vectors $\hat{\mathbf{g}} = \frac{\mathbf{g}}{|\mathbf{g}|}$:

$$S = S_0 e^{-b\hat{\mathbf{g}}^\mathbf{T}\mathbf{D}\hat{\mathbf{g}}} \tag{23}$$

Since $\mathbf{D}$ can be represented like a $3 \times 3$ symmetric matrix, we need six coefficients to define it. Therefore, at least six different non-collinear gradient directions must be applied in order to obtain six independent measures, from which tensor elements can be estimated. It is also required to obtain the image data without diffusion $S_0$, in order to eliminate the spin dependence, as previously noted. Let be $\{S_0, S_1, ..., S_6\}$ the seven images (the one without diffusion weighting, and the six with different gradient directions $\hat{\mathbf{g}}_i, i = 1, ..., 6$). The diffusion tensor can be computed from this equation system:

$$ln(S_i) = ln(S_0) - b\hat{\mathbf{g}}_i^\mathbf{T}\mathbf{D}\hat{\mathbf{g}}_i,\ i = 1, ..., 6 \tag{24}$$

Although six gradients are enough to compute the diffusion tensor, more gradient pulses are often used in the acquisition to reduce the noise influence in the tensor computation.

The information contained in the diffusion tensor is very useful but it is complex and may not be *user-friendly*. Because of this, some scalar parameters are computed from the tensor in order to make easier its use. As an example of these scalar measures, we can point out the apparent diffusion coefficient, that measures the global diffusivity; the fractional anisotropy, that shows how the DT differs from the isotropic case; the relative anisotropy or the volume ratio. All of them are based on rotationally invariant eigenvalues from the matrix representing the tensor. The lattice anisotropy index [9] can be also highlighted, since it takes into account a neighborhood of every voxel. Geometrical diffusion measures [10] (linear, planar and spherical anisotropy) are also very useful, because they include information about directional variation of diffusion.

Scalar parameters from tensors can be visualized as traditional gray images and can be used in a more intuitive way by physicians, as well as traditional processing

techniques can be applied to such images. Nevertheless, it is preferred to adapt these techniques to the special characteristics of tensorial data in order to exploit all the information given by the tensor. Thus, several works about DTI processing have been developed, proposing segmentation, registration and regularization algorithms for tensorial images. A review of these techniques can be found in [11].

Besides processing, several approaches to tensor visualization have been published. The most extended way is representing tensor by an ellipsoid whose shape and size are given by the tensor eigenvectors and eigenvalues [9]. Other ideas are to plot the spherical, elliptic and linear components [10], or to use superquadrics [12].

### 5.1.2 A Riemannian Framework for DTI

In the special case of tensor datasets like those provided, for instance, by DT-MRI, we can identify $PS^+(3)$, the set of $3\times 3$ real symmetric positive definite matrices, with the family of 3-D normal distributions with 0-mean as the 6-D parameter space of variances and covariances, that is a Riemannian manifold.

Following the work by Rao [13] and Burbea-Rao [14], where a Riemannian metric was introduced for $PS^+(3)$ in terms of the Fisher information matrix, it is possible to define notions such as the geodesic distance, the curvature, the mean, and the covariance matrix for elements in $PS^+(3)$. The tangent space $T_{\boldsymbol{\Sigma}}PS^+(3)$ at $\boldsymbol{\Sigma}\in PS^+(3)$ is taken to be $S_{\boldsymbol{\Sigma}}(3)$, the vector space of symmetric matrices [15]. We denote by $\mathbf{E}_i$, $i=1,...,6$ the basis of such a vector space, which is given by:

$$\begin{array}{lll}\mathbf{E}_1=\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix} & \mathbf{E}_2=\begin{pmatrix}0&0&0\\0&1&0\\0&0&0\end{pmatrix} & \mathbf{E}_3=\begin{pmatrix}0&0&0\\0&0&0\\0&0&1\end{pmatrix}\\ \mathbf{E}_4=\begin{pmatrix}0&1&0\\1&0&0\\0&0&0\end{pmatrix} & \mathbf{E}_5=\begin{pmatrix}0&0&0\\0&0&1\\0&1&0\end{pmatrix} & \mathbf{E}_6=\begin{pmatrix}0&0&1\\0&0&0\\1&0&0\end{pmatrix}\end{array} \tag{25}$$

The Riemannian metric tensor for $PS^+(3)$, derived from the Fisher information matrix is given by the following theorem, which is proved in [15]:

The Riemannian metric for the space $PS^+(3)$ of multivariate normal distributions with zero mean is given, $\forall\boldsymbol{\Sigma}\in PS^+(3)$ by:

$$g_{ij}=g(\mathbf{E}_i,\mathbf{E}_j)=\langle\mathbf{E}_i,\mathbf{E}_j\rangle_{\boldsymbol{\Sigma}}=\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{E}_i\boldsymbol{\Sigma}^{-1}\mathbf{E}_j\right)\quad i,j=1,...,6 \tag{26}$$

In practice, this means that for any tangent vectors $\mathbf{A},\mathbf{B}$, their inner product relative to $\boldsymbol{\Sigma}$ is $\langle\mathbf{A},\mathbf{B}\rangle_{\boldsymbol{\Sigma}}=\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{A}\boldsymbol{\Sigma}^{-1}\mathbf{B}\right)$.

We recall that, if $\boldsymbol{\Sigma}:t\mapsto\boldsymbol{\Sigma}(t)\in PS^+(3)$, $\forall t\in[t_1,t_2]\subset\mathbb{R}$ denotes a curve segment in $PS^+(3)$ between two normal distributions parameterized by the matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, its length is expressed as:

$$\mathscr{L}_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2)=\int_{t1}^{t2}\left(\sum_{i,j=1}^{6}g_{ij}(\boldsymbol{\Sigma}(t))\frac{d\theta_i(t)}{dt}\frac{d\theta_j(t)}{dt}\right)^{1/2}dt \tag{27}$$

As stated for example in [16], the geodesic starting from $\boldsymbol{\Sigma}(t_1)\in PS^+(3)$ in the direction $\mathbf{V}=\dot{\boldsymbol{\Sigma}}(t_1)$ of the tangent space $T_{\boldsymbol{\Sigma}}PS^+(3)=S_{\boldsymbol{\Sigma}}(3)$, is given by the exponential map:

$$\boldsymbol{\Sigma}(t)=\boldsymbol{\Sigma}(t_1)^{1/2}\exp\left((t-t_1)\boldsymbol{\Sigma}(t_1)^{-1/2}\mathbf{V}\boldsymbol{\Sigma}(t_1)^{-1/2}\right)\boldsymbol{\Sigma}(t_1)^{1/2}\forall t\in[t_1,t_2] \tag{28}$$

We recall that the geodesic distance $\mathscr{D}$ between any two elements $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ is the length of the minimizing geodesic between $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$:

$$\mathscr{D}(\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2)=\inf_{\boldsymbol{\Sigma}}\{\mathscr{L}_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2):\boldsymbol{\Sigma}_1=\boldsymbol{\Sigma}(t_1),\boldsymbol{\Sigma}_2=\boldsymbol{\Sigma}(t_2)\}$$

It is given by the following theorem, whose original proof is available in an appendix of [17] but different versions can also be found in [15] and [18].

Theorem : (S.T. Jensen, 1976, originally proved in [17]) Consider the family of multivariate normal distributions with common mean vector but different covariance matrices. The geodesic distance between two members of the family with covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ is given by

$$\mathscr{D}(\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2)=\sqrt{\frac{1}{2}\mathrm{tr}\left(\log^2\left(\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1/2}\right)\right)}=\sqrt{\frac{1}{2}\sum_{i=1}^{3}\log^2(\eta_i)} \tag{29}$$

where $\eta_i$ denote the 3 eigenvalues of the matrix $\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1/2}$.

### 5.1.3 DTI Usage

DT-MRI offers important benefits to clinicians. The main application has been in brain imaging since it allows to identify fibers. Nevertheless, it has been used also in myocardial analysis (for instance, in [19]) and in the research of other tissues microstructure, like cartilage [20].

In brain, DTI allows the examination of its microstructure in vivo, what cannot be done with older techniques. Hence, it has a lot of clinical applications [5]. Among others, we can mention its use in the study of the structure and development of white matter. Furthermore, it is a helpful tool in the research of several brain diseases, such as cerebral ischemia, multiple sclerosis, epilepsy or Alzheimer diseases. Its usefulness is also remarkable in tumor diagnosis, since it allows to distinguish between different sorts of tumors. The advances in this medical research are related to the improvements in processing and visualization of tensorial data. A visualization option is to represent the diffusion tensor as a color image by considering the principal diffusion direction (eigenvector corresponding to the higher eigenvalue) like a RGB

vector, and the value of FA like the color intensity. In fiber tracking or tractography, visualization of fibers is usually carried out by streamlines or streamsurfaces.

The probability density function (pdf) of this anisotropic motion can be approximated at each voxel by an anisotropic 3D Gaussian function that can be parameterized by the diffusion tensor to create a 3D diffusion tensor field [4]. A diffusion tensor (DT) is a second order tensor, i.e. a $3 \times 3$ matrix. Since this tensor is symmetric and semi-positive definite the eigenvalues are always positive as long as they are unaffected by noise. A diffusion tensor D can then be expressed in terms of its eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ and eigenvectors $(\overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3})$ as follow:

$$\mathscr{D} = (\overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3}) \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} (\overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3}) \tag{30}$$

when performed within a compact tract with parallel running axonal trajectories, the DT is very anisotropic and the direction of the fiber tract correspond to the principal eigenvector (see figure 4). This was the basic lead for several fiber tracking algorithms that all have the same objective of inferring the fibers bundles trajectories from the DT field.

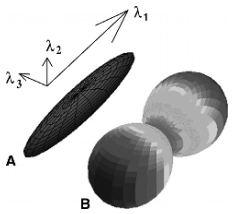



**Fig. 4** Example of diffusion tensors. In A, the diffusion is shown as an ellipsoid with its principal eigenvectors. in B, the diffusion tensor is shown an an orientation distributed tensor [21].

### 5.1.4 DTI Tractography

Several researches have proposed tractography methods and have reported success in fiber tracking [22, 23, 24, 25, 26, 27] . For connecting different regions in the brain the fiber tracking methods are basically based on inferring continuity from voxel to voxel using an estimate of local tract direction. Some recent researches have reviewed the most of the fiber-tracking methods [23]. The basic fiber tracking algorithm is the *streamlines tracking* method that models propagation in the major eigenvector field. Actually, this method consists on effering the tracing by connecting each voxel to the adjacent one toward which the fiber direction is pointing [24][25][23]. This direction is collinear to the eigenvector corresponding to the largest eigenvalue of the diffusion tensor in this voxel. In 1999 Mori et al. [28] proposed the *Fiber Assignment by Continuous Tracking* (FACT) method which changes

the direction of the propagation at the interface of the voxels boundary. In 2000 Basser et al. [25] proposed a continuous tensor field estimation method where a fiber is computed as a streamline of the major eigenvector field corresponding to the solution of the ordinary differential equation using the $4^{th}$ order Runge-Kutta method

$$\frac{dr(s)}{ds} = \varepsilon_1(r(s)) \tag{31}$$

where $r(s)$ is a vector parameterized by the arc length, $s$, of the trajectory, and $\varepsilon_1$ is the unit eigenvector calculated at this latter position.

Another way to determine tract direction as described in [26] is to use the entire DT to deflect the incoming vector direction $V_{in}$ (propagation direction from the previous integration step)

$$V_{out} = V_{in} \tag{32}$$

This tensor operator deflects the incoming vector towards the major eigenvector direction. This algorithm called *tensor deflection* (TEND) enables a smoother tract reconstruction, on one hand the incoming vector is deflected to the direction of the main eigenvector and on the other hand the deflection curvature is limited.

The tensorline tracking technique combine the streamline and the TEND methods. It does not consider simply the main diffusion direction of the local tensor, but also the nearby local tensor information.

$$V_{out} = fe_1 + (1-f)((1-g)V_{in} + gDV_{in}) \tag{33}$$

where, $f$ and $g$ are user defined in interval [0,1]. For most of fiber-tracking methods, a simulated fiber is propagated in both positive and negative direction of the eigenvector direction of the local diffusion. The tracking continues until the boundary of the data set is reached or the mean diffusion anisotropy at the current curve point is less than an a priori defined threshold.

As presented in [29] and [30], a voxel that contains several populations of axons with different directions has a tensor which shape will change according to the proportion of each fiber population, and the principal eigenvector looses then its signification and tracking becomes more hazardous. The principal eigenvector is also very sensitive to noise, especially in areas of low anisotropy [31]. The family of tracking algorithms described above could globally be described as deterministic, and most of them reduce the tensor to a vector field [25, 24, 32, 33, 34] and consequently do not take into account the uncertainty of the fiber direction. To address this issue Hagmann et al. have developed in [27] a statistical fiber tracking algorithm based on two hypotheses: the solutions of this tracking algorithm are a set of lines or virtual fiber trajectories. Each line is the result of a random walk in a diffusion tensor field. This walk, indexed by a time dependent position vector $q_i$, is driven by the local diffusion properties, i.e. the diffusion tensor $D_i$, and a regularity constraint vector to ensure smooth trajectories. More precisely we have:

$$q_{i+1} = q_i + \mu\Omega_i \tag{34}$$

The displacement of size $\mu = 1mm$ at each step occurs along direction $\Omega_i$ that is defined as:

$$\Omega_i = \begin{cases} \frac{\lambda d_i + \Omega_{i-1}}{\|\lambda d_i + \Omega_{i-1}\|} \\ \Omega_{i-1}.\Omega_{i-1} > 0 \end{cases} \tag{35}$$

where $d_i = D_i^{\alpha} r_i$ with $r_i$ a random vector uniformly distributed over the unit sphere. $\alpha$ is an anisotropy enhancing exponent set to 3. $d_i$ is a random vector distributed according to the local diffusion properties. As shown in Eq. (35), the curve that particle propagation generate grows along a unit vector $\Omega_i$ that is a weighted sum of $d_i$ and the previous displacement vector enhancing collinearity. A large set of fibers thus generated provides an estimate of axonal trajectories throughout the brain. An example of some virtual fibers generated by the streamline algorithm are presented in figure 5.

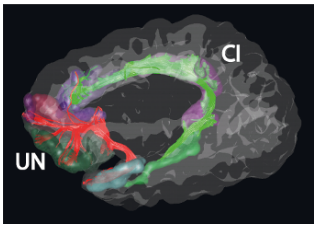



**Fig. 5** Example of virtual tracts: the red bundle corresponds to the uncinate, and the green one is the cingular one.

## *5.2 Tensorial Elastography*

Elastography is well established in the literature [35], and can aid the diagnosis of several diseases such as breast and prostate cancer, or cardiovascular dysfunctions [36, 37]. Ultrasound elastography measures the elastic properties of soft tissues using ultrasound signals. The scalar representation of the axial strain (in the direction of an applied compression) leads to the visualization of some cancers that otherwise would be blind for the US scanner due to their isoechoic response with respect to the surrounding tissue. Researchers have introduced new techniques such as MR-elastography, optical elastography, etc. and the tensorial approach presented here as well as its visualization, can be extended to all of them.

Tensorial elastography is a rather new research area. Investigations trying to get new information for the physicians out of a tensorial approach, and visualization in elastography, has its starting point in papers such as [38]. The fact that other modalities as cardiac strain, has succeeded in contributing to characterize the myocardial

motion and therefore to determine physiological factors affecting the behavior of the cardiac cycle, make us assume the importance of tensorial elastography in spite of its early stage of development.

Several techniques have been proposed to estimate the tensor strain components. All of them estimate a displacement vector field $\mathbf{u}$, due to an applied deformation load. From the displacement field, the individual components of the tensor are calculated and visualized separately. However, the spatial variation, i.e. the gradient, of this displacement defines a second order tensor field given by:

$$\frac{du}{dA} = J_u n \tag{36}$$

where $\frac{du}{dA}$ is the column matrix of unit relative displacement vector components, $n$ is the column matrix of direction cosines and $J_u$ is the the *displacement gradient matrix*, a $3 \times 3$ matrix, also called the *Jacobian matrix*.

In mechanical engineering it is well known that this tensor can be decomposed into the strain tensor $\boldsymbol{\varepsilon}$ (the symmetric part), with the elongational strains on the diagonal and the shearing strains on the off-diagonal, and the vorticity tensor $\boldsymbol{\omega}$ (the antisymmetric part), that contains only the vorticity components.

The strain tensor, $\boldsymbol{\varepsilon}$, measures the changes of shape locally (elongations or stretching), and relates to the stress tensor through the stiffness of the material, the Young modulus, see equation (13).

On the other hand $\boldsymbol{\omega}$ informs about modifications related to rotations and not to the shape.

This decomposition, $J_u = \boldsymbol{\varepsilon} + \boldsymbol{\omega}$, is very important in the tensorial approach to elastography. On one hand, we have an estimation of the strain tensor, as all the works in bibliography, and besides, an estimation of the structures inertia to rotate is obtained.

### 5.2.1 Tensor Visualization of Ultrasound Elastography Data

Although the elastography problem is reduced to the estimation of tensor data (the strain tensor), its visualization has been traditionally based on scalar representations, i.e. scalar images (Young modulus, components of the strain tensor, etc.).

However, it is important to know that the visualization of tensor fields improve the understanding and interpretation of tensor data and is therefore of paramount importance for scientist applications (medicine, engineering, etc.). Over the last few years, tensor fields visualization has achieved great interest thanks to improvement of graphics hardware (allows to visualize large amounts of data), and the advances in nervous fiber visualization given by DT-MRI dataset.

The difficulty of interpreting tensor data arises not only from the usually large size of the data sets but also from the fact that each tensor could be represented by a matrix. The aim of tensor field visualization is therefore to transform these large amount of data into a single image which can be easily understood and interpreted by the user.

Two useful tools in visualizing data are data transformation and data reduction. Data transformation retains all the information in the data but presents it in a different form. The transformation of a tensor into its eigenvectors and eigenvalues is an example. Data reduction, on the other hand, extracts only partial information from the original tensor data and so gives an incomplete representation. Scalar data obtained by reducing tensor data includes, in our case, any principal strain, the maximum shearing strain, and the strain energy per unit volume. Tensor algebra has been used because it is the only way in which important tensor properties under study can be preserved, managed and exploited.

Although the DT-MRI visualization techniques are quite well developed, it is not obvious its application to strain tensor fields. Remember that the strain tensor, used in elastography, is symmetric but does not satisfy the positive semidefinite condition. Hence, its visualization as an ellipsoid in each data point of the field is not clear [39, 40]. However, in our case the condition of positive semidefinite is not a barrier to visualize the strain tensor field as an ellipsoid field, because clinically the sign of the strain tensor eigenvalues (positive and negative represents respectively material stretching or compression, in the direction of the corresponding eigenvector) does not corrupt the information given with this representation. Then, we propose here a technique similar to that in [41], used for the visualization of myocardial strain-rate tensors, i.e. visualize tensors as ellipsoids (ellipses in our 2D case) colored according to the sign of the largest eigenvalue representing stretch or compression in the main direction, or if there exists a negative eigenvalue.

In order to validate the usefulness of this representation, let take into account that what we want to discriminate, at first, are tissues with different elastic properties. The theory says that axially, all the points of a compressed tissue should have a negative value of strain. Due to noise decorrelation this is not always true for data obtained from ultrasound. However, the obtained images differentiate between two tissues or materials with different stiffness.

The sign of the deformation is represented with colors and with the size of the ellipses or ellipsoids in their principal directions, we appreciate how much deformation there is in each principal direction and their ratio, and how much deformation from the total has been absorbed by each tissue.

More information can be found in [42]. In figure 6 some different 2D and 3D tensorial elastograms are presented for illustration.

## 6 Tensors Usage in Computer Vision

### *6.1 Representation of Local Structure Using Tensors*

Tensors are nowadays an important tool in the field of computer vision since it provides information about the local structure of an image. We understand by local

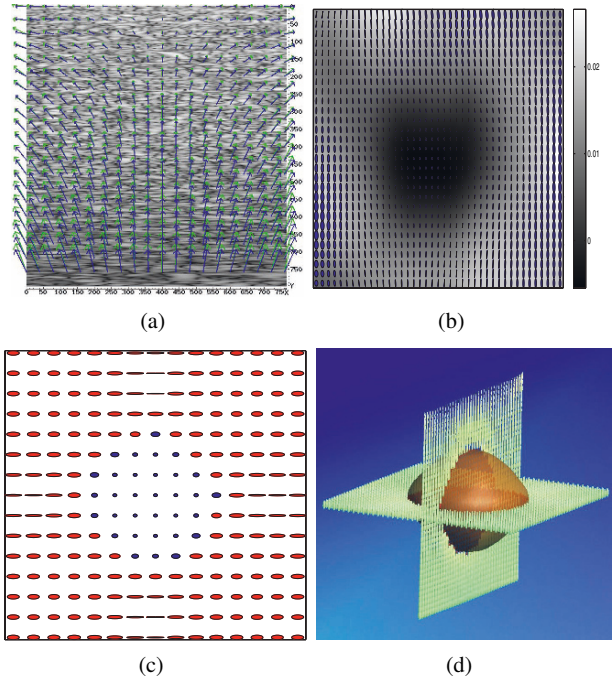



**Fig. 6** a) Displacement field overlaid to the B-mode scan dor a synthetic experimet. The blue vectors are the ground truth and the green ones our estimation. b) Lamé tensorial elastogram overlaid to the axial elastogram for acommercial CIRS 059 breast phantom scanned with the Ultrasonix RP500. c) A novel tensorial elastogram using dephormational strain glyphs (DSG) for the case of a virtual phantom. d) A 3D tensorial elastogram. In all the cases there is a centered circular or spherical inclusion three times stiffer than the background.

structure the meaningful features in the image, which are usually related to those areas where a contour is present [43].

The tensor representation for local orientation was introduced by Knutsson [44] to encode the orientation information for signals with dimensionality greater than one. Tensors, besides allowing a compact representation, have a solid mathematical body that supports further analysis in the tensor domain. The idea is to represent the orientation of a neighborhood by a symmetric and positive semidefinite matrix.

In the case that more than one dominant orientation is present, (for instance, a corner in a scalar image), it should be necessary to use higher order tensors to completely describe local orientation in a strict sense. However, in practice, a second order tensor provides enough information to distinguish the relevant features of an image, including corners [45], although certain ambiguity might appear under certain circumstances.

Knutsson studied the requirements that a continuous representation of orientation should hold [44]. Let $\mathbf{v_o} \in \mathbb{R}^n$ be a known vector which represents the main

orientation and let $\mathscr{M}$ be a mapping $\mathscr{M} : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$ to be defined. Then $\mathscr{M}$ should satisfy the following properties in order to properly describe local structure:

- *Uniqueness:* Antipodal vectors should be mapped onto a singular point. In this way, vectors pointing in opposite directions are treated equally removing the discontinuity at $\pi$:
$$\mathscr{M}(\mathbf{v_o}) = \mathscr{M}(-\mathbf{v_o}) \tag{37}$$
- *Polar separability:* The norm of the mapped vector should be rotation invariant. Information carried by the norm of the vector does not carry information about orientation:
$$\| \mathscr{M}(\mathbf{v_o}) \|_F = f(\| \mathbf{v_o} \|), \tag{38}$$
where $f : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is an arbitrary function.
- *Uniform stretch / Equivariance:* Variations in the original spaces should be proportional to variations in the mapped space. In other words, the tensor representation must be invariant to basic transformations such as translation, rotation or scale:
$$\| \partial\mathscr{M}(\mathbf{v_o}) \|_F = b \| \partial\mathbf{v_o} \| . \tag{39}$$
A mapping that fulfills the above requirements is $\mathbf{M} : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$ [44] such that
$$\mathbf{M}(\mathbf{v_o}) = \frac{\mathbf{v_o}\mathbf{v_o}^T}{\| \mathbf{v_o} \|}. \tag{40}$$

This mapping is a non-linear mapping, therefore filtering operations applied to the elements of $\mathbf{M}$ have to be carefully made. This mapping provides a symmetric positive semidefinite matrix with $\frac{n(n+1)}{2}$ independent components where the dominant orientation given by the vector $\mathbf{v_o}$ is codified by the dominant eigenvector of that matrix. The symmetry condition guarantees the invariance to rotations at the same time that assures that all the eigenvalues of the matrix are real values and the positive semidefiniteness guarantees that all of them are positive. The matrix provided by this mapping is called *local structure tensor* and will be denoted by $\mathbf{T}$. In section 6.2 we discuss different approaches to directly estimate the local structure tensor from data.

The main advantage of the tensor representation is that tensors contain more information than only the dominant orientation, since they also quantify uncertainty about this orientation being actually dominant [46]. In other words, they quantify how much the signal varies along the dominant orientations and also along the orientation orthogonal to the dominant within a certain neighborhood. Thus, through an analysis of eigenvalues $\lambda_1 \geq \ldots \geq \lambda_m$ and eigenvectors $\mathbf{e_1}, \ldots, \mathbf{e_m}$ we can distinguish different cases in 3D $m = 3$ (an analogous classification can be done in 2D):

1. Planar case ($\lambda_1 \gg \lambda_2 \simeq \lambda_3 \simeq 0$): There is only one main direction of signal variation. This neighborhood is approximately a planar structure whose normal vector is given by $\mathbf{e_1}$.

2. Linear case ($\lambda_1 \simeq \lambda_2 \gg \lambda_3 \simeq 0$): In this case, there are two main directions of signal variation which yields a line-like neighborhood oriented along $\mathbf{e_3}$, as the edges of a cube.
3. Point structure case ($\lambda_1 \simeq \lambda_2 \simeq \lambda_3 \gg 0$): There is no preferred orientation of signal variation, which represents a corner or a junction in 3D.
4. Homogeneous case ($\lambda_1 \simeq \lambda_2 \simeq \lambda_3 \simeq 0$): In this case, there is neither a preferred orientation of signal variation nor significant variation, which corresponds to homogeneous regions.

From a statistical point of view, when the signals are regarded as random fields an interesting relationship can be established between the local structure tensor and the covariance matrix $\mathbf{C}$. In fact, when the expectation per element is zero (usually on homogeneous areas such as planes or lines) the local structure tensor and covariance matrix are identical. The equivalence is given by the following formulation:

$$\mathbf{C} = \mathbf{T} - E\{\nabla s\}E\{(\nabla s)^T\} \tag{41}$$

where $\nabla s$ represents the spatial gradient of the signal $s$ and $E\{\cdot\}$ is the expectation operator.

## *6.2 Estimation of the Local Structure Tensor*

In this section we face the problem of local structure estimation from multidimensional images using the tensor representation described in the previous section. As we said before, local structure is related to local orientation and to the most relevant features of an image. As a consequence of this fact, estimation of local structure has been a matter of discussion in literature for years. The works [47], [48] and [49] in 1987 together with [44] in 1989 represent the seed of the tensor representation of local structure. These methods can be classified, as done in [50], in two different groups:

- Gradient method: These methods use the spatial gradient to determine the signal orientation. Among others, the following references are of interest [47, 48, 49, 51, 52].
- Local-energy method: These methods involve the use of quadrature filters to quantify the energy of the signal and, hence, to be able to determine local structure. Among others, the following references are of interest [44, 53, 54, 55]

### 6.2.1 Gradient Method

The computation of the local structure tensor components are given by the following formulation:

$$\mathbf{T}(\mathbf{x_0}) = \{t_{ij}\} = \frac{1}{4\pi^2} \int_{\mathbf{x} \in \Omega(\mathbf{x_0})} \frac{\partial s}{\partial x_i} \frac{\partial s}{\partial y_j} d\mathbf{x} \tag{42}$$

where $\Omega(\mathbf{x_0})$ is a local neighborhood of the function $s(\mathbf{x})$ around the point $\mathbf{x_0}$ where the local structure tensor is going to be estimated. In practice, where we assume we are working on a discrete grid, Eq. (42) can be written using the following expression:

$$\mathbf{T}(\mathbf{x_0}) = G_\sigma(\mathbf{x_0}) * (\nabla s(\mathbf{x_0}) \nabla^T s(\mathbf{x_0})) \tag{43}$$

where $G_\sigma(\mathbf{x_0})$ is a Gaussian kernel with standard deviation $\sigma$, used to regularize the outer product of the gradient introducing the information of the local neighborhood at a given scale $\sigma$ in order to obtain a full rank tensor. Otherwise, the outer product of the gradient itself would not incorporate the uncertainty information since it is a rank-1 tensor. The partial derivatives in the discrete domain are computed with Gaussian derivative filters.

An analogous formulation to Eq. (43) can be obtained by means of a probabilistic interpretation of the signal $s(\mathbf{x})$ [56]. In other words, we can think of the signal $s(\mathbf{x})$ as a field where a random variable is assigned to each position $\mathbf{x} \in \mathbb{R}^n$. Then, to characterize the variation of the signal we make use of the second order joint statistics of the signal gradient. Assuming a zero mean gradient, an estimate of the covariance matrix of the signal gradient is given by Eq. (44), where $V(\mathscr{N}(\mathbf{x_0}))$ stands for the number of samples of the neighborhood $\mathscr{N}(\mathbf{x_0})$ centered at point $\mathbf{x_0}$:

$$\mathbf{C}_{\nabla s}(\mathbf{x_0}) = \frac{1}{V(\mathscr{N}(\mathbf{x_0}))} \sum_{\mathbf{x_i} \in \mathscr{N}(\mathbf{x_0})} \nabla s(\mathbf{x_i}) \nabla s(\mathbf{x_i})^T \tag{44}$$

The elements of the covariance matrix, i.e. local structure tensor, are estimates of the second order cross moments of the signal gradient. Thus, an analysis of the eigenvalues and eigenvectors provide information on the manner in which the gradient changes. In brief, the eigendecomposition helps to discriminate the type local neighborhood among the different cases already presented.

Both formulations are then coherent with the properties that any descriptor of orientation should hold, which were previously outlined in this section. In addition, it is clear that Eqs. (43) and (44) are particular formulations of the proposed mapping $\mathbf{M}$.

### 6.2.2 Local Energy Method

This method, mainly developed by Knutsson [44], involves the use of quadrature filters to quantify the local energy of the signal to infer the local structure, based on the relationship between these two concepts described in [43]. However, local energy itself is not enough to completely describe local structure since it does not provide orientation information, as it is a scalar measure. Then, the estimation of the local structure tensor requires some previous steps.

Using basic theory of tensor algebra, it is known that any tensor $\mathbf{T}(\mathbf{x})$ can be expressed as a linear combination of the elements of a tensor basis $\{\mathbf{M}_k\}$ weighted by the coefficients $q_k(\mathbf{x})$, where $k = 1,\ldots,K$ is the number of elements in the basis of tensors:

$$\mathbf{T}(\mathbf{x}) = \sum_{k=1}^{K} q_k(\mathbf{x})\mathbf{M}_k \tag{45}$$

Knutsson, then, proposed the way to compute the tensor basis $\{\mathbf{M}_k\}$ and the coefficients $q_k(\mathbf{x})$ using a bank of $K$ spherically separable quadrature filters oriented along a given set of directions $\hat{\mathbf{n}}_k$ symmetrically distributed in the $n$-dimensional space of the signal $s(\mathbf{x})$ [44]. The set of quadrature filters is defined in the Fourier domain by:

$$\mathscr{Q}_k(\mathbf{u}) = \begin{cases} e^{\left(\frac{-4}{B^2 ln2}\right)\left(ln^2\left(\frac{\|\mathbf{u}\|}{\|\mathbf{u}_0\|}\right)\right)} \cdot (\hat{\mathbf{u}}^T\hat{\mathbf{n}}_k)^2 & \text{if } \hat{\mathbf{u}}^T\hat{\mathbf{n}}_k > 0 \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

These filters can be seen as oriented Gaussian functions in a logarithmic scale, centered at $\| \mathbf{u}_0 \|$ and with bandwidth $B$.

Assuming a signal model that presents only one main orientation, each coefficient $q_k(\mathbf{x})$ (with $k = 1,\ldots,K$) is given by the magnitude of the complex function given by the convolution of the signal with the quadrature filter, which is more easily computed in the Fourier domain:

$$q_k(\mathbf{x}) = \| \frac{1}{(2\pi)^n} \int_{\Omega} \mathscr{G}(\hat{\mathbf{n}}^T\mathbf{u})\delta^{line}_{\hat{\mathbf{n}}}(\mathbf{u})\mathscr{Q}_k(\mathbf{u})e^{i\mathbf{u}^T\mathbf{x}}d\mathbf{u} \| \tag{47}$$

Concerning the tensor basis, the set $\{\mathbf{M}_k\}$ is defined as the dual or reciprocal basis of the tensor basis $\{\mathbf{N}_k\}$ given by the outer product of the set of directions $\hat{\mathbf{n}}_k$, as it shown in the following expression:

$$\mathbf{N}_k = \hat{\mathbf{n}}_k\hat{\mathbf{n}}_k^T \tag{48}$$

Since the set $\{\mathbf{N}_k\}$ form a basis of tensors which may not be orthonormal, we have to use its dual basis $\{\mathbf{M}_k\}$ in order to get a proper representation for the tensor $\mathbf{T}(\mathbf{x})$.

Finally, the $K$ elements of the basis still have to be determined. The number $K$ is related to the number of directions $\hat{\mathbf{n}}_k$ needed to orientate each quadrature filter in order to compute the local energy over all the orientation space of the signal, i.e. $\mathbb{R}^n$ in an $n$-dimensional space. A practical rule is that the directions $\hat{\mathbf{n}}_k$ should pass through the adjacent vertexes of a regular polytope (a hexagon in 2D, an icosahedron in 3D, etc.). Then, in an $n$-dimensional scalar dataset, the minimum number of directions required for the quadrature filters is $K = \frac{n(n+1)}{2}$ [57]. This requirement is based on the fact that $\mathbf{T}(\mathbf{x})$ by definition is invariant to rotation of the filters [50].

A common practice is to regularize the final result with a Gaussian kernel with standard deviation $\sigma$ to provide a more robust response against noise. Although this regularization is not needed in a strict sense, it also introduces an integration scale

to provide a robust response. Hence, the final expression of the local structure tensor is as follows:

$$\mathbf{T}(\mathbf{x}) = G_\sigma(\mathbf{x}) * \sum_{k=1}^{\frac{n(n+1)}{2}} \parallel s(\mathbf{x}) * h_k(\mathbf{x}) \parallel \mathbf{M}_k \tag{49}$$

where $h_k(\mathbf{x})$ is the spatial formulation of the quadrature filter in Eq. (46).

## *6.3 Nonlinear Structure Tensors*

In the previous section we described two different methods to estimate the local structure tensor. Although the philosophy behind them is completely different, in both cases it is a common practice to regularize the tensor field by means of a Gaussian convolution. In this way, we incorporate spatial information within a certain neighborhood at the desired integration scale and, at the same time, the estimation is more robust against noise.

However, it is known that Gaussian smoothing has two main drawbacks: the blurring effect and delocalization of main structures. This is mainly due to the use of a fixed kernel that does not depend on the local characteristics of the signal. Thus, near boundaries, where the structure information is mainly present, this will yield an inaccurate result since we are mixing different kinds of information.

To avoid these two effects, Brox et al. proposed the definition of the *nonlinear structure tensor* [58]. The main idea is to substitute the Gaussian convolution by a generalized nonlinear matrix-valued heat diffusion equation [59]. As it is done for the scalar case, a diffusivity function $g(\sum_{i,j=1}^{m} \parallel \nabla t_{ij}(\mathbf{x}) \parallel)$ is introduced in the heat equation to correlate the amount of smoothing with the magnitude of the gradient of all tensor components, as it is shown in the following equation:

$$\partial_t t_{ij}(\mathbf{x}) = div(g(\sum_{i,j=1}^{m} \parallel \nabla t_{ij}(\mathbf{x}) \parallel^2) \nabla t_{ij}(\mathbf{x})), \quad t_{ij}(\mathbf{x}, t=0) = t_{ij}(\mathbf{x}) \tag{50}$$

In this way, when the gradient in any tensor component is high due to the presence of structure, the diffusivity is low to avoid smoothing while the diffusivity is high when there is no structure in any component. As a consequence, this technique is different from just applying nonlinear diffusion independently to every component of the tensor field, since a boundary in one component reduces the amount of smoothing in all the others.

There exists another natural extension for the linear structure tensor which consists on the use of an anisotropic matrix-valued diffusion process also proposed in [59]. In the anisotropic case, not only the amount of diffusion is adapted locally to the data, but also the direction of smoothing. Hence, in presence of boundaries, the smoothing will take place along the edges of the boundary but not across them. To do that, we only have to replace the diffusivity function defined in Eq. (50) by a dif-

fusion tensor $\mathbf{G}(\mathbf{x})$ that determines the direction in which the diffusion takes place. The anisotropic structure tensor is given by the following expression:

$$\partial_t t_{ij}(\mathbf{x}) = div(\mathbf{G}(\sum_{i,j=1}^{m}(\nabla t_{ij}(\mathbf{x}))^T(\nabla t_{ij}(\mathbf{x})))\nabla t_{ij}(\mathbf{x})), \quad t_{ij}(\mathbf{x},t=0) = t_{ij}(\mathbf{x}) \tag{51}$$

Concerning the positive semidefiniteness, it can be easily shown that tensors evolving under either Eq. (50) or Eq. (51) stay symmetric positive definite if the initial value $\mathbf{T}(0) = (t(0)_{i,j})$ is symmetric positive definite [58].

## *6.4 Diffusion Filters*

A direct usage of the structure tensors computed based on the data is to use them in diffusion filters. Diffusion filtering is a PDE based image processing technique[1] that is based on the anisotropic diffusion equation. The underliying idea is to interpret the image as an abstract concentration map to which the diffusion is applied. It is directly related to oriented smoothing where the orientation of the smoothing kernel is based on the estimated structure tensor [2]. Diffusion filters are directly related to variational image processing techniques. We would refer the reader to [60] for more details. In this section, we will briefly review the diffusion filters that use the structure tensor explicitly.

Let us start by recalling *the second order symmetric diffusion tensor* $\mathbf{D}$. The 1D diffusion PDE establishes a linear relation between the temporal variation and the second order spatial derivative of the concentration. It namely states that $u_t = \nabla \cdot D\nabla u = Du_{xx}$. We can move onto 2D anisotropic diffusion by defining $D_a$ to be the 1D diffusion along the direction given by the unit vector $\mathbf{a}$ and $D_b$ to be the diffusion coefficient along the unit vector $\mathbf{b}$, where $\mathbf{a} \perp \mathbf{b}$. Then we get,

$$\begin{aligned}
u_t &= D_a u_{aa} + D_b u_{bb} && (52)\\
&= D_a(\mathbf{a}^T\mathbf{H}_u\mathbf{a}) + D_b(\mathbf{b}^T\mathbf{H}_u\mathbf{b}) && (53)\\
&= D_a(\mathbf{a}^T\begin{pmatrix}\partial_{x_1}\\ \partial_{x_2}\end{pmatrix}\nabla u^T\mathbf{a}) + D_b(\mathbf{b}^T\begin{pmatrix}\partial_{x_1}\\ \partial_{x_2}\end{pmatrix}\nabla u^T\mathbf{b}) && (54)\\
&= D_a(\nabla^T\mathbf{a}\mathbf{a}^T\nabla u) + D_b(\nabla^T\mathbf{b}\mathbf{b}^T\nabla u) && (55)\\
&= \nabla^T(D_a\mathbf{a}\mathbf{a}^T + D_b\mathbf{b}\mathbf{b}^T)\nabla u && (56)\\
&= \nabla\cdot\mathbf{D}\nabla u = \nabla\cdot\begin{bmatrix}\mathbf{a} & \mathbf{b}\end{bmatrix}\blacksquare\begin{bmatrix}\mathbf{a}^T\\ \mathbf{b}^T\end{bmatrix}\nabla u && (57)
\end{aligned}$$

[1] Although we are mostly concerned with image processing applications here, diffusion filtering is applicable to a wider class of signals.

[2] It should be noted that this framework is not only used for smoothing but it can also be applied for enhancing operations, as in the shock filters.

So the eigenvectors of the diffusion tensor show the principal diffusion directions and the corresponding eigenvalues are the diffusion coefficients along these directions.

The main strategy is to perform more diffusion (smoothing) along the edges and less (or even negative) diffusion across the edges. We can construct a diffusion tensor that would serve to this purpose, i.e. that would have two eigenvectors $\mathbf{e}_1, \mathbf{e}_2$ s.t. $\mathbf{e}_1 \parallel \nabla u$ and $\mathbf{e}_2 \perp \nabla u$. This corresponds to using the tensor $\mathbf{D} = \nabla u \nabla u^T$. Its eigenvalues are $\lambda_1 = ||\nabla u||^2$ and $\lambda_2 = 0$. It is an extremely local measure of orientation. If we use it directly, we will be using nothing more than the linear isotropic diffusion PDE (to see this, note that $\mathbf{e}_2 \cdot \nabla u = 0$). However, if we gather information from the neighbourhood, and use it to determine a structure tensor $\mathbf{D}$, i.e. the diffusion directions $\mathbf{e}_1$ and $\mathbf{e}_2$, then we can achieve edge enhancing/preserving smoothing. The desired diffusion (filtering) is achieved by forming a diffusion tensor derived from such a *structure tensor* by manipulating its eigenvalues only.

Weickert [61] proposed to use the following smoothed tensor as the structure tensor. He used it to determine the diffusion directions and proposed a set of filters by suggesting different choices of eigenvalue manipulations. Weickert's structure tensor is

$$u_\sigma = k_\sigma * u \Rightarrow \mathbf{D} = k_\rho * \nabla u_\sigma \nabla u_\sigma^T \tag{58}$$

$$= \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \end{bmatrix} \tag{59}$$

where $k_\sigma$ is a Gaussian smoothing kernel. $\mathbf{e}_1$ shows the maximum variation direction (given that $\lambda_1 > \lambda_2$) and $\mathbf{e}_2$ gives the minimum variation direction. Also note that now $\mathbf{e}_2 \cdot \nabla u \neq 0$. The plan is to change $\lambda_1$ and $\lambda_2$ to fit to our needs.

*edge enhancing filter:*

The approach is to reduce diffusion across edges (along maximum variation direction given by $\mathbf{e}_1$). Weickert proposed ($\lambda_1 \geq \lambda_2$),

$$\hat{\lambda}_1 = \begin{cases} 1 & , \lambda_1 = 0 \\ 1 - exp(\frac{-3.315}{\lambda_1^4}) & , o.w. \end{cases} \tag{60}$$

$$\hat{\lambda}_2 = 1 \tag{61}$$

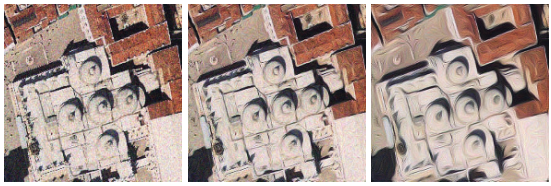
*Coherence Enhancing Filter:*

A measure of coherence is $(\lambda_1 - \lambda_2)^2$ and the maximum coherence direction is given by $\mathbf{e}_2$, i.e. the minimum variation direction. The coherence enhancing filter enhances flowlike structures, closes interrupted lines. The strategy is to diffuse along the coherence direction with a diffusion coefficient increasing with the coherence. Weickert proposed [62],

$$\hat{\lambda}_1 = \alpha \tag{62}$$

$$\hat{\lambda}_2 = \begin{cases} \alpha & ,\lambda_1 = \lambda_2 \\ \alpha + (1-\alpha) exp(\frac{-c}{(\lambda_1 - \lambda_2)^2}) & ,o.w. \end{cases} \tag{63}$$

where $\alpha \in (0,1)$, the chosen isotropic diffusion coefficient (lower bound for diffusion). Figure 7 shows a noisy image and its coherence enhancing filtered versions. Note the change in texture as a result of filtering. Coherence enhancing diffusion filtering identifies the principal edge directions in the image and perform smoothing along these directions selectively throughout the image.



(a) (b) (c)

**Fig. 7** (a) A noisy aerial photograph; the image is filtered using coherence enhancing diffusion filter. (b) Result after a single iteration of diffusion. (c) Output generated after eight iterations; note that the filter has identified and enhanced the coherence directions locally throughout the image.

## 7 Conclusion

Tensors are mathematical entities that are generalizations of more common entities, such as scalars (zero order tensors), vectors (first order tensors) and matrices (second order tensors). As such, tensor analysis provides us a unified framework for the analysis of them all. Such a unified framework makes it possible to generalize the techniques developed for those more common entities. It is of utmost importance to realize that the tensors are not pure abstract entities but unavoidably exist in real world data, such as digital images, elasticity of materials, Brownian motion of particles, etc. We believe that the researchers from a wide spectrum of fields would benefit greatly from the recognition of tensors in their own data and the adaptation of techniques developed in some seemingly unrelated field using the tensor analysis tools.

This review was aimed at introducing the tensors as mathematical entities and stimulating the readers' interest in tensor analysis by highlighting some of the most common fields of research where the tensor analysis is used. In most cases, it is a matter of seeing the tensors in your data, not a matter of their existence.

## References


1. M. Raffel, C. Willert, and J. Kompenhans, *Particle Image Velocimetry. A Practical Guide*, Springer Verlag, 1998.
2. J.C.R. Hunt, "Vorticity and vortex dynamics in complex turbulent flows," in *Transactions Canadian Society for Mechanical Engineering (ISSN 0315-8977)*, 1987, vol. 11, pp. 21–35.
3. R. Haimes and D. Kenwright, "The velocity gradient tensor and fluid feature extraction," in *Proc. AIAA 14th Computational Fluid Dynamics Conference*, 1999.
4. P. Basser, J. Mattiello, and D. Le Bihan, "MR diffusion tensor spectroscopy and imaging," *Biophysical Journal*, vol. 66, pp. 259–267, 1994.
5. P.C. Sundgren, Q. Dong, D. Gómez-Hassan, S.K. Mukherji, P. Maly, and Welsh R., "Diffusion tensor imaging of the brain: review of clinical applications," *Neuroradiology*, vol. 46, pp. 339–350, 2004.
6. A. Einsten, "Ber die von der molekularkinetischen theorie der wärme gefordete bewegung von in ruhenden flüssigkeiten suspendierten teilchen," *Annalen der Physik*, , vol. 17, pp. 549–560, 1905.
7. E.O. Stejskal and T. E. Tanner, "Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient," *Journal of Chemical Physics*, no. 42, pp. 288–292, 1965.
8. D LeBihan, E. Breton, D. Lallemand, P. Grenier, E. Cabanis, and M. Laval-Jeantet, "MR imaging of intravocel incoherents motions: application to diffusion and perfusion in neurological disorders," *Radiology*, vol. 161, pp. 401–407, 1986.
9. C. Pierpaoli and P.J. Basser, "Toward a quantitative assesment of diffusion anisotropy," *Magnetic Resonance in Medicine*, vol. 36.
10. C.F. Westin, S. E. Maier, H. Mamata, F.A. Jolesz, and R. Kikinis, "Processing and visualization for diffusion tensor MRI," *Medical Image Analysis*, vol. 6, pp. 93–108, 2002.
11. J. Weickert and H. Hagen, Eds., *Visualization and Processing of Tensor Fields, part II. Diffusion Tensor Imaging*, pp. 81–187, Springer, 2006.
12. Kindlmann G., "Superquadrics tensor glyphs," in *Proceedings IEEE TVCG/EG Symposium on Visualization, 2004*, May 2004.
13. C.R. Rao, "Information and accuracy attainable in the estimation of statistical parameters," *Bull. Calcutta Math. Soc.*, vol. 37, pp. 81–91, 1945.
14. J. Burbea and C.R. Rao, "Entropy differential metric, distance and divergence measures in probability spaces: A unified approach," *J. Multivariate Anal.*, vol. 12, pp. 575–596, 1982.
15. L.T. Skovgaard, "A Riemannian geometry of the multivariate normal model," Tech. Rep. 81/3, Statistical Research Unit, Danish Medical Research Council, Danish Social Science Research Council, 1981.
16. M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, 2005.
17. C. Atkinson and A.F.S. Mitchell, "Rao's distance measure," *Sankhya: The Indian Journal of Stats.*, vol. 43, no. A, pp. 345–365, 1981.
18. W. Förstner and B. Moonen, "A metric for covariance matrices," Tech. Rep., Stuttgart University, Dept. of Geodesy and Geoinformatics, 1999.
19. D.F. Scollan, A. Holmes, R. Winslow, and J. Forder, "Histological validation of myocardial microstructure obtainde form diffusion tensor magnetic resonance imaging," *American Journal of Physiology*, no. 275, pp. 2308–2318, 1998.
20. E. W. Hsu and Setton L. A., "Diffusion tensor microscopy of the intervertebral disc annulus fibrosus," *Magnetic Resonance in Medicine*, no. 41, pp. 992–999, 1999.
21. P. Hagmann, L. Jonasson, P. Maeder, J. Thiran, V. Wedeen, and R. Meuli, ,"Understanding diffusion MRI imaging techniques: from scalar diffusion-weighted imaging to diffusion tensor imaging and beyond," *Radiographics*, pp. S205–S223.
22. H. Jiang, P.C. Van Zijl, J. Kim, G.D. Pearlson, and S. Mori, "DTIstudio: Resource program for diffusion tensor computation and fiber bundle tracking," *Comput. Methods Programs Biomed.*, vol. 81, no. 2, pp. 106–116, 2006.

23. S. Mori and P.C.M. van Zijl, "Fiber tracking: principles and strategies -a technical review," *NMR in Biomedicine*, vol. 15, no. 7-8, pp. 468–480, 2002.
24. T. E. Conturo, N. F. Lori, T. S. Cull, E. Akbudak, A. Z. Snyder, J. S. Shimony, R. C. Mckinstry, H. Burton, and M. E. Raichle, "Tracking neuronal fiber pathways in the living human brain," in *Proc. Natl. Acad. Sci. USA*, August 1999, pp. 10422–10427.
25. P.J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi, "In vivo fiber tractography using DT-MRI data," *Mag. Res. in Med.*, vol. 44, pp. 625–632, 2000.
26. M. Lazar, D.M.Weinstein, J.S.Tsuruda, K.M.Hasan, K. Arfanakis, M.E. Meyerand, B. Badie, H.A.Rowley, V. Haughton, A. Field, and A. L.Alexander, "White matter tractography using diffusion tensor deflection," *Human Brain Mapping*, vol. 18, pp. 306–321, 2003.
27. P. Hagmann, J.-P. Thiran, L. Jonasson, P. Vandergheynst, S. Clarke, P. Maeder, and R. Meulib, "DTI mapping of human brain connectivity: Statistical fibre tracking and virtual dissection," *NeuroImage*, vol. 19, pp. 545–554, 2003.
28. S. Mori, B.J. Crain, V.P. Chacko, and P.C. van Zijl., "Three dimensional tracking of axonal projections in th brain by magnetic resonance imaging," *Ann. Neurol.*, vol. 45, no. 2, pp. 265–269, 1999.
29. C. Pierpaoli, P. Jezzard, PJ. Basser, A. Barnett, and G. Di Chiro, "Diffusion tensor MR imaging of the human brain," *Radiology*, vol. 201.
30. E. von dem Hagen and R. Henkelman, "Orientational diffusion reflects fiber structure within a voxel," *Magn. Reson. Med.*, vol. 48.
31. D. Jones, "Determining and visualizing uncertainty in estimates of fiber orientation from diffusion tensor MRI," *Magn. Reson. Med.*, vol. 49.
32. D. Jones, A. Simmons, S. Williams, and M. Horsfield, ," .
33. S. Mori, B. Crain, V. Chacko, and P. Van Zijl, "Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging," *Ann. Neurol.*, vol. 45.
34. C. Tench, P. Morgan, M. Wilson, and L. Blumhardt, "White matter mapping using diffusion tensor MRI," *Magn. Reson. Med.*, vol. 47.
35. J. Ophir, I. Céspedes, B. Garra, H. Ponnekanti, Y. Huang, and N. Maklad, "Elastography: A quantitative method for imaging the elasticity of biological tissues," *Ultrasound Imaging*, vol. 13, pp. 111-134, 1991.
36. B.S. Garra, I. Céspedes, J. Ophir, S. Spratt, R. A. Zuurbier, C. M. Magnant, and M. F. Pennanen, "Elastography of breast lesions: initial clinical results," *Radiology*, vol. 202, pp. 79-86, 1997.
37. R. L Maurice, M. Daronat, J. Ohayon, E. Stoyanova1, F. S. Foster, and G. Cloutier, "Non-invasive high-frequency vascular ultrasound elastography," *Phys. Med. Biol.*, no. 50, pp. 1611–1628, 2005.
38. D. Sosa-Cabrera, M.A. Rodriguez-Florido, E. Suarez-Santana, and J. Ruiz-Alzola, *Tensor Elastography: A New Approach for Visualizing the Elastic Properties of the Tissue*, 2006.
39. A. Neeman, B. Jeremic, and A. Pang, "Visualizing tensor fields in geomechanics," in *IEEE Visualization*, 2005, pp. 329–343.
40. B. Wuensche, "The visualization of 3d stress and strain tensor fields," in *Proc. of the 3rd New Zealand Comp. Science Research Student Conf.*, Apr. 1999, vol. 3, pp. 109–116.
41. P. Selskog, E. Heiberg, T. Ebbers, L. Wigstrom, and M. Karlsson, "Kinematics of the heart: strain-rate imaging from time-resolved three-dimensional phase contrast MRI," *IEEE Trans. Med. Imaging*, vol. 21, no. 1, pp. 1105-1109, 2002.
42. D. Sosa-Cabrera, *Novel Processing Schemes and Visualization Methods for Elasticity Imaging*, Phd dissertation, University of Las Palmas de GC, 2008.
43. M.C. Morrone and R.A. Owens, "Feature detection from local energy," *Pattern Recognition Letters*, vol. 6, pp. 303–313, 1987.
44. H. Knutsson, "Representing local structure using tensors," in *6th Scandinavian Conference on Image Analysis. Oulu, Finland*, 1989, pp. 244–251.
45. U. Köthe, "Integrated edge and junction detection with the boundary tensor," in *9th Intl. Conf. on Computer Vision*, Nice, 2003, IEEE Computer Society, vol. 1, pp. 424–431.

46. C.-F. Westin, S.E. Maier, H. Mamata, A. Nabavi, F.-A. Jolesz, and R. Kikinis, "Processing and visualization for diffusion tensor MRI," *Medical Image Analysis*, vol. 6, no. 2, pp. 93–108, Jun. 2002.
47. J. Bigün and G.-H. Granlund, "Optimal orientation detection of linear symmetry.," in *IEEE First International Conference on Computer Vision*, London, UK, Jun. 1987, pp. 433–438.
48. W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features.," in *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, Interlaken, Switzerland, June 1987, pp. 281–305.
49. M. Kass and A. Witkin, "Analyzing oriented patterns," *Computer Vision Graphics and Image Processing*, vol. 37, pp. 362–385, 1987.
50. R. San Jose, *Local Structure Tensor for Multidimensional Signal Processing. Applications to Medical Image Analysis*, Ph.D. thesis, Universidad de Valladolid, E.T.S. Ingenieros de Telecomunicación, February 2005.
51. A.-R. Rao and B.-G. Schunck, "Computing oriented texture fields", *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 2, pp. 157–185, 1991.
52. R. Mester and M. Mühlich, "Improving motion and orientation estimation using an equilibrated total least squares approach," in *Proc. IEEE International Conference on Image Processing (ICIP 2001)*, Thessaloniki, Greece, October 2001, pp. 640–643.
53. H. Knutsson and M. Andersson, "What's so good about quadrature filters?", in *2003 IEEE International Conference on Image Processing*, 2003.
54. H. Knutsson and M. Andersson, "Loglets: Generalized quadrature and phase for local spatiotemporal structure estimation," in *Proc. of SCIA03. LNCS.*, J. Bigün and T. Gustavsson, Eds., 2003, vol. 2749, pp. 741–748, Springer.
55. B. Rieger and L. Vlieet, "A systematic approach to nd orientation representation," *Image and Vision Computing*, vol. 22, no. 6, pp. 453–459, 2004.
56. J. Ruiz-Alzola, R. Kikinis, and C.-F. Westin, "Detection of point landmarks in multidimensional tensor data," *Signal Processing*, vol. 81, no. 10, Oct. 2001.
57. G.-H. Granlund and H. Knutsson, *Signal Processing for Computer Vision*, Kluwer Academic Publishers, 1995.
58. T. Brox, J. Weickert, B. Burgeth, and P. Mrázek, "Nonlinear structure tensors," *Image and Vision Computing*, vol. 24, no. 1, pp. 41–55, January 2006.
59. J. Weickert and T. Brox, "Diffusion and regularization of vector- and matrix-valued images," *Inverse Problems, Image Analysis and Medical Imaging. Contemporary Mathematics*, vol. 313, pp. 251–268, 2002.
60. G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing*, Applied Mathematical Sciences, vol. 147. Springer-Verlang, 2002.
61. J. Weickert, *A Review of Nonlinear Diffusion Filtering*, vol. 1252 of *Lecture Notes in Computer Science - Scale Space Theory in Computer Science*, Springer, Berlin, 1997.
62. J. Weickert, "Coherence enhancing diffusion filtering," *Int. J. of Computer Vision*, vol. 31, pp. 111–127, 1999.

# Part I
# Tensors and Tensor Field Processing

# Segmentation of Tensor Fields: Recent Advances and Perspectives

Rodrigo de Luis-García, Carlos Alberola-López and Carl-Fredrik Westin

**Abstract** The segmentation of tensor-valued images or 3D volumes is a relatively recent issue in image processing, but a significant effort has been made in the last years. Most of this effort has been focused on the segmentation of anatomical structures from DT-MRI (Diffusion Tensor Magnetic Resonance Imaging), and some contributions have also been made for the segmentation of 2D textured images using the Local Structure Tensor (LST). In this chapter, we carefully review the state of the art in the segmentation of tensor fields. We will discuss the main approaches that have been proposed in the literature, with particular emphasis on the importance of the different tensor dissimilarity measures. Also, we will highlight the key limitations of the segmentation techniques proposed so far, and will provide some insight on the directions of current research.

## 1 Introduction

Tensor fields are an extension of scalar or vector fields, and their usage in image processing is becoming more and more common, having also been employed in different areas such as differential geometry, physics, mechanics, chemistry, crystallophysics, engineering and other sciences. Usually, second-order tensors are employed for the description of anisotropic behaviour [81].

Rodrigo de Luis-García
Laboratory of Mathematics in Imaging, Harvard Medical School, Boston, MA, USA, e-mail: rodrigo@bwh.harvard.edu

Carlos Alberola-López
Laboratorio de Procesado de Imagen, Universidad de Valladolid, Valladolid, Spain, e-mail: caralb@tel.uva.es

Carl-Fredrik Westin
Laboratory of Mathematics in Imaging, Harvard Medical School, Boston, MA, USA, e-mail: westin@bwh.harvard.edu

In the area of image processing, and particularly within the field of medical image processing, the use of tensors has gained much relevance lately. Most of this interest is due to the appearance of a relatively new medical imaging modality, the DT-MRI (also referred to as DTI, *Diffusion Tensor Imaging*) [11], and the use of tensor descriptors for feature extraction, as in the case of the LST (*Local Structure Tensor*) [41, 42, 35, 13, 27]. These two applications deserve special consideration as they are central elements of this chapter.

When dealing with tensor-valued images, such as in the case of the LST or DTI, traditional techniques like filtering, registration or segmentation have been revisited in order to take into account the special properties of this data modality. This chapter reviews the state of the art in the segmentation of tensor-valued images. To do so, we will follow the course of the main contributions in the literature, and the chapter is organized as follows: in next section, we introduce the two main tensor modalities employed in image processing, that is, the local structure tensor and the diffusion tensor. We then review the most important scalar parameters that have been employed in the analyis of tensor images prior to the proposal of truly tensor processing of this kind of data. As tensor processing heavily relies on the choice of a suitable tensor metric, in Section 4 we present the main tensor dissimilarity measures that have been proposed for segmentation purposes. Section 5 carefully reviews the different methods in the literature for the segmentation of tensor-valued data, and we finally summarize in Section 6 with a special emphasis in the main limitations of the segmentation methods proposed so far and the future trends in tensor segmentation.

## 2 Tensors in Image Processing

### *2.1 Local Structure Tensor*

Texture is one of the most important features in images, and therefore its consideration can greatly improve their analysis and understanding. Besides, this feature space is deeply connected with important biological vision properties [52]. The representation and modeling of textures is a difficult and open issue, and different approaches have been proposed over the last decades [24, 74].

Local orientation is a major component of textures in images, and therefore can be used as feature space. The tensor representation for local orientation in multidimensional signals [41, 42, 35, 13, 27], known as the LST, is widely accepted to provide a compact representation of orientation, and has been used for motion analysis besides texture representation [13]. This representation yields a symmetric and positive semidefinite second order tensor that encodes the local orientation.

For a scalar 2D image $I$, the LST is defined as follows [35, 13, 27, 32, 50, 62]:

$$\mathbf{T}_\rho = K_\rho * (\nabla I \nabla I^T) = \begin{pmatrix} K_\rho * I_x^2 & K_\rho * I_x I_y \\ K_\rho * I_x I_y & K_\rho * I_y^2 \end{pmatrix} \quad (1)$$

where $K_\rho$ is a Gaussian kernel with standard deviation $\rho$, and subscripts denote partial derivatives. The tensor yields three feature channels ($K_\rho * I_x^2$, $K_\rho * I_y^2$ and $K_\rho * I_x I_y$) which, for a fixed scale, may be as powerful as a whole set of Gabor filters for the discrimination of different textures [65]. In the case of a vector-valued (i.e. colour) image, all channels are taken into account by summing the tensor products of the particular channels [88]:

$$\mathbf{T}_\rho = K_\rho * \left( \sum_{i=1}^{N} \nabla I_i \nabla I_i^T \right) \tag{2}$$

This definition of the LST is appropriate for any number of dimensions, and can be extended in a straightforward manner. As it was shown in [80, 64], smoothing with a Gaussian kernel causes blurring of each of the tensor components, which may make the structure tensor suffer from the dislocation of edges, leading to inaccurate segmentation results near region boundaries. Indeed, the convolution with a Gaussian kernel is equivalent to the linear matrix-valued diffusion process

$$\partial_t u_{ij} = \mathrm{div}(\nabla u_{ij}) \; \forall i \tag{3}$$

where $u_{ij}(t=0)$ are the components resulting from the tensor product $\nabla I \nabla I^T$. In order to solve the limitations of the classic linear structure tensor, Brox and Weickert [17, 18] proposed to replace the Gaussian smoothing (i.e. linear diffusion) by nonlinear diffusion. Nonlinear diffusion was introduced by Perona and Malik [59], and aims at reducing the smoothing in the presence of edges. The resulting diffusion equation is, for the scalar case:

$$\partial_t u = \mathrm{div}(g(|\nabla u|)\nabla u) \tag{4}$$

with $u(t=0)$ being the image $I$ and $g(\cdot)$ a decreasing function. This diffusion equation is only suitable for scalar-valued data. The version for vector-valued data was introduced by Gerig *et al.* in [31], while the nonlinear matrix-valued process is [80, 18]:

$$\partial_t u_{ij} = \mathrm{div}\left( g\left( \sum_{kl} |\nabla u_{kl}|^2 \right) \nabla u_{ij} \right) \; \forall i, j \tag{5}$$

where $u_{ij}(t=0)$ are the initial tensor product components. It is worth noticing that, for the equation above, the diffusivity $g(\cdot)$ is a scalar, and thus the diffusion process is isotropic. A very interesting property of this matrix-valued diffusion equation relies on its preservation of positive semidefiniteness [80, 18]. For implementation, the AOS (*additive operator splitting*) scheme, proposed in [82], allows for a much more efficient and faster computation than a straightforward implementation scheme, and is consequently preferred.

## 2.2 Diffusion Tensor Magnetic Resonance Imaging

DT-MRI is a medical imaging modality based on *Magnetic Resonance Imaging* (MRI) which is able to quantify the anisotropic diffusion of water molecules in biological tissues [14]. This property is usually applied to visualize highly structured tissues. Currently, brain imaging is the most common application of diffusion MRI, as the brain has a complex structure of gray matter areas connected by white matter fibres. Diffusion MRI can be therefore employed for the visualization of the fibre tracts in the white matter of the brain, and has been applied in Neurology and Neurosurgery (see [71, 34] for a comprehensive introduction to the applications of DT-MRI to brain diseases).

Water is a major component of biological tissue. Due to Brownian motion, water molecules experiment random motion within the tissue. The diffusion is then measured as the PDF $p$ of particle displacements $\mathbf{x}$ over a fixed time $t$. To measure diffusion from *Diffusion Weighted Images* (DWI), the Stejskal-Tanner imaging sequence is employed [70], which allows for a controlled diffusion weighting. If, for instance, a measurement is done without diffusion weighting and one is done with diffusion weighting in a certain direction, then diffusion can be calculated using the Stejskal-Tanner equation:

$$S = S_0 e^{-bD} \tag{6}$$

where $S$ is the measured diffusion weighted image, $S_0$ is the baseline image (i.e. with no diffusion weighting), $b$ is the diffusion weighting factor, introduced by Le Bihan *et al.* [14]. The diffusion values, $D$, are also known as *Apparent Diffusion Coefficients* (ADC). It is worth emphasizing that the diffusion values generated depend on the direction of the sensitizing gradient and other parameters [85].

In the more general case of considering anisotropic diffusion, Eq. 6 can be rewritten in the following way:

$$S_k = S_0 e^{-b\mathbf{g_k}^T \mathbf{D} \mathbf{g_k}} \tag{7}$$

where $S_k$ are the diffusion weighted images in the gradient directions $\mathbf{g_k}$, and $\mathbf{D}$ is the diffusion tensor, represented by a SPD (symmetric positive definite) $3 \times 3$ matrix. This formula reverts to the isotropic case of Eq. 6 with $\mathbf{D} = D\mathbf{I}$, being $\mathbf{I}$ the identity tensor.

The representation of diffusion by means of the diffusion tensor yields the DT-MRI imaging modality, which implicitly implies modeling the PDF of the three-dimensional molecular motion, $p_t(\mathbf{x})$ by a zero-mean trivariate Gaussian distribution whose covariance matrix is given by the diffusion tensor:

$$p(\mathbf{x}|\mathbf{D},t) = \frac{1}{\sqrt{(4\pi t)^3 |\mathbf{D}|}} \exp\left(\frac{-\mathbf{x}^T \mathbf{D}^{-1} \mathbf{x}}{4t}\right) \tag{8}$$

For the estimation of a field of $3 \times 3$ diffusion tensors, at least six measurements $S_k$ must be made using different non-collinear gradient directions, besides the baseline image $S_0$, taken in absence of a diffusion sensitizing field gradient. This is due to the fact that the diffusion tensor has six free components in a matrix representation,

that is, six degrees of freedom. Different methods for the estimation of the diffusion tensor have been proposed [55], such as the direct estimation proposed by Westin *et al.* [85, 84], the *least squares estimation* [10, 60] or a variational approach [72].

Even though the Gaussian assumption that DT-MRI employs is valid in most cases, sometimes this Gaussian model does not accurately fit the data. DT-MRI provides only one fibre-orientation estimate for each voxel, whereas in regions where different fibres cross inside a voxel, there are multiple fibre-orientations. In these points, obtaining the anisotropy from the diffusion tensor underestimates the true directional anisotropy and, furthermore, the orientation estimates are not correct. Several diffusion MRI modalities have been proposed that go beyond the tensor representation of the water diffusion in order to overcome these limitations [1, 28, 29, 3, 79, 73, 36, 2].

## 3 Scalar Diffusion and Anisotropy Measures

The necessity of developing new methods for the segmentation of tensor images arises from the fact that traditional segmentation techniques operate with scalars or vectors, and therefore cannot be directly applied to tensors. Initially, most of the methods intended for the analysis of tensor fields were based on scalar or vector values extracted from the tensors.

Apart from segmentation, other image processing techniques developed for DT-MRI, such as smoothing, have been primarily addressed by using derived expressions from the diffusion tensor such as eigenvalues, eigenvectors or rotationally-invariant scalar measures [23, 61]. Also for fibre tracking, most techniques work on scalar or vector-valued data [19, 76].

In [12], Basser *et al.* proposed the use of some scalar quantities derived from the diffusion tensor $\mathbf{D}$. First, they decomposed $\mathbf{D}$ into isotropic and anisotropic parts:

$$\mathbf{D} = D\mathbf{I} + (\mathbf{D} - D\mathbf{I}) \tag{9}$$

where $\mathbf{D_i} = D\mathbf{I}$ is an isotropic tensor ($\mathbf{I}$ is the identity tensor) and $\mathbf{D_a} = \mathbf{D} - D\mathbf{I}$ is an anisotropic tensor. The scalar $D$ is called the mean diffusivity, and can be computed in the following way:

$$D = \frac{\text{Trace}(\mathbf{D})}{3} = \frac{d_{11} + d_{22} + d_{33}}{3} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \tag{10}$$

where $\lambda_1$, $\lambda_2$ and $\lambda_3$ are the eigenvalues of $\mathbf{D}$. The authors also reinterpret $D$ with relation to the square root of the (scalar) generalized tensor product or tensor dot product[1] of the isotropic part of the tensor:

[1] The generalized tensor product of a tensor, which is equivalent to the square of the Frobenius norm ($\mathbf{T} : \mathbf{T} = \| \mathbf{T} \|_F^2$), is defined as $\mathbf{T}$, as $\mathbf{T} : \mathbf{T} = \sum_{i=1}^{3} \sum_{j=1}^{3} t_{ij}^2 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$

$$\sqrt{D\mathbf{I} : D\mathbf{I}} = D\sqrt{\mathbf{I} : \mathbf{I}} = D\sqrt{3} \tag{11}$$

It can be seen that this *magnitude* of the isotropic part of $\mathbf{D}$ is proportional to the mean diffusivity. The magnitude of the anisotropic part of $\mathbf{D}$ is given by:

$$\sqrt{\mathbf{D_a} : \mathbf{D_a}} = \sqrt{\sum_{i=1}^{3}\sum_{j=1}^{3}(d_{ij} - Di_{ij})^2} = \sqrt{\frac{1}{3}\left((\lambda_1-\lambda_2)^2 + (\lambda_2-\lambda_3)^2 + (\lambda_3-\lambda_1)^2\right)} \tag{12}$$

Starting from these definitions, it is then possible to obtain the well-known *relative anisotropy*, RA, and *fractional anisotropy*, FA:

$$RA = \frac{\sqrt{\mathbf{D_a} : \mathbf{D_a}}}{\sqrt{D\mathbf{I} : D\mathbf{I}}} = \frac{1}{\sqrt{3}}\frac{\sqrt{\mathbf{D_a} : \mathbf{D_a}}}{D} \tag{13}$$

$$FA = \sqrt{\frac{3}{2}}\frac{\sqrt{\mathbf{D_a} : \mathbf{D_a}}}{\sqrt{\mathbf{D} : \mathbf{D}}} \tag{14}$$

In [86], the authors proposed a decomposition of the diffusion tensor based on its symmetry properties that results in measures that describe the geometry of the diffusion ellipsoid (i.e. an ellipsoid whose principal axes correspond to the directions of the eigenvector system, and whose dimensions are the corresponding eigenvalues). From this analysis, a simple anisotropy measure was proposed. Specifically, let $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ be the eigenvalues of the diffusion tensor $\mathbf{D}$ and let $\hat{\mathbf{e}}_\mathbf{i}$ be the eigenvector corresponding to $\lambda_i$. Then, $\mathbf{D}$ can be decomposed as follows:

$$\mathbf{D} = \lambda_1 \hat{\mathbf{e}_1}\hat{\mathbf{e}_1}^T + \lambda_2 \hat{\mathbf{e}_2}\hat{\mathbf{e}_2}^T + \lambda_3 \hat{\mathbf{e}_3}\hat{\mathbf{e}_3}^T \tag{15}$$

A classification of the tensor shape in three different cases can be considered:

- **Linear case:** This case occurs when $\lambda_1 \gg \lambda_2 \simeq \lambda_3$ , and diffusion occurs only along one direction:
$$\mathbf{D} \simeq \lambda_1 \mathbf{D_l} = \lambda_1 \hat{\mathbf{e}_1}\hat{\mathbf{e}_1}^T \tag{16}$$
- **Planar case:** In this case, $\lambda_1 \simeq \lambda_2 \gg \lambda_3$, and diffusion is restricted to planes:
$$\mathbf{D} \simeq \lambda_1 \mathbf{D_p} = \lambda_1 (\hat{\mathbf{e}_1}\hat{\mathbf{e}_1}^T + \hat{\mathbf{e}_2}\hat{\mathbf{e}_2}^T) \tag{17}$$
- **Spherical case:** This is the isotropic case, where $\lambda_1 \simeq \lambda_2 \simeq \lambda_3$.
$$\mathbf{D} \simeq \lambda_1 \mathbf{D_s} = \lambda_1 (\hat{\mathbf{e}_1}\hat{\mathbf{e}_1}^T + \hat{\mathbf{e}_2}\hat{\mathbf{e}_2}^T + \hat{\mathbf{e}_3}\hat{\mathbf{e}_3}^T) \tag{18}$$

The geometrical shapes corresponding to these basic cases can be seen in Figure 1. One can expand the diffusion tensor using these cases as a basis:

$$\mathbf{D} = (\lambda_1 - \lambda_2)\mathbf{D_l} + (\lambda_2 - \lambda_3)\mathbf{D_p} + \lambda_3 \mathbf{D_s} \tag{19}$$

where $(\lambda_1 - \lambda_2)$, $(\lambda_2 - \lambda_3)$ and $\lambda_3$ are the coordinates of $\mathbf{D}$ in the tensor basis $\{\mathbf{D_l}, \mathbf{D_p}, \mathbf{D_s}\}$. If we normalize these coordinates so that each of them lies within

the range $[0,1]$ and the they all sum one, the linear, planar and spherical coefficients $c_l$, $c_p$ and $c_s$ are obtained:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \tag{20}$$

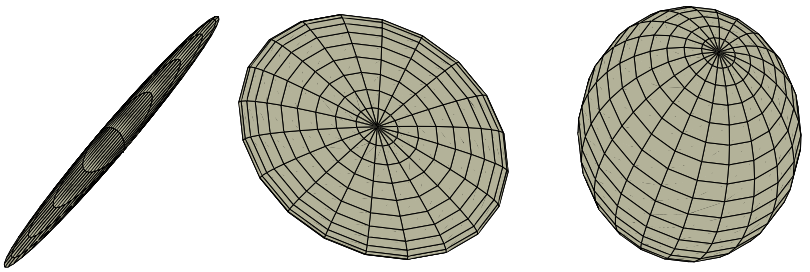
$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \tag{21}$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{22}$$

Finally, an anisotropy measure that describes the deviation from the spherical case is achieved by summing the normalized linear and planar coefficients:

$$c_a = c_l + c_p = \frac{\lambda_1 + \lambda_2 - 2\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} = 1 - c_s \tag{23}$$

In [57], the authors employed these coefficients to visualize and quantitatively an-



(a) (b) (c)

**Fig. 1** Geometric shapes of tensors for the linear, planar and spherical case ((a), (b), (c) respectively).

alyze the anisotropy properties and different parts of the brain, such as the corpus callosum and the internal capsule.

Zhukov *et al.* presented a slightly different approach in [89]. In their work, they argue that using eigenvalues/eigenvectors to compute scalar measures of diffusivity or anisotropy requires a considerable computational expense, and stability problems can arise as a small amount of noise will greatly affect not only the values but also the ordering of the eigenvalues (many of the anisotropy measures seen above depend on the ordering of the eigenvalues). Therefore, an anisotropy measure was proposed that does not require eigenvalue computations and is stable with respect to noise, besides of being rotationally invariant. This anisotropy measure is derived from three tensor invariants, that is, combination of tensor elements which are rotationally invariant, proposed in [75]. These invariants can be obtained without the computation of the eigenvalues (although they can also be obtained from them), and are the following:

1. First invariant or trace:

$$I_1 = d_{11} + d_{22} + d_{33} = \lambda_1 + \lambda_2 + \lambda_3 \tag{24}$$

2. Second invariant:

$$\begin{aligned} I_2 &= d_{11}d_{22} + d_{11}d_{33} + d_{22}d_{33} - d_{12}d_{21} - d_{13}d_{31} - d_{23}d_{32} \\ &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 \end{aligned} \tag{25}$$

3. Third invariant or determinant:

$$\begin{aligned} I_3 &= d_{11}(d_{22}d_{33} - d_{32}d_{23}) - d_{12}(d_{21}d_{33} - d_{31}d_{23}) \\ &+ d_{13}(d_{21}d_{32} - d_{31}d_{22}) = \lambda_1\lambda_2\lambda_3 \end{aligned} \tag{26}$$

The first invariant is proportional to the sum of the square of the radii of the ellipsoid, the second is proportional to the square of its surface area, and the third one is proportional to the square of its volume. From the three of them, the following anisotropy measure was proposed in [89]:

$$C_a = \frac{1}{6}\left(\frac{I_1 I_2}{I_3} - 3\right) \tag{27}$$

For isotropic diffusion, when $\lambda_1 = \lambda_2 = \lambda_3$, $C_a = 1$. In the anisotropic case, both for linear or planar diffusion, $C_a$ is always $\sim \lambda_1/\lambda_3$.

## 4 Tensor Dissimilarity Measures

Representing tensor fields using scalar diffusion or anisotropy measures extracted from the full tensor presents serious limitations derived from the substantial loss of information that this reduction implies. Therefore, in order to exploit the information present in all the components of the structure tensor, Rousson, Brox *et al.* proposed in [16, 66, 65] to represent the tensor in a vector form consisting of the nonlinearly diffused free components of the LST, and then using an Euclidean distance between them. This way, the image is treated as though it were multiespectral, being the channels each of the components of the tensor.

This approach presents an evident advantage over the schemes described in the preceding section that uniquely considered a scalar value extracted from the full tensor, as it uses the information corresponding to all the components in the tensor. However, it also suffers from a major drawback, namely, it misses the tensor structure of the information; making explicit use of the tensor structure implies performing the segmentation directly in the tensor domain.

In order to overcome the limitations of scalar diffusion or anisotropy measures or the vector processing of the tensor information, segmentation approaches have been

proposed based on the use of different dissimilarity measures between tensors. Next, we present and discuss the most relevant tensor distances that have been introduced in the literature.

In [4], Alexander *et al.* proposed a number of similarity measures for tensor images for their application in the registration of diffusion tensor data. These similarity measures include the squared trace of the difference of the tensors, the squared anisotropy indices, and have been the inspiration for subsequent work in the segmentation of tensor images.

Wiegell *et al.* proposed in [87] a tensor metric which is a combination of the Mahalanobis voxel distance and the Frobenius distance tensor distance. Specifically, the distance $E_{jk}$ between a voxel $j$ and the cluster centroid $k$ is as follows:

$$E_{jk} = \| \mathbf{x_j} - \bar{\mathbf{x}}_k \|_{\mathbf{W_k}} + \gamma \| \mathbf{D_j} - \bar{\mathbf{D}}_k \|_F \tag{28}$$

where $\mathbf{x_j}$ is the location of voxel $j$, $\bar{\mathbf{x}}_k$ is the mean voxel location for cluster $k$, $\mathbf{W_k}$ is the covariance matrix for the voxels in cluster $k$, $\gamma$ is a weighting factor to control the relative importance of the tensor distance and the voxel distance, $\mathbf{D_j}$ is the diffusion tensor for voxel $j$, and $\bar{\mathbf{D}}_k$ is the mean diffusion tensor. The Frobenius norm for a tensor $\mathbf{T}$ is defined as $\| \mathbf{T} \|_F = \left[\text{Trace}(\mathbf{T}^T\mathbf{T})\right]^{1/2}$. This way, the procedure labels as belonging to the same region voxels which are close both in position and in diffusion properties. To the best of our knowledge, this method is the first to propose a metric working directly on the tensor data to perform the segmentation.

Wang and Vemuri employed the Frobenius norm of the difference of tensors as a tensor distance in their segmentation approach [78]. Indeed, this is equivalent to the consideration of the tensor as a vector with an Euclidean distances. Although the experimental results were good, the Frobenius norm of the difference of tensors is not completely appropriate, as it uses the same weights for different components of the tensor and ignores the fact that tensors have structure.

Jonasson *et al.*, in [38, 37], proposed a new tensor dissimilarity measure called the *normalized tensor scalar product* (NTSP), which was specifically designed to be very sensitive to small changes of the orientation of the tensors:

$$\text{NTSP}(\mathbf{D}_1, \mathbf{D}_2) = \frac{\mathbf{D}_1 : \mathbf{D}_2}{\text{Trace}(\mathbf{D}_1)\text{Trace}(\mathbf{D}_2)} \tag{29}$$

where $\mathbf{D}_1 : \mathbf{D}_2$ stands for the well known tensor scalar product that is a measure of overlap between two tensors:

$$\mathbf{D}_1 : \mathbf{D}_2 = \text{Trace}(\mathbf{D}_1\mathbf{D}_2) \tag{30}$$

## *4.1 Kulback-Leibler Distance*

In order to overcome the limitations of the Frobenius norm or other tensor dissimilarity measures used in the preceding segmentation approaches, Wang and Vemuri

introduced in [77] a new tensor dissimilarity measure. Recalling that, in the context of DT-MRI, the displacement of water molecules over a time $t$ follows a Gaussian distribution whose covariance matrix is the diffusion tensor, it is naturally justified to use the distance between Gaussian distributions to induce a distance between the tensors. The most frequently used information theoretic dissimilarity measure is the Kullback-Leibler (KL) divergence, which is defined as:

$$KL(p \parallel q) = \int p(\mathbf{x}) \log \left( \frac{p(\mathbf{x})}{q(\mathbf{x})} \right) d\mathbf{x} \tag{31}$$

As the KL divergence is not symmetric, it is commonly symmetrized as:

$$J(p,q) = \frac{1}{2} \left( KL(p \parallel q) + KL(q \parallel p) \right) \tag{32}$$

This symmetrized KL divergence is also called J-Divergence. Based on it, Wang and Vemuri proposed in [77] a dissimilarity measure for SPD tensors, given by the square root of the J-divergence between two Gaussian distributions with zero mean and covariances $\mathbf{T_1}$ and $\mathbf{T_2}$:

$$d(\mathbf{T_1}, \mathbf{T_2}) = \sqrt{J(p(\mathbf{x}|\mathbf{T_1}, t), p(\mathbf{x}|\mathbf{T_2}, t))} \tag{33}$$

Taking the square root in Eq. 33 is justified because the KL divergence, and thus twice the J-divergence, is the square distance of two infinitesimally nearby points on a Riemannian manifold of parametrized distributions. It turns out that the J-divergence has a simple form for the Gaussian distribution considered [77]:

$$d(\mathbf{T_1}, \mathbf{T_2}) = \frac{1}{2} \sqrt{\text{trace}(\mathbf{T_1^{-1}} \mathbf{T_2} + \mathbf{T_2^{-1}} \mathbf{T_1}) - 2n} \tag{34}$$

where $n$ is the size of the tensors.

In order to employ the J-divergence for segmentation purposes, one usually needs to compute the mean values of the tensor field over different regions. In [77], the mean value $\bar{\mathbf{T}}$ of a tensor field $\mathbf{T}$ over a region $\Omega$ is defined as

$$\bar{\mathbf{T}}(\mathbf{T}, \Omega) = \min_{M \in S^+(n,\mathbb{R})} \int_{\Omega} d^2(\mathbf{T}(\mathbf{x}), \mathbf{M}) d\mathbf{x} \tag{35}$$

where $S^+(n,\mathbb{R})$ stands for the set of real SPD matrices of size $n$. Wang and Vemuri proved that there is a closed form expression for $\bar{\mathbf{T}}$ when using the J-Divergence as a distance measure, which is given by

$$\bar{\mathbf{T}} = \sqrt{\mathbf{B^{-1}}} \left[ \sqrt{\sqrt{\mathbf{B}} \mathbf{A} \sqrt{\mathbf{B}}} \right] \sqrt{\mathbf{B^{-1}}} \tag{36}$$

where $\mathbf{A} = \int_{\Omega} \mathbf{T}(\mathbf{x}) d\mathbf{x}$ and $\mathbf{B} = \int_{\Omega} \mathbf{T^{-1}}(\mathbf{x}) d\mathbf{x}$.

### 4.2 Geodesic Distance

In order to define a new tensor dissimilarity measure, Lenglet *et al.* considered in [46, 47] the statistical manifold $\mathscr{M}$ representing the family of 3-D, zero-mean Gaussian PDFs described through the 6 free parameters of their covariance matrix[2]. A Riemannian metric can then be introduced in terms of the Fisher information matrix [63] that allows for the definition of geodesic distances on this manifold.

The geodesic distance induced by the Riemannian metric derived from the Fisher information matrix has been investigated for several parametric distributions. If we concentrate on the family of multivariate normal distributions with common mean vector $\xi$ but different covariance matrices $\mathbf{T}$ or, equivalently, on the manifold $S^+(n,\mathbb{R})$, i.e. the set of $n \times n$ real SPD matrices, the information geodesic distance between any two element $\mathbf{T_1}$ and $\mathbf{T_2}$ is given by (theorem by S. T. Jensen, 1976, originally proved in [7]):

$$d(\mathbf{T_1},\mathbf{T_2}) = \sqrt{\frac{1}{2}\text{trace}\left(\log^2\left({\mathbf{T_1}}^{-1/2}\mathbf{T_2}{\mathbf{T_1}}^{-1/2}\right)\right)} = \sqrt{\frac{1}{2}\sum_{i=1}^{m}\log^2(\lambda_i)} \tag{37}$$

where the $\lambda_i$ are the $m$ eigenvalues of the determinantal equation $|\lambda\mathbf{T_2} - \mathbf{T_1}| = 0$. The information geodesic distance was used in the segmentation algorithm that will be described in next section (Eqs. 44-47) replacing the J-divergence [47, 43]. For this purpose, the mean value of the tensor field over each region with respect to this new geodesic distance is needed. Fréchet [30], Karcher [39] and Pennec [58] define this mean value, also known as the Riemannian barycenter, as follows:

$$\bar{\mathbf{T}} = \min_{T\in S^+(n,\mathbb{R})}\frac{1}{N}\sum_{k=1}^{N} d^2(\mathbf{T},\mathbf{T_k}) = \mathbb{E}[d^2(\mathbf{T},\mathbf{T_k})] \tag{38}$$

While it has been shown that the Riemannian barycenter exists and is unique for $S^+(n,\mathbb{R})$, a closed form expression cannot be obtained [53]. However, a gradient descent algorithm was proposed in [46] where, starting from an initial guess $\hat{\mathbf{T}}(t=0)$, the following evolution equation is obtained:

$$\partial_t\hat{\mathbf{T}}(t) = -\frac{\hat{\mathbf{T}}(t)}{N}\sum_{k=1}^{N}\log\left(\mathbf{T_k^{-1}}\hat{\mathbf{T}}(t)\right) \tag{39}$$

More details and the corresponding numerical implementation can be found in [46].

[2] Throughout the theoretical derivation of the geodesic distance, $3 \times 3$ diffusion tensors are considered.

### *4.3 Log-Euclidean Metrics*

Even though Riemannian geodesic distances have excellent theoretical properties and have shown very good results for segmentation purposes, they yield in practice slow algorithms because of their complexity. In order to overcome this limitation, the Log-Euclidean metrics were proposed [5, 6] that produce similar results but using simpler and faster computations.

Euclidean distances (also referred to as Frobenius distance distances) are well adapted to general square matrices, but they are not for the tensor specific case. If Euclidean operations are performed on tensors, null or negative eigenvalues can appear, as these Euclidean operations are not convex [6]. Besides, Euclidean averaging of tensors produces what is called *tensor swelling effect* [22], meaning that the determinant of the Euclidean mean of several tensors can be larger than the determinants of the original tensors.

With respect to Riemannian metrics, the computational burden is related to the intensive use of matrix inverses, square roots, logarithms and exponentials. Besides, there is not a closed form for the tensor mean, and the computation needs to be done in an iterative manner.

In order to define the Log-Euclidean metrics, the notions of matrix logarithm and exponential are first needed. For any matrix $\mathbf{M}$, its exponential is given by:

$$\exp(\mathbf{M}) = \sum_{k=0}^{\infty} \frac{\mathbf{M}^k}{k!} \tag{40}$$

The matrix logarithm is defined as the inverse of the exponential. The existence and uniqueness of the logarithm is not guaranteed for a general invertible matrix, but it is well defined for a positive definite tensor, yielding a symmetric matrix. Conversely, the exponential of any symmetric matrix is a positive definite tensor.

The introduction of Log-Euclidean metrics is based on the idea of defining a novel vector space structure on tensors. It corresponds to Euclidean metrics in the domain of logarithms. Using the Euclidean norm $||\cdot||$ on symmetric matrices, the distance between two tensors can be written as:

$$d(\mathbf{T_1}, \mathbf{T_2}) = ||\log(\mathbf{T_1}) - \log(\mathbf{T_2})|| \tag{41}$$

Log-Euclidean metrics do not satisfy full affine-invariance as the Riemannian metric introduced in the previous section. However, a number of them are invariant by similarity, that is, orthogonal transformation and scaling. The simplest similarity-invariant Log-Euclidean metric is given by:

$$d(\mathbf{T_1}, \mathbf{T_2}) = \left[\text{Trace}\left((\log(\mathbf{T_1}) - \log(\mathbf{T_2}))^2\right)\right]^{1/2} \tag{42}$$

Log-Euclidean have been mainly employed for interpolation and regularization of tensor fields [6], but some authors have also used for segmentation purposes, as will be seen in next section.

## 5 Tensor Field Segmentation Techniques

Whereas most of the scalar diffusion and anisotropy measures previously described in this chapter were proposed mainly for visualization or some quantitative purposes from brain tensor data, the diffusivity measure $I_1$ (Eq. 24) and the anisotropy measure $C_a$ (Eq. 27) were both employed in [89] for the segmentation of tensor brain data through the evolution of a level set following an edge-based approach similar to the *Geodesic Active Contour* (GAC) model [20, 40]. This can be considered to be one of the very first approaches to the segmentation of tensor data.

Rousson, Brox *et al.* proposed in [16, 66, 65] to apply the vector-valued version of the *Geodesic Active Regions* (GAR) model [56] for multivariate Gaussian distributions [67, 68] to the vector consisting of the nonlinearly diffused free components of the LST. This way, the image is treated as though it were multiespectral, being the channels each of the components of the tensor.

The first level set segmentation approach directly working on tensor data was proposed by Feddern *et al.* [26]. In their work, they present a structure tensor for tensor data as a generalization of the concept of image gradient to tensor data. Starting from this concept an active contour model is presented, based on the GAC model and adapted to tensor data by using the trace of the LST of the tensor data as an edge detector that stops the evolution of the contour in the presence of edges.

In [87], a modified *k-means* algorithm [49, 33, 15] was proposed to segment the thalamic nuclei from DT-MRI. The metric the authors propose is a combination of the Mahalanobis voxel distance and the Frobenius distance tensor distance (see Section 4).

After the pioneering works by Feddern *et al.* [26] —who first run a level set algorithm on tensor data, as previously indicated—, and by Wiegell *et al.* [87] —who first employed tensor dissimilarity measures for segmentation—, most recent approaches for the segmentation of tensor data are based on the combination of both elements, that is, the use of variational methods and level sets based on the information given by tensor dissimilarity measures.

In [78], Wang and Vemuri proposed the use of a region-based active contour model for the segmentation of tensor fields. Their approach is an extension of the level set implementation of the Mumford-Shah functional [54] to matrix-valued images, where the Frobenius norm of the difference of tensors is employed. This segmentation method was tested on synthetic tensor field segmentation as well as for the segmentation of textures by means of the LST, and MRI data.

Jonasson *et al.*, in [38, 37], face the segmentation of DT-MRI data from a perspective somehow inspired on a fibre tracking approach. In a level set framework, the speed propagation of the front is proportional to the similarity between the tensors lying on the front and its neighbors. They employed the NTSP as a tensor dissimilarity measure (see Section 4). In order for the mean curvature flow —usually introduced in the level set evolution for regularization purposes— not to destroy the tubular structure of the fibre tracts that are sought, the *smaller principal curvature* is employed, which is a combination of the mean and the Gaussian curvatures and was introduced in [51].

There is an important group of segmentation algorithms that have been proposed using the J-divergence as tensor dissimilarity measure. The first approach was presented by Wang and Vemuri [77], and is based on the minimization of the following energy functional:

$$E(\mathscr{C},\mathbf{T_1},\mathbf{T_2}) = \int_{\Omega_1} d^2(\mathbf{T}(\mathbf{x}),\mathbf{T_1})d\mathbf{x} + \int_{\Omega_2} d^2(\mathbf{T}(\mathbf{x}),\mathbf{T_2})d\mathbf{x} + \nu|\mathscr{C}| \tag{43}$$

This energy functional is the tensor extension, with the introduction of the J-divergence as dissimilarity measure, of the Chan and Vese model for the segmentation of piecewise constant regions [21].

The energy functional in Eq. 43 depends on the segmenting contour, $\mathscr{C}$, and on the mean values of the tensor field over regions $\Omega_1$ and $\Omega_2$, $\mathbf{T_1}$ and $\mathbf{T_2}$, respectively. These are computed as shown in Section 4.1. Through its implementation in a level set framework, this segmentation approach was successfully tested on synthetic tensor fields as well as for the segmentation of textures using the LST and the segmentation of real DT-MRI data. Its main drawback is, coherently with the Chan and Vese model on which it is grounded, its limitation to a piecewise constant model. With the aim to overcome this limitation, Lenglet *et al.* presented in [43, 47] a segmentation approach that, with the twofold inspiration of the GAR model for segmentation and the use of the J-divergence as a tensor dissimilarity measure, proposes the minimization of the following energy functional:

$$E(\mathscr{C},\bar{\mathbf{T}}_i,\sigma_i^2) = -\sum_{i=1}^{2}\int_{\Omega_1} \log p_{d,i}\left(d(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_i)\right) d\mathbf{x} \tag{44}$$

where $\mathscr{C}$ is the segmenting contour that splits the image domain $\Omega$ into regions $\Omega_1$ and $\Omega_2$. The equation above is a derivation from the region term in the GAR model where, instead of modeling the PDF of the image values, the distances between the tensor at each point and the mean value of the tensor field over the corresponding region are modeled as Gaussian distributions[3] with zero mean and variances $\sigma_i^2$, that is:

$$p_{d,i}\left(d(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_i)\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-d^2(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_i)}{2\sigma_i^2}\right) \tag{45}$$

If we reformulate the energy term by representing the segmenting contour $\mathscr{C}$ by the zero level set of the distance function $\phi$, and make use of the regularized versions of the Dirac and Heaviside functions, $\delta_\varepsilon(\phi(\mathbf{x}))$ and $H_\varepsilon(\phi(\mathbf{x}))$, and add the regularity constraint on the length of the contour, the energy functional becomes:

$$\begin{aligned} E(\phi,\bar{\mathbf{T}}_i,\sigma_i^2) = \int_\Omega & -\log p_{d,1}\left(d(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_1)\right) H_\varepsilon(\phi(\mathbf{x})) \\ & - \log p_{d,2}\left(d(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_2)\right)(1-H_\varepsilon(\phi(\mathbf{x}))) + \nu|\nabla H_\varepsilon(\phi(\mathbf{x}))|d\mathbf{x} \end{aligned} \tag{46}$$

[3] Since the tensor distances are always non-negative, strictly speaking the expression in Eq. 45 should be multiplied by 2 so that $p_{d,i}\left(d(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_i)\right)$ remains a PDF. Anyway, as this applies for both distributions over $\Omega_1$ and $\Omega_2$, this consideration has no practical effect.

The derivation of the Euler-Lagrange equations for this energy functional, which was studied in [67], yields the following evolution equation for the level set function:

$$\frac{\partial \phi(\mathbf{x})}{dt} = \delta_\varepsilon(\phi(\mathbf{x}))\left[\nu\nabla\cdot\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|} + \frac{1}{2}\log\left(\frac{p_{d,2}\left(d(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_2)\right)}{p_{d,1}\left(d(\mathbf{T}(\mathbf{x}),\bar{\mathbf{T}}_1)\right)}\right)\right] \tag{47}$$

The derivation of the energy with respect to $\sigma_i^2$ and $\bar{\mathbf{T}}_i$ indicates that the variances must be updated by their estimations, while the mean values for the tensor field over both regions are updated following Eq. 36.

It is worth noticing that this segmentation approach is an extension of the method proposed in [67] by Rousson *et al.* that applies the GAR model with Gaussian distributions on the intensity values. In the present approach, it is not the data that is modeled as indicated, but the distances between the tensors at each point and their mean values, and a zero mean constraint is imposed.

Together with this segmentation framework, the use of a new tensor distance was also proposed in [47]. Taking into account the Riemannian geometry of the space of Gaussian PDFs based upon which the K-L distance was defined, they rigorously define a geodesic distance on this Riemannian manifold.

The described segmentation approach was able to achieve impressive results for the segmentation of both synthetic and real DT-MRI data and it was made clear in [47] the superiority of the J-divergence and the geodesic distance over the consideration of the tensor components in a $\mathbb{R}^6$ vector representation.

While the J-divergence and the information geodesic distance between tensors are indeed appropriate intrinsic tensor dissimilarity measures, their use for tensor field segmentation purposes through the modeling of the distances between the tensors and their mean values as a zero-mean Gaussian distribution entails an important limitation. Indeed, the complexity of the space of SPD tensors is reduced to the 1-D space of the J-divergence or geodesic distances.

On the other hand, the previous vector representations in $\mathbb{R}^d$ for the tensors components considered in [65, 66] allowed for the statistical modeling directly on the domain of the data, but the vector representation was shown not to be completely adequate.

Because of this, Lenglet *et al.* proposed in [48, 44, 45] an approach for the statistical modeling of the data directly on the tensor domain, through the definition of Gaussian distributions between diffusion tensors. These distributions are then incorporated into the probabilistic setting of the GAR model. Making use of the necessary elements from differential geometry, and once a suitable metric has been defined, a Gaussian distribution between tensors belonging to the manifold $S^+(3,\mathbb{R})$ of the $3\times 3$ real, SPD matrices is defined with the following PDF:

$$p(\mathbf{T_i}|\bar{\mathbf{T}},\Lambda) = \frac{1}{\sqrt{(2\pi)^6|\Lambda|}}\exp\left(-\frac{\varphi(\beta_i)^T\Lambda^{-1}\varphi(\beta_i)}{2}\right) \tag{48}$$

The following elements need to be specified to complete the description of the Gaussian distribution:

- $\mathbf{T_i}$ is the $3 \times 3$ tensor located at voxel $i$ in $\Omega$.
- $\bar{\mathbf{T}}$ is the empirical mean tensor over a set of $N$ diffusion tensors.
- $\Lambda$ is the associated covariance matrix, whose size is $6 \times 6$ for a $3 \times 3$ tensor (that is, the number of free components).
- The symmetric matrix $\beta_i$ is defined, for a given metric $\mathscr{D}_x$, by $\beta_i = -\nabla_{\mathbf{T_i}} \mathscr{D}_x(\mathbf{T_i}, \bar{\mathbf{T}})$.
- The map $\varphi : S^+(3,\mathbb{R}) \mapsto \mathbb{R}^6$ associates to each matrix $\beta_i$ its 6 free components.

As can be seen, the definition of the Gaussian distribution is general in terms of the metric employed. In their work, Lenglet *et al.* consider three different choices: Euclidean metric, J-divergence and geodesic metric. For further details, we refer the reader to [48, 44, 45].

Using the described Gaussian distribution on tensor data, and choosing one of the proposed metrics, an energy functional was then proposed in that is based on modeling the data directly on the tensor domain:

$$E(\mathscr{C}, \bar{\mathbf{T}}_{1,2}, \Lambda_{1,2}) = -\int_{\Omega_1} \log p(\mathbf{T}(\mathbf{x})|\bar{\mathbf{T}}_1, \Lambda_1) d\mathbf{x} - \int_{\Omega_2} \log p(\mathbf{T}(\mathbf{x})|\bar{\mathbf{T}}_2, \Lambda_2) d\mathbf{x} + \nu|\mathscr{C}| \tag{49}$$

In the original work by Lenglet *et al.*, a boundary based term was also employed, based on the norm of the spatial gradient of the tensor field, which is also derived for the Euclidean, J-divergence and Geodesic metrics (see [44]).For given statistical parameters, the final level set evolution equation employed for segmentation is:

$$\frac{\partial \phi(\mathbf{x})}{dt} = \delta_\varepsilon(\mathbf{x}) \left[ \mu \nabla \cdot \frac{\nabla \phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|} + \frac{1}{2} \log \left( \frac{p(\mathbf{T}(\mathbf{x})|\bar{\mathbf{T}}_1, \Lambda_1)}{p(\mathbf{T}(\mathbf{x})|\bar{\mathbf{T}}_2, \Lambda_2)} \right) \right] \tag{50}$$

Experimental results made with this segmentation approach on both synthetic and real MR-DTI images proved that the choice of the tensor dissimilarity measure has a critical effect on the segmentation results. The Euclidean metric presents serious limitations that become evident as the complexity of the data increases. All in all, empirical evidence showed that the geodesic distance outperforms the J-divergence which, in turn, achieves better results than the Euclidean metric.

Following the idea of applying a statistical model directly on the tensor data for segmentation, de Luis-Garcia *et al.* proposed in [25] an extension of the previous method through the definition of Gaussian mixtures on tensor fields, which are then incorporated in the segmentation framework seen above. Through an automatic method for the selection of the number of components of the mixture, the statistical model is able to adapt to the complexity of the data. This technique was successfully tested for the segmentation of the corpus callosum from DTI over a big dataset, showing a slight higher accuracy and robustness than the use of a single Gaussian.

A completely different approach was proposed by Ziyan *et al.* for the segmentation of the talamic nuclei from DTI data [90]. This method uses a spectral segmentation algorithm [69] with some modifications. In spectral clustering, the segmentation problem is posed in terms of a graph partitioning problem. The nodes of the graphs

are the diffusion tensors, and the links connecting the nodes are based on the tensor similarities of the data points being linked. In their work, several distances were employed: Frobenius norm, K-L divergence, and also the angular difference between the principal eigenvector directions. Finally, in order to incorporate spatial relations into the scheme, Markovian relaxation was employed. Results showed an accurate segmentation on a dataset of 10 DTI volumes, starting from the manual delineation of thalamus masks by experts.

Weldeselassie and Hamarneh proposed a related approach in [83], where graph cuts segmentation technique operates using either the Log-Euclidean or the J-divergence as tensor distances. The user interactively selects certain tensors as belonging to the different regions, thus imposing hard constraints to the segmentation. This technique was applied to the segmentation of the corpus callosum as well as the cardiac wall from DTI data.

Another recent approach to DTI segmentation was proposed by Awate *et al.* [9, 8], who presented a fuzzy C-means algorithm that, instead of incorporating Gaussian class models, uses nonparametric data-driven statistical models. The motivation underlying this approach lies in the fact that, because of the anatomical characteristics of fiber bundles, they change their orientation significantly. Thus, the tensor statistics do not accurately fit Gaussian models, whereas nonparametric statistical models can efectively adjust to these situations. Using a Log-Euclidean metric, the segmentation of different structures in the white matter, such as the cingulum and the corticospinal tract, is performed.

In order to provide a comparative insight on all the segmentation methods that have been described in this section, we summarize them in Tables 1 and 2, where the main features of each method are listed together with their main advantages and disadvantages.

## 6 Summary

In this chapter, the use of tensor fields for image processing, and for medical image processing in particular, has been discussed. Second-order tensors constitute a convenient mathematical tool to describe anisotropic behaviours; specifically, we have focused on the LST and the diffusion tensor from DT-MRI.

Once the LST and the diffusion tensor have been introduced, we have reviewed the main scalar measures that have been used for tensor analysis in the literature. These were precursors of the segmentation techniques that were proposed afterwards directly in the tensor domain. These techniques rely on the definition of suitable tensor distances, which have thus been a central element in this chapter.

With regard to the different tensor segmentation methods that have been studied in this chapter, a two-fold evolution is worth noticing. First, methods have evolved from the usage of basic tensor dissimilarity measures, such as Euclidean distance, towards more advanced distances, such as the geodesic distance or the Log-Euclidean metric. Second, these distances have been incorporated into segmen-

tation framework with an increasing level of refinement. Indeed, DT-MRI has been the leading application for the development of new segmentation techniques, with the final goal of the automatic segmentation of anatomical structures within the white matter of the brain.

Even though the most recent segmentation methods have successfully proved to be able to segment different structures such as the corpus callosum or the thalamus nuclei, some important limitations still remain unsolved. First, these segmentation methods need some degree of manual initialization, which makes the automatic analysis of populations hard to perform. Automatic initialization procedures need to be developed, either through a coarse-to-fine segmentation approach, the use of anatomical knowledge to drive the segmentation or the usage of brain atlases to guide the initialization process. Second, the relative performance of the different tensor metrics has not completely been understood. Some comparisons have been made, but they are limited and do not allow to assess the superiority of certain metric for segmentation purposes in general, or for the segmentation of a certain anatomical structure. The nature of the interfaces between different fiber bundles in the white matter is not homogeneous, which suggests that different metrics could be appropriate for different specific applications.

All in all, the topic of segmentation of tensor-valued data keeps receiving an increasing interest. Given the rapid evolution of the segmentation techniques in the last years, it can be expected that this trend will go on, providing useful tools for the analysis of DT-MRI images and tensor-valued data in general.

**Acknowledgements** The authors acknowledge the Comisión Interministerial de Ciencia y Tecnología of Spain for research grant TEC 2007-67073/TCM. The first author is funded by the Spanish MEC/Fulbright Comission grant 2007-1238. This is also acknowledged.

## References

1. A. L. Alexander, K. M. Hasan, M. Lazar, J. S. Tsuruda, and D. L. Parker. Analyis of partial volume effects in diffusion-tensor MRI. *Magnetic Resonance in Medicine*, 45:770–780, 2001.
2. D. C. Alexander. Introduction to diffusion MRI. in (J. Weickert and H. Hagen, Eds.) *Visualization and Processing of Tensor Fields*, Springer, 2006, pp. 83-106.
3. D. C. Alexander, G. J. Barker, and S. R. Arridge. Detection and modeling of non-Gaussian apparent diffusion coefficient profiles in human brain data. *Magnetic Resonance in Medicine*, 48:331–340, 2002.
4. D. C. Alexander, J. Gee, and R. Bajcsy. Similarity measures for matching diffusion tensor images. In *Proc. of the British Machine Vision Conference (BMVC)*, Nottingham, UK, Sep. 1999.
5. V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Fast and simple calculus on tensors in the Log-Euclidean framework. In *Proc. of the MICCAI'05, volume 3749 of Lecture Notes in Computer Science*, pages 115–122, Palm Springs, CA, USA, Oct. 2005.
6. V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, 2006.

7. C. Atkinson and A. Mitchell. Rao's distance measure. *Sankhya: The Indian Journal of Statistics*, 43(A):345–365, 1981.
8. S. P. Awate and J. C. Gee. A fuzzy, nonparametric segmentation framework for DTI and MRI analysis. In *Proc. of Information Processing in Medical Imaging (IPMI)*, Kerkrade, Netherlands, Jul. 2007.
9. S. P. Awate, H. Zhang, and J. C. Gee. A fuzzy, nonparametric segmentation framework for DTI and MRI analysis: with applications to DTI-tract extraction. *IEEE Transactions on Medical Imaging*, 16(1):1525–1536, 2007.
10. P. Basser, J. Mattiello, and D. L. Bihan. Estimation of the effective self-diffusion tensor from the NMR spin echo. *Journal of Magnetic Resonance*, B(103):247–254, 1994.
11. P. Basser, J. Mattiello, and D. L. Bihan. MR diffusion tensor spectroscopy and imaging. *Biophysica*, 66:259–267, 1994.
12. P. Basser and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion tensor MRI. *Journal of Magnetic Resonance*, B(111):209–219, 1996.
13. J. Bigun, G. H. Grandlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.
14. D. Le Bihan, E. Breton, D. Lallemand, P. Grenier, E. Cabanis, and M Laval-Jeantet. Mr imaging of intravoxel incoherent motions: Application to diffusion and perfusion in neurologic disorders. *Radiology*, 161(2):401–407, 1986.
15. C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, New York, USA, 1997.
16. T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. Technical Report 4760, INRIA, Mar. 2003.
17. T. Brox and J. Weickert. Nonlinear matrix diffusion for optic flow estimation. In *Proc. of the 24th DAGM Symposium, volume 2449 of Lecture Notes in Computer Science*, pages 446–453, Zurich, Switzerland, Sep. 2002.
18. T. Brox, J. Weickert, B. Burgeth, and P. Mrázek. Nonlinear structure tensors. Technical Report Preprint No 113, Deparatment of Mathematics, Saarland University, Saarbrücken, Germany, Oct. 2004.
19. J. Campbell, K. Siddiqi, B. Vemuri, and G. B. Pike. A geometric flow for white matter fibre tract reconstruction. In *Proc. of the IEEE International Symposium on Biomedical Imaging*, pages 505–508, Washington DC, USA, Jul. 2002.
20. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal on Computer Vision*, 22:61–79, 1997.
21. T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001.
22. C. Chefd'hotel, D. Tschumperlé, R. Deriche, and O. Faugeras. Regularizing flows for constrained matrix-valued images. *Journal of Mathematicl Imaging and Vision*, 10(1-2):147–162, 2004.
23. O. Coulon, D. C. Alexander, and S. A. Arridge. A regularization scheme for diffusion tensor magnetic resonance images. In *Proc. of the 17th IPMI Conference, volume 2082 of Lecture Notes in Computer Science*, pages 92–105, Davis, CA, USA, Jun. 2001.
24. G. Cross and A. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:25–39, 1983.
25. R. de Luis-Garcia and C. Alberola-Lopez. Mixtures of Gaussians on tensor fields for the segmentation of DT-MRI. In *Proc. of the MICCAI'07, volume 4791 of Lecture Notes in Computer Science*, pages 1117–1127, Brisbane, Australia, Oct. 2007.
26. C. Feddern, J. Weickert, and B. Burgeth. Level set methods for tensor-valued images. In *Proc. of the 9th IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 65–72, Nice, France, Oct. 2003.
27. W. Foerstner and E. Gulch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, Jun. 1987.

28. L. R. Frank. Characterization of anisotropy in high angular resolution diffusion-weighted MRI. In *Proc. 9th ISMRM (International Society for Magnetic Resonance in Medicine)*, page 1531, Glasgow, Scotland, Apr. 2001.
29. L. R. Frank. Characterization of anisotropy in high angular resolution diffusion-weighted MRI. *Magnetic Resonance in Medicine*, 47:1083–1099, 2002.
30. M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'Institut Henri Poincaré*, X(IV):215–310, 1948.
31. G. Gerig, O. Kubler, R. Kikinis, and F. A. Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11:221–232, 1992.
32. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
33. J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.
34. M. Horsfield and D. Jones. Applications of diffusion-weighted and diffusion tensor MRI to white matter diseases-a review. *NMR in Biomedicine*, 15(7-8):570–577, 2002.
35. G. H. Grandlund J. Bigun. Optimal orientation detection of linear symmetry. In *Proc. of the 1st IEEE International Conference on Computer Vision*, London, Jun. 1987.
36. K. M. Jansons and D. C. Alexander. Persistent angular structure: New insights from diffusion MRI data. *Inverse Problems*, 19:1031–1046, 2003.
37. L. Jonasson, X. Bresson, P. Hagmann, O. Cuisenaire, R. Meuli, and J.-P. Thiran. White matter fiber tract segmentation in DT-MRI using geometric flows. *Medical Image Analysis*, 9:223–236, 2005.
38. L. Jonasson, P. Hagmann, X. Bresson, R. Meuli, O. Cuisenaire, and J.-P. Thiran. White matter mapping in DT-MRI using geometric flows. In *Proc. Eurocast 2003, volume 2809 of Lecture Notes in Computer Science*, pages 585–595, Las Palmas de Gran Canaria, Spain, Feb. 2003.
39. H. Karcher. Riemannian centre of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30:509–541, 1977.
40. S Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. of the 5th International Conference on Computer Vision*, pages 810–815, Orlando, FL, USA, Jun. 1995.
41. H. Knutsson. A tensor representation of 3-D structures. In *Proc. 5th IEEE-ASSP and EURASIP Workshop on Multidimensional Signal Processing*, Noordwijkerhout, The Netherlands, Sep. 1987.
42. H. Knutsson. Representing local structure using tensors. In *Proc. 6th Scandinavian Conference on Image Analysis*, pages 244–251, Oulu, Finland, Jun. 1989.
43. C. Lenglet, M. Rousson, and R. Deriche. Segmentation of 3D probability density fields by surface evolution: Application to diffusion MRI. In *Proc. of the Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Saint Malo, France, Sep. 2004.
44. C. Lenglet, M. Rousson, and R. Deriche. DTI segmentation by statistical surface evolution. Technical Report 5843, INRIA, Feb. 2006.
45. C. Lenglet, M. Rousson, and R. Deriche. DTI segmentation by statistical surface evolution. *IEEE Transactions on Medical Imaging*, 25(6):685–700, 2006.
46. C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras. Statistics on multivariate normal distributions: A geometric approach and its application to diffusion tensor MRI. Technical Report 5242, INRIA, Jun. 2004.
47. C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras. Toward segmentation of 3D probability density fields by surface evolution: Application to diffusion MRI. Technical Report 5243, INRIA, jun. 2004.
48. C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor MRI processing. *Journal of Mathematical Imaging and Vision*, 25:423–444, 2006.
49. Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.
50. T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Boston, USA, 1994.

51. L. M. Lorigo, O. Faugeras, W. E. L. Grimson, R. Keriven, R. Kikinis, and C. F. Westin. Co-dimension 2 geodesic active contours for MRA segmentation. In *Proc. of Information Processing in Medical Imaging*, pages 126–139, Visegrád, Hungary, Jun. 1999.
52. J. Malik and J. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am. A*, 7:923–932, 1990.
53. M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 26(3):735–747, 2005.
54. D. Mumford and Shah. Boundary detection by minimizing functionals. In *Proc. of the International Conference on Computer Vision and Pattern Recognition*, pages 22–26, San Francisco, CA, USA, Jun. 1985.
55. C. A. Casta no Moraga. *Nuevos esquemas para el procesado de señales tensoriales*. PhD dissertation, University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain, 2006.
56. N. Paragios and R. Deriche. Geodesic active regions: A new framework to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13:249–268, 2002.
57. S. Peled, H. Gudbjartsson, C.-F. Westin, R. Kikinis, and F. A. Jolesz. Magnetic resonance imaging shows orientation and asymmetry of white matter fiber tracts. *Brain Research*, 780:27–33, 1998.
58. X. Pennec. Probabilities and statistics on Riemannian manifolds. Technical Report 5093, INRIA, Jan. 2004.
59. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.
60. C. Poupon. *Détection des faisceaux de fibres de la substance blanche pour l'etude de la connectivité anatomique cérébrale*. PhD dissertation, Ecole Nationale Supérieure des Télécommunications, 1999.
61. C. Poupon, J.-F. Mangin, V. Frouin, J. Régis, F. Poupon, M. Pachot-Clouard, D. Le Bihan, and I. Bloch. Regularization of MR diffusion tensor maps for tracking brain white matter bundles. In *Proc. of the MICCAI'98, volume 1496 of Lecture Notes in Computer Science*, pages 489–498, Cambridge, MA, USA, Oct. 1998.
62. A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.
63. C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–91, 1945.
64. M. Rousson. *Cues Integrations and Front Evolutions in Image Segmentation*. PhD dissertation, University of Nice - Sophia Antipolis, Sophia Antipolis, France, 2004.
65. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Madison, Wisconsin, USA, Jun. 2003.
66. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. Technical Report 4695, INRIA, Jan. 2003.
67. M. Rousson and R. Deriche. A variational framework for active and adaptative segmentation of vector valued images. In *Proc. of the IEEE Workshop on Motion and Video Computing*, pages 56–62, Orlando, Florida, USA, Dec. 2002.
68. M. Rousson and R. Deriche. A variational framework for active and adaptative segmentation of vector valued images. Technical Report 4515, INRIA, Jul. 2002.
69. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
70. E. Stejskal and J. Tanner. Spin diffusion measurements: spin echoes in presence of a time-dependent field gradient. *Journal of Chemical Physics*, 42:288–292, 1965.
71. P. C. Sundgren, Q. Dong, D. Gómez-Hassan, S. K. Mukherji, P. Maly, and R. Welsh. Diffusion tensor imaging of the brain: review of clinical applications. *Neuroradiology*, 46(5):339–350, Aug. 2004.

72. D. Tsumperlé and R. Deriche. Variational frameworks for DT-MRI estimation, regularization and visualization. In *Proc. of the International Conference on Computer Vision*, volume 1, pages 116–121, Nice, France, 2000.
73. D. S. Tuch, T. G. Reese, M. R. Wiegell, and V. J. Wedeen. Diffusion MRI of complex neural architecture. *Neuron*, 40:885–895, 2003.
74. M. R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82, 1986.
75. A. M. Uluğ and P. C. M. van Zijl. Orientation-independent diffusion imaging without tensor diagonalization: anisotropy definitions based on physical attributes of the diffusion ellipsoid. *Journal of Magnetic Resonance Imaging*, 9(6):804–813, 1999.
76. B. Vemuri, Y. Chen, M. Rao, T. McGraw, Z. Wang, and T. Mareci. Fiber tract mapping from diffusion tensor MRI. In *Proc. of the 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 73–80, Vancouver, Canada, Jul. 2001.
77. Z. Wang and B. Vemuri. An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 228–233, Washington DC, USA, 2004.
78. Z. Wang and B. Vemuri. Tensor field segmentation using region based active contour model. In *Proc. of the European Conference on Computer Vision*, pages 304–315, Prague, Czech Republic, 2004.
79. V. J. Wedeen, T. G. Reese, D. S. Tuch, J. G. Dou, R. M. Weiskoff, and D. Chessler. Mapping fiber orientation spectra in cerebral matter with fourier-transform diffusioin MRI. In *Proc. of the 7th ISMRM (International Society for Magnetic Resonance in Medicine)*, page 321, Philadelphia, USA, 1999.
80. J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. Technical Report 58, Department of Mathematics, Saarland University, Saarbrucken, Germany, 2002.
81. J. Weickert and H. Hagen (Eds.) *Visualization and Processing of Tensor Fields*. Springer, Berlin Heidelberg, Germany, 2006.
82. J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998.
83. Y. T. Weldeselassie and G. Hamarneh. DT-MRI segmentation using graph cuts. In J. P. W. Pluim and J. M. Reinhardt, editors, *Proceedings of Medical Imaging, SPIE*, San Diego, CA, USA, February 2007.
84. C.-F. Westin and S. E. Maier. A dual tensor basis solution to the stejskal-tanner equations for DT-MRI. In *Proc. of the ISMRM (International Society for Magnetic Resonance in Medicine)*, Honolulu, Hawaii, USA, 2002.
85. C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor mri. *Medical Image Analysis*, 6:93–108, 2002.
86. C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz. Geometrical diffusion measures for MRI from tensor basis analysis. In *Proc. of the ISMRM (International Society for Magnetic Resonance in Medicine)*, page 1742, Vancouver, Canada, Apr. 1997.
87. M. R. Wiegell, D. S. Tuch, H. B. W. Larsson, and V. J. Wedeen. Automatic segmentation of thalamic nuclei from diffusion tensor magnetic resonance imaging. *NeuroImage*, 19:391–401, 2003.
88. S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing*, 33:116–125, 1986.
89. L. Zhukov, K. Museth, D. Breen, R. Whitaker, and A. H. Barr. Level set segmentation and modeling of DT-MRI human brain data. *Journal of Electronic Imaging*, 12(1):125–133, 2003.
90. U. Ziyan, D. Tuch, and C-F Westin. Segmentation of thalamic nuclei from DTI using spectral clustering. In *Proc. of the MICCAI'06, volume 4191 of Lecture Notes in Computer Science*, pages 807–814, Copenhagen, Denmark, 2006.

| Technique | Application | Tensor Distance | Segmentation Type | Pros | Cons |
|---|---|---|---|---|---|
| Peled *et al.*, 1998 [57] | DTI (corpus callosum, internal capsule) | Linear, planar and spherical components | Quantitative analysis (not really segmentation) | Easy implementation | Computational cost (computing eigenvalues/eigenvectors); Noise affecting eigenvalues |
| Zhukov *et al.*, 2003 [89] | DTI (regions with high diffusivity/ anisotropy) | Diffusivity measure $I_1$ Anisotrpy measure $C_a$ | GAC (level sets) | Faster and less sensitive to noise than [57] | GAC needs close inizialization Only scalar values employed Needs to adjust parameters and to combine results heuristically |
| Rousson *et al.*, 2003 [16, 66, 65] | LST (unsupervised segmentation) | Euclidean distance tensor components as vector | GAR(level sets) | Introduces statistical modeling of data Uses all tensor components | Does not take into account the tensor nature of the data |
| Feddern *et al.*, 2003 [26] | DTI | LST of the tensor data | GAC | | Only segmented brain contour in 2D |
| Wiegell *et al.*, 2003 [87] | DTI (talamic nuclei) | Combination of Frobenius tensor distance and Mahalanobis voxel distance | k-means | Simple method | Frobenius distance is not best option |
| Jonasson *et al.*, 2003 [38, 37] | DTI (corpus callosum, corticospinal tract | Normalized tensor scalar product (NTSP) | Level set, curve propagation based on tensor distance | Tensor distance specifically designed | Tensor distance appropriate for other structures? |
| Wang & Vemuri, 2004 [78] | LST | Frobenius | Mumford-Shah functional (level sets) | First method to combine curve evolution and tensor distances | Frobenius distance |

**Table 1** Summary of tensor segmentation methods.

| Technique | Application | Tensor Distance | Segmentation Type | Pros | Cons |
|---|---|---|---|---|---|
| Wang & Vemuri, 2004 [77] | DTI<br>LST | K-L distance | Chan andVese model<br>(level sets) | Better tensor distance<br>than Frobenius | Piecewise constant model |
| Lenglet *et al.*, 2004 [43, 47] | DTI | K-L distance<br>geodesic distance | GAR on the distance<br>to the mean tensor | Incorporates statistical<br>model | Does not directly model<br>the data, but the distances |
| Lenglet *et al.*, 2006 [48, 44, 45] | DTI | Euclidean distance<br>K-L distance<br>Geodesic distance | GAR with Gaussian<br>distribution over tensor<br>fields | Incorporates statistical<br>model directly on the<br>tensor domain | Complicated formulation<br>Needs manual initialization |
| Ziyan *et al.*, 2006 [90] | DTI<br>(thalamus nuclei) | Frobenius distance<br>K-L distance<br>Eigenvectors angular diff. | Spectral clustering with<br>Markovian relaxation | Accurate segmentation | Needs careful delineation<br>of thalamus for<br>initialization |
| de Luis-Garcia & Alberola,<br>2007 [25] | DTI<br>(corpus callosum) | K-L distance<br>Geodesic distance | GAR with mixtures of<br>Gaussians on tensors | More flexible than [48, 44, 45]<br>Accurate results | Complicated model<br>Needs initialization |
| Weldeselasie &<br>Hamarneh, 2007 [83] | DTI<br>(corpus callosum and cardiac) | Log-Euclidean<br>K-L distance | Graph-cuts | Allows for interactive<br>segmentation | Results only in 2D slices |
| Awate *et al.*, 2007 [9, 8] | DTI<br>(cingulum,<br>corticospinal tract) | Log-Euclidean | Fuzzy C-means with<br>nonparametric statistical<br>models | Overcomes Gaussian<br>limitations on bending<br>fiber bundles | Need some manual<br>initialization |

**Table 2** Summary of tensor segmentation methods.

# A Variational Approach to the Registration of Tensor-Valued Images

Sebastiano Barbieri, Martin Welk and Joachim Weickert

**Abstract** A variational framework for the registration of tensor-valued images is presented. The underlying energy functional consists of four terms: a data term modelled on a tensor constancy constraint, a compatibility term which couples domain deformations and tensor reorientation on the basis of a physical model, and regularity terms imposing smoothness of deformation and tensor reorientation fieldss in space. A specific feature of our model is the separation of data and compatibility terms which eases an adaptation to different physical models of tensor deformation. A multiscale gradient descent is used to minimise the energy functional with repect to both transformation fields involved. The viability and potential of the approach in the registration of tensor-valued images is demonstrated by experiments.

## 1 Introduction

In medical imaging it is often necessary to fuse data from multiple images depicting the same structure to gather the necessary information for diagnosis or the planning of surgeries or other treatments. The images can originate from different imaging modalities, but also from the same modality at different times. The task of transforming all images into one common coordinate system is called image registration [10]. Analogous problems appear of course also in other imaging application fields.

In a standard setting for registration using two images one aims at transforming the *template image* in such a way that it can be overlaid to the other one, the *reference image,* that corresponding structures are found at equal locations.

Sebastiano Barbieri
School of Engineering Science, Simon Fraser University, 8888 University Drive, V5A 1S6, Burnaby, BC, Canada, e-mail: sba61@sfu.ca

Martin Welk, Joachim Weickert
Mathematical Image Analysis Group, Saarland University, Campus E1.1, 66123 Saarbrücken, Germany, e-mail: welk@mia.uni-saarland.de, weickert@mia.uni-saarland.de

At a closer look, this task decomposes into two parts. First, a map has to be found that indicates for each point in the domain of the template image the corresponding point in the domain of the reference image. Second, the template image has to be deformed in accordance to this map. The first part is what in computer vision is called a *correspondence problem,* similar to stereo vision or optic flow, while the deformation task comes down to a geometric transformation commonly denoted as *warping.*

Tensor fields [16] are more difficult to register than scalar-valued images, such as computerised tomography or standard magnetic resonance (MR). To understand why this is the case, notice that scalar-valued images can be deformed by simply moving values within the image domain. For sake of discretisation, this will usually require to resample the image by some interpolation, but not more. In contrast, tensors in a tensor field are inherently linked to the geometry of the domain of definition. In order to keep this reference intact while transforming the domain, tensor values have to be adapted. Locally, the relevant deformation data are gathered in the Jacobian of the displacement map. To integrate the transformation of tensors properly into the registration procedure, it needs not only to be used in the warping step but should already be taken into account when computing the correspondence map.

**Our contribution.** This chapter is dedicated to a variational framework for the registration of tensor-valued images in which the correspondence and warping problems are joined in a single minimisation task. A gradient descent procedure is used for finding the minimiser. While a preliminary version of this framework has been presented at the CVPR workshop *Tensors in Image Processing and Computer Vision* [3], our present work improves the compatibility term such that it becomes algorithmically simpler, and provides a substantially extended experimental investigation.

Our approach is characterised by the consistent combination of displacement and reorientation information that allows to make full use of the direction information contained in the tensor data. Although the physical model behind the functional used here is motivated by DT-MRI, the approach itself is fairly generic as it can easily accommodate any change to the physical model. This is a typical advantage of variational methods because they make the assumptions of the model explicit and permit straightforward manipulations on them, and our specific model is especially designed to capitalise on this advantage. For example, while we refrained from modelling empirical parameters describing tissue physiology into the functional, a refined model capturing physiological details could easily be plugged in, but also any completely different transformation behaviour that could possibly arise from the physical nature of some other tensor imaging application.

We present here the variational model and demonstrate experimentally its potential as a versatile tool for image registration of tensor data.

**Related work.** A variety of approaches to DT-MRI registration have been proposed in the literature. In some approaches the correspondence and warping problems are treated independently. Often the correspondence is inferred from derived quantities that to not involve directional information, thus completely avoiding reorientation in this step [12, 13, 15, 19]. In extracting from the location map the reorientation

information for the warping step, a frequent assumption is that only a rotation takes place. This can be motivated from a single-fibre model. Two methods to extract a rotation field are discussed in [1]. One is the *finite-strain* approach based on a polar decomposition of the Jacobian of the domain map. The second one is the *preservation of principal directions* approach where the reorientation is governed by the application of the domain transformation on the principal eigenvector. Another way is to apply the Jacobian directly to the tensors, whose shape is thereby changed [13]. For an overview of these techniques, we refer also to [7].

In [8], an algebraic approach to DT-MRI registration is proposed. Restricting again tensor transformation to rotations, transformation parameters are optimised locally to solve the correspondence problem. Furthermore, the authors discuss a multi-scale refinement and the inclusion of feature correspondences. For another feature-based registration approach see [14].

Another category of approaches aims at using directional information from the beginning, i.e. already in computing the correspondence maps. For example, [17, 18] formulate a diffusion tensor constancy constraint in which an affine transformation of the domain is combined with a rotation of the tensor values based on polar decomposition. A parametric model, namely a piecewise affine transformation, is fitted to the image data. In contrast, [5] proposes a gradient descent method coming from a variational model. Here, tensor transformation is made dependent on the deformation field in the sense of the preservation of principal directions. Another variational model presented in [6] contains a data term that codes the deformation of tensors by the Jacobian of the displacement field.

A comparison of tensor field registration using derived quantities without orientation information to an alternative registration method based on the matrix entries of the tensors is found in [11].

Concerning the related problem of registering DT-MRI data with data from other imaging modalities, mutual information approaches have been studied in [9]. This approach again clearly separates the correspondence and warping tasks. Another approach to registration problems of this kind found in [4] is based on fluid dynamics PDEs.

**Structure of the chapter.** We start by describing our variational model in Section 2. We detail on the energy functional and outline the minimisation procedure. Section 3 is devoted to the experimental evaluation. A series of experiments in 2D on synthetic and DT-MRI data demonstrates basic features of the method. By two 3D experiments the viability of the approach for 3D data is verified, and a comparison to a finite-strain based registration procedure is made. Section 4 contains a summary and outlook.

## 2 The Variational Model

In the following, we consider two tensor fields $R, T : \mathbb{R}^d \to \mathrm{SPD}(d)$. Here, $\mathrm{SPD}(d)$ is the cone of symmetric positive definite real $d \times d$-matrices. We consider the tensor

field $R$ as reference image, $T$ as template image. In order to register these images, we want to find two maps: first, the *displacement field* $u : \mathbb{R}^d \to \mathbb{R}^d$ that describes how the image domain $\mathbb{R}^d$ is transformed between $R$ and $T$, and second, the *reorientation field* $P : \mathbb{R}^d \to G$ that controls the change of the tensor values, and whose range $G \subset \mathrm{GL}(d,\mathbb{R})$ is the matrix group of the admissible transformations.

The purpose of this separation is twofold. On the one hand, the specific dependency between these two fields is not coded into the structure of the variational model, thereby enabling an easy adaptation to a different physical setting. On the other hand, it also simplifies the variational formulation, as no algebraic solution of the equations relating both fields needs to be included in the functional.

## 2.1 Energy Functional

The energy functional to be minimised in our variational registration model reads

$$E[u,P] = D[u,P] + w_1 C[u,P] + w_2 S_u[u] + w_3 S_P[P] \; . \tag{1}$$

It consists of four terms, each of which enforces a specific model requirement by generating a nonnegative penalty for violations of this requirement:

- The *data term* $D$ enforces the match between the transformation fields $u$, $P$ and the given tensor fields $R$, $T$.
- The *compatibility term* $C$ encodes the physically motivated relation between the displacement and reorientation fields.
- The *displacement smoothness term* $S_u$ expresses a regularity assumption for the displacement field.
- The *reorientation smoothness term* $S_P$ promotes regularity of the reorientation field.

The contributions of the four terms are combined with weight factors $w_1$, $w_2$, and $w_3$, thereby balancing the influences of the different model requirements. Let us now discuss the four terms in detail.

**Data term.** If the displacement field $u$ and reorientation field $P$ perfectly match the given data, their application to the template $T$ should become equal to the reference image $R$, which leads us to the *diffusion tensor constancy constraint (DTCC)*

$$(P(x)^{\mathrm{T}})^{-1} T(x+u(x)) P(x)^{-1} = R(x) \; . \tag{2}$$

To make the following mathematical derivations simpler, we distribute the actions of $u$ and $P$ to the two tensor fields, yielding $T(x+u) = P^{\mathrm{T}} R P$, compare also [8, 17].

Penalising deviations from the DTCC by the Frobenius norm $\|\cdot\|_{\mathrm{F}}$ of the matrix difference, and integrating over the image domain $\Omega \subset \mathbb{R}^d$, we arrive at the data term

$$D[u,P] = \int_{\Omega} \|T(x+u(x)) - P(x)^{\mathrm{T}} R(x) P(x)\|_{\mathrm{F}}^2 \, \mathrm{d}x \; . \tag{3}$$

**Compatibility term.** With this term, the relation between displacements and reorientation of tensors is introduced into the functional, which requires to specify a physical model of tensor deformation. The shift field $u$ of the domain acts induces locally a deformation given by the Jacobian $\mathrm{J}(x+u)$ of the location map $x \mapsto x+u(x)$. This deformation acts on the tensors. It is evident that pure translations for which the Jacobian vanishes everywhere leave the tensors unchanged. Generally, the action of $\mathrm{J}(x+u)$ can geometrically be interpreted in terms of the three components rotation, shearing, and scaling. The action of each of these components needs to be specified in the deformation model. We discuss in the following simple deformation models for DT-MRI data.

As to scaling, it can be assumed that absolute diffusivities depend rather on the physiology of the tissue on a microscopic scale, and will therefore not change when the tissue is stretched or shrunk. As a result, scaling of the domain does not change the tensors.

Rotation, however, can reasonably be assumed to act on the tensors directly by accordingly rotating their eigensystems.

It remains to specify the effect of shearing, which is the most controversial component of the model. In the case of DT-MRI, the way in which diffusion tensors in tissue change when this tissue is sheared may depend on the underlying fibre composition and the physiology of the fibres, and can therefore not be inferred unambiguously in the context of the second-order tensor model. We suppose that only small shears occur, and consider two models for the action of shear. In the first one the shearing component of $\mathrm{J}(x+u)$ acts directly on the tensors. In the second version of the model no shearing at all is assumed for the tensors. Formally these models correspond to different matrix groups $G$ as ranges of $P$: For the first model it is the group of all matrices with determinant one, $G = \mathrm{SL}(d,\mathbb{R})$, while for the second model $G$ equals the group of (proper) rotation matrices $G = \mathrm{SO}(d)$.

In both cases, the compatibility term reads

$$C[u,P] = \int_\Omega \|(P\,\mathrm{J}(x+u))^p - \det(\mathrm{J}(x+u))^{p/d} I\|_{\mathrm{F}}^2 \,\mathrm{d}x\,, \tag{4}$$

where $I$ is the unit matrix. By penalising once again deviations via the Frobenius norm of a matrix difference, this term enforces approximate equality between $P$ and the inverse of $\mathrm{J}(x+u)$ up to the scaling component expressed via the determinant. Its minimisation has the form of a least-squares fit. In those models which do not include shear in the tensor transformation $P$, this least-squares fit takes place between the rotation matrix $P \in \mathrm{SO}(d)$ and the rescaled Jacobian from $\mathrm{SL}(d,\mathbb{R})$. As a consequence, also the shear comprised in the Jacobian $\mathrm{J}(x+u)$ is subsumed to the least-squares error in this case.

Note that in (4) matrices are raised to the $p$-th power. In this chapter, the standard value for $p$ will be 1. By keeping the exponent $p$ in the formula we subsume the model from our previous work [3], in which $p = 2$ was used for 2D tensor registration. While this seemed advantageous in terms of registration quality in early experiments, the case $p = 1$ is algorithmically simpler, and experimental exploration

has proven its equal qualitative performance. We will include one experiment for comparison in Section 3.

**Smoothness terms.** Imposing regularity by including smoothness terms in a functional is a standard component in variational models. It allows the model to cope with the inevitable noise in input data.

In our model, we need regularisers for both transformation fields. Regularity of the displacement field $u$ is enforced by penalising its spatial gradient with its Euclidean norm $\|\cdot\|_2$, yielding the smoothness term

$$S_u[u] = \int_\Omega \sum_{j=1}^{d} \|\partial_j u\|_2^2 \,\mathrm{d}x \,. \tag{5}$$

In quite the same way the reorientation field is regularised by

$$S_P[P] = \int_\Omega \sum_{j=1}^{d} \|\partial_j P\|_\mathrm{F}^2 \,\mathrm{d}x \,. \tag{6}$$

## *2.2 Minimisation*

We want to minimise the functional (1) by a gradient descent method. We use therefore the Euler-Lagrange formalism from the calculus of variations to obtain variational derivatives of $E$ with respect to the transformation fields $u$ and $P$. As these fields, however, consist of multi-channel quantities, it is necessary to take the derivatives with respect to suitable sets of parameters. For the displacement field $u$ whose values belong to a vector space, the displacements $u_1, \ldots, u_d$ in the coordinate directions lend themselves as such a parameter set. In the case of $P$, however, the Lie group $\mathrm{SO}(d)$ or $\mathrm{SL}(d,\mathbb{R})$ has to be parametrised, dependent on the deformation model in use.

Restricting ourselves to the 2D case with and without shearing, and the 3D case without shearing, and assuming that deformations are small (thus, $P$ close to $I$), we can use the following parameter sets:

- *Case* $\mathrm{SO}(2)$ *(2D, only rotations):* This is a one-parameter rotation group with one rotation angle $\alpha$, such that

$$P(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} . \tag{7}$$

- *Case* $\mathrm{SL}(2,\mathbb{R})$ *(2D, rotations and shear):* Here, we decompose the action of $P$ into two rotations framing between them a shear with invariant $x$ axis. By choosing the rotation angles as $\frac{1}{2}(\alpha \pm \beta)$, we arrive at the parametrisation

$$P(\alpha,\beta,\gamma) = \begin{pmatrix} \cos\frac{\alpha+\beta}{2} & -\sin\frac{\alpha+\beta}{2} \\ \sin\frac{\alpha+\beta}{2} & \cos\frac{\alpha+\beta}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 2\gamma & 1 \end{pmatrix} \begin{pmatrix} \cos\frac{\alpha-\beta}{2} & -\sin\frac{\alpha-\beta}{2} \\ \sin\frac{\alpha-\beta}{2} & \cos\frac{\alpha-\beta}{2} \end{pmatrix}. \quad (8)$$

Here, $\alpha$ can be interpreted as "net rotation", $\beta$ as shear direction, while $\gamma$ gives the magnitude of the shear.

- *Case* SO(3) *(3D, only rotations):* A sensible parametrisation of 3D rotations is given by Euler angles $\alpha$, $\beta$, $\theta$, such that

$$P(\alpha,\beta,\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (9)$$

To write down the gradient descent equations, we abbreviate by $F$ the integrand of the energy functional,

$$\begin{aligned} F := {} & \|T(x+u(x)) - P(x)^{\mathrm{T}} R(x) P(x)\|_{\mathrm{F}}^2 \\ & + \|(P\mathrm{J}(x+u))^p - \det(\mathrm{J}(x+u))^{p/d} I\|_{\mathrm{F}}^2 + \sum_{j=1}^{d} \|\partial_j u\|_2^2 + \sum_{j=1}^{d} \|\partial_j P\|_{\mathrm{F}}^2 , \end{aligned} \quad (10)$$

and introduce the artificial time variable $t$.

The 2D gradient descent then reads

$$\begin{aligned} \frac{\partial}{\partial t} u_1 &= \frac{\mathrm{d}}{\mathrm{d}x} F_{u_{1x}} + \frac{\mathrm{d}}{\mathrm{d}y} F_{u_{1y}} - F_{u_1} , \\ \frac{\partial}{\partial t} u_2 &= \frac{\mathrm{d}}{\mathrm{d}x} F_{u_{2x}} + \frac{\mathrm{d}}{\mathrm{d}y} F_{u_{2y}} - F_{u_2} , \\ \frac{\partial}{\partial t} \alpha &= \frac{\mathrm{d}}{\mathrm{d}x} F_{\alpha_x} + \frac{\mathrm{d}}{\mathrm{d}y} F_{\alpha_y} - F_{\alpha} , \\ \frac{\partial}{\partial t} \beta &= \frac{\mathrm{d}}{\mathrm{d}x} F_{\beta_x} + \frac{\mathrm{d}}{\mathrm{d}y} F_{\beta_y} - F_{\beta} , \\ \frac{\partial}{\partial t} \gamma &= \frac{\mathrm{d}}{\mathrm{d}x} F_{\gamma_x} + \frac{\mathrm{d}}{\mathrm{d}y} F_{\gamma_y} - F_{\gamma} \end{aligned} \quad (11)$$

with rotations and shear. When no shear is used, the last two equations are omitted. Analogously, the equations in the 3D case read

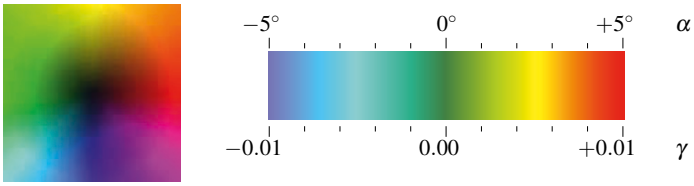



**Fig. 1** Colour coding schemes for visualisation of tensor field transformations. *Left: (a)* Displacement vectors in 2D. *Right: (b)* Scale for rotation angles and shear factors.

$$
\begin{aligned}
\frac{\partial}{\partial t}u_1 &= \frac{\mathrm{d}}{\mathrm{d}x}F_{u_{1x}} + \frac{\mathrm{d}}{\mathrm{d}y}F_{u_{1y}} + \frac{\mathrm{d}}{\mathrm{d}z}F_{u_{1z}} - F_{u_1}\ , \\
\frac{\partial}{\partial t}u_2 &= \frac{\mathrm{d}}{\mathrm{d}x}F_{u_{2x}} + \frac{\mathrm{d}}{\mathrm{d}y}F_{u_{2y}} + \frac{\mathrm{d}}{\mathrm{d}z}F_{u_{2z}} - F_{u_2}\ , \\
\frac{\partial}{\partial t}u_3 &= \frac{\mathrm{d}}{\mathrm{d}x}F_{u_{3x}} + \frac{\mathrm{d}}{\mathrm{d}y}F_{u_{3y}} + \frac{\mathrm{d}}{\mathrm{d}z}F_{u_{3z}} - F_{u_3}\ , \\
\frac{\partial}{\partial t}\alpha &= \frac{\mathrm{d}}{\mathrm{d}x}F_{\alpha_x} + \frac{\mathrm{d}}{\mathrm{d}y}F_{\alpha_y} + \frac{\mathrm{d}}{\mathrm{d}z}F_{\alpha_z} - F_{\alpha}\ , \\
\frac{\partial}{\partial t}\beta &= \frac{\mathrm{d}}{\mathrm{d}x}F_{\beta_x} + \frac{\mathrm{d}}{\mathrm{d}y}F_{\beta_y} + \frac{\mathrm{d}}{\mathrm{d}z}F_{\beta_z} - F_{\beta}\ , \\
\frac{\partial}{\partial t}\theta &= \frac{\mathrm{d}}{\mathrm{d}x}F_{\theta_x} + \frac{\mathrm{d}}{\mathrm{d}y}F_{\theta_y} + \frac{\mathrm{d}}{\mathrm{d}z}F_{\theta_z} - F_{\theta}\ .
\end{aligned}
\tag{12}
$$

To compute these gradient descents, we use explicit (forward Euler) discretisations. As we aim at a steady state, higher order schemes bear no advantage. All spatial derivatives are discretised by Sobel operators.

Note that the evaluation of $T(x+u)$ involves a resampling which is performed here by bilinear interpolation. While some interpolation artifacts implied by this proceeding could be reduced by using a shape interpolation approach, see e.g. [2], bilinear interpolation appears to be closer to the physical sampling process which also leads to partial volume effects in the measurements.

**Multiscale procedure.** Particularly in the presence of displacements that exceed the range of about 1 to 2 pixels, the gradient descent method tends to converge extremely slow, or can even be caught in local minima. We use a multiscale approach to solve this problem: The registration is performed subsequently for a series of image pairs smoothed with Gaussians of decreasing standard deviation, i.e. $(R_k,T_k),(R_{k-1},T_{k-1}),\ldots,(R_0,T_0)$ with $R_0=R$, $T_0=T$ and $R_i=G_{\sigma_i}*R_0$, $T_i=G_{\sigma_i}*T_0$ where $G_\sigma$ denotes the Gaussian of standard deviation $\sigma$, and we choose $\sigma_k>\sigma_{k-1}>\ldots>\sigma_1>0$. The transformation fields $u$ and $P$ in each step $i\leq k-1$ are initialised with the minimisers found in the previous step $i+1$.

## 3 Experiments

First we present a series of experiments in 2D. In the presentation of our results we use a glyph-based representation of tensor fields with ellipse glyphs. The principal axes of each ellipse are oriented in the eigenvector directions of the corresponding tensor, while their lengths equal its eigenvalues. Note that the eigenvalues are real and positive, as we deal with symmetric positive definite matrices.

In visualising displacement fields, we use a colour coding scheme given in Fig. 1(a) to generate colour images. We also represent the scalar quantities, i.e. angles $\alpha$ and shear magnitudes $\gamma$, by colour images. The colour scale for this is shown in Fig. 1(b).

Weight parameters have been chosen manually in all of our experiments. Further investigation will be necessary to address a possible automatic choice of these parameters.

### *3.1 Synthetic 2D Experiment*

In the experiment in Fig. 2, we use a synthetic 2D data set containing a fibre-like structure as reference image (a). The template image (b) has been generated by rotating consistently domain and tensors of the reference image by $3^\circ$. A caveat about this test image pair is that the correspondence map is dramatically underdetermined.

For registration, we set the weights in (1) as $w_1 = 2 \cdot 10^6$, $w_2 = 5 \cdot 10^7$, $w_3 = 4 \cdot 10^7$. Our complete 2D model with rotation and shear yields the registered image shown in Fig. 2(c). No multiscale procedure has been used here. In Fig. 2(d–f) we show the displacement field and two components of the reorientation field. Due to the mentioned underdetermination of the example, the model attains an optimum with nonzero shear. In particular, the vertically periodic structures visible in Fig. 2(e, f) indicate that the process slightly overfits and matches aliasing-type structures generated by the resampling of the rotated data. Fig. 2(g–i) show the results of registration with the rotation-only model.

As a measure of registration quality we use the value of the data term $D[u,P]$ for the computed displacement and reorientation fields. By this means, we can compare different registration models. As this measure could, however, be misled by overfitting, its utilisation is restricted to cases with equal smoothness parameters. Table 1

**Table 1** Data term values measured to assess registration quality for the fibre images from Fig. 2(a,b) by different models.

| | |
|---|---|
| Model with rotation and shear | $2.14 \cdot 10^4$ |
| Model with rotation only | $3.04 \cdot 10^4$ |
| Model without reorientation | $4.60 \cdot 10^4$ |
| Ground truth | $1.29 \cdot 10^4$ |

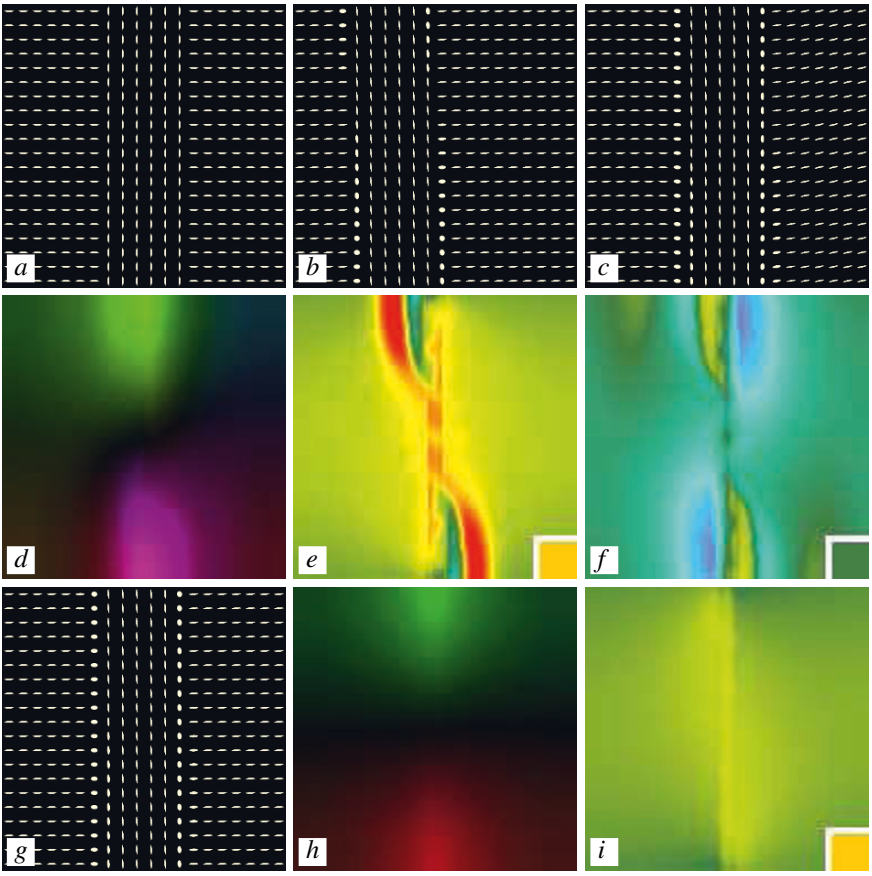



**Fig. 2** Registration of a synthetic tensor field. *(a) Top left:* Clipping (rows 0 to 19, columns 54 to 73) from a $127 \times 127$ synthetic tensor field featuring a simplified "fibre", used as reference image. *(b) Top middle:* Template image obtained by 3° rotation of (a). *(c) Top right:* Result of variational registration with deformation model involving rotation and shear. *(d) Second row, left:* displacement field computed in the registration process (all $127 \times 127$ pixels). *(e) Second row, middle:* Rotation angle $\alpha$ from the reorientation field. *(f) Second row, right:* Shear factor $\gamma$. *(g) Bottom left:* Registration result using the deformation model involving only rotations. *(h) Bottom middle:* displacement field computed in registration with rotation-only model. *(i) Bottom right:* Rotation angle $\alpha$ from rotation-only model. Inserts in (e), (f) and (i) show correct parameter values (3°, 0.0).

comprises values of the data term for registration with and without shearing and, for comparison, for a model that does not account for reorientation as well as for the known correct displacement and reorientation field (ground truth). The nonzero error measured in the latter case is caused by the resampling in generating the data and in the registration itself.

As expected, the model with reorientation reduces the error considerably. Since by construction the data fit a pure rotation model, it is natural that the gain by includ-

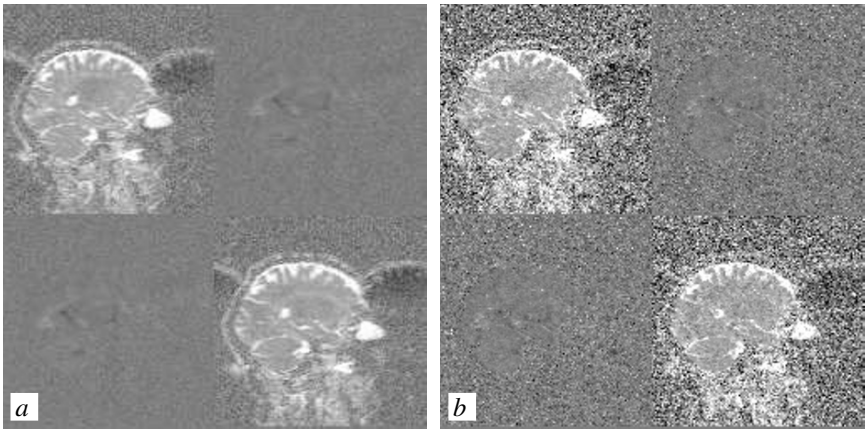



**Fig. 3** $127 \times 127$ fields of 2D tensors extracted from two DT-MRI data sets of the same human brain. The data set in (a) was obtained with 10 times averaging, while that in (b) was measured without averaging, which leads to a great difference in noise level.

ing shear is smaller. Nevertheless, the underdetermination of the correspondence allows a further improvement by a nonzero shear. [1]

## *3.2 Real-World 2D Experiments*

We continue by three experiments based on 2D tensor fields from diffusion tensor imaging. From two DT-MRI data sets of the same human brain, a single plane of $2 \times 2$ tensors is extracted, using only those components of the original $3 \times 3$ tensors that belong to the section plane. Fig. 3 shows the images in a grey-value representation, where each of the $2 \times 2$ tiles of each image represents one matrix component.

In analogy to our synthetic experiment, we register first one image to a rotated version of the same image. The original image from Fig. 3(b) serves as reference, and a copy rotated by $3^\circ$ as template. As the noise level is higher than in the synthetic experiment, we choose $w_1 = 3.6 \cdot 10^7$, $w_2 = 9 \cdot 10^7$, $w_3 = 7.2 \cdot 10^8$. From now on we use the multiscale procedure with $k = 4$ and standard deviations $4.0, 2.0, 1.0, 0.5$.

The resulting displacement and reorientation estimates are shown in Fig. 4 alongside with detail views of the reference, template, and registered image. In Table 2 data term measurements are given that allow a judgement on the registration quality. It can be seen in the colour-coded display, Fig. 4(e,f) that the angle and shear estimates in most of the image area approximate the correct (ground truth) values

[1] In [3], where $p = 2$ was used in the compatibility term, data term measurements for the registration results were significantly lower than those reported here. It is only in this synthetic example that such a large difference between $p = 1$ and $p = 2$ is observed. It can be ascribed to the dramatic underdetermination of the correspondence map by the given data.

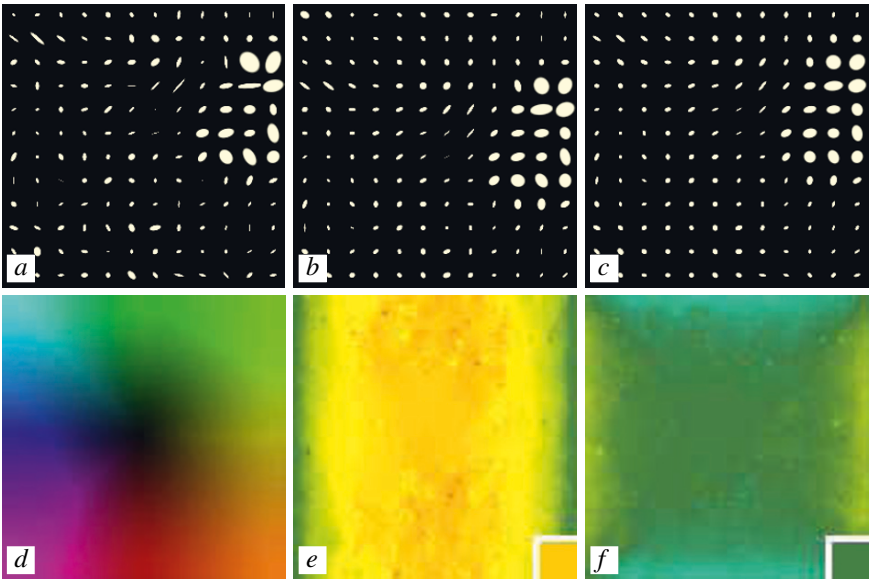



**Fig. 4** Registration of a $127 \times 127$ field of 2D tensors extracted from a DT-MRI dataset of a human brain as reference image and a copy rotated by 3° as template. *(a) Top left:* Reference image, rows 55 to 66, columns 29 to 40 shown. *(b) Top middle:* Template image, same area. *(c) Top right:* Registered image, same area. *(d) Bottom left:* displacement field. *(e) Bottom middle:* Rotation angle from the reorientation field. *(f) Bottom right:* Shear factor. Inserts in (e), (f) show ground truth parameter values.

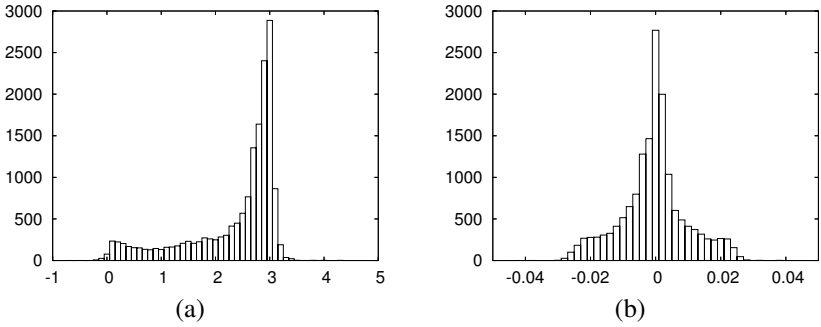



**Fig. 5** Histograms of estimated deformation parameters for 2D tensors. *(a) Left:* Rotation angles from Fig. 4(e). *(b) Right:* Shear factors from Fig. 4(f).

3° and 0, respectively. Histograms shown in Fig. 5 confirm this. Significant deviations occur only in those outer regions of the data set which are dominated by noise tensors close to zero.

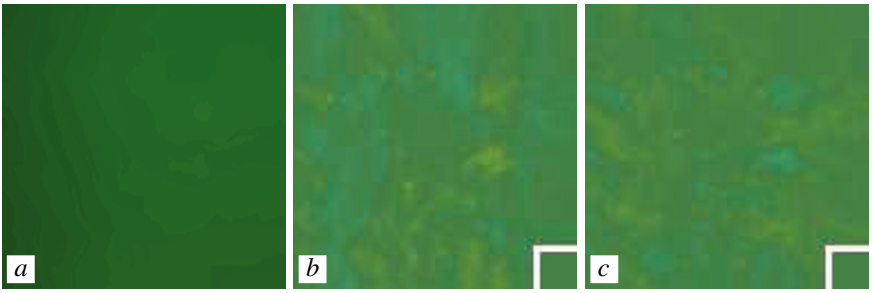



**Fig. 6** Registration of two $127 \times 127$ fields of 2D tensors extracted from different DT-MRI data sets of a human brain. *(a) Left:* Colour-coded displacement field that shows an approximately constant shift between the two images. *(b) Middle:* Rotation angle from the reorientation field. *(c) Right:* Shear factor from the reorientation field. – Inserts in (b), (c) show ground truth values.

While the improvement by a model accounting for reorientation is again evident, the remaining mismatch is much larger than in the synthetic model due to the high spatial variation of the data and resulting large resampling error.

We turn now to register the two different 2D DT-MRI images onto each other. As both data sets were acquired subsequently during the same MR session, there is only a slight displacement, and virtually no reorientation between them. Using the same parameters as before, we obtain the values shown in Fig. 6. In this case, there is no significant influence of the reorientation, which indicates that also no substantial overfitting takes place.

We end our 2D experiments by registering again the two different 2D DT-MRI images as before, but additionally rotating the reference image by $3^\circ$. Using the same weight parameters as before, we obtain the results compiled in Table 3 and Fig. 7. In the top row (standard setting), a good capture of rotation is achieved in the central region where most anisotropic tensors reside (white matter). One observes,

**Table 2** Data term measurements for the registration of a 2D DT-MRI data set and its rotated version by different models.

| | |
|---|---|
| Model with rotation and shear | $1.02 \cdot 10^6$ |
| Model with rotation only | $1.70 \cdot 10^6$ |
| Model without reorientation | $2.39 \cdot 10^6$ |
| Ground truth | $0.89 \cdot 10^6$ |

**Table 3** Data term measurements for the registration of two 2D DT-MRI data sets, one of them rotated, by different models.

| | |
|---|---|
| Model with rotation and shear | $1.16 \cdot 10^6$ |
| Model with rotation only | $1.42 \cdot 10^6$ |
| Model without reorientation | $1.51 \cdot 10^6$ |

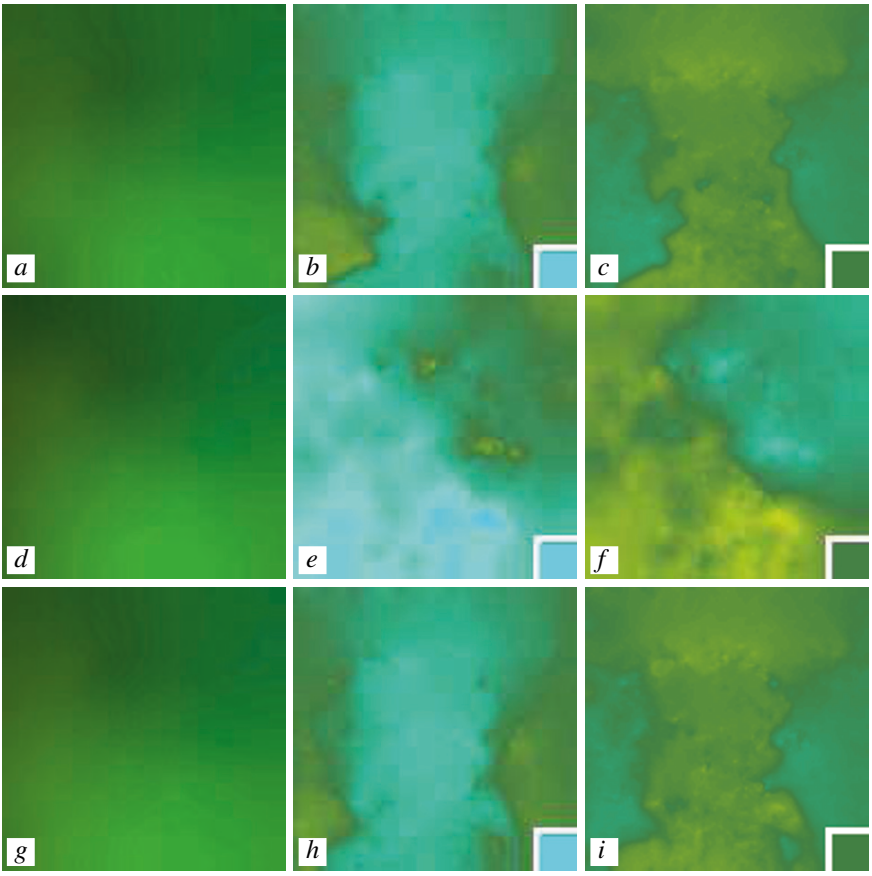



**Fig. 7** Registration of two $127 \times 127$ fields of 2D tensors extracted from different DT-MRI data sets of a human brain. The test images were the same as in Fig. 6 but with the reference image rotated by 3°. *Top row:* Standard model with exponent $p = 1$ and parameters $w_1 = 3.6 \cdot 10^8$, $w_2 = 9 \cdot 10^7$, $w_3 = 7.2 \cdot 10^8$. *(a) Top left:* displacement field showing a rotation superposed to the constant shift. *(b) Top middle:* Rotation angle. *(c) Top right:* Shear factor. *Middle row, (d)–(f):* Same but with suppressed compatibility term, $w_1 = 0$. While the displacement field (d) is almost identical to (a), both deformation components are substantially misestimated. *Bottom row, (g)–(i):* Model with $p = 2$ (compare [3]) and parameters $w_1 = 3.6 \cdot 10^7$, $w_2 = 9 \cdot 10^7$, $w_3 = 7.2 \cdot 10^8$. Note that the compatibility weight $w_1$ is rescaled to accommodate the change in the compatibility energy. Inserts in angle and shear fields show ground truth values.

however, that angles are somewhat underestimated. The middle row demonstrates the importance of the compatibility term in our model: By setting its weight to zero, the displacements are still fairly well estimated but deformations are estimated reasonably only in a part of the image domain. Finally, the bottom row shows a result obtained by the model from [3] with exponent $p = 2$ in the compatibility term. With a suitably adapted value of the compatibility weight $w_1$, results of both models (top row and bottom row) look almost identical.

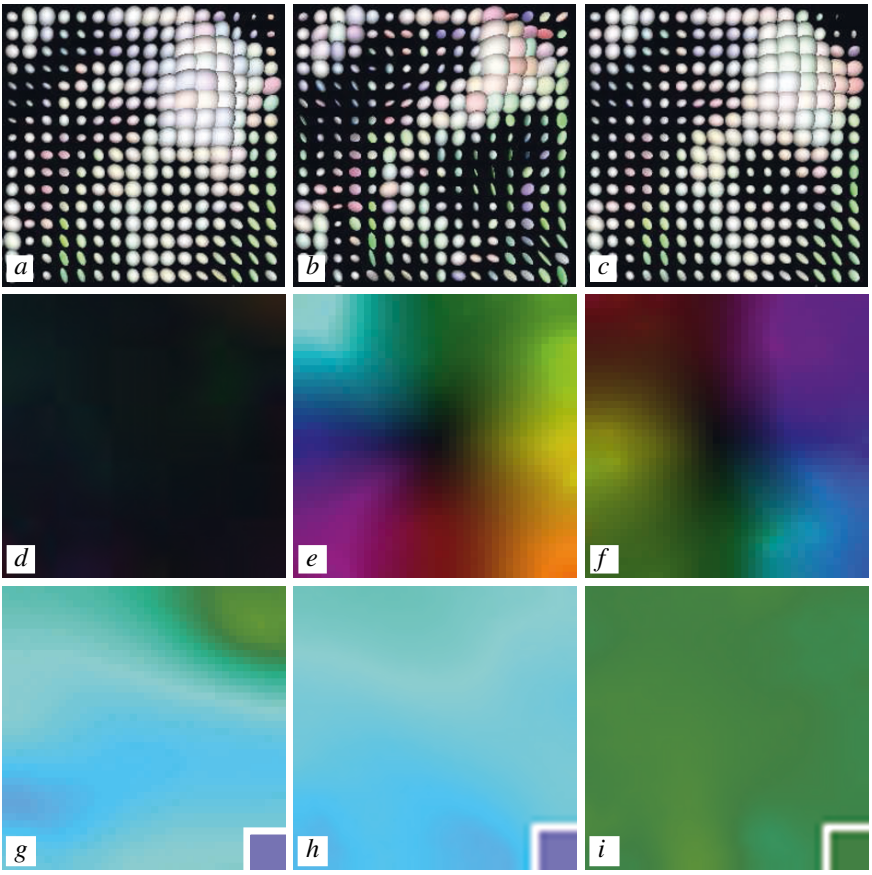



**Fig. 8** Registration of 3D tensor data. A $40 \times 40 \times 40$ volume from a human brain DT-MRI data set serves as reference image. The template is a copy of the same data, rotated first by 5° around the *x*-axis and then by 5° around the *y*-axis. The reorientation model allows only rotations. *Top row: (a) Left:* Voxels 20 to 35 in *x* direction, 5 to 20 in *y* direction from layer 30 in *z* direction of the reference volume. *(b) Middle:* Same voxels from the template volume. *(c) Right:* Same voxels from the registered volume. *Second row: (d) Left:* displacement field in the *x*-*y* central plane. *(e) Middle:* Displacement field in the *x*-*z* central plane. *(f) Right:* Displacement field in the *y*-*z* central plane. *Bottom row:* Euler angles in the *x*-*y* central plane. *(g) Left:* Angle $\alpha$. *(h) Middle:* Angle $\beta$. *(i) Right:* Angle $\theta$. Inserts in (g)–(i): ground truth.

### *3.3 Experiments with 3D Data*

In our first 3D experiment (Fig. 8) we register a rotated volume taken from one of our DT-MRI data sets onto its unrotated counterpart, using the $\mathrm{SO}(3)$ reorientation model. For the visualisation of exemplary data we use ellipsoid glyphs analogous to the ellipse glyphs mentioned above for the 2D case. For the transformation fields we apply the 2D colour coding within the planes shown. The results show that the ho-

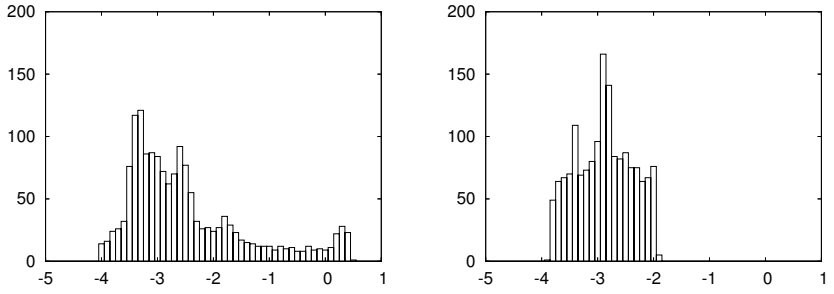



**Fig. 9** 3D tensor registration results. *(a) Left:* Histogram of Euler angles $\alpha$ from Fig. 8(g). *(b) Right:* Histogram of Euler angles $\beta$ from Fig. 8(h).

mogeneous rotation field and corresponding displacements are captured well, again with an underestimation of reorientation angles, see also the histograms in Fig. 9.

Our second 3D experiment (Fig. 10) consists in registering equally sized volumes from both our DT-MRI data sets. First we use our variational model with $\mathrm{SO}(3)$ reorientation. Similarly as in the registration of 2D sections, the displacement field is dominated by an almost constant translation, whereas the rotation angles range around zero, with absolute values not exceeding $0.1^\circ$, see also the histogram in Fig. 11(a). In order to visualise the small deviations from zero, the angles have been amplified by a factor 10 before applying the colour scale.

For comparison, we register the same volumes by a different model: Gradient descent is performed only for the displacement field, while the reorientation component is computed by the finite-strain model in each iteration step. The resulting Euler angles are shown in the bottom row of Fig. 10. It can be seen that the reorientation fields are less smooth than before and vary in fact up to absolute values of approx. $0.3^\circ$, compare also the histogram in Fig. 11(b).

## 4 Summary and Outlook

In this chapter, we have described a variational approach to the registration of tensor data. Using an energy functional consisting of separate terms for data fidelity, compatibility, and smoothness of displacement and reorientation fieldss, this model is apt for a flexible adjustment to possible different physical models of tensor deformation. We have presented an experimental demonstration of the applicability of our model for synthetic and DT-MRI-based real-world data.

A more extensive evaluation of the model in 3D and its comparison to other registration approaches are part of our ongoing work. We also aim at investigating the use of the framework with other physical compatibility models.

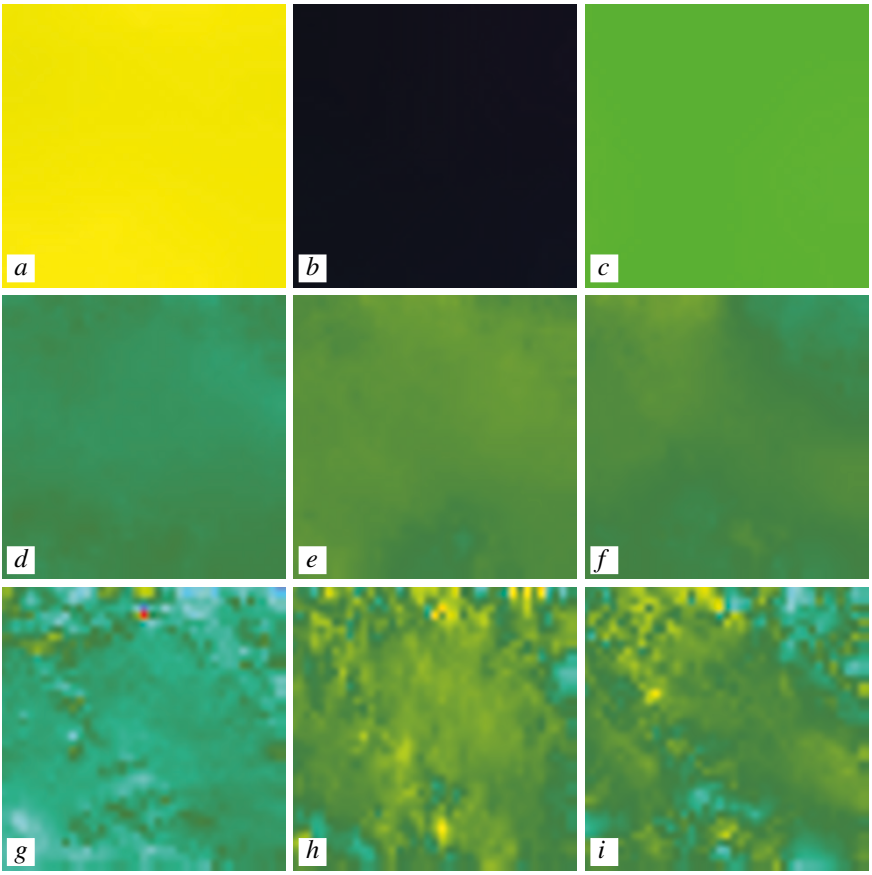



**Fig. 10** Registration of 3D tensor data. Reference and template were $40 \times 40 \times 40$ regions of the 3D data sets from which also the 2D data in Fig. 3 have been extracted. A $40 \times 40 \times 40$ volume from one human brain DT-MRI data set is registered onto an equally sized volume from a different DT-MRI measurement of the same subject. *Top row: (a) Left:* displacement field in the *x*-*y* central plane. *(b) Middle:* Displacement field in the *x*-*z* central plane. *(c) Right:* Displacement field in the *y*-*z* central plane. *Second row:* Euler angles in the *x*-*y* central plane, amplified by a factor 10 (i.e., the colour scale from Fig. 1(b) covers the interval $[-0.5°, 0.5°]$). *(d) Left:* Angle $\alpha$. *(e) Middle:* Angle $\beta$. *(f) Right:* Angle $\theta$. *Bottom row:* Same Euler angles for a registration based on a finite-strain model (see text). *(g) Left:* Angle $\alpha$. *(h) Middle:* Angle $\beta$. *(i) Right:* Angle $\theta$.

**Acknowledgements** We thank O. Gruber and I. Henseler from Saarland University Hospital, Homburg, for providing the DT-MRI measurements. The visualisation tools for tensor images were written by S. Didas and L. Pizarro.

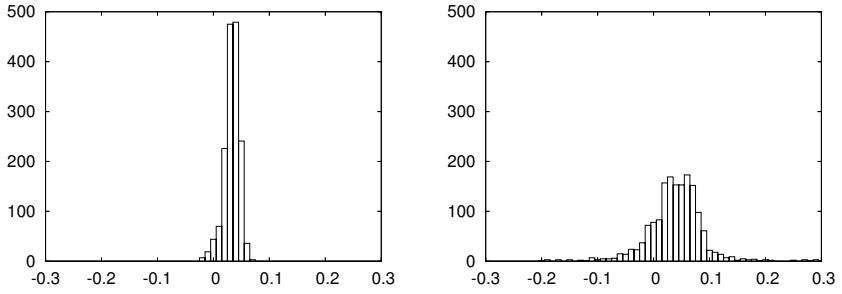



**Fig. 11** Comparison of 3D tensor registration results. *(a) Left:* Histogram of Euler angles $\beta$ estimated by our variational method, see Fig. 10(e). *(b) Right:* Histogram of Euler angles $\beta$ estimated by finite-strain model, see Fig. 10(h).

# References

1. D. C. Alexander, C. Pierpaoli, P. J. Basser, and J. C. Gee. Spatial transformations of diffusion tensor magnetic resonance images. *IEEE Transactions on Medical Imaging*, 20:1131–1139, 2001.
2. V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 20(2):411–421, 2006.
3. S. Barbieri, M. Welk, and J. Weickert. Variational registration of tensor-valued images. In *Proc. CVPR Workshop on Tensors in Image Processing and Computer Vision*, Anchorage, Alaska, USA, 2008.
4. C. Brun, N. Lepore, X. Pennec, Y.-Y. Chou, O. L. Lopez, H. J. Aizenstein, J. T. Becker, A. W. Toga, and P. M. Thompson. Comparison of standard and Riemannian fluid registration for tensor-based morphometry in HIV/AIDS. In *MICCAI Workshop on Statistical Registration: Pair-Wise and Group-Wise Alignment and Atlas Formation*, 2007.
5. Y. Cao, M. I. Miller, S. Mori, R. L. Winslow, and L. Younes. Diffeomorphic matching of diffusion tensor images. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 67, Washington, DC, USA, 2006. IEEE Computer Society.
6. O. Faugeras, C. Lenglet, T. Papadopoulo, and R. Deriche. Non rigid registration of diffusion tensor images. Technical Report 6104, INRIA Sophia-Antipolis, France, 2007.
7. J. C. Gee and D. C. Alexander. Diffusion-tensor image registration. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 327–342. Springer, Berlin, 2006.
8. A. Goh and R. Vidal. Algebraic methods for direct and feature based registration of diffusion tensor images. In H. Bischof, A. Leonardis, and A. Pinz, editors, *Computer Vision – ECCV 2006, Part III*, volume 3953 of *Lecture Notes in Computer Science*, pages 514–525. Springer, Berlin, 2006.
9. C. A. Kemper. Incorporation of diffusion tensor MRI in non-rigid registration for image-guided neurosurgery. Master's thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, Massachusetts, USA, 2003.
10. J. Modersitzki. *Numerical Methods for Image Registration*. Oxford University Press, Oxford, 2004.

11. H.-J. Park, M. Kubicki, M. E. Shenton, A. Guimond, R. W. McCarley, S. E. Maier, R. Kikinis, F. A. Jolesz, and C.-F. Westin. Spatial normalization of diffusion tensor MRI using multiple channels. *NeuroImage*, 20(4):1995–2009, 2003.
12. G. K. Rohde, S. Pajevic, C. Pierpaoli, and P. J. Basser. A comprehensive approach for multi-channel image registration. In J. C. Gee, J. B. A. Maintz, and M. W. Vannier, editors, *Biomedical Image Registration*, volume 2717 of *Lecture Notes in Computer Science*, pages 214–223. Springer, Berlin, 2003.
13. J. Ruiz-Alzola, C.-F. Westin, S. K. Warfield, C. Alberola, S. Maier, and R. Kikinis. Nonrigid registration of 3D tensor medical data. *Medical Image Analysis*, 6(2):143–161, 2002.
14. R. Sierra. Nonrigid registration of diffusion tensor images. Master's thesis, Dept. of Electrical Engineering, ETH, Zürich, Switzerland, 2001.
15. R. Verma and C. Davatzikos. Matching of diffusion tensor images using Gabor features. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 396–399, Arlington, VA, USA, 2004. IEEE Computer Society Press.
16. J. Weickert and H. Hagen, editors. *Visualization and Processing of Tensor Fields*. Springer, Berlin, 2006.
17. H. Zhang, P. Yushkevich, and J. Gee. Deformable registration of diffusion tensor MR images with explicit orientation optimization. In J. Duncan and G. Gerig, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, volume 3749 of *Lecture Notes in Computer Science*, pages 725–732. Springer, Berlin, 2005.
18. H. Zhang, P. A. Yushkevich, D. C. Alexander, and J. C. Gee. Deformable registration of diffusion tensor MR images with explicit orientation optimization. *Medical Image Analysis*, 10(5):764–785, 2006.
19. H. Zhang, P. A. Yushkevich, and J. C. Gee. Towards diffusion profile image registration. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 324–327, Arlington, Virginia, USA, 2004.

# Quality Assessment of Tensor Images

Emma Muñoz-Moreno, Santiago Aja-Fernández and Marcos Martin-Fernandez

**Abstract** The evaluation of tensor image processing algorithms is an open problem that has not been broadly handled, and specific measures have not been described to assess the quality of tensor images. In this chapter, we propose the adaptation of quality measures that have been defined in the case of conventional scalar images to the tensor case, in order to evaluate the quality of the tensor images that are most frequently used in the image processing field. Special attention is paid to the tensor features that made this extension no straightforward. Some general concepts that should be taken into account for the definition of quality indexes for tensor images based on the well-known measures for conventional scalar images are detailed. Then, some of these measures are adapted to deal with tensor images and their behavior is analyzed by means of some examples. Thus, it is shown that structure based measures outperform point-wise measures, as well as the influence of handling all the tensor components.

## 1 Introduction

Due to emerging modalities and advanced processing techniques, tensor data usage has become frequent in the image processing field. We can highlight its application in medical image, where new modalities, such as diffusion tensor imaging (DTI) [3] or tensor elastography [20], allow to visualize structures or tissue properties that are not visible with conventional image techniques. DTI is based on measures of the diffusion anisotropy in the tissues and it has been mainly used in brain imaging, since the nerve fibers can be identified as areas where diffusion is clearly anisotropic. It is also used to analyze the microstructure of muscles such as the myocardium [23] or cartilage [12]. The elastography, on the other hand, is a technique that analyzes the

Emma Muñoz-Moreno, Santiago Aja-Fernández and Marcos Martin-Fernandez
Laboratory of Image Processing (LPI), University of Valladolid, Spain,
e-mail: emunmor@lpi.tel.uva.es, sanaja@tel.uva.es, marcma@tel.uva.es

elastic properties of tissues. These properties can be described by means of tensors, such as the strain and the vorticity tensors, and for this reason, the advanced elastography techniques deal with tensor images. Moreover, the mechanical properties of the tissues can be represented by fields of strain or stress tensors, which can be useful, for instance, to analyze the motion of the heart muscles in the heartbeat [16].

In order to obtain a tensor image, some processing must be carried out to estimate the tensor at each voxel after the acquisition procedure. In consequence, not only acquisition noise but artifacts related to the tensor estimation may appear in the final image. Therefore, likewise in conventional scalar images, many filtering and denoising algorithms have been defined to improve the image quality. As well, other processing techniques, such as registration algorithms, have been proposed for this sort of data. In any case, the evaluation of the algorithm performance requires the definition of quality measures that describe the goodness of the resulting images. This quality evaluation is not trivial, even when working with scalar images, and it becomes more complex when dealing with tensor images. In tensor image processing literature, results are evaluated both quantitatively and qualitatively. Quantitative approaches provide a numerical value of the quality of an image, usually by comparing it with a reference image. When estimating the quality of a tensor image, part of the tensor information is usually discarded, since the tensorial magnitude is often reduced to a scalar value that measures a given property. Once a scalar value is computed for each voxel, it is possible to apply well-known quality or error measures to the obtained images. However, this scalar-based measure may not reflect the real quality of the tensor image. Qualitative approaches, on the other hand, are based on quality assessment by visual inspection. Obviously, this evaluation is subjective and therefore it does not allow to establish rigorous comparison among different processing methods. Moreover, as in quantitative approaches, this visual comparison is usually carried out over intensity images that only represent one of the tensor components or some scalar magnitude related to the tensor, ignoring again part of the tensor information.

All in all, it seems necessary to define measures that take into account every tensor component to describe more realistically the quality of tensor images. Accordingly, in this chapter, we provide an approach to the definition of this kind of measures, based on the existing quality measures for scalar images.

Quality assessment methods for scalar images can be roughly classified into full-reference methods [30], that compare the image with a reference image, and blind-quality methods, which do not use reference images. This chapter is more focused on the former, the so-called full-reference techniques that can be divided into point-wise and structural measures. Tensor point-wise comparison can be found in some tensor image processing works. However, this kind of measures have several drawbacks that have been widely reported in the case of scalar images. They are very sensitive to changes in the intensity range, although these changes may not necessarily affect the image structure. This is also the case when working with tensor images [18]. For this reason, the definition of quality measures based on structural information becomes necessary. Different structure-based quality measures have been

defined to evaluate scalar images. The issues that should be considered to adapt such measures to tensor images are further described.

In the remainder, the chapter is structured as follows: in Section 2, a review of quality measures for scalar images is presented; next, in Section 3 we include a brief definition of what a tensor is, and a description of the most used tensors in the image processing field. Then, in Section 4, the special issues which the definition of quality measures for tensor data involves are highlighted, and some examples of quality measures are defined for tensor images. These measures are analyzed by a set of experiments described in Section 5 and finally, results are discussed in Section 6 and conclusions are drawn in Section 7.

## 2 Background on Quality Assessment in Image Processing

Images can suffer distortion due to several sources, from the acquisition process itself to compression, transmission through noisy channels and others. On the other hand, images can also undergo quality improvement processes, like enhancement or restoration techniques [15]. In each case it is useful to quantify the *quality* of the resulting image. The approaches that do it using a reference image to carry out this task are known as *full-reference methods* [30]. Within them, the most frequently used are those based on error measures, as the Mean Squared Error (MSE) [10, 27].

Let $I(\mathbf{x})$ be the *ground truth image* or *golden standard* and $J(\mathbf{x})$ an image we want to compare with the former. The MSE is defined by

$$\mathrm{MSE}(I,J) = \frac{1}{|\Omega|} \sum_{\mathbf{x}\in\Omega} (I(\mathbf{x}) - J(\mathbf{x}))^2, \tag{1}$$

where $|\Omega|$ denotes the cardinality of space $\Omega$, i.e., the number of elements in image $I(\mathbf{x})$. Although the MSE gives a measure of pixelwise similarity between the images, it does not explicitly take into account any structural information in the images or any sort of subjective measure. Variations of the former are sometimes considered [10], such as the Structural Content (SC)

$$\mathrm{SC}(I,J) = \frac{\frac{1}{|\Omega|} \sum_{\mathbf{x}\in\Omega} [I(\mathbf{x})]^2}{\frac{1}{|\Omega|} \sum_{\mathbf{x}\in\Omega} [J(\mathbf{x})]^2} \tag{2}$$

or the Peak MSE (PMSE)

$$\mathrm{PMSE}(I,J) = \frac{\frac{1}{|\Omega|} \sum_{\mathbf{x}\in\Omega} (I(\mathbf{x}) - J(\mathbf{x}))^2}{\left(\max_{\mathbf{x}} I(\mathbf{x})\right)^2}. \tag{3}$$

The limitations of such methods have been widely reported in the literature. Consequently, some additional variations of the MSE have also been used in order to

better deal with the features of the Human Visual System [10, 27], like the Laplacian MSE (LMSE)

$$\mathrm{LMSE}(I,J) = \frac{\frac{1}{|\Omega|}\sum_{\mathbf{x}\in\Omega}\left(H(I(\mathbf{x})) - H(J(\mathbf{x}))\right)^2}{\frac{1}{|\Omega|}\sum_{\mathbf{x}\in\Omega}\left(H(I(\mathbf{x}))\right)^2} \tag{4}$$

where the images are previously high-pass filtered with the high-pass operator $H(\cdot)$.

In [22], Przelaskowski proposes a vector of six components, mainly based on errors between the ground truth and the degraded image, such as the average pixel error, correlated errors or preservation of high contrast edges. Methods proposed in [8] are also mainly based on error measurements. Other approaches, as the one in [9], assess the quality using a degradation model.

In [17] a new index is proposed, namely, the objective Picture Quality Scale (PQS), basically intended to measure the degradation in coding and compression of images. It takes into account properties of visual perception of both global features and of disturbances. It turns out to be bounded, with a maximum value of 5.797, obtained when an image is compared with itself. Experiments show that although it is a good measure when dealing with compression, it is not so good for other sources of degradation [26]. Recently, some methods based on *Natural Scene Statistics* have also been reported [25, 24].

In [30] Wang *et al.* proposed a full-reference quality assessment method based on the structural similarity of two images, the so-called Structural Similarity (SSIM) index. The method is a modification of their Quality Index, originally proposed in [29]. This method has proved to be versatile and robust in many different environments [26]. It uses three levels of pixelwise comparison:

1. Luminance comparison:

$$l(I,J) = \frac{2\mu_I(\mathbf{x})\mu_J(\mathbf{x}) + C_1}{\mu_I^2(\mathbf{x}) + \mu_J^2(\mathbf{x}) + C_1} \tag{5}$$

   with $\mu_I(\mathbf{x})$ and $\mu_J(\mathbf{x})$ the local means of the images $I(\mathbf{x})$ and $J(\mathbf{x})$ respectively, and $C_1$ a constant.
2. Contrast comparison:

$$c(I,J) = \frac{2\sigma_I(\mathbf{x})\sigma_J(\mathbf{x}) + C_2}{\sigma_I^2(\mathbf{x}) + \sigma_J^2(\mathbf{x}) + C_2} \tag{6}$$

   with $\sigma_I(\mathbf{x})$ and $\sigma_J(\mathbf{x})$ the local standard deviations of the images $I(\mathbf{x})$ and $J(\mathbf{x})$, respectively, and $C_2$ a constant.
3. Structure comparison:

$$s(I,J) = \frac{\sigma_{IJ}(\mathbf{x}) + C_3}{\sigma_I(\mathbf{x})\sigma_J(\mathbf{x}) + C_3} \tag{7}$$

with $\sigma_{IJ}(\mathbf{x})$ the local covariance between the images $I(\mathbf{x})$ and $J(\mathbf{x})$, and $C_3$ a constant.

The local SSIM index then is defined as

$$\mathrm{SSIM}(I,J) = [l(I,J)]^{\alpha} \cdot [c(I,J)]^{\beta} \cdot [s(I,J)]^{\gamma} \tag{8}$$

and with a proper parameter selection [30] it becomes

$$\mathrm{SSIM}(I,J) = \frac{(2\mu_I(\mathbf{x})\mu_J(\mathbf{x}) + C_1)(2\sigma_{IJ}(\mathbf{x}) + C_2)}{(\mu_I^2(\mathbf{x}) + \mu_J^2(\mathbf{x}) + C_1)(\sigma_I^2(\mathbf{x}) + \sigma_J^2(\mathbf{x}) + C_2)}. \tag{9}$$

The overall value is obtained using the mean of the local SSIM (with acronym MSSIM):

$$\mathrm{MSSIM}(I,J) = \frac{1}{|\Omega|} \sum_{\mathbf{x}\in\Omega} \mathrm{SSIM}(I(\mathbf{x}), J(\mathbf{x})). \tag{10}$$

Some variations of the original method have been proposed elsewhere, like using a weighted sum instead of the mean [31],

$$\mathrm{WSSIM}(I,J) = \frac{\sum_{\mathbf{x}\in\Omega} W(\mathbf{x})\mathrm{SSIM}(I(\mathbf{x}), J(\mathbf{x}))}{\sum_{\mathbf{x}\in\Omega} W(\mathbf{x})} \tag{11}$$

with $W(\mathbf{x})$ a weighting function that can be, for example:

$$W(\mathbf{x}) = \sigma_I^2(\mathbf{x}) + \sigma_J^2(\mathbf{x}) + C_2. \tag{12}$$

A somehow different approach is the one by Weken *et al.*; they use fuzzy similarity measures as a way to compare two images. In [34, 35, 36] many distances have been defined, tested and used, either over the images themselves or over their histograms. For instance, using the fuzzy Minkowski distance

$$S_1(I,J) = 1 - \left( \frac{1}{|\Omega|} \sum_{\mathbf{x}\in\Omega} |I(\mathbf{x}) - J(\mathbf{x})|^r \right)^{1/r} \tag{13}$$

or a modified version

$$M_3(I,J) = 1 - \frac{\sum_{\mathbf{x}\in\Omega} |I(\mathbf{x}) - J(\mathbf{x})|}{\sum_{\mathbf{x}\in\Omega} (I(\mathbf{x}) + J(\mathbf{x}))}. \tag{14}$$

To be considered as fuzzy sets, the images must be normalized by their maximum value. The measures are bounded, giving an index in the interval $[0,1]$.

In [1] an alternative method is proposed. It is based on the distribution of the local variance of the data. According to the authors, the quality indexes given by this method correspond more closely to those expected from subjective visual as-

sessment (concerning structural information) than methods previously reported. The index is based on the assumption that a great amount of the structural information of an image is encoded in the distribution of its local variance. The Quality Index based on Local Variance (QILV) between two images $I(\mathbf{x})$ and $J(\mathbf{x})$ is defined as

$$\mathrm{QILV}(I,J) = \frac{2\mu_{V_I}\mu_{V_J}}{\mu_{V_I}^2+\mu_{V_J}^2} \cdot \frac{2\sigma_{V_I}\sigma_{V_J}}{\sigma_{V_I}^2+\sigma_{V_J}^2} \cdot \frac{\sigma_{V_IV_J}}{\sigma_{V_I}\sigma_{V_J}} \tag{15}$$

where $\mu_{V_I}$ and $\mu_{V_J}$ denote the mean of the local variance of images $I(\mathbf{x})$ and $J(\mathbf{x})$, $\sigma_{V_I}$ and $\sigma_{V_J}$ the standard deviation of the local variance of images $I(\mathbf{x})$ and $J(\mathbf{x})$, and $\sigma_{V_IV_J}$ the covariance between the local variance of $I(\mathbf{x})$ and the local variance of $J(\mathbf{x})$.

Note that although there is a purposeful great similarity between Eq. (15) and the SSIM index, the latter relies on the mean of the local statistics of the images, and the former deals with the global statistics of the local variances of the images.

The first term in Eq. (15) carries out a comparison between the means of the local variances of both images. The second one compares the standard deviation of the local variances. This term is related with the *blur* and the *sharpness* of the image. The third term is the one to introduce cross information in the two images. To avoid computational problems with small values, some constants may be added to every term in Eq. (15):

$$\mathrm{QILV}(I,J) = \frac{2\mu_{V_I}\mu_{V_J}+C_4}{\mu_{V_I}^2+\mu_{V_J}^2+C_4} \cdot \frac{2\sigma_{V_I}\sigma_{V_J}+C_5}{\sigma_{V_I}^2+\sigma_{V_J}^2+C_5} \cdot \frac{\sigma_{V_IV_J}+C_6}{\sigma_{V_I}\sigma_{V_J}+C_6}. \tag{16}$$

In order to make the index more sensitive to certain kind of degradations, each of the three components may be weighted by a different positive exponent $\alpha$, $\beta$ and $\gamma$:

$$\mathrm{QILV}(I,J) = \left[\frac{2\mu_{V_I}\mu_{V_J}}{\mu_{V_I}^2+\mu_{V_J}^2}\right]^{\alpha} \cdot \left[\frac{2\sigma_{V_I}\sigma_{V_J}}{\sigma_{V_I}^2+\sigma_{V_J}^2}\right]^{\beta} \cdot \left[\frac{\sigma_{V_IV_J}}{\sigma_{V_I}\sigma_{V_J}}\right]^{\gamma}. \tag{17}$$

## 3 Background on Tensor Images

Although a rigorous definition of tensors is out of the scope of this chapter, in this section we will briefly describe tensors and tensor fields, as well as the most common tensor images used in image processing.

Tensors are mathematical entities that allow to describe some physical or mathematical properties that cannot be fully described by scalar magnitudes. In addition, tensors offer an appropriate behavior under changes of basis. They are expressed as multi-dimensional arrays, and their rank is defined as the number of array indexes that are required to fully describe the tensorial magnitude. Thus, tensors can be viewed as a generalization of scalars (zero order tensors), vectors (first order tensors) and matrices (second order tensors) to high-dimensional objects.

Regarding the tensor fields, they can be described as a geometric space that has a tensor associated with each of their points. This is obviously a non-rigorous definition, but it is enough for the purposes of this chapter. According to this definition, tensor images, that is, images where a tensor is defined at each voxel, can be regarded as discrete tensor fields. Thus, the representation of complex quantities, such as anisotropic magnitudes, is carried out by tensor images.

In the image processing area, tensors appear both in the images themselves and as a tool to image analysis or processing. This is the case of the structure tensors [14] that are often used to identify structured areas in the images, or the diffusion tensors that are used to anisotropically filter the images [32]. Regardless of these applications, this chapter is focused on the tensor images, in order to assess their quality and for this reason we will describe more thoroughly the tensors that appear in the most commonly used images.

One type of tensors that most frequently appear in image are the diffusion tensors (DT), that describe the diffusion of molecules in the space [3]. DTI has been broadly used in medical imaging, especially in neuroimage, since the anisotropy of the diffusion provides information about the nerve fiber structure that cannot be observed by other modalities. In general, DT is a second order tensor, that can be expressed as a $3 \times 3$ positive definite matrix, to describe diffusion in a 3D space, or a $2 \times 2$ positive definite matrix if only a two-dimensional space is considered. However, higher order tensors can be defined for advanced applications [33], since they provide more information about diffusion, specially in areas where fibers cross. Currently, these higher order tensors are not widely used, and the $3 \times 3$ second order tensor is still the most common approach to represent diffusion in tissues.

Other common tensors are the stress and strain tensors, which are related to the mechanical properties of the materials. The stress describes the distribution of forces within a tissue or material, and the strain tensor measures how a body is deformed when a force is applied on. Both are second order tensors and can be represented as $3 \times 3$ symmetric matrices. In medical imaging, they have been used to analyze the forces in the heart muscle in the course of a heartbeat [16], or the behavior of tissues when deformation forces are applied [20]. This technique is known as elastography, and it also takes into account the vorticity tensor [13], that is also a second order tensor, represented as a $3 \times 3$ anti-symmetric matrix.

Since these are the most common tensor images, the quality measures that are defined in this chapter are focused on them. Nevertheless, the topics mentioned in Section 4 intend to be generalizable to other tensor images.

On the other hand, in some applications only part of the information that the tensor provides is required. For instance, some scalar magnitudes are defined to measure some quantities of interest or to make more intuitive the data visualization and interpretation. In other cases, relevant information can be given by vectorial magnitudes computed from the tensor data. This is the case of fiber tracking applications, which consist in the estimation of the nerve fiber trajectories from the DTI [19]. It is usually based on the direction given by the major eigenvector of the matrix that represents the DT. Thus, the quality measure definition can be focused on these magnitudes instead of the tensor itself. However, it will depend on the spe-

cific application and the tensor type, so a general definition is not possible. We will show a comparison between the results based on these magnitudes with the results achieved using every tensor component in Section 5.

## 4 Application of Quality Measures to Tensor Images

In this section we describe how the quality measures can be extended to the case of tensor images. We focus on full-reference methods, which are based on comparison of the image under evaluation with a reference image or golden standard. Among this kind of measures, we distinguish between point-wise measures and structure based measures. Point-wise measures are usually computationally less costly but since they do not consider the image structure, they are less reliable than measures that take into account structural information. In [26] a comparative study of quality measures for scalar images is done, concluding that the results of structure based measures are more coherent with the subjective observer judgement.

The extension of point-wise measures to second order tensors is relatively straightforward. For instance, the MSE consists in the average of the square of the Euclidean distance between the pixel values in both images, as defined in Eq. (1). In the case of second order tensors, which can be expressed as matrices, the Euclidean distance should be computed between matrices, which can be done by computing the Frobenius norm of the difference matrix. Averaging the square of these values all over the image, the MSE value for the tensor images is obtained. Other different metrics could be also defined instead of the Euclidean metric, and thus more appropriate norms for each tensor type could be defined to better assess the error. The same concept can be applied to other point-wise measures: the SC can be defined for tensors by substituting in Eq. (2) the intensity values at each pixel by the Frobenius norm or another appropriate norm of the tensor in each voxel. Likewise, the PMSE can be defined by changing the square difference in Eq. (3) by the square of the chosen norm of the difference between tensors, and computing the maximum value of the tensor norm in image $I(\mathbf{x})$.

Some quality measures have been also defined taking into account the features of the human visual system. However, note that the tensor images do not have a straight visualization, since a tensor is represented at each voxel. It is usual to visualize tensors as glyphs, but the human eye should be accustomed to this representation and its interpretation to assess the quality of the image. Moreover, the image quality appreciated by the human visual system depends on the visualization algorithm and not only on the original image quality itself. Another option is to show a scalar image whose values are somehow related to the tensor. However, these images only reflect part of the tensor, which in most cases is not enough to assess the quality of the tensor image. For these reasons, the methods based on the human visual system cannot be extended to tensor images.

Further, we will show that point-wise measures present the same drawbacks that have been widely reported in the case of scalar images comparison. Therefore, struc-

ture measures are also required to describe the quality of tensor images. Structure measures are usually based on local statistics of the images. The computation of such statistics in tensor fields is the most involved issue to be handled in the extension of the quality measures to tensor images. For this reason, the next subsection is focused on the special tensor features that should be taken into account in the statistics computations, as well as the frameworks that have been developed to carry out this estimation.

## *4.1 Statistics of Tensor Data*

Due to the special nature of tensor data, some issues should be kept in mind to estimate the statistics of a tensor field. Next, we describe the main reasons that prevent a straightforward extension of scalar statistic estimation:

- Let $\mathbf{T}$ be a tensor describing a given transformation. The inverse transformation is described by $\mathbf{T}^{-1}$. The composition of the transformation and its inverse must be the identity tensor, that is, $\mathbf{T}^{-1}\mathbf{T} = \mathbf{I}$. Thus, it will be advisable that the mean of these two tensors would be the identity tensor, what is achieved by the use of geometric means instead of arithmetic ones.
- If the tensors are averaged by a Euclidean metric, a swelling effect appears, that makes the determinant of the mean tensor higher than the determinant of the individual tensors [6]. In order to avoid this effect, non-Euclidean metrics should be defined.
- As aforementioned, the most often used tensors in image processing are constrained to the space of symmetric definite or semipositive definite matrices. When using Euclidean metrics, computation with this kind of data can lead to results that do not belong to the allowed space. It can be avoided by using Riemannian metrics that consider the negative eigenvalues at infinite distances, and therefore, prevent the tensor from having negative eigenvalues.

The problem of how to compute tensor statistics taking into account the previous issues has been tackled by some works in the literature [2, 4, 7, 21], especially for the case of DT. Most of them are focused on the computation of the mean tensor for interpolation purposes. These approaches have been based on the statistic methodology for Riemmanian manifolds that had been previously developed, and affine-invariant Riemmanian metrics have been defined to avoid the undesirable effects of Euclidean metrics [21].

Thus, the statistic estimation can be based on the concepts about statistics on manifolds [5]. Instead of the conventional arithmetic mean, the Fréchet mean [11] is more suitable to compute the mean of a set of tensors, since it avoids the problems previously described. Given a set of tensors $\mathbf{T}_1, ..., \mathbf{T}_N$, the mean tensor $\mathbf{T}_{mean}$ is given by the value that minimizes the sample Fréchet function, $F_n(\mathbf{T})$, [5]:

$$\mathbf{T}_{mean} = \arg\min_{\mathbf{T}} F_n(\mathbf{T}) = \arg\min_{\mathbf{T}} \frac{1}{N} \sum_{i=1}^{N} d^2(\mathbf{T}_i, \mathbf{T}), \tag{18}$$

where $d$ is the distance between tensors. Therefore, the mean tensor should be computed by means of an optimization algorithm, but, if the metric is properly chosen, the problem has an explicit solution. Thus, if a Log-Euclidean metric [2] is defined, the distance between two tensors is given by

$$d(\mathbf{T}_1, \mathbf{T}_2) = \sqrt{\text{trace}\left( (\log(\mathbf{T}_1) - \log(\mathbf{T}_2))^2 \right)}. \tag{19}$$

It has been proved that, with this distance, the Fréchet mean is a generalization of the geometric mean for positive numbers, and can be directly computed as

$$\mathbf{T}_{mean} = \exp\left( \frac{1}{N} \sum_{i=1}^{N} \log(\mathbf{T}_i) \right). \tag{20}$$

Note that Log-Euclidean metric is only defined in the case of positive definite matrices, since the logarithm of a matrix with negative eigenvalues is not real. Therefore, this expression can only be applied to the case of tensors that are represented by positive definite matrices, while for tensors that are not positive definite other metrics must be defined.

The Fréchet variation of a probability measure on a manifold is defined as the mean value of the sample Fréchet value [5]. Thus, the variation $V_n$ of a set of tensors can be computed as

$$V_n = F_n(\mathbf{T}_{mean}) = \frac{1}{N} \sum_{i=1}^{N} d^2(\mathbf{T}_i, \mathbf{T}_{mean}). \tag{21}$$

These definitions must be taken into account in the description of the structural quality measures that are based on the local statistic assessment. Next, some of the quality measures that have been introduced in Section 2 are defined for tensor images.

## *4.2 MSSIM and WSSIM*

MSSIM and WSSIM have been defined in Section 2 for scalar images. Both are derived from the SSIM map. This SSIM index is based on the comparison of the luminance, contrast and structure, which are described by ratios between some local statistics. Namely, the local mean and variance in both images, and the local covariance between them are required. When working with tensor images, the assessment of such statistics should be based on the Fréchet mean and variation previously defined.

First of all, the local mean at a voxel $\mathbf{x}$ of a tensor image $I$ is defined as the Fréchet mean in a given neighborhood $\mathcal{N}(\mathbf{x})$ of the voxel. Thus, let $\eta(\mathbf{x})$ be the estimated local mean, it is computed as

$$\eta_I(\mathbf{x}) = \arg\min_{\mathbf{T}} \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{x}_i \in \mathcal{N}(\mathbf{x})} d^2(I(\mathbf{x}_i), \mathbf{T}), \tag{22}$$

where $|\cdot|$ denotes the cardinality of the neighborhood, that is, its number of elements. In the case of positive definite tensors, with the Log-Euclidean metric given by Eq. (19), can be directly computed as

$$\eta_I(\mathbf{x}) = \exp\left( \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{x_i \in \mathcal{N}(\mathbf{x})} \log(I(\mathbf{x}_i)) \right). \tag{23}$$

The SSIM of an image $J$ is computed by comparing such image with a reference image that we will denote as $I$. Thus, to compute the term of luminance comparison in Eq. (5), the estimation of both $\eta_I(\mathbf{x})$ and $\eta_J(\mathbf{x})$ is required. Let it be noticed that the mean of a tensor set is also a tensor. In order to compute the ratio between local means that describes the luminance comparison, the norm of the mean tensor is computed. If the Log-Euclidean metric has been considered to estimate the local mean, the norm is computed according to this metric. That is,

$$||\mathbf{T}|| = \sqrt{\text{trace}\,(\log(\mathbf{T})^2)}. \tag{24}$$

Thus, $\mu_I(\mathbf{x}) = ||\eta_I(\mathbf{x})||$ and $\mu_J(\mathbf{x}) = ||\eta_J(\mathbf{x})||$ are computed, and the luminance term in the SSIM definition can be obtained by including these local mean images in Eq. (5).

With regard to the contrast comparison in the SSIM expression, it is based on the local standard deviation of images, $\sigma_I(\mathbf{x})$. In the case of tensor images, this statistic is estimated as the square root of the variation of Fréchet in a given neighborhood,

$$\sigma_I(\mathbf{x}) = \sqrt{\frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{x}_i \in \mathcal{N}(\mathbf{x})} d^2(I(\mathbf{x}_i), \eta_I(\mathbf{x}))}. \tag{25}$$

If the Log-Euclidean metric is used to assess the local mean, the distance $d$ in this equation must be again the one defined in Eq. (19). Note that $\sigma_I(\mathbf{x})$ is a scalar value, so the contrast term can be directly computed by means of the Eq. (6), where both $\sigma_I(\mathbf{x})$ and $\sigma_J(\mathbf{x})$ are obtained according to Eq. (25).

Finally, the local covariance between both images is required for the structure comparison. The estimation of this statistic for tensor images can be based on the Fréchet variation, and it is defined as

$$\sigma_{IJ}(\mathbf{x}) = \sqrt{\frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{x}_i \in \mathcal{N}(\mathbf{x})} d(I(\mathbf{x}_i), \eta_I(\mathbf{x})) \cdot d(J(\mathbf{x}_i), \eta_J(\mathbf{x}))}. \tag{26}$$

Again, the Log-Euclidean distance should be considered for positive definite tensors, if it has been used to estimate the mean. With this covariance estimation the structure comparison term can be obtained as defined in Eq. (7). Thus, the SSIM map can be computed by the Eqs. (8) or (9) , where the statistics are computed as we have just described. The values of the SSIM map are scalars, so the MSSIM or WSSIM can be directly computed applying the Eqs. (10) and (11) respectively. If statistics are included in the weighting term, they should be estimated with the previous equations. In the implementation of WSSIM performed in the experiments, the weighting function is the one described in the Eq. (12).

## *4.3 QILV*

The QILV index is based on the statistics of the local variance images, instead of the original images themselves. The estimation of the local variance $V_I(\mathbf{x})$ at each voxel $\mathbf{x}$ is defined as the Fréchet variation of the tensor in a neighborhood $\mathscr{N}(\mathbf{x})$ of the voxel:

$$V_I(\mathbf{x}) = \frac{1}{|\mathscr{N}(\mathbf{x})|} \sum_{\mathbf{x}_i \in \mathscr{N}(\mathbf{x})} d^2(I(\mathbf{x}_i), \eta_I(\mathbf{x})). \tag{27}$$

The local mean $\eta_I(\mathbf{x})$ is computed as described in Eq. (22) for the general case, or by Eq. (23) if positive definite tensors and Log-Euclidean metric are considered. The distance definition used in the Eq. (27) must be coherent with the metric used to estimate the mean.

Thus, the local variance $V_I$ and $V_J$ of both images $I$ and $J$ are computed. Let notice that $V_I$ and $V_G$ are scalar images, and therefore the statistics of such images can be computed as usual, and the QILV value is obtained by directly applying Eq. (15).

## *4.4 Modified Fuzzy Minkowsky Distance*

The so-called M3 quality index is the modified version of the Minkowsky distance, described by Eq. (14). This measure is computed over the image histogram. In our case, the histogram will be computed over the norm of the tensors at each voxel. An appropriate metric should be considered to compute the norm of the tensor. In the case of positive definite tensors, the Log-Euclidean metric can be chosen, so the norm definition is the one given in Eq. (24). Once the norm is computed at each voxel, the histogram of the scalar images so computed can be obtained, and the quality index can be estimated as usual.

The histogram can be based on other scalar magnitudes computed from the tensor, such as its eigenvalues or its trace. The appropriateness of such magnitudes depends on the specific applications. Moreover, multichannel schemes can be pro-

posed, where a vector of quality indexes is built by computing the M3 value for a set of scalar images.

## 5 Experiments

In order to analyze the applicability of quality measures to tensor images, we have designed a set of experiments, whose results are described in this section. Though the appropriateness of a given quality measure mainly depends on the specific application or kind of data, we intend to provide a general view of the behavior of such measures when working with tensor images. Thus, we compute and compare the measures that have been defined for tensor images in Section 4, that is, QILV, MSSIM, WSSIM and M3, as well as the point-wise MSE measure for tensors. Comparison is done with the same measures based on some scalars derived from the tensor data.

In the next subsection the data set used for the experiments is described, and then the experiments are detailed and their results are shown.

### *5.1 Data Set*

The experiments have been carried out on both synthetic and real tensor images. In the case of synthetic data, we have generated $200 \times 200$ tensor fields composed by second order tensors of $2 \times 2$ components and another set of fields of $3 \times 3$ second order tensors. Their construction has been based on DTI, since it is the most common kind of tensor images. For this reason, the synthetic images can be viewed as an isotropic region where some fibers appear. Diffusion inside fibers is supposed to be clearly anisotropic, since molecules mainly diffuse in the direction parallel to the fiber. Three areas can be distinguished in both images:

1. Background: In this region tensors are defined to be isotropic, that is, all their eigenvalues have similar values. These eigenvalues are generated by means of a uniform distribution $U[0.05, 0.15]$.
2. Lines or *fibers*: In these areas the tensors are oriented parallel to the structure, and are clearly anisotropic. The highest eigenvalue corresponds to the eigenvector that is parallel to the line direction, and its generated by a uniform random distribution $U[2, 6]$, while the two other eigenvalues are generated by a uniform distribution $U[0.05, 0.15]$.
3. Crossings: These are the regions where two or more fibers intersect. Here, the tensors are computed as a composition of the tensors defined in each of the crossing lines.

In Fig. 1 (a), an example of the structure of the synthetic field is shown. The gray scale is related with the fractional anisotropy (FA) of the tensor[1]: white stands for isotropic diffusion, and black means that the diffusion occurs only along one direction. Five fibers can be identified, where tensors are oriented parallel to the lines. The same layout is defined for $2 \times 2$ and $3 \times 3$. As an example of how the tensors are in fiber crossing, the squared region in Fig. 1 (a) is zoomed in, and represented in Fig. 1 (b). Tensors are represented by ellipses, where the axis direction is given by the tensor eigenvectors and their size is given by the tensor eigenvalues. In the crossing area, the diffusion is less anisotropic, since it is the sum of the diffusion in both fibers. Since diffusion is higher in the diagonal fiber than in the vertical one, diffusion is higher in the diagonal direction, but some diffusion also appears in the vertical direction.

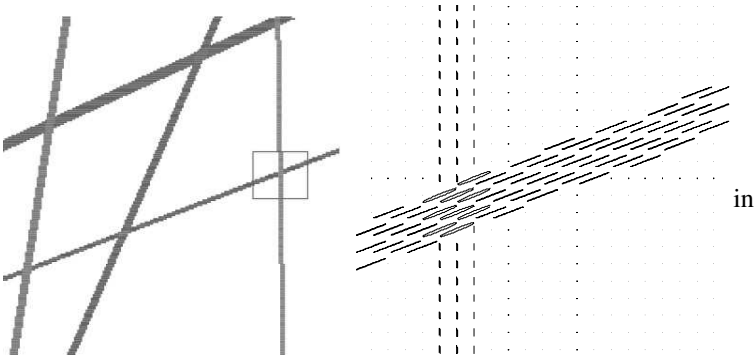



**Fig. 1** Synthetic data: a) Structure of the synthetic field, five fibers can be distinguished; b) Fiber cross: zoom in of the squared area in a). Tensors are represented as ellipses

The original synthetic images are corrupted by different means, and comparison is carried out between these altered images and the original ones, that are considered the ground truth. In order to produce realistic alterations of the images, these degradations have been applied considering the synthetic field to be a DTI. The DT is computed from a set of diffusion weighted images (DWI) [37], which have been acquired by a magnetic resonance scanner. The noise or artifacts in the DWIs are responsible for the decreasing in the quality of the estimated DT field. For this reason, such DWIs have been simulated and the distortions have been applied to these images. DT image is estimated again from the corrupted image by means of the Stejskal-Tanner equations [37]. Note that, in other kind of tensor images, the tensors are also computed from some kind of acquired images, so a similar procedure should be carried out to synthetically alter the images. Next, we described the trans-

[1] Fractional anisotropy is a measure of the anisotropy of the diffusion and it is defined from the tensor eigenvalues, $\lambda_1, \lambda_2$ and $\lambda_3$, as $FA = \frac{\sqrt{(\lambda_1-\lambda_2)^2+(\lambda_1-\lambda_3)^2+(\lambda_2-\lambda_3)^2}}{\sqrt{\lambda_1^2+\lambda_2^2+\lambda_3^2}}$

formations that have been applied to the synthetic dataset and how they influence the quality indexes.

Furthermore, real DTI data are considered to provide an example of a real application of the quality measures. These images are obtained by a General Electric Sigma 1.5 T scanner, with $b$-value$=$ 1000 and 15 gradient directions.

## *5.2 Experiments on the Synthetic Data Set*

To analyze the influence of different kinds of distortion in the quality indexes, the synthetic images previously defined are modified by the next degradation sources:

1. DWIs are blurred by local averaging using a square $N \times N$ window with different $N$ values, and then DTI is computed. Examples of these images are shown in Fig. 2 (a) and 2 (b), where the image has been blurred by a $5 \times 5$ and a $15 \times 15$ windows respectively. The FA images of the corrupted tensor fields are represented. Moreover, a detail of a tensor field blurred by a $3 \times 3$ window is displayed in Fig. 2 (c). The zoomed area is the same region that is highlighted in Fig. 1 (b).
2. DWIs are corrupted by Rician noise, with different values of the $\sigma$ parameter, and the DTI is estimated from the noisy images. Rician noise has been chosen because it is the main kind of distortion in the magnetic resonance images. In Fig. 2 (d) and 2 (e), two examples of the FA maps of noisy DTI are shown. The parameters of the applied Rician noises are $\sigma = 0.1$ and $\sigma = 0.25$ respectively. A detail of the noisy field is shown in Fig. 2 (f), where the effect of noise in the zoomed region in Fig. 1 (b) can be viewed. In this case the parameter of the Rician noise is $\sigma = 0.1$.
3. The intensity level of the DWIs is increased by a percentage $\tau$ of their original level. Different percentages of change are applied. Then, the DT field is computed from the new DWIs. This distortion is more noticeable in the trace image of the tensor field, that is, the voxel intensity is the value of the trace of the corresponding tensor. Trace is directly related with the size of the tensor, and for this reason, it reflects the intensity change better than other magnitudes such as the FA. Thus, in Fig. 2 (g), the trace of the original field is shown, and in in Fig. 2 (h) the field computed from the modified DWI is displayed. In such example, the intensity of the DWIs have been increased in a 50%. The effect of such distortion in the tensor size can be viewed in the Fig. 2 (i).

These transformations are applied to a reference image, so three groups of images are obtained. A set of blurred images is built by averaging the original image by $N \times N$ windows, with $N = 2n+1$, $n = 1,..,10$. These images will be denoted as $I_{b,N}$. A set of images with different levels of noise is also generated. Thus, noisy images are obtained with Rician noise of $\sigma = \{0.005, 0.5\}$. These images will be denoted as $I_{n,\sigma}$. Moreover, DT images are computed from DWI whose intensities have been increased by a percentage $\tau = \{10\%, 100\%\}$ respect to their original value, so a set of $I_{c,\tau}$ images is built.

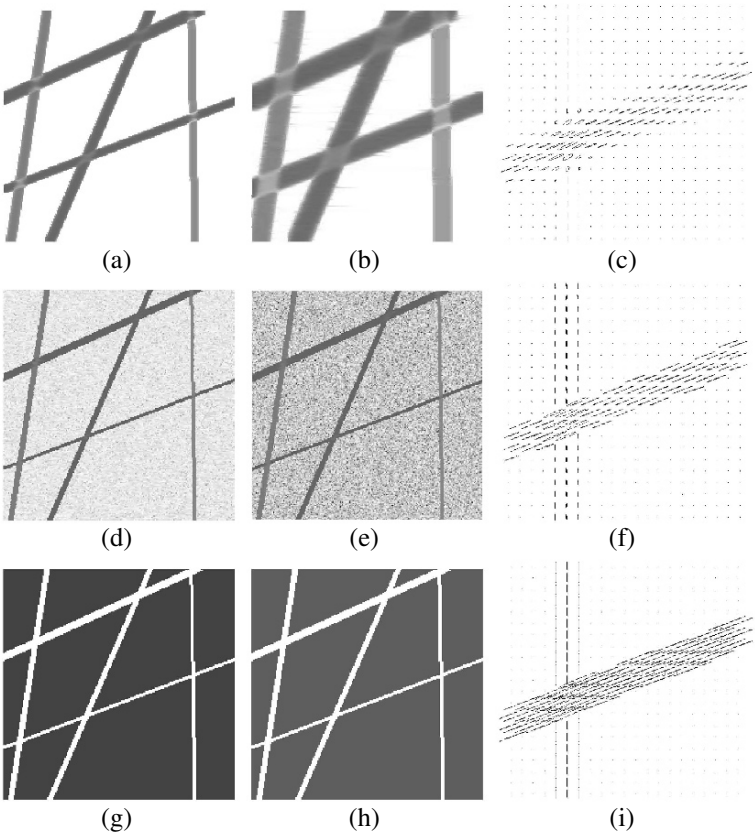



**Fig. 2** Examples of corrupted images used in the experiments: a) FA image of the tensor field blurred with a $5 \times 5$ window; b) FA image of the tensor field blurred with a $15 \times 15$ window; c) Fiber crossing blurred by a $3 \times 3$ window; d) FA image of the tensor field corrupted by Rician noise of parameter $\sigma = 0.01$; e) FA image of the tensor field corrupted by Rician noise of parameter $\sigma = 0.25$; f) Fiber crossing corrupted by Rician noise of parameter $\sigma = 0.1$; g) Trace of the original tensor field; h) Trace of the tensor field whose intensity has been increased by 50%; i) Fiber crossing, where intensity has been increased by 50%.

We compute five quality indexes for the transformed images, taking as reference image the original synthetic field. The results are shown in Fig. 3. Notice that the values of MSSIM, WSSIM, QILV and M3 are bounded between 0 and 1, and higher values correspond to higher quality, whereas the MSE is not bounded, and its value tends to zero as the quality increases. For this reason, in Fig. 3, we represent $1 - MSE$ instead of $MSE$, to more straight comparison of the indexes behavior. Thus, due to the range within which the tensor values are defined, the $1 - MSE$ curve can be seen together with the other measures.

The first conclusion that can be obtained from Fig. 3 is that the indexes based on structural information are more reliable to assess the quality of images. This

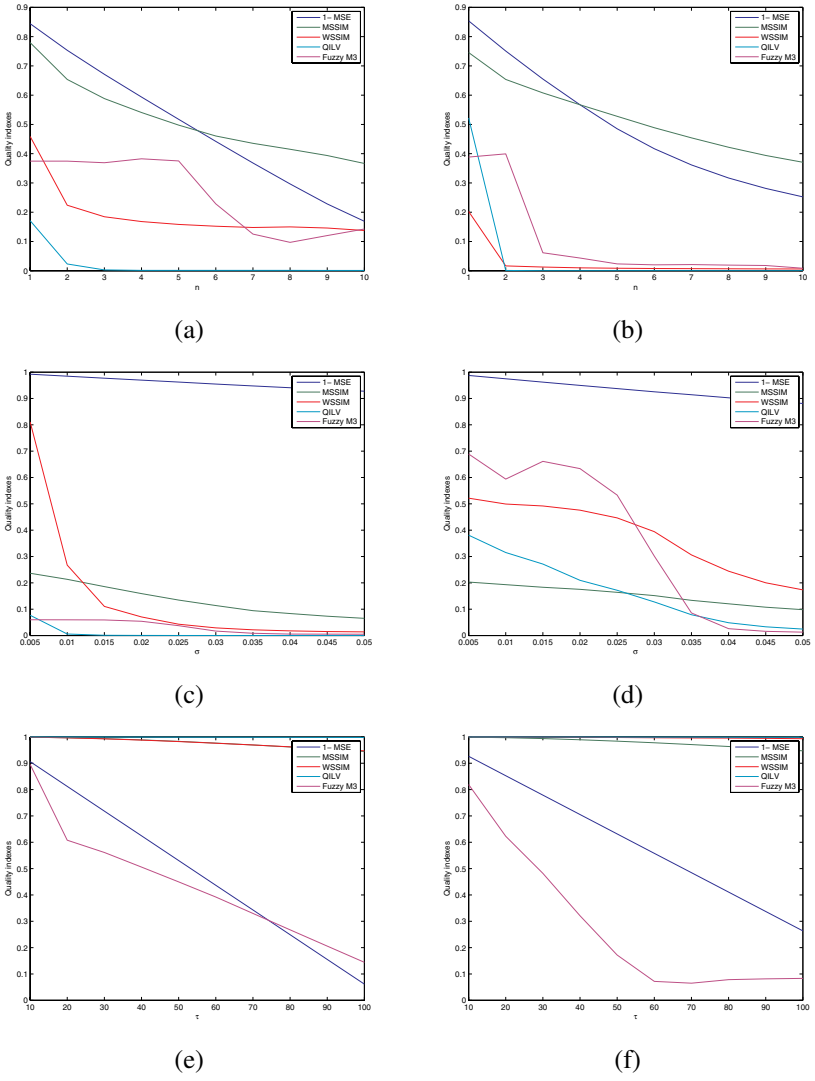



**Fig. 3** Quality indexes for corrupted images: a, b) Blurring by $N \times N$ windows, with $N = 2n+1$; c, d) Noisy images. Rician noise with parameter $\sigma$ has been applied; e, f) Tensor fields computed from a modified set of DWI, whose intensity has been increased by a percentage $\tau$. The right column corresponds to fields of $2 \times 2$ tensors, and the left column to fields of $3 \times 3$ tensors.

issue has already been observed in the analysis of the quality of conventional scalar images [26], and now it can be extended to tensor images. The main drawback of point-wise measures can be noticed in the quality of the $I_{c,\tau}$ images. In Fig. 2 (h) it can be viewed that the structure of the data is not affected by this kind of distortion, and the only effect in the image is the rescaling of the tensors. For this

reason, the QILV, MSSIM and WSSIM indexes decay very slowly and are near one, while the MSE error increases rapidly. On the other hand, the M3 index, that is based on the histogram comparison also decreases faster although the image remains very similar to the reference image, since it is based on the tensor norm, and the difference between the resulting histograms is noticeable. Point-wise measures are more sensitive to changes in the intensity image range, whereas structural measures are supposed to be less sensitive to these changes as long as the image structures are preserved. Let notice that the value of MSE for the image in Fig. 2 (h) is similar to the MSE value for the image in Fig. 2 (b), that is a very blurred image.

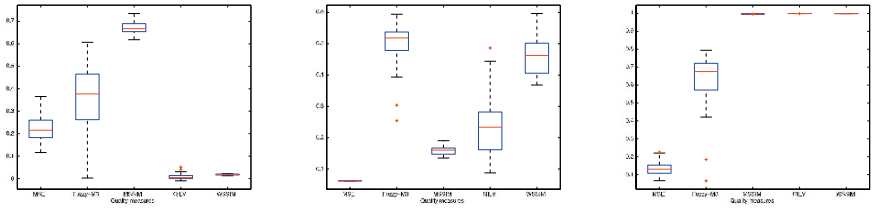
It can also be noticed that the behavior of the structure-based quality indexes is similar to their performance in the scalar case. For instance, the QILV decreases faster for blurred images than for impulsive noise, whereas the MSSIM is very sensitive to noise but it is less influenced by blurring. The WSSIM has an intermediate behavior since its value decreases slowly for little distortions of both types (noise and blurring) but decays more abruptly as the image degradation decreases.

### Variance of the Quality Measures

An experiment has been developed to assess the variance of the quality indexes in a set of images that have been altered by the same source of degradation. It is supposed that if a similar distortion has been applied to two different images, the quality indexes of these images should be similar. Thus, we have generated a set of 40 synthetic images of $3 \times 3$ tensors, that have been modified with the three kinds of distortion previously defined: blurring, noise and intensity change. Then, the quality indexes have been computed for each of the altered images. The variance of the indexes for each of the distortion sources is shown in Fig. 4. It can be highlighted the small variability of the structure based measures QILV, MSSIM and WSSIM in case of intensity changes, which is coherent with the previous comments about Fig. 3. QILV and WSSIM have higher variability when noisy images are evaluated than if blurred images are considered, while MSSIM seems to be more *robust* with both distortion sources. With regard to the fuzzy M3 computed over the norm of the tensors data, it is the less reliable index since it has a great variability whatever the distortion source be. Nevertheless, its behavior for the analysis of noisy images is comparable to the WSSIM and QILV measures.

### Quality Indexes Based on Scalar Magnitudes

Previous experiments have applied the formulation of the quality indexes in Section 4, that takes into account every tensor component. However, as mentioned, it is frequent to compute scalar images from the tensor data, in order to make easier the data visualization or processing. In this experiment, the quality indexes are computed over scalar images derived from the previously described synthetic fields. Two magnitudes are considered: FA and the higher eigenvalue $\lambda_1$. The respective



(a) (b) (c)

**Fig. 4** Quality index variance distribution for three kind of degradation sources: a) Blurring by a $5 \times 5$ window; b) Rician noise with parameter $\sigma = 0.025$; c) Intensity increasing of $\tau = 20\%$

scalar images from every degraded $3 \times 3$ tensor field are computed, and quality is assessed.

In Fig. 5, the behavior of the indexes so computed when noise or intensity changes appear in the images can be viewed. Thus, it can be noticed that the different magnitudes have different sensitivity for noise or intensity changes. For instance, the higher eigenvalue is more sensitive to intensity changes than FA, and therefore, the degradation in the tensor image due to this source will be ignored if a quality index based on FA is considered. On the other hand, the noise has more influence on the FA than in the higher eigenvalue. Therefore, special care must be taken to choose the images over which the quality indexes are computed, in order to not discard relevant information.

## *5.3 Example of Real Application: Fiber Tracking.*

In this experiment, an example of the utility of quality measures in a real application is shown. A fiber tracking algorithm is considered, which estimates the fiber trajectories from a DT-MRI volume. Fibers are computed by tracking the direction given by the major eigenvector in the voxels whose FA is high enough. The reference image is a real DT-MRI data set, whose tensors have been estimated from a filtered set of DWIs [28]. It will be compared with the DT-MRI obtained from the same DWIs, but in this case, the original noisy images are considered. Moreover, a corrupted DT-MRI set is considered, where the tensors are computed with modified gradient directions. Namely, a rotation of $\frac{\pi}{6}$ has been applied to every gradient direction, what produces a rotation of the tensors in the image. This is a problem that can occur if some geometrical transformation is applied to the image, but the tensor data are not transformed coherently to preserve their relation to the underlying structure.

In Fig. 6 a fiber tract computed from the same seed in the three images is shown. Fig. 6 (a), corresponds to the reference image, Fig. 6 (b) is the fiber tract computed from the noisy DT-MRI and Fig. 6 (c) the volume whose tensors have been rotated.

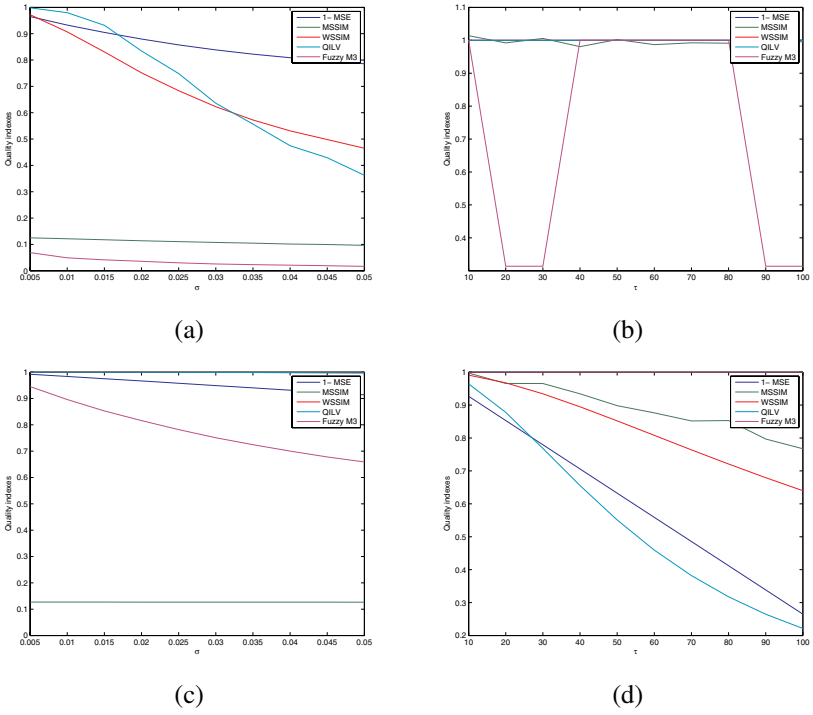
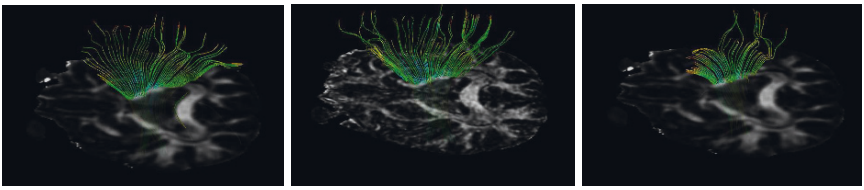



**Fig. 5** Quality indexes for scalar images derived from the tensor: a, b) FA images; c, d) Higher eigenvalue images. The images in the right column correspond to noisy images, with Rician noise of parameter $\sigma$, and those in the left column to changes in the intensity of the DWIs.

It can be viewed that better fiber tracts are obtained from the noisy image than from the image whose tensors are rotated with respect to the original data, that is, the noisy image is *better* for tractography purposes. The values of the quality indexes are compiled in the Table 1, where it can be noticed that some measures are not valid to assess the quality of the DTI for this application. Namely, the FA-based indexes are not sensitive to orientation changes, and they give higher values for the rotated tensors than for the noisy image. Moreover, MSSIM and MSE obtain better quality values for the rotated image, even though the whole tensor is considered, whereas the WSSIM, QILV and fuzzy M3 indexes are coherent with the visual assessment of the quality. Thus, this example shows the importance of the choice of the appropriate quality measure for each specific application.



(a) (b) (c)

**Fig. 6** Fiber tracts computed from the same seed: a) Original image; b) Noisy image; c) DTI computed with rotated gradient directions.

| | 1-MSE | MSSIM | WSSIM | QILV | Fuzzy M3 |
|---|---|---|---|---|---|
| | Tensor | | | | |
| Noisy | 0.9950 | 0.9111 | 0.9383 | 0.9601 | 0.6282 |
| Rotated tensors | 0.9965 | 0.9918 | 0.9198 | 0.9502 | 0.5979 |
| | Fractional Anisotropy | | | | |
| Noisy | 0.9919 | 0.8902 | 0.3739 | 0.8015 | 0.8894 |
| Rotated tensors | 0.9993 | 0.9914 | 0.7755 | 0.9886 | 0.9802 |
| | Highest Eigenvalue | | | | |
| Noisy | 0.9971 | 0.9024 | 0.7450 | 0.9278 | 0.9550 |
| Rotated tensors | 0.9990 | 0.9907 | 0.7342 | 0.8227 | 0.9835 |

**Table 1** Quality indexes for noisy image and the image whose tensors have been rotated.

## 6 Discussion

Quality assessment has proved to be a very important issue when dealing with image processing. As previously stated, image quality may vary due to noise, acquisition artifacts, compression or any kind of processing. Data compression or noise, for instance, are well known sources of degradation. Image enhancement, on the other hand, is a tool that intends to improve the image quality, though sometimes it works reversely, destroying some important structures.

As a result of all the possible worsening and improvements an image may suffer, a need for quality assessment arises. Users need a way to establish the quality of their data. However, the definition of quality measures is an involved issue. It may be difficult to assess when an image is better than other, and it may depend on the specific application. In some cases, a noisy image is *worse* than a blurred image, while in other cases well-defined edges are preferred, although impulsive noise appears. In addition, the quality can be related with the structure of the image, rather than with its intensity levels, and therefore quality indexes should quantify if the structure in two images is similar or not, independently of the intensity values in pixels.

All these issues also appear when tensor images are considered. In this case, once images have been acquired, the tensor at each voxel must be computed, and

therefore the quality of the resulting image depends also on the tensor estimation algorithm. Moreover, since a tensor, and not a scalar intensity value, is defined at each voxel, the issue of how to measure the similarity between tensors arises. Point-wise measures are computed by comparing the values at the same position of two images. In the scalar case, the difference or square difference between the intensity values is usually computed. Now, the difference between tensors can be considered, but then the computation of the norm of the tensor difference will be required to obtain a scalar value. The Frobenius norm has been usually considered in this case, which is equivalent to the Euclidean distance between matrices. However, depending on the application or the tensor class, this measure could not reflect the difference between tensors. For instance, the tractography application is usually based on the fractional anisotropy and the tensor orientation. Thus, two tensors that have the same orientation but different size will lead to the same results for a tractography algorithm. However, the Frobenius norm of the difference may take a value higher than between two tensors of the same size and different orientation. These topics depend on the application and the specific tensor type, and for this reason it is not possible to provide a general definition of how to measure the difference between tensors.

Furthermore, structure-based quality measures usually rely on image statistics. The computation of such statistics in tensor data is not straightforward, and recent works are dealing with this topic. In this chapter, we define a method to assess such statistics, but more rigorous definitions of such estimations could be proposed. By means of these definitions, quality indexes have been extended to tensor images, and it has been shown that their behavior is similar in both the scalar and tensor cases. Thus, some measures are more sensitive to noise and others are more sensitive to blurring, or to intensity changes.

It is important to be aware of the fact that tensor data provide information about the underlying structure. If structure is defined only as the region layout in the images, as is the case of conventional structure-based quality indexes, two images could have exactly the same structure, but if tensors are different, the information that they provide would be different. Suppose a pair of DTIs, each of them composed by the same two regions. Nevertheless, if different tensors are defined in each of the regions of both images, the information provided by the images may be completely different. This would be the case, for instance, if regions in the first DTI are formed by isotropic tensors, and regions in the second DTI are composed by cigar-shaped tensors. In the second case, fiber structures can be found, whereas in the first a grey matter region (where no fibers are present) would be represented. For this reason, the specific nature of the tensor data should be taken into account when describing quality measures.

A usual application of the quality measures is to analyze the performance of image filters. If full-reference indexes are considered, images should be compared with a reference image. Thus, synthetic data or phantoms are defined for providing a reference image which can be corrupted to analyze the filtering results. Although synthetic data sets have already been defined to several image modalities, some of them available on line, to compare algorithm performances, the building of synthetic data set for tensor images is still in its earlier stage. In other processing algorithms

the previous problem with the reference image may not appear. This is the case of image registration, where the registered image is compared with the target in order to assess the goodness of the registration algorithm.

A final question would be what is considered to be *quality*. In the scalar case, it is usual to consider that the quality of an image is related to the visual impression. Thus, proposed measures intend to reduce the gap between the human-based assessment and the machine-based criteria. For this reason, some quality indexes are based on the features of the Human Visual System. On the contrary, in the case of tensor images, the goodness of an image cannot be easily established from the visual perception, since they cannot be directly visualized. The visualization of tensor fields is done by means of scalar images whose intensity values are computed from the tensor, and therefore the observer does not view the tensor field but some part of it. Depending on the scalar magnitude that is represented, some changes in the image quality could not be noticed. Another way to visualize a tensor field can be the use of glyphs that represent the tensor. However, to visually analyze this representation, the observer should be familiarized with this kind of visualization, so the quality assessment may be more difficult than in the scalar case. Thus, there are not clear criteria to assess the *quality* of the quality indexes. When focusing on a specific application the quality can be defined based on the features that have stronger influence on the algorithm.

## 7 Conclusions

In this chapter, we have analyzed the problem of the quality assessment for tensor images. This issue has often been left aside in the works about tensor image processing. However, the definition of measures that allows to compare the performance of different algorithms over tensor images is a task of paramount importance when dealing with tensor data. Although the goodness of a particular method will be related with the particular application or the nature of data, there are some global measures that give useful information about the structural similarity between two tensor images, hence the quality of a processed image. Measures described in this chapter are based on the quality indexes defined for scalar images. They have been extended to cope with tensor images, and their validity has been analyzed. The proposed methods are not exclusive, and they can be viewed as a first and general approach to the problem. Based on the presented results, appropriate modifications may be proposed to face a specific application or to analyze a given sort of tensor images. In addition, specific tensor quality indexes could also be defined. Quality assessment is still an open topic in the analysis of images and a lot of work should be done to find the appropriate way to quantify the quality of a tensor image. Moreover, the quality can be related with different issues in the image (noise, structure, difference between tensors...) and a set of quality indexes could be required, each of which measures the quality with respect to different criteria.

**Acknowledgements** The authors acknowledge the Comisión Interministerial de Ciencia y Tecnología for research grant TEC2007-67073/TCM, the Junta de Castilla y León for grant VA026A07 and the Consejería de Sanidad de Castilla y León for grant GRS 300/A/08. First author is funded by the JCyL and the European Social Fund (ESF).

## References

1. Aja-Fernández, S., San-José-Estépar, R., Alberola-López, C., Westin, C.F.: Image quality assesment based on local variance. In: Proc of the 28th IEEE EMBC, pp. 4815–4818. New York, NY, USA (2006)
2. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance in Medicine **56**(2), 411–421 (2006)
3. Basser, P., Mattiello, J., Le Bihan, D.: MR diffusion tensor spectroscopy and imaging. Biophysical Journal **66**, 259–267 (1994)
4. Batchelor, P.G., Moakher, M., Atkinson, D., Calamante, F., Connelly, A.: A rigorous framework for diffusion tensor calculus. Magnetic Resonance in Medicine **53**, 221–225 (2005)
5. Bhattacharya, A., Bhattacharya, R.: Nonparametric statistics on manifolds with applications to shape spaces. In: B. Clarke, S. Ghosal (eds.) Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh, pp. 282–301. Institute of Mathematical Statistics, Beachwood, OH, USA (2008)
6. Chefd'hotel, C., Tschumperlé, D., Deriche, R., Faugeras, O.: Regularizing flows for constrained matrix-valued images. Journal of Mathematical Imaging and Vision **20**(1-2), 147–162 (2004)
7. Corouge, I., Fletcher, P.T., Joshi, S., Gilmore, J.H., Gerig, G.: Fiber-tract oriented statistics for quantitative diffusion tensor MRI analysis. In: S. Berlin (ed.) Medical Imaging Computing and Computer-Assisted Intervention, vol. 3749, pp. 131–139 (2005)
8. Cosman, P., Gray, R., Olshen, R.: Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. Proceedings of the IEEE **82**(6), 919–932 (1994)
9. Damera-Venkata, N., Kite T.D.; Geisler, W., Evans, B., Bovik, A.: Image quality assessment based on a degradation model. IEEE Transactions on Image Processing **9**(4), 636–650 (2000)
10. Eskicioglu, A.M., Fisher, P.: Image quality measures and their performance. IEEE Transactions on Communications **43**(12), 2959–2965 (1995)
11. Fréchet, M.: Les elements aleatoires de nature quelconque dans un espace distancié. Annales de l'Institute Henri Poincaré **10**(4), 215–310 (1948)
12. Hsu, E.W., Setton, L.A.: Diffusion tensor microscopy of the intervertebral disc anulus fibrosous. Magnetic Resonance in Medicine **41**(5), 992–999 (1999)
13. Hunt, J.C.R.: Vorticity and vortex dynamics in complex and turbulents flows. Transactions Canadian Society for Mechanical Engineering **11**, 21–35 (1987)
14. Knutsson, H.: Representing local structure using tensors. In: Proceedings of the 6th Scandinavian Conference on Image Analysis, pp. 244–251. Oulu, Finland (1989)
15. Lim, J.S.: Two Dimensional Signal and Image Processing. Prentice Hall, Englewood Cliffs, NJ (1990)
16. McVeigh, E.R.: MRI of myocardial function: motion tracking techniques. Magnetic Resonance Imaging **14**(2), 137–150 (1996)
17. Miyahara, M., Kotani, K., Algazi, V.R.: Objective picture quality scale (PQS) for image coding. IEEE Transactions on Communications **46**(9), 1215–1226 (1998)
18. Muñoz Moreno, E., Aja-Fernández, S., Martin-Fernandez, M.: A methodology for quality assessment in tensor images. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–6. Anchorage, AK, USA (2008)
19. Mori, S., van Zijl, P.C.M.: Fiber tracking: principles and strategies - a technical review. NMR in Biomedicine **15**, 468–480 (2002)

20. Parker, K.J., Taylor, L.S., Gracewski, S., Rubens, D.J.: A unified view of imaging the elastic properties of tissue. Journal of Acoustical Society of America **117**(5), 2705–2712 (2005)
21. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. Tech. Rep. 5255, Institut Nat. de Recherche en Inform. et en Autom. (INRIA), France (2004)
22. Przelaskowski, A.: Vector quality measure of lossy compressed medical images. Computers in Biology and Medicine **34**(3), 193–207 (2004)
23. Scollan, D.F., Holmes, A., Winslow, R., Forder, J.: Histological validation of myocardial microstructure obtained from diffusion tensor magnetic resonance imaging. American Journal of Physiology **44**(6), 2308–2318 (1998)
24. Sheikh, H., Bovik, A.: Image information and visual quality. IEEE Transactions on Image Processing **15**(2), 430–444 (2006)
25. Sheikh, H., Bovik, A.C., de Veciana, G.: An information fidelity criterion for image quality assessment using natural scene statistics. IEEE Transactions on Image Processing **14**(12), 2117–2128 (2005)
26. Sheikh, H.R., Sabir, M.F., Bovik, A.C.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Transactions on Image Processing **15**(11), 3340–3451 (2006)
27. Tang, H., Cahill, L.: A new criterion for the evaluation of image restoration quality. In: Proc. of the IEEE Region 10 Conf. Tencon 92, pp. 573–577. Melbourne, Australia (1992)
28. Tristán-Vega, A., Aja-Fernández, S.: Joint LMMSE estimation of DWI data for DTI processing. In: Medical Images Computing and Computer-Assisted Intervention (MICCAI), pp. 27–34. Springer-Verlag, New York, NY, USA (2008)
29. Wang, Z., Bovik, A.C.: A universal image quality index. IEEE Signal Processing Letters **9**(3), 81–84 (2002)
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. on Image Proc. **13**(4), 600–612 (2004)
31. Wang, Z., Bovik, A.C., Simoncelli, E.P.: Handbook of image and video processing, 2nd ed., chap. Structural approaches to image quality assessment. Academic Press (2005)
32. Weickert, J.: A review of nonlinear diffusion filtering. In: Lecture Notes in Computer Science. Scale Space Theory in Computer Science, vol. 1252. Springer-Berlin (1997)
33. Weickert, J., Hagen, H. (eds.): Visualization and Processing of Tensor Fields. Springer-Verlag (2006)
34. Van der Weken, D., Nachtegael, M., De Witte, V., Schulte, S., Kerre, E.: A survey on the use and the construction of fuzzy similarity measures in image processing. In: Proceedings of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, pp. 187–192. Giardini Naxos, Italy (2005)
35. Van der Weken, D., Nachtegael, M., Kerre, E.: Using similarity measures and homogeneity for the comparison of images. Image and Vision Computing **22**(9), 695–702 (2004)
36. Van der Weken, D., Nachtegael, M., Kerre, E.: Combining neighbourhood-based and histogram similarity measures for the design of image quality measures. Image and Vision Computing **25**(2), 184–195 (2007)
37. Westin, C.F., Maier, S.E., Mamata, H., Nabavi, A., Jolesz, F.A., Kikinis, R.: Processing and visualization for diffusion tensor MRI. Medical Image Analysis **6**, 93–108 (2002)

# Algorithms for Nonnegative Tensor Factorization

Stefanos Zafeiriou[1]

**Abstract** Nonnegative Matrix Factorization (NMF) is a decomposition which incorporates nonnegativity constraints in both the weights and the bases of the representation. The nonnegativity constraints in NMF correspond better to the intuitive notion of combining parts in order to create a complete object, since the object is represented using only additions of weighted nonnegative basis images. NMF has proven to be very successful for image analysis, especially for imaged-based object representation, discovery of latent object variables and recognition. A drawback of NMF is that it requires the object tensor (with valence more than one) to be vectorized. This procedure may result in information loss since the local object structure is lost due to vectorization. Recently, in order to remedy this disadvantage of NMF methods, Nonnegative Tensor Factorization (NTF) algorithms that can be applied directly to the tensor representation of object collections, have been introduced. In this chapter, we demonstrate how various algorithms are formulated in order to treat arbitrary valence NTFs and we present the various cost functions that have been used for measuring the quality of the approximation. We discuss the optimization procedures that have been used for deriving the factors of the decomposition. Afterwards, we describe how additional constraints can be incorporated into the cost of the decomposition in order to either enhance the sparsity of the solution or to enhance the discrimination between object classes. The presented NTF schemes are described in a manner that can be easily implemented using, in most cases, only matrix multiplications and publicly available packages for treating tensor representations. Finally, we comment on the various applications of NTF algorithms in visual representation and recognition.

Stefanos Zafeiriou
Communications and Signal Processing Research Group
Department of Electrical and Electronic Engineering
Imperial College, London, UK
e-mail: s.zafeiriou@imperial.ac.uk

## 1 Introduction

The Nonnegative Matrix Factorization (NMF) algorithm [1] has attracted a significant interest in the scientific community in the past decade especially for pattern recognition applications. A recent survey on some NMF algorithms can be found in [2]. In pattern recognition, one disadvantage of NMF is that the objects (e.g., the images) should be vectorized in order to find the nonnegative decomposition. This vectorization leads to information loss, since the local structure of objects is lost. In order to remedy this drawback of NMF representation, arbitrary valence Nonnegative Tensor Factorization (NTF) schemes have been proposed [3, 4, 5, 6, 7, 8, 9].

Nowadays, an increasing interest in analysis of large scale tensors has been observed. Tensors are also called multi-way arrays or multidimensional matrices and are generalizations of vectors and matrices which are first order and second order tensors, respectively. To analyze tensor data, decomposition techniques have been widely used in a variety of fields including psychometric, chemometrics, image analysis, graph analysis and signal processing [10, 11, 12]. The tensorization of well-established vector-based algorithms is not an easy procedure and it constitutes a very active research field with many applications in image processing and computer vision.

In order to achieve the extension of NMF to NTF algorithms the most common formulation of the decomposition are the Tucker tensor model [6], the more restricted Parallel Factor (PARAFAC) analysis model (or CANnonical DECOMPosition (CANDECOMP) [17, 18]) and the equivalent to PARAFAC model using Kruskal tensors model [3, 5, 6]. NTF has been initially proposed using PARAFAC [3] and subsequently using Kruskal tensor formulations [5]. The first algorithms for non-negative Tucker decompositions have been proposed in [13] but the decompositions do not allow for all the factors to be constrained non-negative (i.e., do not allow the core tensor to be constrained nonnegative). Recently, research has been conducted [6, 8, 7] in order to extend NMF to arbitrary valence Tucker [14] decompositions, where data, core and mode matrices are nonnegative and multiplicative update rules for all the factors have been proposed. In order to further reduce the ambiguities of the decomposition, updates that can impose sparseness in any combination of modalities have been also proposed in [8]. Moreover, the notion of nonsmoothness [15] for controlling the sparseness has been extended in NTF algorithms [6].

In [9] a series of NTF methods has been proposed using Kruskal tensor formulation. Moreover, by incorporating discriminant constraints inside the NTF decompositions a series of Discriminant Nonnegative Tensor Factorization (DNTF) methods have been presented. These approaches have been tested for face verification and facial expression recognition, where it is shown that they outperform other popular subspace approaches.

In this chapter we will review the basic tensor formulations and the basic algorithms based on multiplicative update rules that have been applied for setting and solving NTF problems. The reminder of the chapter is organized as following. In Section 2 we will describe the NTF tensor formulations and the various cost func-

tions that have been employed in order to measure the error of the approximation. Afterwards we will show, in Section 3, how multiplicative update rules can be derived in order to find the solution and we will discuss the constraints that can be incorporated in the decomposition in order to control the sparseness. Furthermore, in Section 4 we will describe how discriminant constraints can be incorporated in order to find a supervised nonnegative tensor decomposition. In Section 5 we will comment on the applications of NTFs in the fields of image representation and computer vision. Finally, conclusions are drawn in Section 6.

## 2 Formulating Nonnegative Tensor Factorizations

The two most commonly used tensor decompositions are the Tucker model [14] and the more restricted PARAFAC/CANDECOMP model [17, 18]. The RARAFAC model is also equivalent to the decomposition using 1-valence Kruskal tensors [16] which is another popular formulation. In this Section, we will describe how the problem is formulated using Kruskal and Tucker models and we will present the cost functions that are used for measuring the quality of the approximation. The procedures used for the minimization of the cost will be discussed in the following Sections.

### *2.1 Tensor Representations*

Let $\mathbf{X}$ denotes a database of $L$ objects $\mathbf{X}_i$, $i = 1, 2, \ldots, L$. Every object $\mathbf{X}_i$ has a nonnegative representation as a $(n-1)$-valence tensor, i.e., $\mathbf{X}_i \in \Re_+^{I_1 \times \cdots \times I_{n-1}}$, indexed by an $(n-1)$ tuple of indices $(i_1, i_2, \ldots, i_{n-1})$. Thus, the object database is a $n$-valence tensor $\mathbf{X} \in \Re_+^{I_1 \times \cdots \times I_n}$ with $I_n = L$. The dimension $I_j$ is indexed by $i_j = 1, \ldots, I_j$. For example, the most natural way to model a facial image database using a tensor is by a 3-valence tensor $\mathbf{X} \in \Re_+^{I_1 \times I_2 \times I_3}$ [5], where $I_1 \times I_2$ is the resolution of each image (i.e., height and width) and $I_3 = L$ is the number of images in the database. We will consider the general case of $n$-valence tensors in the expositions that follow.

Firstly we will review some of the tensor algebra and notations that will be used throughout the chapter. Scalars will be denoted with lower case letters (i.e., $a$ denotes a scalar), lower case bold letters will denote vectors (ie., $\mathbf{a}$ denotes a vector) while matrices and tensors with valence more than 2 will be denoted with upper case bold letters (i.e., $\mathbf{A}$ is a matrix or a tensor). A vector $\mathbf{a} \in \Re_+^I$ will also be denoted as $\mathbf{a} = [a_i]$, $i = 1 \ldots I$ and $a_i = [\mathbf{a}]_i$ will be the elements of the vector. An arbitrary $n$-valence tensor $\mathbf{A} \in \Re_+^{I_1 \times \ldots \times I_n}$ will be also denoted as $\mathbf{A} = [a_{i_1,\ldots,i_n}]$ containing elements $a_{i_1,\ldots,i_n} = [\mathbf{A}]_{i_1,\ldots,i_n}$ or for simplicity we may use $a_{i_1,\ldots,i_n} = \mathbf{A}_{i_1,\ldots,i_n}$. The operator **vec** stacks the columns of its matrix argument into a single vector.

Let that $\mathbf{a} \in \Re_+^I$ and $\mathbf{b} \in \Re_+^J$ are two real vectors, then their outer product is a matrix defined as:

$$\mathbf{a} \otimes \mathbf{b} = [a_i b_j] \in \Re_+^{I \times J}. \tag{1}$$

Consequently, the outer product of $n$ vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$, with $\mathbf{a}_i \in \Re_+^{I_i}$, $\mathbf{a}_1 \otimes \ldots \otimes \mathbf{a}_n$, is denoted as $\bigotimes_{i=1}^n \mathbf{a}_i$ and is a tensor in $\Re_+^{I_1 \times \ldots \times I_n}$.

The Kronecker product between two matrices $\mathbf{A} \in \Re_+^{I_1 \times J_1}$ and $\mathbf{B} \in \Re_+^{I_2 \times J_2}$ is defined as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}b_{1,1} & \cdots & \mathbf{A}b_{1,J_2} \\ \vdots & \ddots & \vdots \\ \mathbf{A}b_{I_2,1} & \cdots & \mathbf{A}b_{I_2,J_2} \end{bmatrix} \tag{2}$$

and $\mathbf{A} \otimes \mathbf{B} \in \Re_+^{I_1 I_2 \times J_1 J_2}$. In case we have $n$ matrices $\mathbf{A}_1, \ldots, \mathbf{A}_n$ with $\mathbf{A}_i \in \Re_+^{I_i \times J_i}$ the product $\bigotimes_{i=1}^n \mathbf{A}_i$ is an array of $\prod_{i=1}^n I_i \times \prod_{i=1}^n J_i$.

The Khatri-Rao product between two matrices $\mathbf{A} = [\mathbf{a}_1 | \ldots | \mathbf{a}_K] \in \Re_+^{I \times K}$ and $\mathbf{B} = [\mathbf{b}_1 | \ldots | \mathbf{b}_K] \in \Re_+^{J \times K}$ is given by:

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{vec}(\mathbf{a}_1 \otimes \mathbf{b}_1) \ldots \mathbf{vec}(\mathbf{a}_n \otimes \mathbf{b}_n)] \tag{3}$$

and $\mathbf{A} \odot \mathbf{B} \in \Re_+^{IJ \times K}$. In case we have $n$ matrices $\mathbf{A}_1, \ldots, \mathbf{A}_n$, with $\mathbf{A}_i \in \Re_+^{I_i \times K}$, the Khatri-Rao product between them is given by $\mathbf{A}_1 \odot \ldots \odot \mathbf{A}_n = \bigodot_{i=1}^n \mathbf{A}_i \in \Re_+^{(\prod_{i=1}^n I_i) \times K}$.

An important operation between a tensor $\mathbf{G} \in \Re_+^{J_1 \times \ldots \times J_n}$ and a matrix $\mathbf{A}^{(j)} \in \Re_+^{I_j \times J_j}$ is the $j$-mode product defined as:

$$\begin{array}{l} [\mathbf{G} \times_n \mathbf{A}^{(j)}]_{m_1,\ldots,m_{j-1},i_j,m_{n+1},\ldots,m_n} = \sum_{m_j=1}^{J_j} g_{m_1,\ldots,m_n} a_{i_j,m_n} \\ 1 \leq m_l \leq J_l,\ 1 \leq l \leq n,\ 1 \leq i_l \leq I_l \end{array} \tag{4}$$

which is a tensor in $\Re_+^{J_1 \times \ldots \times J_{j-1} \times I_j \times J_{j+1} \times \ldots \times J_n}$.

We will often make use of transformations that change tensors into matrices ($j$-mode matricization) and matrices into vectors (vectorization). The $j$-mode matricization transforms the tensor $\mathbf{G}$ into a matrix $\mathbf{G}_{(j)}$ defined by:

$$\mathbf{G}_{(j)} \in \Re_+^{J_j \times M} \text{ with } M = \prod_{l=1, l \neq j}^n J_l \tag{5}$$

for each $j = 1, \ldots, n$. Note that the number of rows of $\mathbf{G}_{(j)}$ is equal to the size of the $j$-th dimension of the tensor. The number of columns is expanded to accommodate all the other dimensions of the tensor. The matrix $\mathbf{G}_{(j)}$ can be expressed elementwise as:

$$[\mathbf{G}_{(l)}]_{j_l,k} = g_{j_1,\ldots,j_l,\ldots,j_n} \text{ with } k = 1 + \sum_{m=1, m \neq l}^n [(j_m - 1) \prod_{r=1, r \neq m}^m J_r]. \tag{6}$$

In the remaining of this chapter $\bullet$ and $/$ tensor operators will denote the elementwise multiplication and division between tensors. tr$[\mathbf{A}]$ denotes the trace of the matrix $\mathbf{A}$. The vector of ones is denoted as $\mathbf{1}$ while the tensor and matrix of ones is denoted as $\mathbf{E}$. A tensor which has all elements equal to zero except those for which

all indices are the same is called a superdiagonal tensor. If all nonzero elements equal unity, then is is referred to as the unit superdiagonal tensor $\mathbf{I}$. The identity matrix $\mathbf{I}$ is a 2-valence unit diagonal tensor.

### 2.1.1 Nonnegative Tensor Factorization Using Kruskal tensors

A $n$-valence tensor $\mathbf{X}$ is of rank at most $K$ if it can be expressed as a sum of $K$ rank-1 Kruskal tensors i.e., a sum of $K$ $n$-fold outer-products. In the NTF framework $\mathbf{X}$ can be decomposed as a sum of $K$ $n$-fold outer-products as:

$$\begin{array}{c} \mathbf{X} \approx \sum_{l=1}^{K} \bigotimes_{j=1}^{n} \mathbf{u}_j^l \Leftrightarrow \\ x_{i_1,\ldots,i_n} \approx \sum_{l=1}^{K} u_{i_1,1}^l \ldots u_{i_n,n}^l,\ 0 \le i_j \le I_j,\ 1 \le j \le n \end{array} \tag{7}$$

with $\mathbf{u}_j^l \in \Re_+^{I_j}$ and $\mathbf{u}_j^l = [u_{1,j}^l, \ldots, u_{I_j,j}^l]^T$. That is, NTF aims at finding the best rank $K$ approximation of $\mathbf{X}$ with respect to an approximation cost. The NTF factorization using Kruskal tensors is pictorially described in Figure 1. As can be seen, the tensor $\mathbf{X} \in \Re_+^{I_1 \times I_2 \times I_3}$ is represented as a sum of $K$ outer product tensors $\mathbf{u}_1^l \otimes \mathbf{u}_2^l \otimes \mathbf{u}_3^l$.

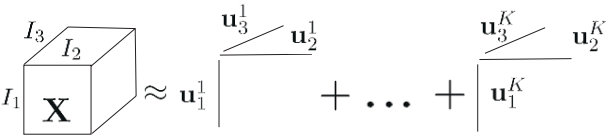



**Fig. 1** Visualization of the rank-$K$ approximation of a 3-valence tensor using Kruskal tensor notation.

Lets define the matrices $\mathbf{U}_j \triangleq [\mathbf{u}_j^1 | \ldots | \mathbf{u}_j^K]$, $j = 1, \ldots, n$ or equivalently $\mathbf{U}_j = [u_{i_j,j}^l] \in \Re_+^{I_j \times K}$, $1 \le i_j \le I_j$, $1 \le j \le n$, $1 \le l \le n$. These matrices will be used for defining NTF algorithms using matrix multiplications. The notation $\mathbf{u}_{i_j,j}$ holds for the column vector that corresponds to the $i_j$-th row of the matrix $\mathbf{U}_j$. A vectorization of the tensor decomposition $\mathbf{vec}(\sum_{l=1}^{K} \bigotimes_{j=1}^{n} \mathbf{u}_j^l)$ is given by summing the columns of the matrix $\bigodot_{l=1}^{K} \mathbf{U}_l$.

The NMF problem $\mathbf{X} \approx \mathbf{ZH}$ in [1] can be easily derived from (7) by selecting $\mathbf{Z} = \mathbf{U}_1$ and $\mathbf{H} = \mathbf{U}_2^T$.

### 2.1.2 Nonnegative Tensor Factorization Using Tucker tensors

In terms of the Tucker tensor model the factorization is written as:

$$\begin{array}{c} \mathbf{X} \approx \mathbf{G} \times_1 \mathbf{A}^{(1)} \ldots \times_n \mathbf{A}^{(n)} \Leftrightarrow \\ x_{i_1,\ldots,i_n} \approx \sum_{j_1=1,\ldots,j_n=1}^{J_1,\ldots,J_n} g_{j_1,\ldots,j_n} a_{i_1,j_1}^{(1)} \ldots a_{i_n,j_n}^{(n)} \\ 1 \le i_l \le I_l,\ 1 \le j_l \le J_l,\ 1 \le l \le n \end{array} \tag{8}$$

where $\mathbf{A}^{(m)} = [a^{(m)}_{i_m,j_m}] \in \Re_+^{I_m \times J_m}$, $m = 1,\ldots,n$ are the so-called mode matrices. The Tucker model is more general than the Kruskal decomposition and it allows the use of the core tensor $\mathbf{G} \in \Re_+^{J_1 \times \ldots \times J_n}$. Using the $j$-mode product the matrix factorization is written as $\mathbf{X} \approx \mathbf{G} \times \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} = \mathbf{A}^{(1)}\mathbf{G}\mathbf{A}^{(2)^T}$.

The decomposition $\mathbf{X} \approx \mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}$ for 3-valence tensor $\mathbf{X} \in \Re_+^{I_1 \times I_2 \times I_3}$ and $\mathbf{G} \in \Re_+^{J_1 \times J_2 \times J_3}$ is pictorially described in Figure 2. The matrix $\mathbf{A}^{(2)}$ has not been transposed, for reasons of symmetry. Multiplication with $\mathbf{A}^{(1)}$ involves linear combination of the "horizontal matrices" (index $j_1$ fixed) in $\mathbf{G}$. Multiplication of $\mathbf{G}$ with $\mathbf{A}^{(1)}$ means that every column of $\mathbf{G}$ (indices $j_2$ and $j_3$ fixed) has to be multiplied from the left with $\mathbf{A}^{(1)}$. Similarly, multiplication with $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$ involves linear combinations of matrices, obtained by fixing $j_2$ and $j_3$, respectively. This can be considered as a multiplication, from the left, of the vectors obtained by fixing the indices $j_3$ and $j_1$ and $j_1$ and $j_2$, respectively.

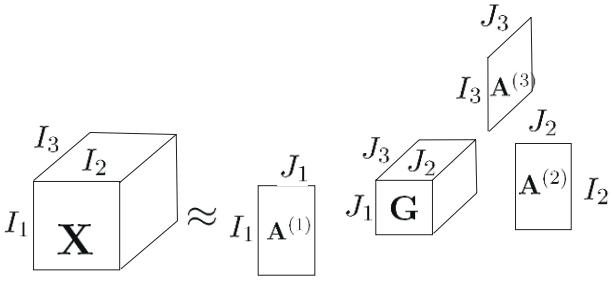



**Fig. 2** Visualization of the multiplication of a 3-valence tensor $\mathbf{G} \in \Re_+^{J_1 \times J_2 \times J_3}$ with matrices $\mathbf{A}^{(1)} \in \Re_+^{I_1 \times J_1}$, $\mathbf{A}^{(2)} \in \Re_+^{I_2 \times J_2}$ and $\mathbf{A}^{(3)} \in \Re_+^{I_3 \times J_3}$.

A simplification or the Tucker model is the CANDECOMP/PARAFAC model. The main difference between these two models is the presence of the core tensor in the Tucker model. The core tensor allows column vectors of mode matrices to interact each other in order to reconstruct the original tensor. In the PARAFAC model $J_1 = J_2 = \ldots = J_n = K$ and the core tensor is selected as $\mathbf{G}_{j_1,\ldots,j_n} \neq 0$ if and only if $j_1 = j_2 = \ldots = j_n$ (diagonal tensor). In case that the tensor $\mathbf{G}$ is the unit super-diagonal tensor, then the nonnegative Tucker decomposition in (8) can be written as:

$$\begin{array}{c} x_{i_1,\ldots,i_n} \approx \sum_{l=1}^{K} a^{(1)}_{i_1,l} \ldots a^{(n)}_{i_n,l}\, , a^{(j)}_{i_j,l} \in \Re_+,\ 1 \leq i_j \leq I_j \Leftrightarrow \\ \mathbf{X} \approx \sum_{l=1}^{K} \bigotimes_{j=1}^{n} \mathbf{a}^{(j)}_l,\ \mathbf{a}^{(j)}_l \in \Re_+^{I_j} \end{array} \tag{9}$$

where $\mathbf{A}^{(j)} = [\mathbf{a}^{(j)}_1 | \ldots | \mathbf{a}^{(j)}_K]$. The above model is equivalent to the Kruskal decomposition in (7) in case we select as rank-1 Kruskal tensors the columns of the matrices $\mathbf{A}^{(j)}$, (i.e., $\mathbf{u}^l_j \equiv \mathbf{a}^{(j)}_1$).

Strictly speaking, only the PARAFAC model has an uniqueness property up to scaling and permutation of the mode matrices [19, 16]. Even though the Tucker model has not the uniqueness property, it is more useful in data compression. That

is, the Tucker model may require a smaller number of component vectors for mode matrices than the PARAFAC model, since it uses every combination of mode vectors (column vectors of mode matrices).

## 2.2 Cost Functions

Many cost functions have been proposed in order to set the NTF decomposition. Every cost function models in a different way the error of the approximation. Among the most popular cost functions is the Least Squares Error (LSE) which corresponds to a homoscedastic gaussian noise model and the Kullback-Leibler (KL) divergence which corresponds to Poisson noise.

In the Kruskal tensor notation the cost function for NTF decomposition is defined as:

$$\begin{aligned} d_{LK}(\mathbf{u}_1^1,\ldots,\mathbf{u}_n^K) &= \tfrac{1}{2}||\mathbf{X}-\textstyle\sum_{l=1}^K \bigotimes_j \mathbf{u}_j^l||_F^2 \\ &= \tfrac{1}{2}\textstyle\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}(\mathbf{X}_{i_1,\ldots,i_n}-\sum_{l=1}^K\prod_j u_{i_j,j}^l)^2 \end{aligned} \tag{10}$$

where $||.||_F$ is the Frobenius norm. The corresponding cost with nonnegative Tucker decomposition is given by:

$$\begin{aligned} d_{LT}(\mathbf{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(n)}) &= \tfrac{1}{2}||\mathbf{X}-\mathbf{G}\times_1\mathbf{A}^{(1)}\ldots\times_n\mathbf{A}^{(n)}||_F^2 \\ &= \tfrac{1}{2}\textstyle\sum_{i_1=1,\ldots,i_n=1,j_1=1,\ldots,j_n=1}^{I_1,\ldots,I_n}(\mathbf{X}_{i_1,\ldots,i_n} \\ &\quad -\textstyle\sum_{j_1=1,\ldots,j_n=1}^{J_1,\ldots,J_n}\mathbf{G}_{j_1,\ldots,j_n}\mathbf{A}_{i_1,j_1}^{(1)}\ldots\mathbf{A}_{i_n,j_n}^{(n)})^2. \end{aligned} \tag{11}$$

The KL divergence for the NTF with Kruskal decomposition is:

$$\begin{aligned} d_{KLK}(\mathbf{u}_1^1,\ldots,\mathbf{u}_n^K) &= \textstyle\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}(\mathbf{X}_{i_1,\ldots,i_n}\ln(\frac{\mathbf{X}_{i_1,\ldots,i_n}}{\sum_{m=1}^K\prod_{j=1}^n u_{i_j,j}^m})-\mathbf{X}_{i_1,\ldots,i_n} \\ &\quad +\textstyle\sum_{m=1}^K\prod_{j=1}^n u_{i_j,j}^m) \end{aligned} \tag{12}$$

and with Tucker model:

$$\begin{aligned} &d_{KLT}(\mathbf{G},\,\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(n)}) \\ &\quad=\textstyle\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}(\mathbf{X}_{i_1,\ldots,i_n}\ln(\frac{\mathbf{X}_{i_1,\ldots,i_n}}{\sum_{j_1=2,\ldots,j_n=1}^{J_1,\ldots,J_n}\mathbf{G}_{j_1,\ldots,j_n}\mathbf{A}_{i_1,j_1}^{(1)}\ldots\mathbf{A}_{i_n,j_n}^{(n)}}) \\ &\quad-\mathbf{X}_{i_1,\ldots,i_n}+\textstyle\sum_{j_1=1,\ldots,j_n=1}^{J_1,\ldots,J_n}\mathbf{G}_{j_1,\ldots,j_n}\mathbf{A}_{i_1,j_1}^{(1)}\ldots\mathbf{A}_{i_n,j_n}^{(n)}). \end{aligned} \tag{13}$$

The NTF optimization problems are formulated by requiring the minimization of one of the above cost functions subject to nonnegativity constraints for all factors.

# 3 Solving the Optimization Problems Using Multiplicative Update Rules

The most popular way for solving constrained optimization problems like for NMF and NTF is via multiplicative update rules. In order to derive the update rules for minimizing the cost function we should first define a proper auxiliary function and afterwards optimize it. A function $W$ is an auxiliary function for $Y(\mathbf{F})$ if $W(\mathbf{F},\mathbf{F}^{(t)}) \geq Y(\mathbf{F})$ and $W(\mathbf{F},\mathbf{F}) = Y(\mathbf{F})$. If $W$ is an auxiliary function of $Y$, then $Y$ is nonincreasing under the update $\mathbf{F}^{(t+1)} = \arg\min_{\mathbf{F}} W(\mathbf{F},\mathbf{F}^{(t)})$[1]. All the multiplicative update rules that are described in this section have been derived from defining proper auxiliary functions.

## *3.1 Update Rules For Least Squares Error and Kruskal tensors*

The update rules that guarantee a nonincreasing behavior of (10) under nonnegativity constraints are [5]:

$$u^{l}_{i_j,j}{}^{(t+1)} = u^{l}_{i_j,j}{}^{(t)} \frac{\sum_{i_1=1,\ldots,i_{j-1}=1,i_{j+1}=1,\ldots,i_n=1}^{I_1,\ldots,I_{j-1},I_{j+1},\ldots,I_n} \mathbf{X}_{i_1,\ldots,i_n} \prod_{r\neq j} u^{l}_{i_r,r}}{\sum_{m=1}^{K} u^{m}_{i_j,j}{}^{(t)} \prod_{r\neq j} (\mathbf{u}^{m}_{r})^{T} (\mathbf{u}^{l}_{r})}. \tag{14}$$

For all the factors $u^{l}_{i_p,p}$ with $1 \leq p \leq j-1$ we use the estimation at time $t+1$ while for the rest of the factors we use the estimation at time $t$. Let that we allow a total of $r$ iterations, then the complexity of the NTF algorithm is $O(rnK\prod_{j=1}^{n} I_j)$.

The algorithm that is described by the update rules in (14) can be implemented using only matrix operations as following. In the following we will describe it for the 3D NTF case. However this algorithm can be easily extended for treating arbitrary valence NTF decompositions. Initially $\mathbf{U}_1^{(0)}$, $\mathbf{U}_2^{(0)}$ and $\mathbf{U}_3^{(0)}$ are assumed to be initialized with random nonnegative entries. At every iteration the matrix $\mathbf{U}_1$ is updated as:

$$\mathbf{U}_1^{(t+1)} = \mathbf{U}_1^{(t)} \bullet (\mathbf{X}_{(1)}(\mathbf{U}_2^{(t)} \odot \mathbf{U}_3^{(t)}) / (\mathbf{U}_1^{(t)}(\mathbf{U}_2^{(t)} \odot \mathbf{U}_3^{(t)})^{T}(\mathbf{U}_2^{(t)} \odot \mathbf{U}_3^{(t)}))). \tag{15}$$

For the second matrix $\mathbf{U}_2$ we have:

$$\mathbf{U}_2^{(t+1)} = \mathbf{U}_2^{(t)} \bullet (\mathbf{X}_{(2)}(\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_3^{(t)}) / (\mathbf{U}_2^{(t)}(\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_3^{(t)})^{T}(\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_3^{(t)}))). \tag{16}$$

and for the third matrix:

$$\begin{aligned}\mathbf{U}_3^{(t+1)} = \mathbf{U}_3^{(t)} \bullet (\mathbf{X}_{(3)}(\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_2^{(t+1)}) / (\mathbf{U}_3^{(t)}(\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_2^{(t+1)})^{T} \\ (\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_2^{(t+1)}))).\end{aligned} \tag{17}$$

### 3.2 Update Rules for KL divergence and Kruskal Tensors

Lets define the following optimization problem for solving NTF using KL divergence:

$$\begin{aligned} &\min_{\mathbf{u}_{i_j}^m} d_{KLK}(\mathbf{u}_1^1 \dots \mathbf{u}_n^K) \\ &\text{subject to } u_{i_j,j}^m \geq 0,\ 1 \leq i_j \leq I_j, 1 \leq j \leq n, 1 \leq m \leq K \\ &\text{and } \sum_{i_j=1}^{I_j} u_{i_j,j}^m = 1,\ j \neq n. \end{aligned} \tag{18}$$

The above additional constraint (i.e., $\sum_{i_j=1}^{I_j} u_{i_j,j}^m = 1\ , j \neq n$) is added by following the same reasoning as the constraint of the bases of the NMF to sum up to one. That is, the normalization of the matrices $\mathbf{U}_1, \dots, \mathbf{U}_{n-1}$ is a generalization of the normalization of the bases matrix in the NMF algorithms for eliminating the degeneracy associated with the invariance of $\mathbf{ZH}$ under the transformation $\mathbf{Z} \rightarrow \mathbf{ZP}$, $\mathbf{H} \rightarrow \mathbf{P}^{-1}\mathbf{H}$, where $\mathbf{P}$ is a diagonal matrix with positive diagonal elements.

In that case the update rules for $j \neq n$ are given by:

$$(\acute{u}_{i_j,j}^m)^{(t)} = u_{i_j,j}^{m\ (t)} \frac{\sum_{i_1=1,\dots,i_{j-1}=1,i_{j+1}=1,\dots,i_n=1}^{I_1,\dots,I_{j-1},I_{j+1},\dots,I_n} \mathbf{X}_{i_1,\dots,i_n} \frac{\prod_{r=1,r\neq j}^{n} u_{i_r,r}^m}{\sum_{l=1}^{K} u_{i_j,j}^{l\ (t)} \prod_{r=1,r\neq j}^{n} u_{i_r,r}^m}}{\sum_{i_n=1}^{I_n} u_{i_n,n}^m}. \tag{19}$$

We also normalize the terms $(\acute{u}_{i_j,j}^m)^{(t)}$ so as:

$$u_{i_j,j}^{m\ (t+1)} = \frac{(\acute{u}_{i_j,j}^m)^{(t)}}{\sum_{i_j=1}^{I_j} (\acute{u}_{i_j,j}^m)^{(t)}} \tag{20}$$

in order to obtain $u_{i_j,j}^{m\ (t+1)}$. The corresponding update rules for the factors $u_{i_n,n}^m$ are given by:

$$u_{i_n,n}^{m\ (t+1)} = u_{i_n,n}^{m\ (t)} \sum_{i_1=1,\dots,i_{n-1}=1}^{I_1,\dots,I_{n-1}} \mathbf{X}_{i_1,\dots,i_n} \frac{u_{i_1,1}^m \dots u_{i_{n-1},n-1}^m}{\sum_{l=1}^{K} u_{i_1,1}^l \dots u_{i_{n-1},n-1}^l u_{i_n,n}^{l\ (t)}}. \tag{21}$$

The above algorithm using only matrix operation for the 3D NTF case is written as follows. At every iteration the following updates should be repeated for each of the matrices $\mathbf{U}_1$, $\mathbf{U}_2$ and $\mathbf{U}_3$. The matrix update rules presented in the following are equivalent to the update rules in (19), (20) and (21). The matrices $\mathbf{U}_1$, $\mathbf{U}_2$ and $\mathbf{U}_3$ are initialized using random nonnegative entries and the columns of $\mathbf{U}_1$ and $\mathbf{U}_2$ are normalized in such a way that every column sums up to one.

The matrix $\mathbf{U}_1 \in \Re_+^{I_1 \times K}$ is updated as:

$$\tilde{\mathbf{U}}_1^{(t)} = \mathbf{U}_1^{(t)} \bullet ((\mathbf{X}_{(1)} / (\mathbf{U}_1^{(t)} (\mathbf{U}_2^{(t)} \odot \mathbf{U}_3^{(t)})^T)(\mathbf{U}_2^{(t)} \odot \mathbf{U}_3^{(t)})) / \mathbf{B}^{(t)}). \tag{22}$$

The matrix $\mathbf{B}_1^{(t)} \in \Re_+^{I_1 \times K}$ is formulated by repeating the vector

$$\mathbf{b}^{(t)} = [\sum_{i_3} {u^1_{i_3,3}}^{(t)} \dots \sum_{i_3} {u^K_{i_3,3}}^{(t)}]$$

as a row $I_1$ times, i.e., $\mathbf{B}_1^{(t)} = \begin{bmatrix} {\mathbf{b}^{(t)}}^T \\ \vdots \\ {\mathbf{b}^{(t)}}^T \end{bmatrix}$. The estimation $\mathbf{U}_1^{(t+1)}$ is obtained from the matrix $\tilde{\mathbf{U}}_1^{(t)}$ when it is normalized in such a way that every column sums up to one.

Afterwards, for $\mathbf{U}_2 \in \Re_+^{I_2 \times K}$:

$$\tilde{\mathbf{U}}_2^{(t)} = \mathbf{U}_2^{(t)} \bullet ((\mathbf{X}_{(2)} / (\mathbf{U}_2^{(t)} (\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_3^{(t)})^T)(\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_3^{(t)}))/\mathbf{B}_2^{(t)}. \tag{23}$$

$\mathbf{B}_2^{(t)}$ is obtained in the same way as $\mathbf{B}_1$ by in this case repeating the vector $\mathbf{b}^{(t)}$ $I_2$ times. For obtaining $\mathbf{U}_2^{(t+1)}$ the matrix $\tilde{\mathbf{U}}_2^{(t)}$ is normalized so that every column sums up to one.

Finally, the matrix $\mathbf{U}_3 \in \Re_+^{I_3 \times K}$ is updated as:

$$\mathbf{U}_3^{(t+1)} = \mathbf{U}_3^{(t)} \bullet ((\mathbf{X}_{(3)} / (\mathbf{U}_3^{(t)} (\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_2^{(t+1)})^T)(\mathbf{U}_1^{(t+1)} \odot \mathbf{U}_2^{(t+1)}))). \tag{24}$$

### *3.3 Update Rules for Least Squares Error and Tucker Model*

Using Tucker tensor formulation the minimization of (11) under nonnegativity constraints is formulated as:

$$\begin{array}{l} \text{minimize}_{\mathbf{G},\mathbf{A}^{(1)},\dots,\mathbf{A}^n} \ \frac{1}{2}||\mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \dots \times_n \mathbf{A}^{(n)} - \mathbf{X}||^2 \\ \text{subject to } [\mathbf{G}]_{j_1,\dots,j_n} \geq 0, [\mathbf{A}^{(m)}]_{i_m,j_m} \geq 0 \\ 1 \leq m \leq n,\ 1 \leq i_m \leq I_m,\ 1 \leq j_m \leq J_m. \end{array} \tag{25}$$

In order to solve the above optimization problem and derive a set of update rules for all factors, the LSE should be written in such a form that the factors $\mathbf{A}^{(j)}$ and $\mathbf{G}$ will be isolated. The properties of $j$-mode matricization can be used for this purpose and the LSE cost can be expanded as:

$$\begin{array}{l} d_{LST}(\mathbf{G}, \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(n)}) \\ \quad = \frac{1}{2}||\mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \dots \times_n \mathbf{A}^{(n)} - \mathbf{X}||_F^2 \\ \quad = \frac{1}{2}||(\mathbf{I} \otimes (\bigotimes_{l=n,l\neq j}^1 \mathbf{A}^{(l)} \mathbf{G}_{(j)}^T))\mathbf{vec}(\mathbf{A}^{(j)})^T - \mathbf{vec}(\mathbf{X}_{(j)})^T||_2^2. \end{array} \tag{26}$$

As can be seen, the Kronecker product $\bigotimes_{l=n,l\neq j}^1 \mathbf{A}^{(l)}$ does not include $\mathbf{A}^{(j)}$ which is the optimization variable. The above vectorization leads to $n$ least squares subproblems (one for each $\mathbf{A}^{(j)}$) over the vectors $\mathbf{vec}(\mathbf{A}^{(j)})$.

The $j$-th subproblem is a constrained linear least-squares problem with respect to the vector $\mathbf{vec}(\mathbf{A}^{(j)})$

$$\begin{array}{l}\text{minimize}_{\mathbf{A}^{(j)}} \ \frac{1}{2}||(\mathbf{I}\otimes(\bigotimes_{l=1,l\neq j}\mathbf{A}^{(l)}\mathbf{G}_{(j)}^{T}))\text{vec}(\mathbf{A}^{(j)})^{T}-\text{vec}(\mathbf{X}_{(j)})^{T}||_{2}^{2}\\ \text{subject to } [\mathbf{A}^{(j)}]_{i_j,j_j}\geq 0,\ 1\leq i_j\leq I_j,\ 1\leq j_j\leq J_j.\end{array} \tag{27}$$

Update rules that guarantee the nonincreasingness of the cost in (27) can be derived from an auxiliary function similar to the one used in [1].

By using the $j$-mode matricization of the tensor $\mathbf{X}$ into the matrix $\mathbf{X}_{(j)}\in\Re_{+}^{I_j\times\prod_{l=1,l\neq j}^{n}I_l}$ the Tucker model may lead to the factorization:

$$\mathbf{X}_{(j)}\approx\mathbf{R}_j=\mathbf{A}^{(j)}\mathbf{G}_{(j)}(\bigotimes_{l=n,l\neq j}^{1}\mathbf{A}^{(l)})^{T}=\mathbf{A}^{(j)}\mathbf{Z}_{(j)} \tag{28}$$

where $\mathbf{Z}_{(j)}=\mathbf{G}_{(j)}(\bigotimes_{l=n,l\neq j}^{1}\mathbf{A}^{(l)})^{T}$. The matrix $\bigotimes_{l=n,l\neq j}^{1}\mathbf{A}^{(l)}\mathbf{G}_{(n)}^{T}$ arises at several key points in the analysis that follows. This overall product has dimensions $\prod_{l=n,l\neq j}^{1}I_l\times J_j$ and usually $J_j\ll I_l$. The intermediate matrix $\bigotimes_{l=n,l\neq j}^{1}\mathbf{A}^{(n)}$ has dimensions $\prod_{l=n,l\neq j}^{1}I_l\times\prod_{l=n,l\neq j}^{1}J_l$. Then, for $j=1,\ldots,n$ we have the following update rules:

$$\begin{array}{rl}\mathbf{R}_{(j)} &= \mathbf{A}^{(j)^{(t)}}\mathbf{Z}_{(j)}\\ \mathbf{A}^{(j)^{(t+1)}} &= \mathbf{A}^{(j)^{(t)}}\bullet(\frac{\mathbf{X}_{(j)}\mathbf{Z}_{(j)}{}^{T}}{\mathbf{R}_{(j)}\mathbf{Z}_{(j)}{}^{T}}).\end{array} \tag{29}$$

For the matrix $\mathbf{Z}_{(j)}$ we use the estimation $\mathbf{A}^{(l)^{(t+1)}}$ for $1\leq l\leq j-1$ and $\mathbf{A}^{(l)^{(t)}}$ for $j+1\leq l\leq n$.

In order to obtain the core tensor $\mathbf{G}$ we may proceed as in the case of $\mathbf{A}^{(j)}$ and similarly we may transform the cost as:

$$d_{LT}(\mathbf{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(n)})=\frac{1}{2}||\bigotimes_{j=n}^{1}\mathbf{A}^{(j)}\mathbf{vec}(\mathbf{G}_{(1)})-\mathbf{vec}(\mathbf{X}_{(1)})||_{2}^{2}. \tag{30}$$

The optimization problem is as follow:

$$\begin{array}{l}\text{minimize}_{\mathbf{G}} \ \frac{1}{2}||\bigotimes_{j=n}^{1}\mathbf{A}^{(j)}\mathbf{vec}(\mathbf{G}_{(1)})-\mathbf{vec}(\mathbf{X}_{(1)})||_{2}^{2}\\ \text{subject to } [\mathbf{G}]_{j_1,\ldots,j_n}\geq 0,\ 1\leq j_l\leq J_l,\ 1\leq l\leq n.\end{array} \tag{31}$$

By using $\mathbf{vec}(\mathbf{G}_{(1)})$ in an auxiliary function in [1] the following multiplicative update rules:

$$\begin{array}{rl}\mathbf{R} &= \mathbf{G}\times_1\mathbf{A}^{(1)}\times_2\ldots\times_n\mathbf{A}^{(n)}\\ \mathbf{B} &= \mathbf{X}\times_1\mathbf{A}^{(1)}\times_2\ldots\times_n\mathbf{A}^{(n)}\\ \mathbf{C} &= \mathbf{R}\times_1\mathbf{A}^{(1)}\times_2\ldots\times_n\mathbf{A}^{(n)}\\ \mathbf{G}^{(t+1)} &= \mathbf{G}^{(t)}\bullet(\frac{\mathbf{B}}{\mathbf{C}})\end{array} \tag{32}$$

guarantee the nonincreasingness of the cost (30). For all the matrices $\mathbf{A}^{(j)}$ for $1\leq j\leq n$ in (32) we use the estimation in time $t+1$. Let that the algorithm be iterated $r$ times. Then, the complexity of the Tucker decomposition is $O(r(n+1)\prod_{l=1}^{n}I_l\prod_{l=1}^{n}J_l)$.

### 3.3.1 Imposing Sparseness

One of the most popular ways for enhancing the sparseness is by requiring the minimization of $L_1$ norm of the various factors [8]. Let that for the $j$-th matrix $\mathbf{A}^{(j)}$ we enforce sparseness constraints by incorporating the additional cost $||\mathbf{vec}(\mathbf{A}_{(j)}^T)||_1$ in the main LSE cost function as:

$$\begin{array}{l} \text{minimize}_{\mathbf{A}^{(j)}} \ \frac{1}{2}||(\mathbf{I}\otimes(\bigotimes_{l=n,l\neq j}^{1}\mathbf{A}^{(l)}\mathbf{G}_{(j)}^T))\text{vec}\mathbf{A}^{(j)^T} - \text{vec}\mathbf{X}_{(j)}^T||_2^2 \\ +\gamma||\mathbf{vec}(\mathbf{A}^{(j)})||_1 \\ \text{subject to } \mathbf{A}^{(j)} \geq 0,\ 1 \leq j \leq n \end{array} \tag{33}$$

where $\gamma$ is a nonnegative constant that controls the effect of $||\mathbf{vec}(\mathbf{A}^{(j)})||_1$ in the total cost. Then, the update rules are the following:

$$\mathbf{A}^{(j)^{(t+1)}} = \mathbf{A}^{(j)^{(t)}} \bullet (\frac{\mathbf{X}_{(j)}\mathbf{Z}_{(j)}{}^T}{\mathbf{R}_{(j)}\mathbf{Z}_{(j)}{}^T+\gamma}). \tag{34}$$

For the core tensor the optimization problem with additional sparseness constraints for $\mathbf{G}$ are:

$$\begin{array}{l} \text{minimize}_{\mathbf{G}} \ \frac{1}{2}||\bigotimes_{j=n}^{1}\mathbf{A}^{(j)}\mathbf{vec}(\mathbf{G}_{(1)}) - \mathbf{vec}(\mathbf{X}_{(1)})||_2^2 + \gamma||\mathbf{G}||_1 \\ \text{subject to } [\mathbf{G}]_{j_1,\ldots,j_n} \geq 0,\ 1 \leq j_l \leq J_l,\ 1 \leq l \leq n \end{array} \tag{35}$$

and the corresponding update rules for the core tensor $\mathbf{G}$ are:

$$\mathbf{G}^{(t+1)} = \mathbf{G}^{(t)} \bullet (\frac{\mathbf{B}}{\mathbf{C}+\gamma}). \tag{36}$$

## *3.4 Update Rules for KL Divergence and Tucker Model*

Using a similar procedure as in the minimization of LSE, we could write the cost function (13) by isolating all $\mathbf{A}^{(j)}$ and $\mathbf{G}$ defining in that way $n+1$ optimization problems (an optimization problem for obtaining every mode matrix $\mathbf{vec}(\mathbf{A}^{(j)})$ and one for obtaining the core tensor $\mathbf{vec}(\mathbf{G}_{(1)})$, respectively). Then, by using $\mathbf{vec}(\mathbf{A}^{(j)})$ and $\mathbf{vec}(\mathbf{G}_{(1)})$ in the auxiliary function of KL divergence the following update rules, for $j = 1,\ldots n$ can be defined:

$$\begin{array}{rl} \mathbf{R}_{(j)} &= (\mathbf{A}^{(j)})^{(t)}\mathbf{Z}_{(n)} \\ (\mathbf{A}^{(j)})^{(t+1)} &= (\mathbf{A}^{(j)})^{(t)} \bullet (\frac{(\frac{\mathbf{X}_{(j)}}{\mathbf{R}_{(j)}})\mathbf{Z}_{(j)}{}^T}{\mathbf{E}_{(j)}\mathbf{Z}_{(j)}{}^T}). \end{array} \tag{37}$$

The corresponding update rules for $\mathbf{G}$ are:

$$\begin{aligned} \mathbf{R} &= \mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \ldots \times_n \mathbf{A}^{(n)} \\ \mathbf{D} &= \tfrac{\mathbf{X}}{\mathbf{R}} \times_1 \mathbf{A}^{(1)} \times_2 \ldots \times_n \mathbf{A}^{(n)} \\ \mathbf{F} &= \mathbf{E} \times_1 \mathbf{A}^{(1)} \times_2 \ldots \times_n \mathbf{A}^{(n)} \\ \mathbf{G}^{(t+1)} &= \mathbf{G}^{(t)} \bullet (\tfrac{\mathbf{D}}{\mathbf{F}}). \end{aligned} \tag{38}$$

A similar problem to the minimization of LSE with $L_1$ constraints can be formulated for KL divergence [8], as well. But, for KL divergence a more powerful scheme for controlling the sparseness has been proposed [15, 6], initially for NMF, and afterwards extended for NTF. This scheme is described in the following.

### 3.4.1 Nonsmooth Nonnegative Tucker Factorization

One of the simplest and most effective methods for controlling the sparseness of NMF has been proposed in [15] and has been incorporated in NMF with KL divergence, the so-called nonsmooth NMF (nsNMF). The nsNMF model [6] is described by $\mathbf{X} \approx \mathbf{ZMH}$, introducing a smoothing matrix $\mathbf{M}$ given by:

$$\mathbf{M} = (1-\theta)\mathbf{I} + \frac{\theta}{K}\mathbf{1}\mathbf{1}^T \tag{39}$$

where the parameter $\theta$ satisfies $0 \leq \theta \leq 1$. The parameter $\theta$ controls the extent of smoothness of the matrix operator $\mathbf{M}$. For $\theta = 0$, the model (39) is equivalent to the original NMF (i.e., $\mathbf{X} \approx \mathbf{ZH}$). As $\theta \to 1$, stronger smoothing is imposed on $\mathbf{M}$, leading to a strong sparseness on both $\mathbf{Z}$ and $\mathbf{H}$ in order to maintain the faithfulness of the model to the data.

The notion of nonsmoothness can be extended in order to enforce sparseness in nonnegative tucker factorizations. The nonsmooth NTF (nsNTF) using Tucker model is given by:

$$\begin{aligned} \mathbf{X} &\approx (\mathbf{G} \times_1 \mathbf{M}_1 \times_2 \ldots \times_n \mathbf{M}_n) \times_1 \mathbf{A}^{(1)} \times_2 \ldots \times_n \mathbf{A}^{(n)} \\ &\approx \mathbf{G} \times_1 \mathbf{A}^{(1)}\mathbf{M}_1 \times_2 \ldots \times_n \mathbf{A}^{(n)}\mathbf{M}_n \end{aligned} \tag{40}$$

where $\mathbf{M}_1, \ldots, \mathbf{M}_n$ are the smoothing matrices for each mode defined similar to (39). They smooth the core tensor and mode matrices simultaneously as $\mathbf{M}$ smoothes $\mathbf{Z}$ and $\mathbf{H}$ in the nsNMF. For the same reason, a sparseness on both $\mathbf{G}$ and $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(n)}$ is obtained. The update rules can be derived from (37) by replacing $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(n)}$ into $\mathbf{A}^{(1)}\mathbf{M}^{(1)}, \ldots, \mathbf{A}^{(n)}\mathbf{M}^{(n)}$ in original mode matrices. The update rules for $\mathbf{G}$ can be derived by replacing $\mathbf{G}$ in (38) into $\mathbf{G} \times_1 \mathbf{M}^{(1)} \times_2 \ldots \times_n \mathbf{M}^{(n)}$.

## 4 Supervised Nonnegative Tensor Factorization

In [20, 21] discriminant constraints have been incorporated in the NMF decomposition motivated by minimizing the Fisher criterion [22]. These decompositions have

been motivated by the need of finding basis images that correspond to discriminant parts of faces. The discriminant cost functions that have been implemented refer to the matrix $\mathbf{H}$ which has as elements the coefficients of the NMF decomposition.

In DNMF the discriminant constraints concern the coefficients of the decomposition. The problem involves the selection of proper coefficients of the NTF decomposition on which the discriminant analysis should be applied. In order to answer this question lets examine the decomposition of a 3-valence tensor. In this case the tensor $\mathbf{X} \in \Re_+^{I_1 \times I_2 \times I_3}$ is a 3D matrix that is built of slices that are the images $\mathbf{X}_1, \ldots, \mathbf{X}_{I_3}$ and every image $\mathbf{X}_i \in \Re_+^{I_1 \times I_2}$. Let $\mathbf{U}_1 = [\mathbf{u}_1^1, \ldots, \mathbf{u}_1^K]$, $\mathbf{U}_2 = [\mathbf{u}_2^1, \ldots, \mathbf{u}_2^K]$ and $\mathbf{U}_3 = [\mathbf{u}_3^1, \ldots, \mathbf{u}_3^K]$. Let $(\mathbf{U}_1 \odot \mathbf{U}_2)\mathbf{U}_3^T = [\mathbf{x}_1, \ldots, \mathbf{x}_{I_3}]$, then each vector $\mathbf{x}_i$ is the vectorized image $\mathbf{X}_i$, i.e. the image $\mathbf{X}_i$ scanned columnwise. That is, each vectorized image $\mathbf{x}_i$ is a linear combination of the $\mathbf{vec}(\mathbf{u}_1^j \mathbf{u}_2^{jT})$ with the coefficients of the decomposition taken from the $i$-th row of the matrix $\mathbf{U}_3$. Thus, the weights of the representation are stored in the matrix $\mathbf{U}_3$, while the bases are found by combining the two matrices $\mathbf{U}_1$ and $\mathbf{U}_2$.

In the same way as in the DNMF decomposition (or like other methods, e.g. like Principal Component Analysis (PCA) plus Linear Discriminant Analysis (LDA) [22]), the discriminant analysis is incorporated in the coefficients of the decomposition, which in the 3-valence tensor case corresponds to the elements of the matrix $\mathbf{U}_3$. The $j$-th row $u_{j,3}^1, \ldots, u_{j,3}^K$ of the matrix $\mathbf{U}_3$ corresponds to the coefficients of the image $\mathbf{X}_j$. Following the same reasoning we can generalize for the case of $n$-valence tensors. The matrix $\mathbf{U}_n \in \Re_+^{I_n \times K}$ has rows that correspond to coefficients of the decomposition of the objects $\mathbf{X}_i \in \Re_+^{I_1 \times I_2 \ldots \times I_{n-1}}$. Let $\mathbf{v}_i = [u_{i_n,n}^1, \ldots, u_{i_n,n}^K]$ be a column vector with elements the $i$-th row of the matrix $\mathbf{U}_n$. This vector contains the representation coefficients of the object $\mathbf{X}_i$.

The objects $\mathbf{A}_i$ are separated to $R$ different object classes. The coefficient vectors $\mathbf{v}_i$ are separated to $R$ classes $(\mathscr{V}_1, \ldots, \mathscr{V}_R)$, as well. The within and between class scatter matrices, for these vectors are defined as:

$$\acute{\mathbf{S}}_w = \sum_{r=1}^{R} \sum_{\mathbf{v} \in \mathscr{V}_r} (\mathbf{v} - \acute{\mu}^{(r)})(\mathbf{v}_i - \acute{\mu}^{(r)})^T \text{ and } \acute{\mathbf{S}}_b = \sum_{r=1}^{R} N_r (\acute{\mu}^{(r)} - \acute{\mu})(\acute{\mu}^{(r)} - \acute{\mu})^T \quad (41)$$

where $\acute{\mu}^{(r)} = [\acute{\mu}_1^{(r)}, \cdots, \acute{\mu}_{I_n}^{(r)}]$ is the mean vector of the class of vectors $\mathbf{v} \in \mathscr{V}_r$ and the $\acute{\mu} = [\acute{\mu}_1, \cdots, \acute{\mu}_{I_n}]$ is the grand mean of $\mathscr{V}$. The divergence with the discriminant cost function is:

$$d_d(\mathbf{u}_1^1, \ldots, \mathbf{u}_n^K) = d_{KLK}(\mathbf{u}_1^1, \ldots, \mathbf{u}_n^K) + \gamma \mathrm{tr}[\acute{\mathbf{S}}_w] - \delta \mathrm{tr}[\acute{\mathbf{S}}_b]. \quad (42)$$

and the corresponding optimization problem of the factorization is:

$$\begin{array}{l} \min_{\mathbf{u}_{i_j}^m} d_d(\mathbf{u}_1^1, \ldots, \mathbf{u}_n^K) \\ \text{subject to} \sum_{i_j=1}^{I_j} u_{i_j,j}^m = 1 \text{ and } u_{i_j,j}^m \geq 0,\ 1 \leq i_j \leq I_j, 1 \leq j \leq n, 1 \leq m \leq K. \end{array} \quad (43)$$

The update rules that guarantee a non increasing behavior of (42), under nonnegativity constraints, for $j \neq n$, are given by equations (19) and (20). For $j = n$, let $\mathscr{F}_r = \{\sum_{\rho=1}^{r-1} N_\rho + 1, \cdots, \sum_{\rho=1}^{r} N_\rho\}$. Then, for the objects of the $r$-th class (i.e., $i_n \in \mathscr{F}_r$) the update rules are given by:

$$u_{i_n,n}^{m\ (t+1)} = \frac{\acute{T} + \sqrt{\acute{T}^2 + 4\varepsilon \sum_{i_1,\ldots,i_{n-1}} \mathbf{G}_{i_1,\ldots,i_n} \frac{u_{i_1,1}^{m\ (t+1)} \ldots u_{i_n,n}^{m\ (t)}}{\sum_{l=1}^{K} u_{i_1,1}^{l\ (t+1)} \ldots u_{i_n,n}^{l\ (t)}}}}{2\varepsilon} \tag{44}$$

where $\acute{T}$ is given by:

$$\acute{T} = (2\gamma + 2\delta)(\frac{1}{N_r} \sum_{\lambda \neq i_n, \lambda \in \mathscr{F}_r} (u_{\lambda,n}^m)) - 2\delta\acute{\mu}_m - 1 \tag{45}$$

and $\varepsilon = 2\gamma - (2\gamma + 2\delta)\frac{1}{N_r}$. The above update rules can be obtained by using a similar to [20] procedure and when having an 2-valence tensor $\mathbf{X}$ it degenerates to the DNMF method in [20].

## 5 Applications of Nonnegative Tensor Factorizationss

Even though the applications of NTF span several disciplines in this chapter we are particularly interested in the applications of NTF in image processing and computer vision.

### *5.1 Application to Image Representation and Unmixing*

We demonstrate the power of NTF framework providing an empirical verification of the fact that NTF can produce better bases in terms of interpretability and sparseness than NMF and its effect on the success of recreating the underlying model. One dataset that has been used to this end is the Swimmer database. The Swimmer database has been used in [23] for demonstrating a case when the NMF can provide a unique decomposition into parts. Some of the images of the Swimmer dataset can be seen in Figure 3 (a). The Swimmer image set is comprised of 256 images of dimensions $32 \times 32$ forming a tensor $\mathbf{X} \in \Re_+^{32\times32\times256}$. Each image contains a "torso" (the invariant part) of 12 pixels in the center and four "limbs" of 6 pixels that can be in one of 4 positions. In [23] the NMF scheme [1] correctly resolves the local parts but fails on the torso, which is shown as a "ghost" part in all images (also shown in Figure 3 (b) ). The NTF on the other hand, contains a unique factorization and correctly resolves the parts (Figure 3) ).

Another example that demonstrates the potential of NTF is the problem of resolving local parts from a single image (a similar experiment has been considered

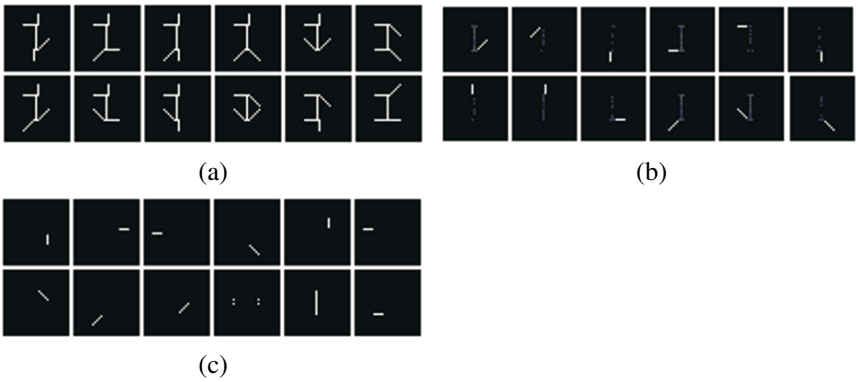


**Fig. 3** Some images of the Swimmer database and the corresponding bases for NMF and NTF, a) Swimmer images; b) NMF bases images; c) NTF bases images .

in [5, 4]). In an NMF framework this cannot be achieved as a single image, even if copied multiple times, would still be decomposed into itself. With NTF on the other hand, the single image has been copied 20 times forming a tensor $\mathbf{X} \in \Re_{+}^{32\times32\times20}$. The NTF algorithm, described in Section 3.2, has been applied on this tensor. The used image and the bases of the decomposition can be seen in Figure 4 (a). As can be seen by using only one image certain parts of the original image can be retrieved through NTF. In Figure 4 (b) some of the bases of NTF when applied to a facial image database are shown. These bases have been retrieved by the application of NTF to the set of $19 \times 19$, facial images of the MIT CBCL database.

**Fig. 4** a) The decomposition of one image using NTF; b) Bases of the CBCL database from NTF.

A consequence of representing the image set as a 3D tensor is that the spatial redundancy is factored in the decomposition (which is not the case when the images are vectorized as in the NMF framework) – therefore one should expect a more efficient representation (higher compression rate). This has been demonstrated providing in [5] simple experiments where the authors have compared the number of NTF and NMF factors that are needed in order to reconstruct images. It has been validated that there is an increased coding efficiency of the tensor representation compared to a 2D NMF representations. Similar experiments have been conducted in [7] where an NTF algorithm has been proposed using the Tucker model and $\alpha$

divergence has been minimized. The role of the parameter $\alpha$ has been also experimentally explored for facial image reconstruction. Finally, in [6] the authors have proposed NTF schemes using the Tucker tensor model and the superiority of NTF in comparison to NMF has been demonstrated in facial image reconstruction.

Another experiment in [6] shows the resistance of NTF decomposition to 'salt & pepper' noise. As shown the learned basis images of NMF are very sensitive to noise while the bases images of NTF have shown great insensitivity. A possible explanation of the fact that NTF is superior to NMF in this problem is that NMF breaks the neighborhood structure since it needs to reshape 2D images into 1D vectors. In contrast to NTF, 2D neighboring structure is preserved in the framework of 3-way tensor, leading to better reconstructed images in the case of the 'salt & pepper' noise. In [7] the authors demonstrated a simple experiment of estimating the source facial images in randomly mixed 3D tensors. Moreover, in [24] NTF have been applied for spectral unmixing.

The sparsity of NTF representations has been studied in [8, 6]. In [6] a method for controlling the sparseness of NTF by extending the notion of nonsmoothness to arbitrary valence tensors has been proposed. It has also been demonstrated that the $\theta$ parameter in the nonsmooth NTF (nsNTF, described in Section 3.4.1) controls in an efficient way the sparseness of the bases. In [8] the methods for enhancing the sparsity of NTF have been proposed and it has also been shown that they control in an efficient way the sparseness of tensor decompositions.

## *5.2 Application to Clustering*

The application of NMF to clustering has been initiated in [25] where it has been shown that there is a close relationship between spectral clustering and NMF. In [26] the problem of clustering data given complex relations (beyond pairwise relationships) between the data points has been considered. The complex $n$-wise relationships between data can be modelled by a $n$-valence tensor where each entry corresponds to an affinity measure (usually nonnegative) over an $n$-tuple of data points. It has been shown that a probabilistic assignment of data points to clusters is equivalent, under mild conditional independence assumptions, to a super-symmetric nonnegative factorization of the closest hyper-stochastic version of the input $n$-valence affinity tensor. NTF decomposition has been applied to a number of visual interpretation problems including 3D multi-body segmentation and illumination-based clustering of human faces.

## *5.3 Facial Image Characterization Applications*

The usefulness of the NTF algorithm in the extraction of facial features for classification has been demonstrated in [4, 9]. In [4] the NTF has been applied for feature

extraction for face detection. The feature vector representing an image was the inner-product between the factors and the input image. Those measurement vectors over positive (faces) and negative (non-faces) examples were fed into various classifiers (Support Vector Machines (SVM) and Adaboost). The MIT CBCL face has been used and it has been shown that the features extracted by the NTF factors generated the higher classification accuracy when compared to NMF and PCA.

NMF and DNMF algorithms have been proven useful for facial features extraction for face recognition and face verification [20, 27]. In [9] NTF and DNTF approaches have been applied to face verification problems. The experiments were conducted in the XM2VTS database using the protocol described in [9, 20]. The authors in [9] have compared the proposed methods with other popular subspace methods like [12, 22] and has been verified that the best results have been achieved using the DNTF and NTF plus LDA frameworks.

Apart from face recognition/verification and detection NMF based algorithms have been proven very useful for facial expression analysis and recognition [21, 28]. In [9] NTF based algorithms have been applied to the facial expression recognition problem since DNMF has shown superior performance in that problem [21, 28]. The database used for the facial expression recognition experiments in [9] was created using the Cohn-Kanade database. The differences images have been used for facial expression recognition. The differences images are created by subtracting the neutral image intensity values from the corresponding values of the fully expressive facial expression image, were calculated. Each differences images was initially normalized, resulting in an image built only from positive values. The differences images are used instead of the original facial expressive images, due to the fact that in the differences images, the facial parts in motion are emphasized [28]. It has been shown that DNTF algorithm described in Section 4 achieved the best results in facial expression recognition.

## 6 Conclusions and Discussion

In this chapter we have described various NTF methods and we have discussed on their applications to image representation and computer vision. We have described various tensor models used for formulating the NTF problems. We have presented the most popular cost functions that are used for measuring the error of approximation. We have shown how multiplicative update rules can be derived for solving the optimization problem. Subsequently, we have shown how constraints can be incorporated in the cost function in order to enhance the sparseness of the NTF decomposition. Moreover, we have described a method for supervised Nonnegative Tensor Factorization by incorporating discriminant cost functions in the NTF decomposition. Finally, we have described the applications of NTF in computer vision and image representation problems.

The NTF-based optimization problems are non-convex. They may have several local minima and produce a sequence of iterations. A common misunderstanding is

that the limit points of this sequence are local minima. In optimization theory most non-convex optimization methods guarantee only the stationarity of the limit points. Such a property is very useful as a local minimum must be a stationary point. On the other hand the multiplicative update rules proposed that far for arbitrary valence NTF algorithms they only guarantee that the cost function is nonincreasing under these rules.

Another problem for NTF methods is that the quality of the decomposition highly depends on the initialization of the various factors. The most common approach for initialization is by using random nonnegative values. In [6] a first approach for initiliazing NTF using Tucker decomposition has been proposed. Finally, another open problem is the optimal choice of $K$. Actually, this a problem that all decomposition methods face and actually there is no standard way for setting $K$ a priori. Typically in pattern recognition problems $K$ is selected as the one that minimizes the classification error.

## References


[1] Lee, D.D. and Seung, H.S.: Algorithms for Non-negative Matrix Factorization. NIPS, 556-562 (2000)

[2] M. Berry, M. Browne, A. Langville, P. Pauca and R. Plemmons: Algorithms and Applications for Approximate Nonnegative Matrix Factorization. Computational Statistics and Data Analysis, **52,1**, 155–173 (2007)

[3] M. Welling and M. Weber: Positive tensor factorization. Pattern Recognition Letters, **22,12**, 1255–1261 (2001)

[4] T. Hazan, S. Polak and A. Shashua: Sparse image coding using a 3D non-negative tensor factorization. Tenth IEEE International Conference on Computer Vision, 2005 (ICCV 2005), **1**, 50–57 (2005)

[5] A. Shashua and T. Hazan: Non-Negative Tensor Factorization with Applications to Statistics and Computer Vision. International Conference of Machine Learning (ICML), Bonn, Germany, 2005.

[6] Y.-D. Kim and S. Choi: Nonnegative Tucker Decomposition. Proceedings of the IEEE CVPR-2007 Workshop on Component Analysis Methods, Minneapolis, Minnesota (2007)

[7] Y.-D. Kim, A. Cichocki and S. Choi: Nonnegative Tucker decomposition with alpha-divergence. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2008), Las Vegas, Nevada (2008)

[8] M. Mørup, L. K. Hansen and S. M. Arnfred: Algorithms for Sparse Nonnegative Tucker Decomposition. Neural Computation, **20,8**, 2112–2131 (2008)

[9] S. Zafeiriou: Discriminant Nonnegative Tensor Factorization Algorithms. IEEE Transactions on Neural Networks accepted for publication

[10] L. De Lathauwer, B. De Moor and J. Vandewalle: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl., **21,4**, 1253–1278 (2000)

[11] D. Tao, X. Li, X. Wu and S. J. Maybank : General Tensor Discriminant Analysis and Gabor Features for Gait Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 10, pp. 1700–1715, 2007.

[12] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang and H.-J. Zhang: Multilinear Discriminant Analysis for Face Recognition. IEEE Transactions on Image Processing, **16,1**, 212–220 (2007)

[13] C. A. Andersson and R. Bro: The N-way Toolbox for MATLAB. Chemometrics and Intelligent Laboratory Systems, **52**, 1–4 (2000)

[14] L. R. Tucker: Some mathematical notes on three-mode factor analysis. Psychometrika, **31**, 279–311 (1996)

[15] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann and R. D. Pascual-Marqui: Nonsmooth Nonnegative Matrix Factorization (nsNMF). IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 3, pp. 403–415 (2006)

[16] J.B. Kruskal : Three way arrays: rank and uniqueness of trilinear decomposition, with application to arithmetic complexity and statistics. Linear Algebra and its Applications, **18**, 95–138 (1977)

[17] R. A. Harshman: Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis, UCLA Working Papers in Phonetics (1970)

[18] J. Carroll and J. Chang: Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. Psychometrika, **35**, 283–319 (1970)

[19] R. A. Harshman: Determination and proof of minimum uniqueness conditions for PARAFAC1. UCLA Working Papers in Phonetics, 111–117 (1972)

[20] S. Zafeiriou, A. Tefas, I. Buciu and I. Pitas: Exploitining Discriminant Information in Nonnegative Matrix Factorization with Application tom Frontal Face Verification. IEEE Transactions on Neural Networks, **14,8**, 1063–1073 (2006)

[21] I. Kotsia, S. Zafeiriou and I. Pitas: A Novel Discriminant Nonnegative Matrix Factorization Method with Application to Facial Image Characterization Problems. IEEE Transactions on Information Forensics and Security, **2,3**, 588–595 (2007)

[22] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, **19,7**, 711–720 (1997)

[23] D. Donoho and V. Stodden: When does non-negative matrix factorization give a correct decomposition into parts?. Advances in Neural Information Processing Systems, **17**, (2004)

[24] Q. Zhang, H. Wang, R. Plemmons and P. Pauca: Tensor Methods for Hyperspectral Data Analysis of Space Objects. Preprint (2008)

[25] C. Ding, X. He and H. D. Simon: On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. Proc. SIAM Intl. Conf. Data Mining, 606–610 (2005)

[26] A. Shashua, R. Zass and T. Hazan : Multi-way Clustering Using Super-symmetric Nonnegative Tensor Factorization. Proceedings of the European Conference on Computer Vision (ECCV), Graz, Austria (2005)

[27] S. Zafeiriou, A. Tefas and I. Pitas: Discriminant NMFfaces for Frontal Face Verification. Proceedings of IEEE International Workshop on Machine Learning for Signal Processing ì (2005)

[28] I. Kotsia, S. Zafeiriou and I. Pitas: Texture and Shape Information Fusion for Facial Expression and Facial Action Unit Recognition. Pattern Recognition, **41,3**, 833–851 (2008)

# PDE-based Morphology for Matrix Fields: Numerical Solution Schemes

Bernhard Burgeth, Michael Breuß, Stephan Didas, and Joachim Weickert

**Abstract** Tensor fields are important in digital imaging and computer vision. Hence there is a demand for morphological operations to perform e.g. shape analysis, segmentation or enhancement procedures. Recently, fundamental morphological concepts have been transferred to the setting of fields of symmetric positive definite matrices, which are symmetric rank two tensors. This has been achieved by a matrix-valued extension of the nonlinear morphological partial differential equations (PDEs) for dilation and erosion known for grey scale images. Having these two basic operations at our disposal, more advanced morphological operators such as top hats or morphological derivatives for matrix fields with symmetric, positive semidefinite matrices can be constructed. The approach realises a proper coupling of the matrix channels rather than treating them independently. However, from the algorithmic side the usual scalar morphological PDEs are transport equations that require special upwind-schemes or novel high-accuracy predictor-corrector approaches for their adequate numerical treatment. In this chapter we propose the non-trivial extension of these schemes to the matrix-valued setting by exploiting the special algebraic structure available for symmetric matrices. Furthermore we compare the performance and juxtapose the results of these novel matrix-valued high-resolution-type (HRT) numerical schemes by considering top hats and morphological derivatives applied to artificial and real world data sets.

Bernhard Burgeth, Michael Breuß, Stephan Didas*, and Joachim Weickert
Department of Mathematics and Computer Science, Saarland University, Germany
email: {burgeth,breuss,didas,weickert}@mia.uni-saarland.de

* Funded by the Excellence Cluster 'Multimodal Computing and Interaction' at the Department of Mathematics and Computer Science, Saarland University, Germany

# 1 Introduction

Initiated by Serra's and Matheron's work on binary morphology [33, 46] in the sixties, mathematical morphology has developed into a powerful discipline that provides versatile tools to the image processing community. Over the last four decades morphological concepts have been generalised and extended not only to greyscale images but also to vector-valued images and image sequences. Numerous monographs, e.g. [26, 34, 47, 48, 51], and conference proceedings, e.g. [21, 22, 27, 32, 53], bear testimony to the variety and the success of mathematical morphology.

Recently, the fundamental morphological operations of dilation and erosion, and with it some other basic morphological operators, have been made available for matrix-valued data, *matrix fields* for short [11].

The interest of the image processing community in matrix-valued data has been triggered mainly by the advent of diffusion tensor magnetic resonance imaging (DT-MRI) [3]. This 3-D medical imaging technique assigns to each voxel a tensor, i.e., a positive semidefinite $3 \times 3$-matrix, which provides in vivo information about the diffusion of water molecules in biological tissue. As such it mirrors the geometry and organisation of the tissue under examination and is a very valuable diagnostic tool [39]. Furthermore, in image analysis itself tensors turned out to be a useful concept [23]: The *structure tensor* [18], for instance, (also known as Förstner interest operator, second moment matrix or scatter matrix) is used for corner detection [25], but also for motion [4] and texture analysis [40]. Finally, in physics and engineering anisotropic behaviour of quantities is described by tensors such as inertia, diffusion, permittivity and stress-strain tensors. Hence, modern image processing should provide appropriate tools to analyse matrix fields.

To fix notation we consider a matrix field as a mapping

$$F : \Omega \subset \mathbb{R}^d \longrightarrow M_k(\mathbb{R}) \tag{1}$$

from a $d$-dimensional image domain into the set of $k \times k$-matrices with real entries, $F(x) = (f_{p,q}(x))_{p,q=1,\ldots,k}$. Unless otherwise stated we will concentrate throughout this chapter on matrix fields with values in the set $\mathrm{Sym}(k)$ of all *symmetric* matrices, $\mathrm{Sym}(k) \subset M_k(\mathbb{R})$. In most applications this type of matrices is of practical relevance.

In [10] novel matrix-valued partial differential equations governing dilation and erosion for matrix fields have been proposed and numerically solved by a first-order scheme of Osher and Sethian generalised to the setting of matrices.

However, in the case of scalar images the morphological differential equations of dilation and erosion can also be solved numerically by a flux corrected transport (FCT) scheme as introduced in [8]. This scheme follows a predictor-corrector strategy which allows for an almost perfect preservation of edges and discontinuities outperforming even a high-resolution scalar variant of the Osher-Sethian scheme. In what follows, we denote the corresponding Osher-Sethian schemes by OS-I and OS-II, respectively.

As a novelty we propose in this chapter high-resolution-type matrix-valued extensions of OS-II as well as of the FCT scheme. Furthermore, we compare in experiments on matrix fields the performance of OS-I, OS-II, FCT, and we juxtapose the results to the output of the ordering based morphological operators which were introduced and investigated in [15, 12].

This chapter is organized as follows: First we deal in Section 2 with mathematical morphology for grey scale images. We present the ordering and the PDE-based approach for the two fundamental operations, dilation and erosion. For the nonlinear hyperbolic PDEs governing dilation and erosion we present three schemes for their numerical solution. The section ends with a short review of some basic morphological derivative operators. Section 3 gives a compact description how the operations of dilation and erosion can be defined for matrix fields. To this end we utilise the Loewner partial ordering for symmetric matrices to define a supremum and infimum of a finite set of symmetric matrices. In Section 4 we establish matrix-valued PDEs for dilation and erosion, and we introduce matrix-valued counterparts of the three schemes for the numerical solution of the dilation/erosion equations in the setting of matrix fields. In our experiments we apply morphological operations to artificial and real DT-MRI data sets. The operations are realised via ordering and via the three numerical schemes of the PDE-based approach. We report on the comparison of the results in Section 5. The remarks in Section 6 conclude this chapter.

## 2 Morphology for Grey Scale Images

In this section we briefly recall the definitions of some basic scalar-valued morphological operators whose matrix-valued generalisations will be of interest in this chapter. First we focus on the two very different approaches to the fundamental operations of mathematical morphology: dilation and erosion.

The first approach is based on ordering leading to the so-called flat morphology while the continuous-scale morphology relies on partial differential equations. After introducing these approaches, we briefly recall some important morphological operations based on subsequent applications of dilation/erosion.

A second topic addressed here are numerical schemes for scalar-valued PDE-based morphology. After a general discussion, we introduce the methods we use in this paper, giving them a formulation easily carried over to the tensor-valued setting discussed later.

### *2.1 Erosion and Dilation Based on Ordering*

In flat morphology for a scalar image $f(x,y)$ the so-called *structuring element* (SE) is a set $B \subset \mathbb{R}^2$ that determines the neighbourhood relation of pixels with respect to

a shape analysis task. Often convex sets such as disks, ellipses or squares are used as structuring elements.

Grey scale *dilation* $\oplus$ replaces the greyvalue of the image $f(x,y)$ by its supremum/maximum within a mask defined by $B$,

$$(f \oplus B)(x,y) := \sup\{f(x-x', y-y') \mid (x',y') \in B\}, \tag{2}$$

while *erosion* $\ominus$ is determined by taking the infimum/minimum,

$$(f \ominus B)(x,y) := \inf\{f(x+x', y+y') \mid (x',y') \in B\}. \tag{3}$$

The notions of supremum and infimum only make sense if an ordering of the grey values is possible. The following PDE-based approach is in principle free from this requirement.

## 2.2 Erosion and Dilation by PDEs

In [43, 44] nonlinear partial differential equations were proposed that mimic the process of dilation and erosion of an image $f$ with a ball as structuring element. These Eikonal equations read

$$\begin{aligned} \partial_t u &= \pm\|\nabla u\| := \sqrt{(\partial_x u)^2 + (\partial_y u)^2} \quad \text{on} \quad \Omega \times (0,+\infty) \\ \partial_n u &= 0 \quad \text{on} \quad \partial\Omega \times (0,+\infty) \\ u(x,y,0) &= f(x,y) \quad \text{for all} \quad (x,y) \in \Omega \end{aligned} \tag{4}$$

The evolution process governed by (4) is initialised with the original image $f$ and yields transformed versions $u(\cdot,t)$ for any $t \in (0,+\infty)$. Here $\partial_n u$ denotes the outward normal derivative of $u$ at the boundary $\partial\Omega$ of the image domain $\Omega$. The plus sign $+$ realises the dilation, while the minus sign $-$ corresponds to erosion.

The dilation/erosion PDEs (4) belong to the class of *hyperbolic* PDEs, see e.g. [16, 17] for introductions. Hyperbolic processes describe transport processes and are strongly linked to wave propagation. An important property of solutions to hyperbolic PDEs is that discontinuities, often called *shocks*, generally arise. Note in the context of this work, that the resolution of shocks requires specifically tailored numerical schemes, see e.g. [31].

## 2.3 Morphological Operations

The combination of dilation and erosion lead to various other morphological operators such as *opening* and *closing*,

$$f \circ B := (f \ominus B) \oplus B, \qquad f \bullet B := (f \oplus B) \ominus B, \tag{5}$$

the *white top-hat* and its dual, the *black top-hat*,

$$\mathrm{WTH}(f) := f - (f \circ B), \qquad \mathrm{BTH}(f) := (f \bullet B) - f, \tag{6}$$

and finally, the *self-dual top-hat*,

$$\mathrm{SDTH}(f) := (f \bullet B) - (f \circ B). \tag{7}$$

Boundaries of objects are loci of high grey value variations in an image, and as such they can be detected by derivative operators. The so-called *Beucher gradient*

$$\rho_B(f) := (f \oplus B) - (f \ominus B), \tag{8}$$

as well as the *internal* and *external gradient*,

$$\rho_B^-(f) := f - (f \ominus B), \qquad \rho_B^+(f) := (f \oplus B) - f, \tag{9}$$

are morphological counterparts of the norm of the gradient $f$, $\|\nabla f\|$, if $f$ is considered as a differentiable image.

In [55] a *morphological Laplacian* has been introduced. But we define a variant by

$$\Delta_B f := \rho_B^+(f) - \rho_B^-(f) = (f \oplus B) - 2 \cdot f + (f \ominus B). \tag{10}$$

This Laplacian is a morphological equivalent of the second derivative $\partial_{\eta\eta} f$ where $\eta$ stands for the unit vector in the direction of the steepest slope. It allows us to distinguish between influence zones of minima and maxima of the image $f$. This is a vital property for the construction of so-called *s*hock filters [24, 30, 37]. Shock filtering amounts to applying either a dilation or an erosion to an image, depending on whether the pixel is located within the influence zone of a minimum or a maximum:

$$S_B f := \begin{cases} f \oplus B, & \Delta_B f < 0, \\ f, & \Delta_B f = 0, \\ f \ominus B, & \Delta_B f > 0. \end{cases} \tag{11}$$

A considerable number of variants of shock filters have been considered in the literature [1, 20, 36, 41, 45, 56]. When they are applied iteratively, experiments show that their steady state is given by a piecewise constant image with discontinuities ("shocks") between adjacent segments of constant grey value. For more details about the morphological shock filter as introduced above see [12].

These operators are at our disposal once we have succeeded to performing dilation and erosion on matrix fields. Depending on the quality of the discrete realisations these two operations we will see considerable differences in the output of the composed morphological operators.

## *2.4 Numerical Schemes for PDEs of Erosion or Dilation*

In the context of PDE-based mathematical morphology, first-order finite difference methods such as the *Osher-Sethian scheme* [35, 38, 49] and the *Rouy-Tourin method* [42, 54] are adequate choices. A typical design feature of such PDE-based algorithms for mathematical morphology consists of diffusive numerical effects necessary to capture propagating shocks. Unfortunately, this also leads to a blurring of edges.

The construction of an accurate method yielding sharp edges is a non-trivial task. In [50, 52], the attempt to circumvent this blurring by means of using higher-order ENO[1] interpolants within numerical schemes was investigated in scalar-valued morphology. However, schemes like these are very difficult to implement in a tensor-valued setting since the mathematical concept behind ENO interpolants does not carry over. On the other hand, as we show in this work, it is possible to define a reasonable tensor-valued analogue of the high-resolution extension of the first-order Osher-Sethian scheme, and also the *flux-corrected transport* (FCT) scheme introduced in [8] for scalar-valued morphology can be extended to the tensor-valued setting. Consequently, we denote the resulting schemes as being of *high-resolution-type* (HRT), and these are to the knowledge of the authors the first schemes in the area of tensor-valued data constructed for the purpose of a high-quality resolution.

In this section, we briefly review the first-order Osher-Sethian scheme (OS-I) as well as its high-resolution extension (OS-II), and the FCT scheme for the scalar-valued 2-D case. Restricting the presentation to this setting, all numerical aspects will become evident while the notation is not overloaded. We employ the notation $u_{ij}^n$ as the grey value of the image $u$ at the pixel centred at $(ih_x, jh_y) \in \mathbb{R}^2$ at the time-level $n\tau$ of the evolution. For the convenience of the reader, the formulae are already given in a format so that the coding procedure is extendable to the 3-D tensor-valued setting in a straightforward fashion:

- Instead of grey values $u_{ij}^n$ the reader may employ tensors $U^n(ih_x, jh_y)$.
- Instead of the minmod-function defined below in a scalar-valued setting, the reader may employ its tensor-valued generalisation discussed in section 4.
- The formulae can be extended straightforwardly to tensors in 3-D $U^n(ih_x, jh_y, kh_z)$.

Also, we only describe the schemes for morphological dilation, as algorithms for erosion just incorporate a switch of sign, compare (4).

For a compact notation, we employ the usual abbreviations for forward and backward difference operators, i.e.,

$$D_+^x u_{i,j}^n := u_{i+1,j}^n - u_{i,j}^n \quad \text{and} \quad D_-^x u_{i,j}^n := u_{i,j}^n - u_{i-1,j}^n \,. \tag{12}$$

These operators can be defined analogously with respect to the $y$-direction, and they can be formally concatenated yielding the obvious results. It also turns out to be advantageous to define the central difference operator

[1] ENO means *essentially non-oscillatory*

$$D_c^x u_{i,j}^n := u_{i+1,j}^n - u_{i-1,j}^n \tag{13}$$

which should be understood accordingly.

## 2.5 The basic Osher-Sethian Scheme

The first-order-accurate Osher-Sethian scheme for morphological dilation referred to as OS-I is given by

$$\begin{aligned} u_{i,j}^{n+1} = u_{i,j}^n + \tau \Bigg( & \left( \frac{1}{h_x} \min\left(D_-^x u_{i,j}^n, 0\right) \right)^2 + \left( \frac{1}{h_x} \max\left(D_+^x u_{i,j}^n, 0\right) \right)^2 \\ & + \left( \frac{1}{h_y} \min\left(D_-^x u_{i,j}^n, 0\right) \right)^2 + \left( \frac{1}{h_y} \max\left(D_+^x u_{i,j}^n, 0\right) \right)^2 \Bigg)^{1/2} . \end{aligned} \tag{14}$$

The scheme (14) is largely identical to the first-order upwind scheme of Rouy and Tourin [42, 54], but with the exception of the treatment of extrema of $u_{i,j}^n$.

## 2.6 The Osher-Sethian Scheme with High-Resolution Correction

For notational convenience, we first write down this enhanced scheme, named OS-II, in semidiscrete form keeping the time derivative, compare [38]. Note, that it is at this stage identical to (14) with the exception of terms tweaking the numerical derivatives:

$$\begin{aligned} \frac{\partial}{\partial t} u_{i,j}(t) = \Bigg( & \left( \frac{1}{h_x} \min\left( D_-^x u_{i,j}^n + \frac{1}{2} \mathrm{mm}\left(D_-^x D_+^x u_{i,j}^n, D_-^x D_-^x u_{i,j}^n\right), 0 \right) \right)^2 \\ & + \left( \frac{1}{h_x} \max\left( D_+^x u_{i,j}^n - \frac{1}{2} \mathrm{mm}\left(D_+^x D_+^x u_{i,j}^n, D_-^x D_+^x u_{i,j}^n\right), 0 \right) \right)^2 \\ & + \left( \frac{1}{h_y} \min\left( D_-^y u_{i,j}^n + \frac{1}{2} \mathrm{mm}\left(D_-^y D_+^y u_{i,j}^n, D_-^y D_-^y u_{i,j}^n\right), 0 \right) \right)^2 \\ & + \left( \frac{1}{h_y} \max\left( D_+^y u_{i,j}^n - \frac{1}{2} \mathrm{mm}\left(D_+^y D_+^y u_{i,j}^n, D_-^y D_+^y u_{i,j}^n\right), 0 \right) \right)^2 \Bigg)^{1/2} . \end{aligned} \tag{15}$$

In (15), the function $\mathrm{mm}(\cdot,\cdot)$ denotes the so-called minmod-function defined as

$$\mathrm{mm}(a,b) := \begin{cases} \min(a,b) & \text{if } a > 0 \text{ and } b > 0, \\ \max(a,b) & \text{if } a < 0 \text{ and } b < 0, \\ 0 & \text{else .} \end{cases} \tag{16}$$

It is left to define the time stepping method discretising the time derivative in (15), whereby we denote the right hand side of (15) as $L(u^n, i, j)$. This is done by the method of Heun, yielding the final update formula

$$\bar{u}_{i,j}^{n+1} = u_{i,j}^n + \tau L(u^n, i, j) \tag{17}$$

$$u_{i,j}^{n+1} = \frac{1}{2}u_{i,j}^n + \frac{1}{2}\bar{u}_{i,j}^{n+1} + \frac{\tau}{2}L\left(\bar{u}^{n+1}, i, j\right). \tag{18}$$

## 2.7 The Rouy-Tourin Scheme

Another first-order-accurate scheme has been proposed by Rouy and Tourin in [42]. The variant we employ in this chapter reads

$$u_{i,j}^{n+1} = u_{i,j}^n + \tau \left( \max\left( \frac{1}{h_x}\max\left(-D_-^x u_{i,j}^n, 0\right), \frac{1}{h_x}\max\left(D_+^x u_{i,j}^n, 0\right)\right)^2 \right.$$
$$\left. + \quad \max\left( \frac{1}{h_y}\max\left(-D_-^y u_{i,j}^n, 0\right), \frac{1}{h_y}\max\left(D_+^y u_{i,j}^n, 0\right)\right)^2 \right)^{1/2} \tag{19}$$

It displays a performance very similar to that of the first-order scheme OS-I, hence we refrain from showing experiments based on this scheme alone. However, we use it as a predictor step in the FCT scheme as it will be pointed out in the following subsection.

## 2.8 The FCT Scheme

The FCT scheme summarised below is by construction a new variant of a technique originally proposed by Boris and Book [5, 6, 7] in the context of fluid flow simulation. As shown in [8], the FCT scheme results in accurate and (largely) rotationally invariant discrete representations of continuous-scale morphological dilation/erosion.

The proposed FCT scheme relies on one-sided upwind differences as both schemes previously presented above. Instead of terms refining the gradient approximation as in (15), the idea behind the FCT scheme is to quantify the undesirable blurring effects introduced by upwinding and to negate the corresponding quantity in a corrector step by *stabilised inverse diffusion* (SID). Note that due to the stabilisation, the SID process is not ill-posed, compare [9].

In order to define the FCT scheme, let us give the abbreviations

$$\lceil^x u_{i,j}^n := \frac{\tau}{2h_x}\left|D_c^x u_{i,j}^n\right| + \frac{\tau}{2h_x} D_+^x u_{i,j}^n - \frac{\tau}{2h_x} D_-^x u_{i,j}^n, \tag{20}$$

$$\lceil^y u_{i,j}^n := \frac{\tau}{2h_y}\left|D_c^y u_{i,j}^n\right| + \frac{\tau}{2h_y} D_+^y u_{i,j}^n - \frac{\tau}{2h_y} D_-^y u_{i,j}^n. \tag{21}$$

Let us stress, that the quantities $\lceil^x u_{i,j}^n$ and $\lceil^y u_{i,j}^n$ just describe the upwinding incorporated in (19), see [8] for details. An important observation is, that the central differences above incorporate a second-order error which is non-diffusive, while the one-sided differences are discrete diffusive fluxes. Thus, a spatial discretisation relying on (20)–(21) such as the Rouy-Tourin scheme (19) introduces exactly these diffusive fluxes. The FCT procedure then inverts the corresponding numerical diffusion using the predicted data in the corrector step.

Using the method (19) as a *predictor* denoting the result pointwise as $u_{i,j}^{n+1,\mathrm{pred}}$, we are now concerned with the *corrector step*, which will finally read as

$$u_{ij}^{n+1} = u_{ij}^{n+1,\mathrm{pred}} + q_h^{n+1,\mathrm{pred}} - q_d^{n+1,\mathrm{pred}}. \tag{22}$$

The FCT scheme then consists of a subsequent application of (19) and (22).

We now define the terms occurring in (22). As indicated, it is essential for the FCT procedure to split the diffusive part from the second-order part. To this end, let us note that the discretisation of the dilation PDE using central differences only,

$$u_{i,j}^{n+1,\mathrm{pred}} = u_{ij}^n + \sqrt{\left(\frac{\tau}{2h_x}\left|D_c^x u_{i,j}^n\right|\right)^2 + \left(\frac{\tau}{2h_y}\left|D_c^y u_{i,j}^n\right|\right)^2}, \tag{23}$$

incorporates no numerical diffusion in the spatial discretisation part.

Let us now consider *predicted data* as arguments in the formulae of our numerical schemes. Then, adding zero by adding and subtracting the square root below (23) on the right hand side of (19), we can easily identify the higher-order part $q_h^{n+1,\mathrm{pred}}$ in (22) as

$$q_h^{n+1,\mathrm{pred}} := \sqrt{\left(\frac{\tau}{2h_x}\left|D_c^x u_{i,j}^{n+1,\mathrm{pred}}\right|\right)^2 + \left(\frac{\tau}{2h_y}\left|D_c^y u_{i,j}^{n+1,\mathrm{pred}}\right|\right)^2}. \tag{24}$$

For the lower-order term $q_d^{n+1,\mathrm{pred}}$ in (22) we have to take into account the *stabilisation* of the backward diffusive fluxes. This is done making use of

$$g_{i+1/2,j} := \mathrm{mm}\left(D_-^x u_{i,j}^{n+1,\mathrm{pred}}, \frac{\tau}{2h_x} D_+^x u_{i,j}^{n+1,\mathrm{pred}}, D_+^x u_{i+1,j}^{n+1,\mathrm{pred}}\right), \tag{25}$$

$$g_{i,j+1/2} := \mathrm{mm}\left(D_-^y u_{i,j}^{n+1,\mathrm{pred}}, \frac{\tau}{2h_y} D_+^y u_{i,j}^{n+1,\mathrm{pred}}, D_+^y u_{i,j+1}^{n+1,\mathrm{pred}}\right), \tag{26}$$

where $\mathrm{mm}(\cdot,\cdot,\cdot)$ is a straightforward extension of the minmod-function defined in (16) to three arguments. Employing then the stabilised fluxes within the formulae of (20)-(21), but applied at predicted data, we obtain

$$\delta^x u_{i,j}^{n+1,\mathrm{pred}} := \frac{\tau}{2h_x}\left|D_c^x u_{i,j}^{n+1,\mathrm{pred}}\right| + g_{i+1/2,j} - g_{i-1/2,j}, \tag{27}$$

$$\delta^y u_{i,j}^{n+1,\mathrm{pred}} := \frac{\tau}{2h_y}\left|D_c^y u_{i,j}^{n+1,\mathrm{pred}}\right| + g_{i,j+1/2} - g_{i,j-1/2}\,, \tag{28}$$

yielding the second new term in (22) as

$$q_d^{n+1,\mathrm{pred}} := \sqrt{\left(\delta^x u_{i,j}^{n+1,\mathrm{pred}}\right)^2 + \left(\delta^y u_{i,j}^{n+1,\mathrm{pred}}\right)^2}. \tag{29}$$

We now conclude our review of morphology in the scalar setting and proceed with the transfer to the matrix-valued case.

## 3 Ordering Based Morphology for Matrix Fields

Since dilation respectively erosion of flat morphology are defined via supremum and infimum, see (2) and (3), a suitable ordering on the set of image values is necessary. Dealing with symmetric matrices as image values, the so-called *Loewner ordering* is a natural choice. We introduce this partial ordering and other useful concepts from matrix analysis in the next subsection.

### *3.1 Matrix Analysis*

Of particular importance for us is the subset $\mathrm{Sym}(k)$ of symmetric $k \times k$-matrices with real entries. They form a vector space endowed with the scalar product

$$\langle A, B\rangle := \sqrt{\mathrm{trace}(A^\top B)}\,. \tag{30}$$

Note that at each point the matrix $F(x)$ of a field of symmetric matrices can be diagonalised and decomposed into its spectral components yielding

$$F(x) = V(x)^\top D(x) V(x) = \sum_{i=1}^{k} \lambda_i(x)\; v_i(x) v_i^\top(x)\,. \tag{31}$$

Here $V(x) \in \mathscr{O}(k)$ is a matrix field of orthogonal matrices $V(x)$ with column vectors $v_i(x)$, $i = 1,\ldots,k$, while $D(x)$ is a matrix field of diagonal matrices with entries $\lambda_i(x)$, $i = 1,\ldots,k$. In the sequel we will denote $k \times k$ - diagonal matrices with entries $\lambda_1,\ldots,\lambda_k \in \mathbb{R}$ from left to right simply by $\mathrm{diag}(\lambda_1,\ldots,\lambda_k)$, and $\mathscr{O}(k)$ stands for the matrix group of orthogonal $k \times k$-matrices.

We need to define functions $h$ of symmetric matrices. The most common way to do this is as follows [29]. Let $\mathrm{diag}(\alpha_1,\ldots,\alpha_k)$ denote a diagonal matrix with entries $\alpha_1,\ldots,\alpha_k$. We define for a symmetric matrix $A \in \mathrm{Sym}(k)$ with eigenvalue decom-

position $A = V\,\mathrm{diag}(\alpha_1,\ldots,\alpha_k)\,V^\top$ and orthogonal matrix $V \in \mathcal{O}(k)$ the matrix $h(A)$ by

$$h(A) := V\,\mathrm{diag}(h(\alpha_1),\ldots,h(\alpha_k))\,V^\top \tag{32}$$

provided the $\alpha_i$'s lie in the domain of definition of $h$. Note that the outcome of that operation is rotational invariant, $h(WAW^\top) = Wh(A)W^\top$, $W \in \mathcal{O}(k)$, and preserves symmetry, $h(A) \in \mathrm{Sym}(k)$.

For example, specifying $h$ as the absolute value function, $h(x) = |x|$, associates with a matrix $A$ its absolute value $|A|$. This $|A|$ denotes a positive semidefinite matrix and must not be confused with the norm or determinant of $A$.

The set of positive (semi-)definite matrices, denoted by $\mathrm{Sym}^+(k)$ ($\mathrm{Sym}^{++}(k)$, resp.), consists of all symmetric matrices $A$ with $\langle v,Av\rangle := v^\top Av > 0 \quad (\geq 0,\ \text{resp.},)$ for
$v \in \mathbb{R}^k \setminus \{0\}$.

The set $\mathrm{Sym}^+(k)$ forms a cone, that is, a set that is invariant under addition of matrices as well as multiplication with a positive scalar. This cone is used to define a partial ordering on $\mathrm{Sym}(k)$, the Loewner ordering:

$$A,B \in \mathrm{Sym}(k): \quad A \geq B :\Leftrightarrow A - B \in \mathrm{Sym}^+(k), \tag{33}$$

i.e. $A \geq B$ if and only if $A - B$ is positive semidefinite.

A subset $K$ of a cone $C$ is called *base* if every $y \in C, y \neq 0$ is uniquely representable as $y = r \cdot x$ with $x \in K$ and $r > 0$. For instance, the set of positive semidefinite matrices with trace 1 form a convex and compact base $K_1$ of $\mathrm{Sym}^+(k)$: $K_1 := \{M \in \mathrm{Sym}^+(k) : \mathrm{trace}(M) = 1\}$.

A point $x$ is an extreme point of a convex subset $S \subset V$ of a vector space $V$ if and only if $S \setminus \{x\}$ is still convex. The set of all extreme points of $S$ is denoted $\mathrm{ext}(S)$.

All the important information of a convex compact set is captured in its extreme points. The theorems of Minkowski and Krein-Milman state that each convex compact set $S$ in a finite dimensional vector space can be reconstructed as the set of all finite convex combinations of its extreme points [2, 28]:

$$\begin{aligned} S &= \mathrm{convexhull}(\mathrm{ext}(S)) \\ &= \left\{ \sum_{i=1}^{N} \lambda_i e_i \,|\, N \in \mathbb{N}, e_i \in \mathrm{ext}(S), \lambda_i \geq 0, \text{ for } i = 1,\ldots,N, \sum_{i=1}^{N} \lambda_i = 1 \right\}. \end{aligned}$$

It is known [2] that the matrices $vv^\top$ with unit vectors $v \in \mathbb{R}^k$, $\|v\| = 1$ are exactly the extreme points of a base $K_1$ of $\mathrm{Sym}^+(k)$. Because of this extremal property the matrices $vv^\top$ with $\|v\| = 1$ carry the complete information about the base of the Loewner ordering cone and hence the cone itself: $\mathrm{convexhull}(\{vv^\top : v \in \mathbb{R}^k, \|v\| = 1\})$ is a base for the Loewner ordering cone. Such extreme points of bases of translated Loewner cones will play a decisive role in the explicit calculation of the supremum/infimum of a finite number of symmetric matrices.

### *3.2 Maximal/Minimal Matrices in the Loewner Ordering*

The supremum of two symmetric matrices $A_1$ and $A_2$ is obtained easily. As it was pointed out in [10] the quantity

$$\sup(A_1,A_2) \;=\; \frac{1}{2}\,(A_1+A_2)+\frac{1}{2}\,|A_1-A_2|\,, \tag{34}$$

well known to hold for real numbers, indeed provides the supremum of the two matrices with respect to the Loewner ordering. The infimum of two matrices we obtain through

$$\inf(A_1,A_2) \;=\; \frac{1}{2}\,(A_1+A_2)-\frac{1}{2}\,|A_1-A_2|\,. \tag{35}$$

We will need in Section 5 a minmod-function for matrix fields, and to this end the supremum/infimum of three symmetric matrices $A_1,A_2,A_3$ has to be calculated. However, the iteration of (34) leads to the upper bounds

$$\begin{gathered} S_1 := \sup(A_1,\sup(A_2,A_3))\,, \quad S_2 := \sup(A_2,\sup(A_3,A_1))\,, \\ S_3 := \sup(A_3,\sup(A_1,A_2))\,, \end{gathered} \tag{36}$$

for the set $\{A_1,A_2,A_3\}$ that in general do not coincide:

$$S_1 \neq S_2 \neq S_3\,. \tag{37}$$

We construct an approximate supremum of $\{A_1,A_2,A_3\}$ in the following manner. Since each $S_i$ dominates $\{A_1,A_2,A_3\}$ so does their arithmetic mean:

$$S_m := \frac{1}{3}(S_1+S_2+S_3) \geq A_i, \quad i=1,2,3\,. \tag{38}$$

We can improve this upper bound $S_m$ by finding an optimal $\tau \geq 0$ such that

$$S_m - \tau I \geq A_i, \quad i=1,2,3, \tag{39}$$

holds, where $I$ denotes the identity matrix. If $\mu_{ij} \geq 0$, $j=1,\ldots,k$, are the eigenvalues of $S_m - A_i$ for $i=1,2,3$, this optimal $\tau$ is given by the minimum of all these eigenvalues

$$\tau_{opt} = \min_{\substack{i=1,\ldots,3 \\ j=1,\ldots,k}} (\mu_{ij}) \tag{40}$$

yielding a suitable supremum of three matrices

$$\sup_{opt}(A_1,A_2,A_3) = S_m - \tau_{opt} I\,. \tag{41}$$

The infimum of three matrices is calculated by

$$\inf_{opt}(A_1,A_2,A_3) \;=\; -\sup_{opt}(-A_1,-A_2,-A_3)\,. \tag{42}$$

It is clear that an extension of this approach to four or more matrices is not feasible. To extend the notions of dilation (2) and erosion (3) to matrix fields a different method to calculate the supremum/infimum of a larger number of matrices $A_i$ has to be employed. Such a technique has been developed and described in detail in [10, 12]. However, in order to keep this presentation as self-contained as possible we sketch this approach in the following.

Using the customary notation $a + rS := \{a + r \cdot s : s \in S\}$ for a point $a \in V$, a scalar $r \in \mathbb{R}$ and a subset $S \subset V$, we define the *penumbra* $P(M)$ of a matrix $M \in \mathrm{Sym}(k)$ as the set of matrices $N$ that are smaller than $M$ with respect to the Loewner ordering:

$$P(M) := \{N \in \mathrm{Sym}(k) : N \leq M\} = M - \mathrm{Sym}^{+}(k)\,. \tag{43}$$

The penumbra $P(M)$ is a reverted and translated version of the Loewner cone geometrically characterising all matrices that are smaller than the matrix $M$ marking its vertex.

Using this geometric description the problem of finding the maximum of a set of matrices $\{A_1,\ldots,A_m\}$ amounts to determining the minimal penumbra covering their penumbras $P(A_1),\ldots,P(A_m)$. Its vertex represents the matrix supremum $\overline{A} := \sup(A_1,\ldots,A_m)$ we are searching for and that dominates all $A_i$ w.r.t the Loewner ordering.

To this end we associate with each matrix $M \in \mathrm{Sym}(k)$ a *ball* in the subspace $\{A : \mathrm{trace}(A) = 0\}$ of all matrices with zero trace as a *completely descriptive set*. For the sake of simplicity we will assume that $\mathrm{trace}(M) \geq 0$. We determine the center and the radius of this enclosing ball: First, we note that the set $\big\{M - \mathrm{trace}(M) \times \mathrm{convexhull}\{vv^{\top} : v \in \mathbb{R}^k, \|v\| = 1\}\big\}$ is a base for $P(M)$ contained in the subspace $\{A : \mathrm{trace}(A) = 0\}$. The orthogonal projection of $M$ onto $\{A : \mathrm{trace}(A) = 0\}$ is given by

$$m := M - \frac{\mathrm{trace}(M)}{k} I\,. \tag{44}$$

Second, the extreme points of the base of $P(M)$ are lying on a sphere with center $m$ and radius

$$r := \|M - \mathrm{trace}(M) vv^{\top} - m\| = \mathrm{trace}(M)\sqrt{1 - \frac{1}{k}}\,. \tag{45}$$

If the center $m$ and radius $r$ of a sphere in $\{A \in \mathrm{Sym}(k) : \mathrm{trace}(A) = 0\}$ are given the vertex $M$ of the associated penumbra $P(M)$ is obtained by

$$M = m + \frac{r}{k} \frac{1}{\sqrt{1 - \frac{1}{k}}} I\,. \tag{46}$$

With this geometric interpretation in mind we may reformulate the task of finding a suitable maximal matrix $\overline{A}$ dominating the matrices $\{A_1,\dots,A_m\}$: The *smallest* sphere enclosing the spheres associated with $\{A_1,\dots,A_m\}$ determines the matrix $\overline{A}$ that dominates the $A_i$. It is minimal in the sense, that there is no smaller one w.r.t. the Loewner ordering which has this "covering property" of its penumbra. For each $i = 1,\dots,m$, we sample within the set of extreme points $\{A_i - \mathrm{trace}(A_i)vv^\top\}$ of the base of $P(A_i)$ by expressing $v$ in 3-D spherical coordinates, $v = (\sin\phi\cos\psi, \sin\phi\sin\psi, \cos\phi)$ with $\phi \in [0,2\pi), \psi \in [0,\pi)$. Vectorising these matrices, that is, writing the entries of each of these matrices in a $n^2$-dimensional vector provides us with points for which a smallest enclosing ball has to be found. This is a non-trivial problem of computational geometry and we tackle it by using a sophisticated algorithm implemented by B. Gaertner [19]. The algorithm returns the center and the radius of the smallest enclosing ball from which we obtain with the help of the relations (44) – (46) the corresponding maximal matrix $\overline{A}$. As in (42) we set

$$\inf(A_1,\dots,A_m) = -\sup(-A_1,\dots,-A_m)\,. \tag{47}$$

As a consequence the notion of dilation/erosion and with them many other morphological operations are available now for matrix fields.

In the next section we turn to the PDE-based approach to dilation and erosion in the matrix setting.

## 4 PDE-based morphology for Matrix Fields

In order to find the matrix-valued counterparts of the scalar morphological PDEs (4) it is necessary to establish a rudimentary calculus for fields of symmetric matrices. For a more extended calculus for matrix fields the reader is referred to [13, 14].

### *4.1 Matrix-valued PDEs for Dilation and Erosion*

The nonlinear PDEs (4) create a dilation and erosion process corresponding to a ball-shaped SE for grey value images. These equations contain the gradient operator $\nabla := (\partial_x, \partial_y, \partial_z)^\top$ with its partial derivatives and the Euclidean vector norm $\|(v_1,v_2,v_3)^\top\| := \sqrt{v_1^2+v_2^2+v_3^2}$. For both we have to find suitable analogs for matrices. To this end we have to clarify what a partial derivative, the absolute value and a square root of a symmetric matrix is. It is important *not* to consider a matrix norm as the extension of the vector norm in (4).

It is natural to define the equivalent $\overline{\partial}_\alpha$ of the partial derivative $\partial_\alpha$, spatial ($\alpha \in \{x,y,z\}$) or temporal ($\alpha = t$), of a scalar function for a matrix-valued func-

tion $U(x,y,z,t) = (u_{i,j}(x,y,z,t))_{i,j=1,\dots,k}$ by componentwise application of $\partial_\alpha$:

$$\overline{\partial}_\alpha U := (\partial_\alpha u_{i,j})_{i,j=1,\dots,k}\,. \tag{48}$$

Note that the subscripts indicate the matrix components and not the grey value of an image $u$ at pixel $(ih_x, jh_y)$. Due to the linearity of matrix multiplication and differentiation the application of $\overline{\partial}_\alpha$ preserves symmetry, $U \in \mathrm{Sym}(k) \Longrightarrow \overline{\partial}_\alpha U \in \mathrm{Sym}(k)$, and is rotational invariant:
$\overline{\partial}_\alpha(WUW^\top) = W(\overline{\partial}_\alpha U)W^\top$ holds for any constant orthogonal matrix $W$.

With definition (32) the notion of a function of a symmetric matrix is already at our disposal. Hence, specifying the functions $h(x) = |x|^2$ and $h(x) = \sqrt{x}$ we have equipped the matrix-valued expression $\sqrt{|\overline{\partial}_x U|^2 + |\overline{\partial}_y U|^2 + |\overline{\partial}_z U|^2}$ with meaning. The latter is in fact a positive definite matrix if $U$ is a non-constant matrix field, and it can be seen as a direct analog of the Euclidean norm of a vector.

Now we are in the position to establish the matrix-valued counterpart of (4):

$$\overline{\partial}_t U = \pm\sqrt{|\overline{\partial}_x U|^2 + |\overline{\partial}_y U|^2 + |\overline{\partial}_z U|^2}\,, \tag{49}$$

where "+" governs the dilation-like, and "–" rules the erosion-like evolution process.

## *4.2 Numerical Schemes for Matrix-valued PDEs of Dilation and Erosion*

In order to solve the matrix-valued PDEs of dilation/erosion we transfer the numerical schemes OS-I, OS-II and FCT presented in the previous section to the setting of matrix fields. Linear combinations and elementary functions such as the square, square-root or absolute value function for matrix fields are now at our disposal. Hence it is straightforward to define one sided differences in $x$-direction for 3-D matrix fields of $k \times k$-matrices:

$$D_+^x U^n(i,j,k) := U^n((i+1)h_x, jh_y, kh_z) - U^n(ih_x, jh_y, kh_z) \in \mathrm{Sym}(k)\,, \tag{50}$$

$$D_-^x U^n(i,j,k) := U^n(ih_x, jh_y, kh_z) - U^n((i-1)h_x, jh_y, kh_z) \in \mathrm{Sym}(k)\,. \tag{51}$$

In order to avoid confusion with the subscript notation for matrix components we used the notation $U(i,j,k)$ to indicate the (matrix-) value of the matrix field evaluated at the voxel centred at $(ih_x, jh_y, kh_z) \in \mathbb{R}^3$. The central difference operator in $x$-direction is interpreted as

$$D_c^x U^n(i,j,k) := U^n((i+1)h_x, jh_x, kh_x) - U^n((i-1)h_x, jh_y, kh_z) \in \mathrm{Sym}(k)\,. \tag{52}$$

The $y-$ and $z-$directions are treated accordingly. The notion of supremum and infimum of two matrices (as needed in a matrix variant of OS-I) has been provided

by (34) and (35). However, care has to be taken for functions that are defined piecewise such as the minmod functions for two or three arguments. We generalise the minmod functions to the matrix setting by invoking the Loewner ordering

$$\mathrm{mm}(A_1,A_2) := \begin{cases} \inf(A_1,A_2) & \text{for } A_1 > 0 \text{ and } A_2 > 0\,, \\ \sup(A_1,A_2) & \text{for } A_1 < 0 \text{ and } A_2 < 0\,, \\ 0 & \text{else}\,, \end{cases} \tag{53}$$

in the case of two matrices, while for three matrices we set

$$\mathrm{mm}(A_1,A_2,A_3) := \begin{cases} \inf_{opt}(A_1,A_2,A_3) & \text{for } A_i > 0,\ i = 1,2,3\,, \\ \sup_{opt}(A_1,A_2,A_3) & \text{for } A_1 < 0,\ i = 1,2,3\,, \\ 0 & \text{else}\,, \end{cases} \tag{54}$$

with $\sup_{opt}$ and $\inf_{opt}$ given in (41) and (42).

Having these generalisations at our disposal the numerical schemes OS-I, OS-II and finally the FCT scheme are available now in the setting of matrix fields.

The case differentiation necessary for shock filtering (11) is handled differently and utilises the *trace*-function: The sign of $\mathrm{tr}(\Delta_B U)$ of the matrix $\Delta_B U$ provides the switching mechanism for shock filtering in the matrix field setting.

## 5 Experimental Comparison of the Numerical Schemes

The hyperbolic morphological PDEs of dilation/erosion are numerically tackled with three specialised schemes: The first- and second-order schemes of Osher and Sethian (OS-I, OS-II) and the FCT scheme of Breuß and Weickert. We compare the results of basic morphological operations on images for ordering-based morphology and PDE-based methods obtained with the numerical schemes OS-I, OS-II and FCT. We restrict ourselves to the self-dual top hat (SDTH), the Beucher gradient, and the matrix-valued variant of a morphological Laplacian and shock filter. Note that the dilation and erosion are performed with respect to a *ball-shaped* structuring element. In extension of the experiments in [8] we first turn our attention to the scalar case in the next subsection.

### *5.1 Scalar Valued Data*

In the scalar case we apply the morphological self-dual top hat and the Beucher gradient to the grey value test image of size $256 \times 256$ depicted in Figure 1.

We juxtapose the results of these operations when the underlying dilation/erosion operations stem from the classical ordering-based definitions, or from the PDE-based approach with resolved with the three numerical schemes OS-I, OS-II and FCT discussed above.

In the case of the self-dual top hat we performed 40 steps with time step size 0.1 resulting in an evolution time of 4 which corresponds to a ball with 4 pixel radius as SE. In Figure 2 we clearly see the superior performance of FCT when compared to OS-I and OS-II: The FCT-based SDTH selects the smaller details of the image almost as good as the ordering-based SDTH. Both OS-I and OS-II produce blurring artefacts due to the inherent numerical dissipation.

Let us now turn to the Beucher gradient $\rho_B$ as our last example in the scalar setting. We performed 20 steps with time step size 0.1 resulting in an evolution time of 2 which corresponds to ball with 2 pixel radius as SE. The results displayed in Figure 3 confirm the advantage of the FCT scheme over the standard schemes. The edges are enhanced as in the ordering-based Beucher gradient while the schemes OS-I and OS-II suffer from blurring.

## *5.2 Matrix-valued Data*

In order to assess the quality of our numerical approach, we first show two experiments with matrix-valued data in two dimensions. Afterwards, a corresponding 3-D experiment is given.

In our first two numerical experiments for matrix data we use an artificial $20 \times 20$-field as well as a $128 \times 128$ slice of 3-D positive definite matrices originating from a 3-D DT-MRI data set of a human head, see figures 4, 6, 7, and 8.

The data are represented as ellipsoids via the level sets of the quadratic form $\{x^\top A^{-2} x = const. : x \in \mathbb{R}^3\}$ associated with a matrix $A \in \mathrm{Sym}^+(3)$. Using $A^{-2}$ instead of the matrix $A$ the length of the semi-axes of the ellipsoid correspond directly with the three eigenvalues of the positive definite matrix.

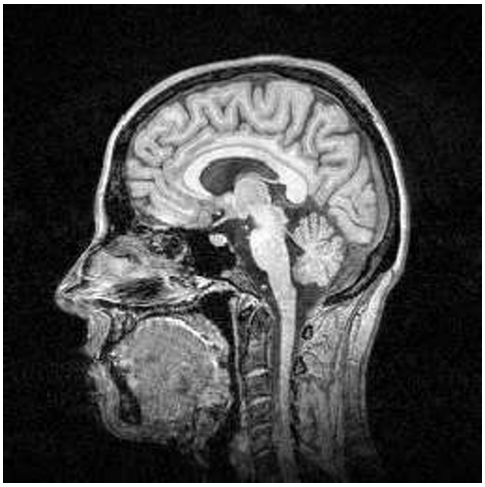


**Fig. 1** The grey value test image of size $256 \times 256$ used for the experiments in the scalar setting.

The artificial data constitute a circular structure where the ellipsoids in the center are elongated, while those outside the circle are simple balls.

In Figure 5, the results with artificial test data are displayed. It is clearly visible that the ordering-based approach leads to sharp edges while the elongated tensors change their shape towards a round appearance. Note that the sharp edges stem from a discrete approximation of the structuring element in the pixel grid. The PDE-based methods better preserve the initial shape of the tensors, but the Osher-Sethian approaches introduce numerical blurring of the edges. The FCT scheme performs better with this respect. The small tensors near the edges arise by geometrical effects as the structuring element is a perfect ball given by the Euclidean norm.

Exactly the analogous numerical behaviour is observable in our experiment with real-world data shown in Figure 6. The same assertion applies with respect to Figure

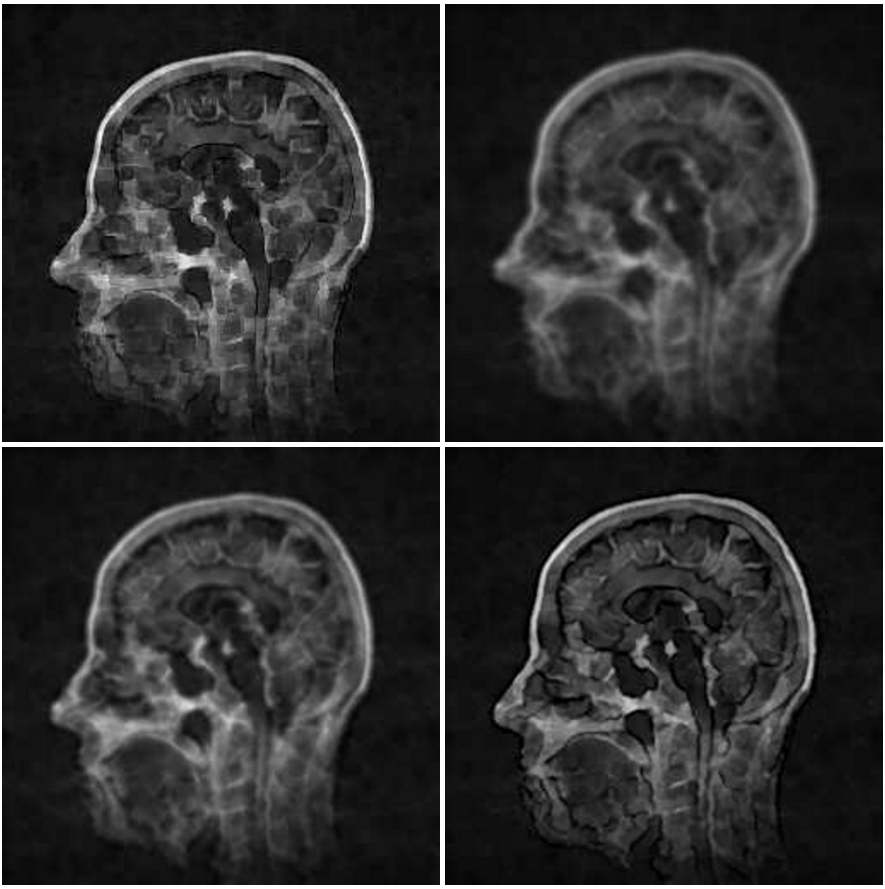


**Fig. 2 Top left**: Ordering-based SDTH with ball-shaped SE of size 4. **Top right**: PDE-based SDTH via OS-I with time step size 0.1 and total evolution time 4. **Bottom left**: The same with OS-II. **Bottom right**: The same with FCT scheme.

7. However, in the latter experiment, it becomes even clearer than in Figure 5 that the ordering-based approach and the PDE-based approach in the matrix-valued setting are not equivalent and thus cannot be expected to yield the same results.

In our last experiment in two dimensions we investigate the shock filter. For this morphological operation the differences between the numerical schemes are hardly visible. We included this example in order to show that it depends on the underlying process of interest if it pays off to use a high-resolution-type scheme.

In our last experiment, we show that the schemes for matrix-valued morphology can also be applied in three dimensions. Figure 9 shows that the qualitative behaviour of the FCT scheme worked out before carries over to the 3-D setting.

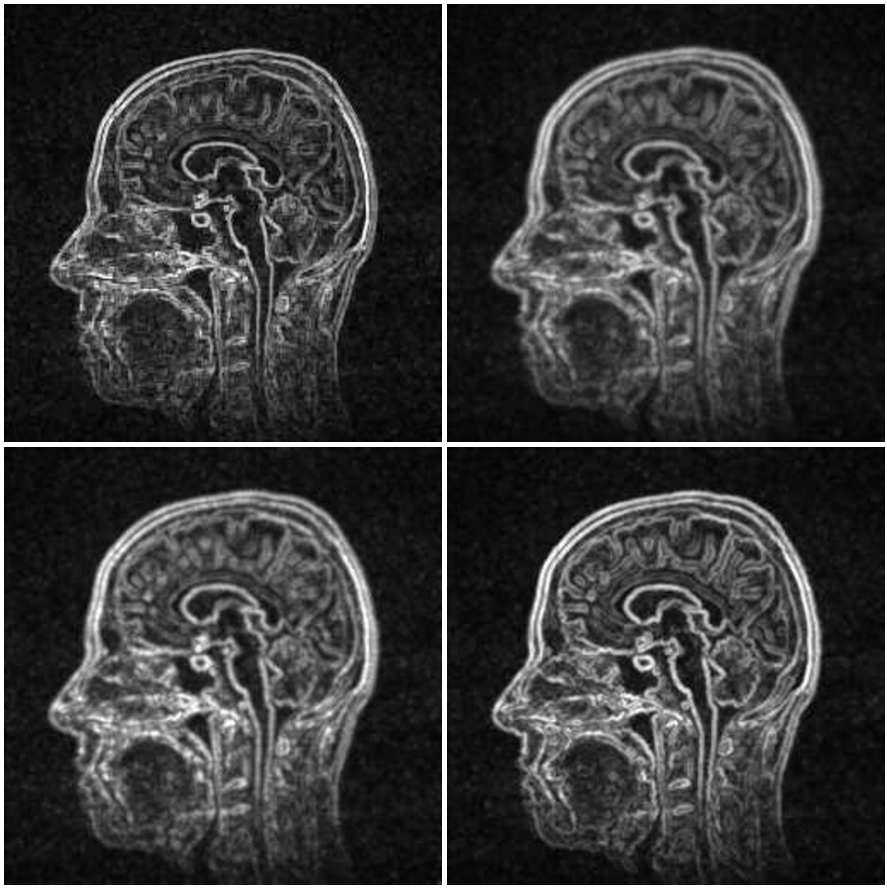


**Fig. 3** **Top left**: Ordering-based Beucher gradient with ball-shaped SE of size 2. **Top right**: PDE-based Beucher gradient via OS-I with time step size 0.1 and total evolution time 2. **Bottom left**: The same with OS-II. **Bottom right**: The same with FCT scheme.

## 6 Conclusion

In this work we were concerned with numerical solution schemes for the morphological PDEs for dilation and erosion in the setting of matrix fields. It has been demonstrated that, firstly, it is possible to extend even sophisticated high-resolution schemes for morphological PDEs, such as the FCT, in a rather straightforward manner once the suitable matrix-algebraic foundations a properly prepared. Secondly, we showed in experiments that the use of a computationally more expensive FCT indeed pays off in the matrix setting: It preserves or enhances edges and contours in matrix fields transformed with top-hats and morphological derivatives better than the first- or even second-order schemes of Osher and Sethian. The findings reported on in this work confirm that the elaborate numerical machinery for PDE-based scalar morphology is now at our disposal for matrix-valued morphology as well.

## References

1. L. Alvarez and L. Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, 31:590–605, 1994.
2. A. Barvinok. *A Course in Convexity*, volume 54 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, 2002.
3. P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66:259–267, 1994.
4. J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, August 1991.
5. J. P. Boris and D. L. Book. Flux corrected transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11(1):38–69, 1973.

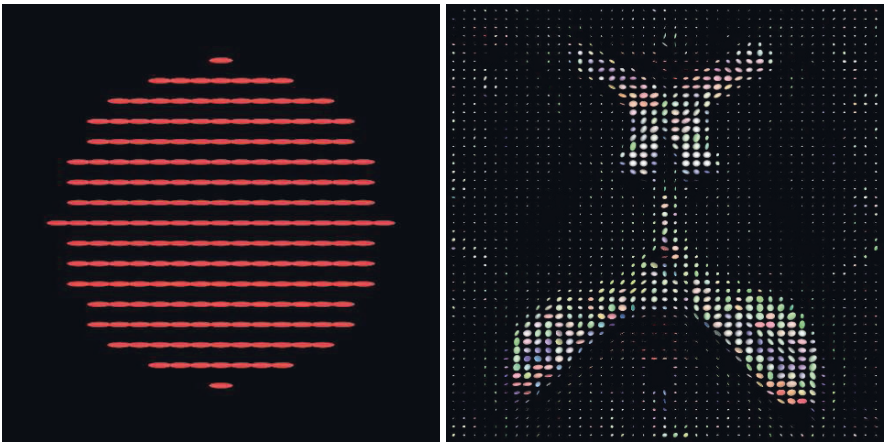


**Fig. 4** **Left**: Artificial matrix field. **Right**: 2-D-slice of a real 3D DT-MRI data set.

6. J. P. Boris and D. L. Book. Flux corrected transport. III. Minimal error FCT algorithms. *Journal of Computational Physics*, 20:397–431, 1976.
7. J. P. Boris, D. L. Book, and K. Hain. Flux corrected transport. II. Generalizations of the method. *Journal of Computational Physics*, 18:248–283, 1975.
8. M. Breuß and J. Weickert. A shock-capturing algorithm for the differential equations of dilation and erosion. *Journal of Mathematical Imaging and Vision*, 25(2):187–201, September 2006.
9. M. Breuß and M. Welk. Staircasing in semidiscrete stabilised inverse diffusion algorithms. *Journal of Computational and Applied Mathematics*, 206(1):520–533, 2007.
10. B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk. Morphology for tensor data: Ordering versus PDE-based approach. *Image and Vision Computing*, 25(4):496–511, 2007.
11. B. Burgeth, A. Bruhn, N. Papenberg, M. Welk, and J. Weickert. Mathematical morphology for matrix fields induced by the Loewner ordering in higher dimensions. *Signal Processing*, 87(2):277–290, 2007.

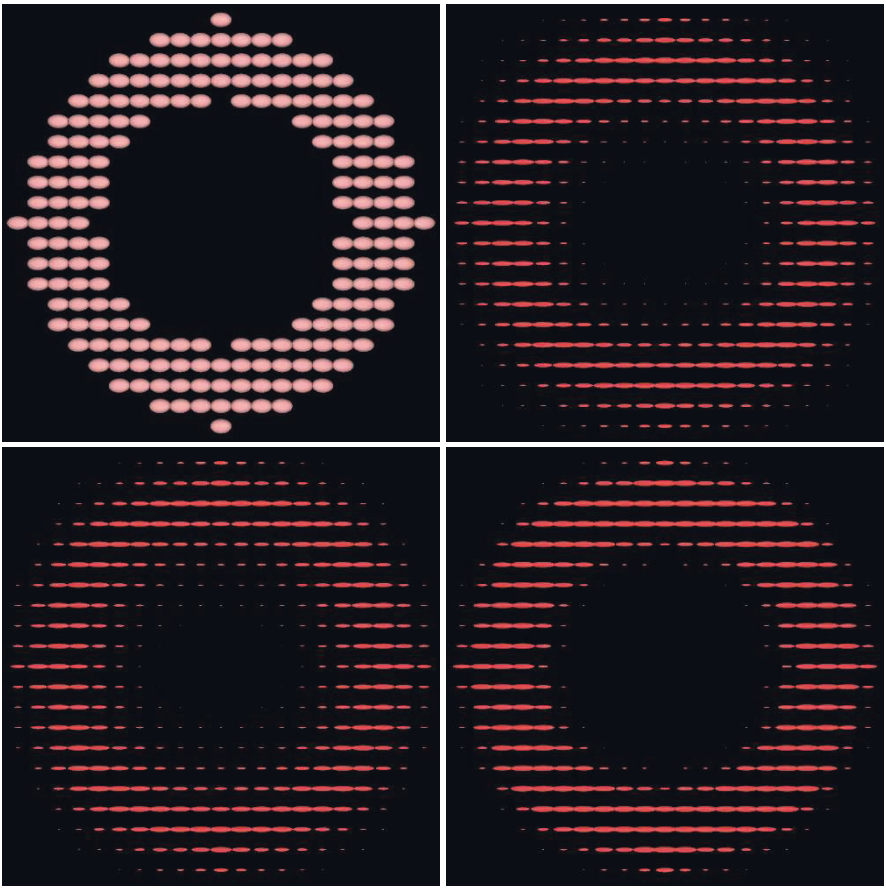


**Fig. 5** **Top left**: Ordering-based Beucher gradient with ball-shaped SE of size 2. **Top right**: PDE-based Beucher gradient via OS-I with time step size 0.1 and total evolution time 2. **Bottom left**: The same with OS-II. **Bottom right**: The same with FCT scheme.

12. B. Burgeth, A. Bruhn, N. Papenberg, M. Welk, and J. Weickert. Mathematical morphology for tensor data induced by the Loewner ordering in higher dimensions. *Signal Processing*, 87(2):277–290, February 2007.
13. B. Burgeth, S. Didas, L. Florack, and J. Weickert. A generic approach to diffusion filtering of matrix-fields. *Computing*, 81:179–197, 2007.
14. B. Burgeth, S. Didas, L. Florack, and J. Weickert. A generic approach to the filtering of matrix fields with singular PDEs. In F. Sgallari, F. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 556–567. Springer, Berlin, 2007.
15. B. Burgeth, N. Papenberg, A. Bruhn, M. Welk, C. Feddern, and J. Weickert. Morphology for higher-dimensional tensor data via Loewner ordering. In C. Ronse, L. Najman, and E. Decencière, editors, *Mathematical Morphology: 40 Years On*, volume 30 of *Computational Imaging and Vision*, pages 407–418. Springer, Dordrecht, 2005.
16. L. C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, USA1998.

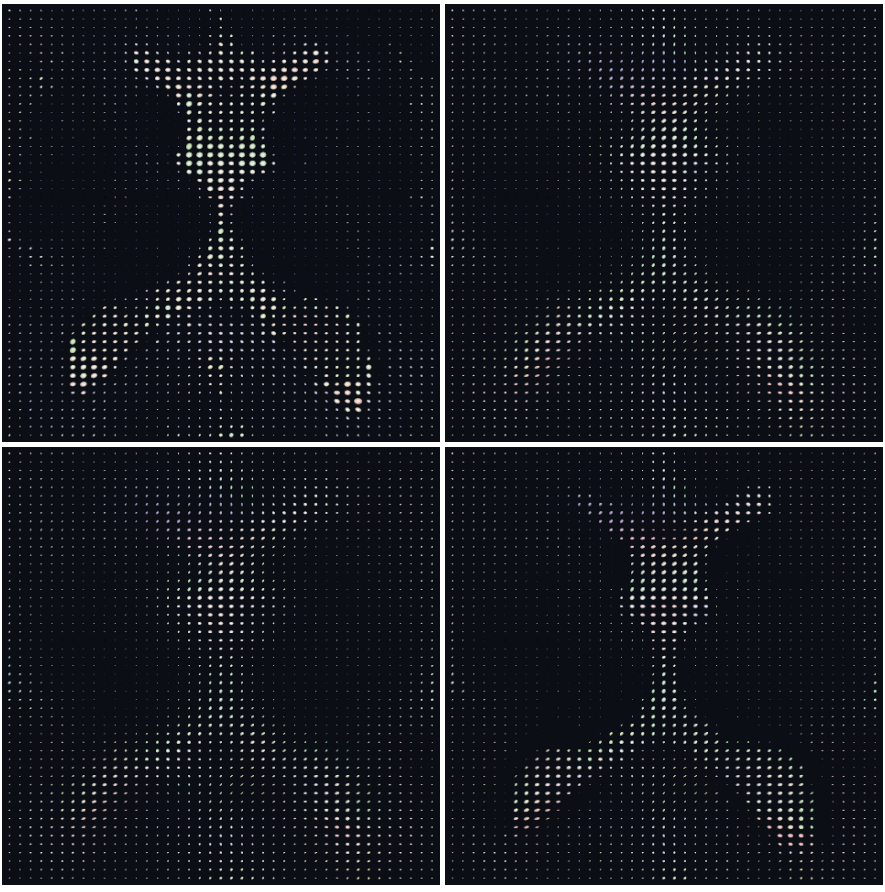


**Fig. 6 Top left**: Ordering-based SDTH with ball-shaped SE of size 3. **Top right**: PDE-based SDTH via OS-I with time step size 0.1 and total evolution time 3. **Bottom left**: The same with OS-II. **Bottom right**: The same with FCT scheme.

17. S. J. Farlow. *Partial Differential Equations for Scientists and Engineers*. Dover, New York, 1993.
18. W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proceedings of the ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.
19. B. Gärtner. http://www.inf.ethz.ch/personal/gaertner. 2005.
20. G. Gilboa, N. A. Sochen, and Y. Y. Zeevi. Regularized shock filters and complex diffusion. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision – ECCV 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 399–413. Springer, Berlin, 2002.
21. J. Goutsias, H. J. A. M. Heijmans, and K. Sivakumar. Morphological operators for image sequences. *Computer Vision and Image Understanding*, 62:326–346, 1995.
22. J. Goutsias, L. Vincent, and D. S. Bloomberg, editors. *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18 of *Computational Imaging and Vision*. Kluwer, Dordrecht, 2000.

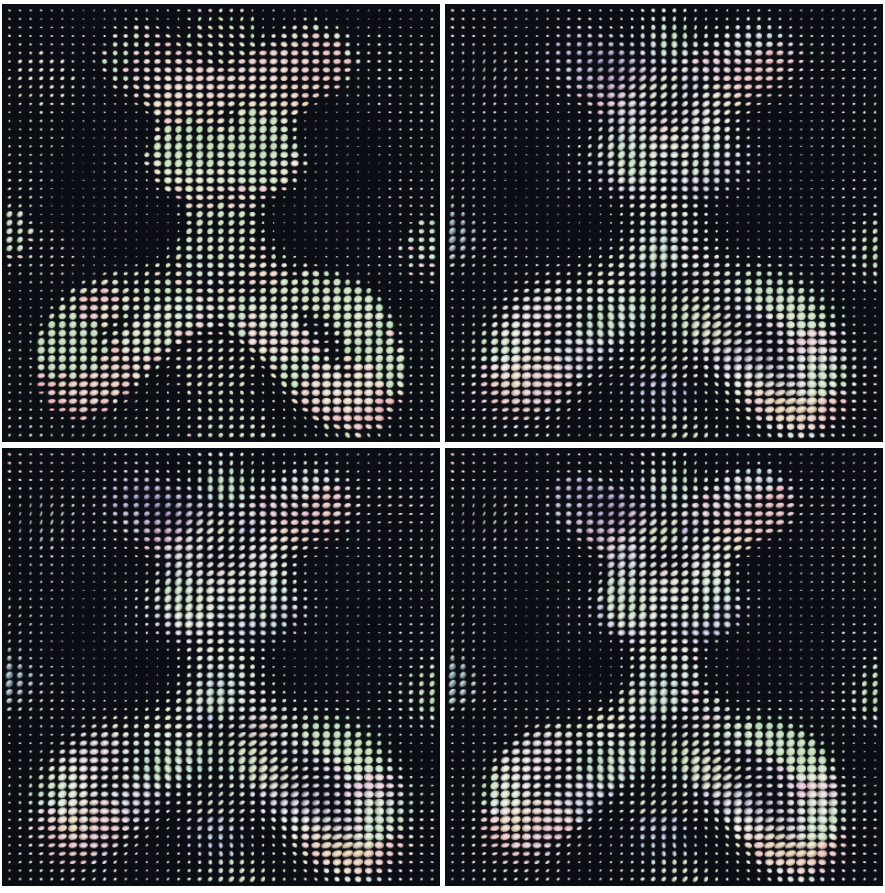


**Fig. 7** **Top left**: Ordering-based Beucher gradient with ball-shaped SE of size 3. **Top right**: PDE-based Beucher gradient via OS-I with time step size 0.1 and total evolution time 3. **Bottom left**: The same with OS-II. **Bottom right**: The same with FCT scheme.

23. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer, Dordrecht, 1995.
24. F. Guichard and J.-M. Morel. A note on two classical enhancement filters and their associated PDE's. *International Journal of Computer Vision*, 52(2/3):153–160, 2003.
25. C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–152, Manchester, UK, August 1988.
26. H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, Boston, 1994.
27. H. J. A. M. Heijmans and J. B. T. M. Roerdink, editors. *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 12 of *Computational Imaging and Vision*. Kluwer, Dordrecht, 1998.
28. J.-B. Hiriart-Urruty and C. Lemarechal. *Fundamentals of Convex Analysis*. Springer, Heidelberg, 2001.
29. R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1990.

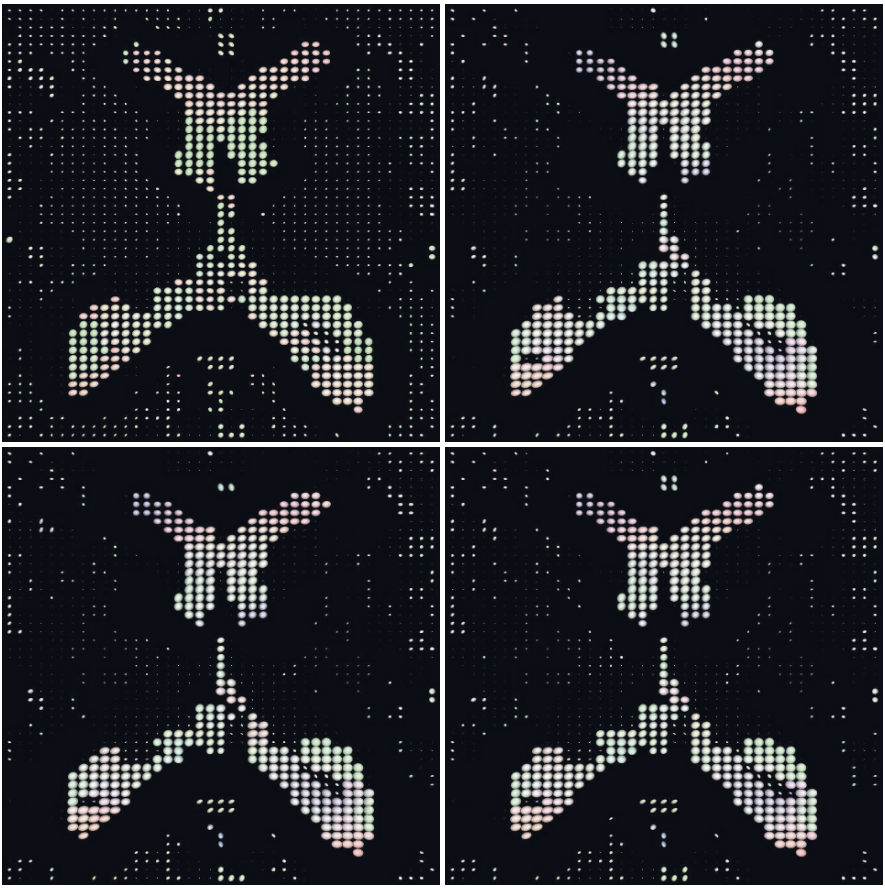


**Fig. 8** **Top left**: Ordering-based shock filter with ball-shaped SE of size 3. **Top right**: PDE-based shock filter via OS-I with time step size 0.1 and total evolution time 3. **Bottom left**: The same with OS-II. **Bottom right**: The same with FCT scheme.

30. H. P. Kramer and J. B. Bruckner. Iterations of a non-linear transformation for enhancement of digital images. *Pattern Recognition*, 7:53–58, 1975.
31. R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, UK, 2002.
32. G. Louverdis, M. I. Vardavoulia, I. Andreadis, and P. Tsalides. A new approach to morphological color image processing. *Pattern Recognition*, 35:1733–1741, 2002.
33. G. Matheron. *Eléments pour une théorie des milieux poreux*. Masson, Paris, 1967.
34. G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.
35. S. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, New York, 2002.
36. S. Osher and L. Rudin. Shocks and other nonlinear filtering applied to image processing. In A. G. Tescher, editor, *Applications of Digital Image Processing XIV*, volume 1567 of *Proceedings of SPIE*, pages 414–431. SPIE Press, Bellingham, 1991.
37. S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27:919–940, 1990.
38. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
39. C. Pierpaoli, P. Jezzard, P. J. Basser, A. Barnett, and G. Di Chiro. Diffusion tensor MR imaging of the human brain. *Radiology*, 201(3):637–648, December 1996.
40. A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.
41. L. Remaki and M. Cheriet. Numerical schemes of shock filter models for image enhancement and restoration. *Journal of Mathematical Imaging and Vision*, 18(2):153–160, March 2003.
42. E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29:867–884, 1992.
43. G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, Cambridge, UK, 2001.
44. G. Sapiro, R. Kimmel, D. Shaked, B. B. Kimia, and A. M. Bruckstein. Implementing continuous-scale morphology via curve evolution. *Pattern Recognition*, 26:1363–1372, 1993.

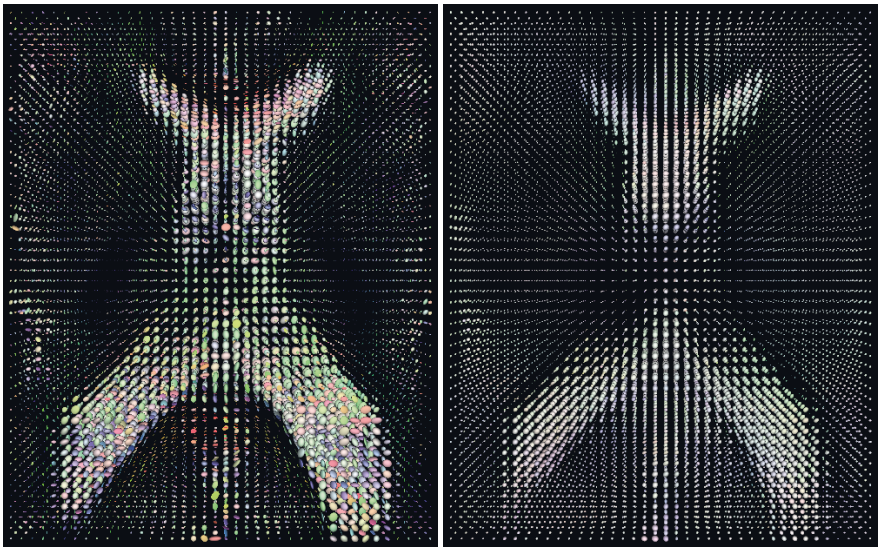


**Fig. 9** **Left**: Real-world 3-D matrix-valued of size $40 \times 50 \times 5$ voxels. **Right**: PDE-based self-dual top hat via FCT scheme with time step size 0.1 and total evolution time 3.

45. J. G. M. Schavemaker, M. J. T. Reinders, and R. van den Boomgaard. Image sharpening by morphological filtering. In *Proceedings of the 1997 IEEE Workshop on Nonlinear Signal and Image Processing*, Mackinac Island, MI, USA, September 1997. www.ecn.purdue.edu/NSIP/.
46. J. Serra. *Echantillonnage et estimation des phénomènes de transition minier*. PhD thesis, University of Nancy, France, 1967.
47. J. Serra. *Image Analysis and Mathematical Morphology*, volume 1. Academic Press, London, 1982.
48. J. Serra. *Image Analysis and Mathematical Morphology*, volume 2. Academic Press, London, 1988.
49. J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, UK, second edition, 1999. Paperback edition.
50. K. Siddiqi, B. B. Kimia, and C.-W. Shu. Geometric shock-capturing ENO schemes for subpixel interpolation, computation and curve evolution. *Graphical Models and Image Processing*, 59:278–301, 1997.
51. P. Soille. *Morphological Image Analysis*. Springer, Berlin, second edition, 2003.
52. P. Stoll, C.-W. Shu, and B. B. Kimia. Shock-capturing numerical methods for viscosity solutions of certain PDEs in computer vision: The Godunov, Osher–Sethian and ENO schemes. Technical Report LEMS-132, Division of Engineering, Brown University, Providence, RI, 1994.
53. H. Talbot and R. Beare, editors. *Proceedings of the Sixth International Symposium on Mathematical Morphology and its Applications*. CSIRO Publishing, Sydney, Australia, April 2002. http://www.cmis.csiro.au/ismm2002/proceedings/.
54. R. van den Boomgaard. Numerical solution schemes for continuous-scale morphology. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 199–210. Springer, Berlin, 1999.
55. L. J. van Vliet, I. T. Young, and A. L. D. Beckers. A nonlinear Laplace operator as edge detector in noisy images. *Computer Vision, Graphics and Image Processing*, 45(2):167–195, 1989.
56. J. Weickert. Coherence-enhancing shock filters. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 1–8. Springer, Berlin, 2003.

# Part II
# Tensors in Image Processing

# Spherical Tensor Calculus for Local Adaptive Filtering

Marco Reisert and Hans Burkhardt

**Abstract** In 3D image processing tensors play an important role. While rank-1 and rank-2 tensors are well understood and commonly used, higher rank tensors are rare. This is probably due to their cumbersome rotation behavior which prevents a computationally efficient use. In this chapter we want to introduce the notion of a spherical tensor which is based on the irreducible representations of the 3D rotation group. In fact, any ordinary cartesian tensor can be decomposed into a sum of spherical tensors, while each spherical tensor has a quite simple rotation behavior. We introduce so called tensorial harmonics that provide an orthogonal basis for spherical tensor fields of any rank. It is just a generalization of the well known spherical harmonics. Additionally we propose a spherical derivative which connects spherical tensor fields of different degree by differentiation. Based on the proposed theory we present two applications. We propose an efficient algorithm for dense tensor voting in 3D, which makes use of tensorial harmonics decomposition of the tensor-valued voting field. In this way it is possible to perform tensor voting by linear-combinations of convolutions in an efficient way. Secondly, we propose an anisotropic smoothing filter that uses a local shape and orientation adaptive filter kernel which can be computed efficiently by the use spherical derivatives.

## 1 Introduction

In 3D image processing tensors play an important role. While rank-1 and rank-2 tensors are well understood and commonly used, higher rank tensors are rare. This

Marco Reisert
Albert-Ludwigs University, Freiburg, Germany,
e-mail: reisert@informatik.uni-freiburg.de

Hans Burkhardt
Albert-Ludwigs University, Freiburg, Germany,
e-mail: burkhardt@informatik.uni-freiburg.de

is probably due to their cumbersome rotation behavior which prevents a computationally efficient use. In this chapter we want to introduce the notion of a spherical tensor which is based on the irreducible representations of the 3D rotation group. In fact, any ordinary cartesian tensor can be decomposed into a sum of spherical tensors, while each spherical tensor has a quite simple rotation behavior. We introduce so called tensorial harmonics that provide an orthogonal basis for spherical tensor fields of any rank. It is just a generalization of the well known spherical harmonics. Additionally we propose a spherical derivative which connects spherical tensor fields of different degree by differentiation.

We will use the proposed theory for local adaptive filtering. By local adaptive filtering we mean that during the filtering process the filter kernels may change their shape and orientation depending on local quantities which were derived from the image. Typically there are two ways to do this which are in a certain sense dual to each other. Consider the classical linear filtering process. There are two interpretations, on the one hand the convolution: each pixel (impulse) in the image is replaced by a predefined filter kernel (impulse response) while the filter kernel itself is weighted by the intensity of the observed pixel. The contribution from all pixels are combined by summation. This is the interpretation we know from signal processing, where the filter kernel is known as the impulse response. For Gaussian filter kernels the physical interpretation of this is simple isotropic diffusion. The second interpretation is to compute a kind of correlation or blurring of the image: at each pixel we compute an inner product of the filter kernel with its local neighborhood, i.e. a kind of correlation. If the filter kernel is positive, then it may be interpreted as an average of the surrounding pixels while the filter kernel determines the shape and size of local window in which the average is taken. In the linear case both interpretations are identical up to a point reflection of the filter kernel. But, if the filter kernel is spatially dependent (or locally adaptive) both approaches are not identical anymore. Let us formalize this. Let $m(\mathbf{r})$ be the intensity of an image at position $\mathbf{r}$ and $V^{\mathbf{n}}(\mathbf{r})$ a filter kernel at position $\mathbf{r}$, where the superscript $\mathbf{n}$ is a parameter that determines the orientation and shape of the kernel. Now suppose that we have also given a parameter field $\mathbf{n}(\mathbf{r})$, i.e. the appearance of the kernel is spatially dependend. Then, the 'convolution' integral looks as

$$\mathbf{U}_{\text{conv}}(\mathbf{r}) = \int_{\mathbb{R}^3} V^{\mathbf{n}(\mathbf{r}')}(\mathbf{r}-\mathbf{r}')\, m(\mathbf{r}')\, d\mathbf{r}'.$$

It formulates the above described intuition. We attach to each position $\mathbf{r}' \in \mathbb{R}^3$ the filter kernel while the filter kernel depends on the kernel parameter $\mathbf{n}$ at position $\mathbf{r}'$. Then, the filter kernel is weighted by the observed image intensity $m(\mathbf{r}')$ and the contributions from all positions $\mathbf{r}'$ are superimposed additively by the integral. On the other hand we can write down the 'correlation' integral as

$$\mathbf{U}_{\text{corr}}(\mathbf{r}) = \int_{\mathbb{R}^3} V^{\mathbf{n}(\mathbf{r})}(\mathbf{r}'-\mathbf{r})\, m(\mathbf{r}')\, d\mathbf{r}',$$

which again covers the above presented picture. The value of the result at position $\mathbf{r}$ is just the standard inner product of the image with the filter kernel modified by the parameter $\mathbf{n}(\mathbf{r})$.

The 'convolution'-approach is related to the so called tensor voting framework (TV) [6, 8]. In TV the filter kernel is denoted as the voting function and is typically tensor-valued. For example, rank-2 tensors are use to enhance feature images for fiber detection. In TV the intensity image $m(\mathbf{r})$ is interpreted as a probability for the presence of a fiber, while the kernel parameter $\mathbf{n}(\mathbf{r})$ is the orientation of the fiber at the specific position. On the other hand, the 'correlation'-approach is related to anisotropic smoothing filters, which are typically used to denoise images while preserving edges and discontinuities. Here the filter kernel is for example a squeezed Gaussian, tablet-like function, which is during the filter process oriented along the intensity gradients. In this way the smoothing is not performed across edges and, hence, the discontinuities are preserved.

In this chapter we propose how to use spherical tensor calculus to expand the filter kernel in an advantageous manner, such that the orientational steering of the filter kernel can be performed efficiently. For scalar filter kernels this expansion is the well-known Spherical Harmonics expansion. To generalize this idea to tensor-valued images we propose the so called tensorial harmonics. In this way arbitrary filter kernels can be expanded in tensorial harmonics and the computation of the filter integral turns out to be a sum of convolutions. Although the convolutions can be computed efficiently by the Fast Fourier Transform, the convolution is still the bottleneck in the computation for very large volumes. Another problem of this approach is the severe memory consumption, because one has to store the tensorial harmonic decomposition in a quite wasteful manner to allow an efficient computation. Hence, we introduce so called spherical derivatives that allow to compute the convolutions with special type of kernels efficiently.

## *1.1 Related Work*

The tensor voting (TV) framework was originally proposed by Medioni et al. [6] and has found several applications in low-level vision in 2D and 3D. For example, it is used for perceptual grouping and extraction of line, curves and surfaces [8]. The key idea of TV is to make unreliable measurements more robust by incorporating neighborhood information in a consistent and coherent manner. To compute the TV-integral in a reasonable time the initial measurements in TV are typically sparse. Recently, Franken et al. [3] proposed an efficient way to compute a dense Tensor Voting in 2D. The idea makes use of a steerable expansion of the voting field. Steerable filters are an efficient architecture to synthesize filters for arbitrary angles from linear combinations of basis filters [4]. Perona generalized this concept in [9] and introduced a methodology to decompose a given filter kernel optimally in a set of steerable basis filters. The idea of Franken et al. [3] is to use the steerable decomposition of the voting field to compute the voting process by convolutions in

an efficient way. Complex calculus and 2D harmonic analysis are the major mathematical tools that make this approach possible.

Anisotropic filtering is a low-level image processing technique that is used to denoise and enhance images. The applied algorithms can be separated into iterative and non-iterative methods. Iterative algorithms [15] are based on solutions of partial differential equations. The motivation of the idea has its roots in the physical modelling of an anisotropic diffusion process. The equations are tailored such that particles tend to diffuse along edges rather than across edges. Consequently, the discontinuities of the images are preserved while the isotropic regions are smoothed. The second class of algorithms [19, 5] treats the problem as a local adaptive blurring process. Depending on a local orientation analysis the blurring kernels are steered for each pixels such that the blurring is not performed across edges. In [5] a technique for fast anisotropic filtering in 2D is proposed; unfortunately the idea is not extendable to 3D. In [10] local adaptive filters were implemented with the help of complex derivatives; this work may be seen as a generalization of this ideas to 3D. In [11] complex derivatives were used to compute nonlinear rotation-equivariant filters in 2D efficiently.

## 2 Spherical Tensor Analysis

We will assume that the reader is familiar with the basic notions of the harmonic analysis of $SO(3)$. For introductory reading we recommend mainly literature [2, 13, 14, 17, 18] concerning the quantum theory of the angular momentum ang chemical qunatum theory, while our representation tries to avoid terms from quantum theory to also give the non-physicists a chance to follow. See e.g. [7, 16] for an introduction from an engineering or mathematical point of view.

In the following we just repeat the basic notions and introduce our notations.

### *2.1 Preliminaries*

Let $\mathbf{D}_g^j$ be the unitary irreducible representation of a $g \in SO(3)$ of order $j$ with $j \in \mathbb{N}$. They are also known as the *Wigner D-matrices* (see e.g. [13]). The representation $\mathbf{D}_g^j$ acts on a vector space $V_j$ which is represented by $\mathbb{C}^{2j+1}$. We write the elements of $V_j$ in bold face, e.g. $\mathbf{u} \in V_j$ and write the $2j+1$ components in unbold face $u_m \in \mathbb{C}$ where $m = -j, \ldots j$. For the transposition of a vector/matrix we write $\mathbf{u}^T$; the adjoint representation (a joint complex conjugation and transposition) is denoted by $\mathbf{u}^\top = \overline{\mathbf{u}}^T$. In this terms the unitarity of $\mathbf{D}_g^j$ is expressed by the formula $(\mathbf{D}_g^j)^\top \mathbf{D}_g^j = \mathbf{I}$.

Note, that we treat the space $V_j$ as a real vector space of dimensions $2j+1$, although the components of $\mathbf{u}$ might be complex. This means that the space $V_j$ is only closed under weighted superpositions with real numbers. As a consequence of this we always have that the components are interrelated by $\overline{u_m} = (-1)^m u_{-m}$. From

a computational point of view this is an important issue. Although the vectors are elements of $\mathbb{C}^{2j+1}$ we just have to store just $2j+1$ real numbers.

We denote the standard basis of $\mathbb{C}^{2j+1}$ by $\mathbf{e}_m^j$, where the $n$th component of $\mathbf{e}_m^j$ is $\delta_{mn}$. In contrast, the standard basis of $V_j$ is written as $\mathbf{c}_m^j = \frac{1+\mathbf{i}}{2}\mathbf{e}_m^j + (-1)^m \frac{1-\mathbf{i}}{2}\mathbf{e}_{-m}^j$. We denote the corresponding 'imaginary' space by $\mathbf{i}V_j$, i.e. elements of $\mathbf{i}V_j$ can be written as $\mathbf{iv}$ where $\mathbf{v} \in V_j$. So, elements $\mathbf{w} \in \mathbf{i}V_j$ fulfill $\overline{w_m} = (-1)^{m+1} w_{-m}$. Hence, we can write the space $\mathbb{C}^{2j+1}$ as the direct sum of the two spaces $\mathbb{C}^{2j+1} = V_j \oplus \mathbf{i}V_j$. The standard coordinate vector $\mathbf{r} = (x,y,z)^T \in \mathbb{R}^3$ has a natural relation to elements $\mathbf{u} \in V_1$ by

$$\mathbf{u} = \frac{x-y}{\sqrt{2}}\mathbf{c}_1^1 + z\mathbf{c}_0^1 - \frac{x+y}{\sqrt{2}}\mathbf{c}_{-1}^1 = \begin{pmatrix} \frac{1}{\sqrt{2}}(x-\mathbf{i}y) \\ z \\ -\frac{1}{\sqrt{2}}(x+\mathbf{i}y) \end{pmatrix} = \mathbf{S}\mathbf{r} \in V_I$$

Note, that $\mathbf{S}$ is a unitary coordinate transformation. The representation $\mathbf{D}_g^1$ is directly related to the real valued rotation matrix $\mathbf{U}_g \in SO(3) \subset \mathbb{R}^{3\times 3}$ by $\mathbf{D}_g^1 = \mathbf{S}\mathbf{U}_g\mathbf{S}^\top$.

**Definition 2.1** *A function $\mathbf{f} : \mathbb{R}^3 \mapsto \mathbb{C}^{2j+1}$ is called a spherical tensor field of rank $j$ if it transforms with respect to rotations as*

$$(g\mathbf{f})(\mathbf{r}) := \mathbf{D}_g^j \mathbf{f}(\mathbf{U}_g^T \mathbf{r})$$

*for all $g \in SO(3)$. The space of all spherical tensor fields of rank $j$ is denoted by $\mathscr{T}_j$.*

## 2.2 Spherical Tensor Coupling

Now, we define a family of bilinear forms that connect tensors of different ranks.

**Definition 2.2** *For every $j \geq 0$ we define a family of bilinear forms of type*

$$\circ_j : V_{j_1} \times V_{j_2} \mapsto \mathbb{C}^{2j+1}$$

*where $j_1, j_2 \in \mathbb{N}$ has to be chosen according to the triangle inequality $|j_1 - j_2| \leq j \leq j_1 + j_2$. It is defined by*

$$(\mathbf{e}_m^j)^\top (\mathbf{v} \circ_j \mathbf{w}) := \sum_{m=m_1+m_2} \langle jm \mid j_1 m_1, j_2 m_2 \rangle v_{m_1} w_{m_2}$$

*where $\langle jm \mid j_1 m_1, j_2 m_2 \rangle$ are the Clebsch-Gordan coefficients.*

For references concerning the Clebsch-Gordan coefficients see the appendix. The characterizing property of these products is that they respect the rotations of the arguments, namely

**Proposition 2.3** *Let $\mathbf{v} \in V_{j_1}$ and $\mathbf{w} \in V_{j_2}$, then for any $g \in SO(3)$*

$$(\mathbf{D}_g^{j_1}\mathbf{v}) \circ_j (\mathbf{D}_g^{j_2}\mathbf{w}) = \mathbf{D}_g^{j}(\mathbf{v} \circ_j \mathbf{w})$$

*holds.*

*Proof.* The components of the left-hand side look as

$$\begin{aligned}(\mathbf{e}_m^j)^\top ((\mathbf{D}_g^{j_1}\mathbf{v}) \circ_j (\mathbf{D}_g^{j_2}\mathbf{w})) \\ = \sum_{\substack{m=m_1+m_2 \\ m_1' m_2'}} \langle jm | j_1 m_1, j_2 m_2 \rangle D^{j_1}_{m_1 m_1'} D^{j_2}_{m_2 m_2'} v_{m_1'} w_{m_2'}\end{aligned}$$

First, one has to insert the identity by using the orthogonality relation (17) with respect to $m_1'$ and $m_2'$. Then we can use relation (25) and the definition of $\circ_j$ to prove the assertion.

**Proposition 2.4** *If $j_1 + j_2 + j$ is even, than $\circ$ is symmetric, otherwise antisymmetric. The spaces $V_j$ are closed for the symmetric product, for the antisymmetric product this is not the case.*

$$\begin{aligned} j + j_1 + j_2 \text{ is even } &\Rightarrow \mathbf{v} \circ_j \mathbf{w} \in V_j \\ j + j_1 + j_2 \text{ is odd } &\Rightarrow \mathbf{v} \circ_j \mathbf{w} \in \mathbf{i}V_j, \end{aligned}$$

*where $\mathbf{v} \in V_{j_1}$ and $\mathbf{w} \in V_{j_2}$.*

*Proof.* The symmetry and antisymmetry is founded in the symmetry properties of the Clebsch-Gordan coefficients in equation (23). To show the closure property consider

$$\begin{aligned}(\mathbf{e}_m^j)^\top \overline{\mathbf{v} \circ_j \mathbf{w}} &= \sum_{m=m_1+m_2} \langle jm | j_1 m_1, j_2 m_2 \rangle \overline{v_{m_1} w_{m_2}} \\ &= \sum_{m=m_1+m_2} (-1)^m \langle jm | j_1 m_1, j_2 m_2 \rangle v_{-m_1} w_{-m_2} \\ &= \sum_{m=m_1+m_2} (-1)^{m+j+j_1+j_2} \langle j(-m) | j_1 m_1, j_2 m_2 \rangle v_{m_1} w_{m_2} \\ &= (-1)^{m+j+j_1+j_2} (\mathbf{e}_{-m}^j)^\top \overline{\mathbf{v} \circ_j \mathbf{w}},\end{aligned}$$

where we used the symmetry property given in equation (24). Hence, we have for even $j + j_1 + j_2$ the 'realness' condition complying to $V_j$ and for odd $j + j_1 + j_2$ the 'imaginariness' condition for $\mathbf{i}V_j$, which prove the statements.

We will later see that the symmetric product plays an important role, in particular, because we can normalize it in a special way such that it shows a more gentle behavior with respect to the spherical harmonics.

**Definition 2.5** *For every $j \geq 0$ with $|j_1 - j_2| \leq j \leq j_1 + j_2$ and even $j + j_1 + j_2$ we define a family of symmetric bilinear forms by*

$$\mathbf{v} \bullet_j \mathbf{w} := \frac{1}{\langle j0 | j_1 0, j_2 0 \rangle} \mathbf{v} \circ_j \mathbf{w}$$

For the special case $j = 0$ the arguments have to be of the same rank due to the triangle inequality. Actually in this case the symmetric product coincides with the standard inner product

$$\mathbf{v} \bullet_0 \mathbf{w} = \sum_{m=-j}^{m=j} (-1)^m v_m w_{-m} = \mathbf{w}^\top \mathbf{v},$$

where $j$ is the rank of $\mathbf{v}$ and $\mathbf{w}$.

**Proposition 2.6** *The products $\circ$ and $\bullet$ are associative in the following manner.*

$$\mathbf{v}^{j_1} \circ_\ell (\mathbf{w}^{j_2} \circ_{j_2+j_3} \mathbf{y}^{j_3}) = (\mathbf{v}^{j_1} \circ_{j_1+j_2} \mathbf{w}^{j_2}) \circ_\ell \mathbf{y}^{j_3} \tag{1}$$

*holds if $j_1 + j_2 + j_3 = \ell$. And*

$$\mathbf{v}^{j_2} \circ_\ell (\mathbf{w}^{j_1} \circ_{j_1+j_3} \mathbf{y}^{j_3}) = (\mathbf{v}^{j_1} \circ_{j_2-j_1} \mathbf{w}^{j_2}) \circ_\ell \mathbf{y}^{j_3} \tag{2}$$

*holds with $\ell = j_2 - (j_1 + j_3) \geq 0$.*

*Proof.* Both statements are proved by using the explicit formulas for the special cases of the Clebsch-Gordan coefficients as given in equation (20) and (21).

The introduced product can also be used to combine tensor fields of different rank by point-wise multiplication.

**Proposition 2.7** *Let $\mathbf{v} \in \mathscr{T}_{j_1}$ and $\mathbf{w} \in \mathscr{T}_{j_2}$ and $j$ chosen such that $|j_1 - j_2| \leq j \leq j_1 + j_2$, then*

$$\mathbf{f}(\mathbf{r}) = \mathbf{v}(\mathbf{r}) \circ_j \mathbf{w}(\mathbf{r})$$

*is in $\mathscr{T}_j$, i.e. a tensor field of rank $j$.*

In fact, there is another way to combine two tensor fields: by convolution. The evolving product respects the translation in a different sense.

**Proposition 2.8** *Let $\mathbf{v} \in \mathscr{T}_{j_1}$ and $\mathbf{w} \in \mathscr{T}_{j_2}$ and $j$ chosen such that $|j_1 - j_2| \leq j \leq j_1 + j_2$, then*

$$(\mathbf{v} \widetilde{\circ}_j \mathbf{w})(\mathbf{r}) := \int_{\mathbb{R}^3} \mathbf{v}(\mathbf{r}' - \mathbf{r}) \circ_j \mathbf{w}(\mathbf{r}')\, d\mathbf{r}'$$

*is in $\mathscr{T}_j$, i.e. a tensor field of rank $j$.*

## 2.3 Relation to Cartesian Tensors

The correspondence of spherical and cartesian tensors of rank 0 is trivial. For rank 1 it is just the matrix $\mathbf{S}$ that connects the real-valued vector $\mathbf{r} \in \mathbb{R}^3$ with the spherical coordinate vector $\mathbf{u} = \mathbf{S}\mathbf{r} \in V_1$. For rank 2 the consideration gets more intricate. Consider a real-valued cartesian rank-2 tensor $\mathbf{T} \in \mathbb{R}^{3\times 3}$ and the following unique decomposition

$$\mathbf{T} = \alpha \mathbf{I}_3 + \mathbf{T}_{\text{anti}} + \mathbf{T}_{\text{sym}},$$

where $\alpha \in \mathbb{R}$, $\mathbf{T}_{\text{anti}}$ is an antisymmetric matrix and $\mathbf{T}_{\text{sym}}$ a traceless symmetric matrix. In fact, this decomposition follows the same rules as the spherical tensor decomposition. A rank 0 spherical tensor corresponds to the identity matrix in cartesian notation, while the rank 1 spherical tensor to corresponds to an antisymmetric $3 \times 3$ matrix or, equivalently, to a vector. The rank 2 spherical tensor corresponds to a traceless, symmetric matrix. Let us consider the spherical decomposition. For convenience let $\mathbf{T}^s = \mathbf{STS}^\top$, then the components of the corresponding spherical tensors $\mathbf{b}^j \in V_j$ with $j = 0, 1, 2$ look as

$$b^j_m = \sum_{m_1+m_2=m} \langle 1m_1, 1m_2 | jm \rangle (-1)^{m_1} T^s_{(-m_1)m_2},$$

where $\mathbf{b}^0$ corresponds to $\alpha$, $\mathbf{b}^1$ to $\mathbf{T}_{\text{anti}}$ and $\mathbf{b}^2$ to $\mathbf{T}_{\text{sym}}$. The inverse of this 'cartesian to spherical'-transformation is

$$T^s_{m_1 m_2} = \sum_{j=0}^{2} \sum_{m=-j}^{m=j} \langle 1(-m_1), 1m_2 | jm \rangle (-1)^{m_1} b^j_m.$$

In particular, consider a cartesian symmetric 2-tensor and its eigensystem. In spherical tensor notation the spherical tensor $\mathbf{b}^2$ is decomposed into products of three 1-tensors $\mathbf{v}_k \in V_1$ as

$$\mathbf{b}^2 = \sum_{k=-1}^{1} \lambda_k \, \mathbf{v}_k \circ_2 \mathbf{v}_k,$$

where $\mathbf{v}_k$ are the eigenvectors of $\mathbf{T}^s$ and $\lambda_k$ the eigenvalues. Note that $\mathbf{b}^2$ is invariant against a common shift of the eigenvalues by some offset $\gamma$. It is 'traceless' in the sense that

$$\sum_{k=-1}^{1} \mathbf{v}_k \circ_2 \mathbf{v}_k = \mathbf{0},$$

for any set of orthogonal vectors $\mathbf{v}_{-1}, \mathbf{v}_0, \mathbf{v}_1 \in V_1$. This offset, namely the trace of $\mathbf{T}$ is covered by the zero-rank $\mathbf{b}^0$. It corresponds to the 'ballness' or 'isotropy' of $\mathbf{T}$.

## *2.4 Spherical Harmonics*

We denote the well-known spherical harmonics by $\mathbf{Y}^j : S^2 \to V_j$ (see appendix). We always write $\mathbf{Y}^j(\mathbf{r})$, where $\mathbf{r}$ may be an element of $\mathbb{R}^3$, but $\mathbf{Y}^j(\mathbf{r})$ is independent of the magnitude of $r = \|\mathbf{r}\|$, i.e. $\mathbf{Y}^j(\lambda \mathbf{r}) = \mathbf{Y}^j(\mathbf{r})$ for any $\lambda \in \mathbb{R}$. We know that the $\mathbf{Y}^j$ provide an orthogonal basis of scalar function on the 2-sphere $S^2$. Thus, any real scalar field $f \in \mathscr{T}_0$ can be expanded in terms of spherical harmonics in a unique manner:

$$f(\mathbf{r}) = \sum_{j=0}^{\infty} \mathbf{a}^j(r)^\top \mathbf{Y}^j(\mathbf{r}),$$

where the $\mathbf{a}^j(r)$ are expansion coefficients just depending on the radius $r = \|\mathbf{r}\|$. In the following, we always use Racah's normalization (also known as semi-Schmidt normalization), i.e.

$$\langle Y_m^j, Y_{m'}^{j'} \rangle = \frac{1}{4\pi} \int_{S^2} Y_m^j(\mathbf{s}) \, \overline{Y_{m'}^{j'}}(\mathbf{s}) d\mathbf{s} = \frac{1}{2j+1} \delta_{jj'} \delta_{mm'}$$

where the integral ranges over the 2-sphere using the standard measure. One important property of the Racah-normalized spherical harmonics is that $\mathbf{Y}^{j\top} \mathbf{Y}^j = 1$. Another important and useful property is that

$$\mathbf{Y}^j = \mathbf{Y}^{j_1} \bullet_j \mathbf{Y}^{j_2} \tag{3}$$

if $j + j_1 + j_2$ is even. We can use this formula to iteratively compute higher order $\mathbf{Y}^j$ from given lower order ones. Note that $\mathbf{Y}^0 = 1$ and $\mathbf{Y}^1 = \mathbf{Sr}$, where $\mathbf{r} \in S^2$.

The spherical harmonics have a variety of nice properties. One of the most important ones is that each $\mathbf{Y}^j$, interpreted as a tensor field of rank $j$ is a fix-point with respect to rotations, i.e.

$$(g\mathbf{Y}^j)(\mathbf{r}) = \mathbf{D}_g^j \mathbf{Y}^j(\mathbf{U}_g^T \mathbf{r}) = \mathbf{Y}^j(\mathbf{r})$$

or in other words $\mathbf{Y}^j(\mathbf{U}_g \mathbf{r}) = \mathbf{D}_g^j \mathbf{Y}^j(\mathbf{r})$. A consequence of this is that the expansion coefficients of the rotated function $(gf)(\mathbf{r}) = f(\mathbf{U}_g^T \mathbf{r})$ just look as $\mathbf{D}_g^j \mathbf{a}^j(r)$.

Note that the spherical harmonics arise as solutions of the Laplace equation $\Delta f = 0$. One set of solutions are the homogeneous polynomials

$$\mathbf{R}^j(\mathbf{r}) := r^j \mathbf{Y}^j(\mathbf{r}),$$

i.e. the $\mathbf{R}^j$ fulfill $\mathbf{R}^j(\lambda \mathbf{r}) = \lambda^j \mathbf{R}^j(\mathbf{r})$ and the components solve the Laplace equation $\Delta R_m^j = 0$. In literature these functions are called the solid harmonics. They will get important in the context of the spherical tensor derivatives.

## 3 Tensorial Harmonic Expansion

We propose to expand a tensor field $\mathbf{f} \in \mathscr{T}_\ell$ of rank $\ell$ as follows

$$\mathbf{f}(\mathbf{r}) = \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} \mathbf{a}_k^j(r) \circ_\ell \mathbf{Y}^j(\mathbf{r}),$$

where $\mathbf{a}_k^j(r) \in \mathscr{T}_{j+k}$ are expansion coefficients. Note, that for $\ell = 0$ the expansion coincides with the ordinary scalar expansion from above. We can further observe

that

$$
\begin{aligned}
(g\mathbf{f})(\mathbf{r}) &= \mathbf{D}_g^{\ell}\mathbf{f}(\mathbf{U}_g^{\top}\mathbf{r}) \\
&= \sum_{j=0}^{\infty}\sum_{k=-\ell}^{k=\ell}(\mathbf{D}_g^{j+k}\mathbf{a}_k^j(r))\circ_\ell \mathbf{Y}^j(\mathbf{r}) \qquad (4)
\end{aligned}
$$

i.e. a rotation of the tensor field affects the expansion coefficients $\mathbf{a}_k^j$ to be transformed by $\mathbf{D}_g^{j+k}$.

By setting $\mathbf{a}_k^j(r) = \sum_{m=-(j+k)}^{m=j+k} a_{km}^j(r)\mathbf{e}_m^{j+k}$ we can identify the functional basis $\mathbf{Z}_{km}^j$ as

$$
\mathbf{f}(\mathbf{r}) = \sum_{j=0}^{\infty}\sum_{k=-\ell}^{k=\ell}\sum_{m=-(j+k)}^{m=j+k} a_{km}^j(r)\underbrace{\mathbf{e}_m^{j+k}\circ_\ell \mathbf{Y}^j(\mathbf{r})}_{\mathbf{Z}_{km}^j},
$$

**Proposition 3.1 (Tensorial Harmonics)** *The functions $\mathbf{Z}_{km}^j : S^2 \mapsto V_\ell$ provide an complete and orthogonal basis of the angular part of $\mathscr{T}_\ell$, i.e.*

$$
\int_{S^2}(\mathbf{Z}_{km}^j(\mathbf{s}))^{\top}\mathbf{Z}_{k'm'}^{j'}(\mathbf{s})d\mathbf{s} = \frac{4\pi}{N_{j,k}}\delta_{j,j'}\delta_{k,k'}\delta_{m,m'},
$$

*where*

$$
N_{j,k} = \frac{1}{2\ell+1}(2j+1)(2(j+k)+1).
$$

*The functions $\mathbf{Z}_{km}^j$ are called the tensorial harmonics.*

*Proof.* We first show the orthogonality by elementary calculations:

$$
\begin{aligned}
&\frac{1}{4\pi}\int_{S^2}(\mathbf{Z}_{km}^j(\mathbf{s}))^{\top}\mathbf{Z}_{k'm'}^{j'}(\mathbf{s})d\mathbf{s} \\
&= \sum_{M=-\ell}^{\ell}\langle \ell M|(j+k)m,j(M-m)\rangle\langle \ell M|(j'+k')m',j'(M-m')\rangle \underbrace{\tfrac{1}{4\pi}\textstyle\int_{S^2}\overline{Y_{M-m}^j}Y_{M-m'}^{j'}}_{\frac{\delta_{j,j'}\delta_{m,m'}}{2j+1}} \\
&= \frac{\delta_{j,j'}\delta_{m,m'}}{2j+1}\underbrace{\sum_{M=-\ell}^{\ell}\langle \ell M|(j+k)m,j(M-m)\rangle\langle \ell M|(j+k')m,j(M-m)\rangle}_{\frac{2\ell+1}{2(j+k)+1}\delta_{(j+k),(j+k')}} \\
&= \delta_{j,j'}\delta_{k,k'}\delta_{m,m'}\frac{1}{2(j+k)+1}\frac{2\ell+1}{2j+1}
\end{aligned}
$$

In line 2 we use the orthogonality of the Racah-normalized spherical harmonics. In the third line we use the orthogonality relation for the Clebsch-Gordan coefficients given in (19).

Secondly, we want to show that the expansion of a spherical tensor field $\mathbf{f} \in \mathscr{T}_\ell$ in terms of tensorial harmonics is unique and complete. Everybody agrees that the expansion of the individual components $(\mathbf{e}_M^\ell)^\top \mathbf{f}$ in spherical harmonics is complete. That is, we can write the expansion as

$$(\mathbf{e}_M^\ell)^\top \mathbf{f}(\mathbf{r}) = \sum_{j=0}^{\infty} \sum_{n=-j}^{j} \mathbf{b}_M^j(r)^\top \mathbf{Y}^j(\mathbf{r}),$$

where $\mathbf{b}_M^j(r) \in V_j$ are the expansion coefficients for the $M$th component. We show the completeness of the tensorial harmonics by connecting them in an one-to-one manner with this ordinary spherical harmonic expansion of the spherical tensor field. For convenience we just consider the $j$th term in the expansion, i.e. the homogeneous part of $\mathbf{f}$ of order $j$ that we denote by $\mathbf{f}^j$. We start with the expansion in terms of tensorial harmonics and rewrite them to identify the elements of $\mathbf{b}_M^j(r)$ written as $b_{M,n}^j(r)$ in terms of the $a_{km}^j(r)$. And so,

$$\begin{aligned}
(\mathbf{e}_M^\ell)^\top \mathbf{f}^j(\mathbf{r}) &= \sum_{k=-\ell}^{\ell} \sum_{m+n=M} a_{km}^j(r) \langle \ell M | (j+k)m, jn \rangle Y_n^j(\mathbf{r}) \\
&= \sum_{n=-j}^{j} Y_n^j(\mathbf{r}) \underbrace{\sum_{k=-\ell}^{\ell} \sum_{m} a_{km}^j(r) \langle \ell M | (j+k)m, jn \rangle}_{b_{M,n}^j(r)} \\
&= \sum_{n=-j}^{j} b_{M,n}^j(r) Y_n^j(\mathbf{r}).
\end{aligned}$$

Now, we just have to give the inverse relation that computes the $a_{km}^j$ out of the $b_{Mn}^j$. This can be accomplished by

$$\begin{aligned}
&\sum_{M,n} b_{M,n}^j(r) \langle \ell M | (j+k')m', jn \rangle \\
&\quad = \sum_{M,n} \sum_{k=-\ell}^{\ell} \sum_{m} a_{km}^j(r) \langle \ell M | (j+k)m, jn \rangle \langle \ell M | (j+k')m', jn \rangle \\
&\quad = \sum_{k=-\ell}^{\ell} \sum_{m} a_{km}^j(r) \underbrace{\sum_{M,n} \langle \ell M | (j+k)m, jn \rangle \langle \ell M | (j+k')m', jn \rangle}_{\delta_{k,k'} \delta_{m,m'} \frac{2\ell+1}{2(j+k')+1}} \\
&\quad = \frac{2\ell+1}{2(j+k')+1} a_{k'm'}^j(r),
\end{aligned}$$

where we used again the orthogonality relation for the Clebsch-Gordan coefficients given in (19). This provides the one-to-one relation between the tensorial harmonic

expansion with the component-wise spherical harmonic expansion and proves the statement.

### *3.1 Symmetric Tensor Fields*

Typical filter kernels show certain symmetry properties. We figured out three symmetries that result in a vanishing of specific terms in the tensorial expansion: the rotationally symmetry with respect to a certain axis, the absence of torsion and reflection symmetry.

The rotation symmetry of a spherical tensor field $\mathbf{f} \in \mathscr{T}_\ell$ about the $z$-axis is expressed algebraically by the fact that $g_\phi \mathbf{f} = \mathbf{f}$ for all rotation $g_\phi$ around the $z$-axis. Such fields can easily be obtained by averaging a general tensor field $\mathbf{f}$ over all these rotations

$$\mathbf{f}_\mathrm{s} = \frac{1}{2\pi} \int_0^{2\pi} g_\phi \mathbf{f} \, d\phi .$$

It is well known that the representation $\mathbf{D}^j_{g_\phi}$ of such a rotation is diagonal, namely $D^j_{g_\phi,mm'} = \delta_{mm'} e^{\mathrm{i}m\phi}$. Hence, the expansion coefficients $a^j_{km}$ of $\mathbf{f}_\mathrm{s}$ vanish for all $m \neq 0$. Thus, we can write any rotation symmetric tensor field as

$$\mathbf{f}_\mathrm{s}(\mathbf{r}) = \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} a^j_k(r) \, \mathbf{e}_0^{j+k} \circ_\ell \mathbf{Y}^j(\mathbf{r}). \tag{5}$$

We call such a rotation symmetric field torsion-free if $g_{yz}\mathbf{f}_\mathrm{s} = \mathbf{f}_\mathrm{s}$, where $g_{yz} \in O(3)$ is a reflection with respect to the $yz$-plane (or $xz$-plane). In Figure 1 we give an example of such a field. The action of such a reflection on spherical tensors is given by $D^j_{g_{yz},mm'} = (-1)^m \delta_{m(-m')}$. Similar to the rotational symmetry we can obtain such fields by averaging over the symmetry operation

$$\mathbf{f}_\mathrm{stf} = \frac{1}{2}(\mathbf{f}_\mathrm{s} + g_{yz}\mathbf{f}_\mathrm{s}).$$

Note, that the mirroring operation for a spherical harmonic is just a complex conjugation, that is $\mathbf{Y}^j(\mathbf{U}^T_{g_{yz}}\mathbf{r}) = \overline{\mathbf{Y}^j}(\mathbf{r})$. The consequence for equation (5) is that all terms where the $k+\ell$ are odd vanish. The reason for that is mainly Proposition 2.4 because with its help we can show that

$$\mathbf{D}^\ell_{g_{yz}}(\mathbf{e}_0^{j+k} \circ_\ell \mathbf{Y}^j(\mathbf{U}^T_{g_{yz}}\mathbf{r})) = (-1)^{(k+\ell)}(\mathbf{e}_0^{j+k} \circ_\ell \mathbf{Y}^j(\mathbf{r}))$$

holds.

Finally, consider the reflection symmetry with respect to the $xy$-plane. This symmetry is particularly important for rank 2 spherical tensor fields. In TV such fields are typically aligned or 'steered' with quantities of the same, even rank. For even rank tensors the parity of the underlying quantity is lost, so the voting field has to

invariant under such parity changes. This symmetry is algebraically expressed by $g_{xy}\mathbf{f}_s = \mathbf{f}_s$ where $g_{xy} \in O(3)$ is a reflection with respect to the $xy$-plane, whose action on spherical tensors is given by $D^j_{g_{yz},mm'} = (-1)^j \delta_{mm'}$. Averaging over this symmetry operation has the consequence that expansion terms with odd $j$ are vanishing. For odd rank tensor fields the reflection symmetry is not imperative. But there is typically an antisymmetry of the form $g_{xy}\mathbf{f}_s = -\mathbf{f}_s$. This antisymmetry let vanish the expansion terms with even index $j$.

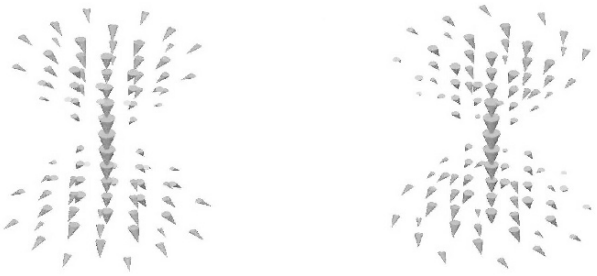


**Fig. 1** Rotation symmetric vector fields. Left: torsion-free. Right: with torsion.

## *3.2 Expanding Rotation-Symmetric Fields in Polar Representation*

We write the spherical tensor field in polar representation $\mathbf{f}(r,\theta,\phi)$, where $\cos(\theta) = z/r$ and $\phi = \arg(x + \mathbf{i}y)$. Consider a field of rank $\ell$. In polar representation the rotation symmetry with respect to the $z$-axis is expressed by the fact that for all $m = -\ell,\ldots,\ell$ the components $f_m(r,\theta,\phi)$ of the field $\mathbf{f}$ can be written as

$$f_m(r,\theta,\phi) = \alpha_m(r,\theta)e^{\mathbf{i}m\phi},$$

where $\alpha_m(r,\theta) \in \mathbb{C}$ is the colatitudinal/radial dependency of the field. This rotation symmetry is easy to verify because $f_m(r,\theta,\phi-\phi')e^{\mathbf{i}m\phi'} = f_m(r,\theta,\phi)$. For torsion-free tensor fields we additionally know that $\alpha_m(r,\theta) \in \mathbb{R}$. To project such a symmetric kind of field on the tensorial harmonics consider the $m$th component of the tensorial harmonic $\mathbf{Z}^j_{k0}$:

$$
\begin{aligned}
(\mathbf{e}_m^\ell)^\top \mathbf{Z}_{k0}^j(\theta,\phi) &= (\mathbf{e}_m^\ell)^\top (\mathbf{e}_0^{j+k} \circ_\ell \mathbf{Y}^j(\theta,\phi)) \\
&= \langle \ell m \mid (j+k)0, jm \rangle Y_m^j(\theta,\phi) \\
&= \langle \ell m \mid (j+k)0, jm \rangle e^{\mathbf{i}m\phi} \sqrt{\frac{(j-m)!}{(j+m)!}} P_m^j(\cos(\theta)) \\
&= C_{\ell jm} e^{\mathbf{i}m\phi} P_m^j(\cos(\theta))
\end{aligned}
$$

Now, using this expression the projection on $\mathbf{Z}_{k0}^j$ yields

$$
\begin{aligned}
\langle \mathbf{Z}_{k0}^j, \mathbf{f} \rangle_{S^2} &= \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} \mathbf{Z}_{k0}^j(\theta,\phi)^\top \mathbf{f}(r,\theta,\phi) \sin(\theta)\, d\phi d\theta \\
&= 2\pi \sum_{m=-\ell}^{\ell} C_{\ell jm} \int_{-\pi/2}^{\pi/2} \alpha_m(r,\theta) P_m^j(\cos(\theta)) \sin(\theta) d\theta
\end{aligned}
$$

The residue integral may be computed numerically or analytically.

### *3.3 Rotational Steering*

By equation (4) the tensorial harmonics are very well suited to rotate the expanded spherical tensor field. We want to show how to steer a rotation symmetric field efficiently in a certain direction.

Consider a general rotation $g_{\mathbf{n}} \in SO(3)$ that rotates the $z$-axis $\mathbf{r}_z = (0,0,1)^\top$ to some given orientation $\mathbf{n} \in \mathbb{R}^3$, i.e. $\mathbf{R}_{g_{\mathbf{n}}} \mathbf{r}_z = \mathbf{n}$. Of course, there are several rotations that can accomplish this. But, if we apply such a rotation on a rotational symmetric field $\mathbf{f}_\mathrm{s}$ this additional freedom does not have an influence on the result. Starting from the general rotation behavior of the tensorial harmonic expansion in Eq. (4) one can derive that the symmetric tensor field $\mathbf{f}_\mathrm{s}$ rotates as

$$
(g_{\mathbf{n}} \mathbf{f}_\mathrm{s})(\mathbf{r}) = \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} a_k^j(r)\, \mathbf{Y}^{j+k}(\mathbf{n}) \circ_\ell \mathbf{Y}^j(\mathbf{r}) \tag{6}
$$

This expression is the basis for the algorithm proposed in the next section. To prove equation (6) one needs to know that $\mathbf{Y}^j(\mathbf{r}_z) = \mathbf{e}_0^j$.

## 4 Local Adaptive Filtering with Tensorial Harmonics

We already described the two dual ideas of local adaptive filtering in the introduction. In this section we describe how tensorial harmonics can be used to compute

the filter integrals efficiently. (application examples can be found in [12]) For both cases we assume that the filter kernel is tensor-valued of rank $\ell$, i.e. a function $\mathbf{V}^n : \mathbb{R}^3 \to V_\ell$. The intensity image is still represented by the function $m : \mathbb{R}^3 \to \mathbb{R}$ and an orientation image $\mathbf{n} : \mathbb{R}^3 \to V_1$ of normalized vectors is given. We also assume a rotation symmetric filter kernel as given in equation (6). The expansion coefficients $a_k^j(r)$ can be obtained by a projection of the filter kernel onto the tensorial harmonics

$$a_k^j(r) = N_{j,k} \langle \mathbf{Z}_{k0}^j, \mathbf{V}^{\mathbf{r}_z} \rangle_{S_r^2} \tag{7}$$

due to the symmetry only $\mathbf{Z}_{k0}^j$ are involved. For the numerical integration scheme see Section 3.2.

## *4.1 The Convolution Integral*

The key expression that has to be computed is

$$\mathbf{U}_{\text{conv}}(\mathbf{r}) = \int_{\mathbb{R}^3} \mathbf{V}^{\mathbf{n}(\mathbf{r}')}(\mathbf{r} - \mathbf{r}')\, m(\mathbf{r}')\, d\mathbf{r}', \tag{8}$$

Following the last section we set the voting field to $\mathbf{V}^{\mathbf{n}}(\mathbf{r}) = (g_{\mathbf{n}}\mathbf{f}_{\text{s}})(\mathbf{r})$, where $\mathbf{f}_{\text{s}}$ is the rotational symmetric field. Inserting this expression in (8) and using Eq. (6) yields

$$\begin{aligned}
\mathbf{U}_{\text{conv}}(\mathbf{r}) &= \int_{\mathbb{R}^3} \mathbf{V}^{\mathbf{n}(\mathbf{r}')}(\mathbf{r} - \mathbf{r}')\, m(\mathbf{r}')\, d\mathbf{r}' = \int_{\mathbb{R}^3} (g_{\mathbf{n}(\mathbf{r}')}\mathbf{f}_{\text{s}})(\mathbf{r} - \mathbf{r}')\, m(\mathbf{r}')\, d\mathbf{r}' \\
&= \int_{\mathbb{R}^3} \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} a_k^j(|\mathbf{r} - \mathbf{r}'|)\; \mathbf{Y}^{j+k}(\mathbf{n}(\mathbf{r}')) \circ_\ell \mathbf{Y}^j(\mathbf{r} - \mathbf{r}')\, m(\mathbf{r}')\, d\mathbf{r}' \\
&= \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} \int_{\mathbb{R}^3} \underbrace{m(\mathbf{r}')\mathbf{Y}^{j+k}(\mathbf{n}(\mathbf{r}'))}_{\mathbf{E}^{j+k}(\mathbf{r}')} \circ_\ell \underbrace{a_k^j(|\mathbf{r} - \mathbf{r}'|)\mathbf{Y}^j(\mathbf{r} - \mathbf{r}')}_{\mathbf{A}_k^j(\mathbf{r}-\mathbf{r}'))}\, d\mathbf{r}' \\
&= \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} \mathbf{E}^{j+k} \widetilde{\circ}_\ell \mathbf{A}_k^j
\end{aligned}$$

where $\mathbf{E}^j(\mathbf{r}) := m(\mathbf{r})\mathbf{Y}^j(\mathbf{n}(\mathbf{r}))$ are combined tensor-valued evidence images and $\mathbf{A}_k^j(\mathbf{r}) := a_k^j(r)\mathbf{Y}^j(\mathbf{r})$ is the harmonic expansion of the voting field. In Algorithm 1 we give pseudo-code for implementation.

**Algorithm 1** Convolution Algorithm

**Input:** $m \in \mathscr{T}_0$, $\mathbf{n}(\mathbf{r}) \in \mathscr{T}_1$, $\mathbf{A}_k^j \in \mathscr{T}_j$
**Output:** $\mathbf{U} \in \mathscr{T}_\ell$

```
Let E^0 := m
for j = 1 : (j_max + ℓ) do
    E^j := (E^{j-1} ∘_j n) / ⟨j0|10, (j-1)0⟩
end for
for j = 0 : j_max do
    for k = -ℓ : 2 : ℓ do
        Compute U := U + E^{j+k} ~∘_ℓ A_k^j
    end for
end for
```

**Algorithm 2** Correlation Algorithm

**Input:** $m \in \mathscr{T}_0$, $\mathbf{n}(\mathbf{r}) \in \mathscr{T}_1$, $\mathbf{A}_k^j \in \mathscr{T}_j$
**Output:** $\mathbf{U} \in \mathscr{T}_\ell$

```
Let N^0 := 1
for j = 1 : (j_max + ℓ) do
    N^j := (N^{j-1} ∘_j n) / ⟨j0|10, (j-1)0⟩
end for
for j = 0 : j_max do
    for k = -ℓ : 2 : ℓ do
        Compute U := U + N^{j+k} ∘_ℓ (m * A_k^j)
    end for
end for
```

## *4.2 The Correlation Integral*

Let us now consider the correlation integral

$$\mathbf{U}_{\text{corr}}(\mathbf{r}) = \int_{\mathbb{R}^3} \mathbf{V}^{\mathbf{n}(\mathbf{r})}(\mathbf{r}' - \mathbf{r})\, m(\mathbf{r}')\, d\mathbf{r}'. \tag{9}$$

Following the same approach as in the previous section we can write

$$\begin{aligned}
\mathbf{U}_{\text{corr}}(\mathbf{r}) &= \int_{\mathbb{R}^3} \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} a_k^j(|\mathbf{r}' - \mathbf{r}|)\, \mathbf{Y}^{j+k}(\mathbf{n}(\mathbf{r})) \circ_\ell \mathbf{Y}^j(\mathbf{r}' - \mathbf{r})\, m(\mathbf{r}')\, d\mathbf{r}' \\
&= \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} \underbrace{\mathbf{Y}^{j+k}(\mathbf{n}(\mathbf{r}))}_{\mathbf{N}^{j+k}(\mathbf{r})} \int_{\mathbb{R}^3} m(\mathbf{r}') \circ_\ell \underbrace{a_k^j(|\mathbf{r}' - \mathbf{r}|)\mathbf{Y}^j(\mathbf{r}' - \mathbf{r})}_{\mathbf{A}_k^j(\mathbf{r}' - \mathbf{r}))}\, d\mathbf{r}' \\
&= \sum_{j=0}^{\infty} \sum_{k=-\ell}^{k=\ell} \mathbf{N}^{j+k} \circ_\ell (m * \mathbf{A}_k^j)
\end{aligned}$$

The final expression enables us to give an efficient computation scheme as depicted in Algorithm 2.

## 5 Spherical Tensor Derivatives

In this Section we propose derivative operators that connects spherical tensor fields of different ranks. We call them spherical tensor derivatives (STD). A similar operator was already proposed by Weniger [17]. In his work the coordinate functions in the harmonic polynomials $\mathbf{R}^\ell(\mathbf{r})$ are replaced by the partial derivatives, symbolically $\mathbf{R}^\ell(\nabla)$.

We will use the spherical derivatives to compute local adaptive filters for special types of filter kernels efficiently. The explicit convolutions used in the last section are replaced by finite difference operations.

The idea is to represent the filter kernel by superpositions of STDs of radial symmetric functions. Due to the commuting property of convolution and differentiation the computation of the filter response will just involve one explicit convolution with the radial symmetric functions, the rest of the computations consists of repeated applications of STDs.

In particular we will consider spherical derivatives of the Gaussian. We will see that the resulting polynomials are just solid harmonics (see Section 2.4). Based on this we will present a special type of filter kernel which can be defined for arbitrary tensor ranks and has a very simple parameter dependency controlling its shape and orientation.

**Proposition 5.1 (Spherical Tensor Derivatives)** *Let $\mathbf{f} \in \mathscr{T}_\ell$ be a tensor field. The spherical up-derivative $\nabla^1 : \mathscr{T}_\ell \to \mathscr{T}_{\ell+1}$ and the down-derivative $\nabla_1 : \mathscr{T}_\ell \to \mathscr{T}_{\ell-1}$ are defined as*

$$\nabla^1 \mathbf{f} := \nabla \bullet_{\ell+1} \mathbf{f} \tag{10}$$

$$\nabla_1 \mathbf{f} := \nabla \bullet_{\ell-1} \mathbf{f}, \tag{11}$$

*where*

$$\nabla = (\frac{1}{\sqrt{2}}(\partial_x - \mathbf{i}\partial_y), \partial_z, -\frac{1}{\sqrt{2}}(\partial_x + \mathbf{i}\partial_y))$$

*is the spherical gradient and $\partial_x, \partial_y, \partial_z$ the standard partial derivatives.*

*Proof.* We have to show that $\nabla^1 \mathbf{f} \in \mathscr{T}_{\ell+1}$, i.e.

$$\nabla^1(\mathbf{D}_g^\ell \mathbf{f}(\mathbf{U}_g^T \mathbf{r})) = \mathbf{D}_g^{\ell+1}(\nabla^1 \mathbf{f})(\mathbf{U}_g^T \mathbf{r})$$

and $\nabla_1 \mathbf{f} \in \mathscr{T}_{\ell-1}$

$$\nabla_1(\mathbf{D}_g^\ell \mathbf{f}(\mathbf{U}_g^T \mathbf{r})) = \mathbf{D}_g^{\ell-1}(\nabla_1 \mathbf{f})(\mathbf{U}_g^T \mathbf{r})$$

Both statements are proved just by using the properties of $\bullet$.

Note, that for a scalar function the spherical up-derivative is just the spherical gradient, i.e. $\nabla f = \nabla^1 f$.

In the Fourier domain the spherical derivatives act by point-wise $\bullet$-multiplications with a solid harmonic $\mathbf{i}k\mathbf{Y}^1(\mathbf{k}) = \mathbf{i}\mathbf{R}^1(\mathbf{k}) = \mathbf{i}\mathbf{S}\mathbf{k}$ where $k = \|\mathbf{k}\|$ the frequency magnitude:

**Proposition 5.2 (Fourier Representation)** *Let $\widetilde{\mathbf{f}}(\mathbf{k})$ be the Fourier transformation of some $\mathbf{f} \in \mathscr{T}_\ell$ and $\widetilde{\nabla}$ representations of the spherical derivative in the Fourier domain that are implicitly defined by $\widetilde{(\nabla \mathbf{f})} = \widetilde{\nabla}\widetilde{\mathbf{f}}$, then*

$$\widetilde{\nabla}^1 \widetilde{\mathbf{f}}(\mathbf{k}) = \mathbf{R}^1(\mathbf{i}\mathbf{k}) \bullet_{\ell+1} \widetilde{\mathbf{f}}(\mathbf{k}) \tag{12}$$

$$\widetilde{\nabla}_1 \widetilde{\mathbf{f}}(\mathbf{k}) = \mathbf{R}^1(\mathbf{i}\mathbf{k}) \bullet_{\ell-1} \widetilde{\mathbf{f}}(\mathbf{k}). \tag{13}$$

*Proof.* By the ordinary Fourier correspondence for the partial derivative, namely $\widetilde{\partial_x \mathbf{f}} = \mathbf{i}k_x \widetilde{\mathbf{f}}$, we can verify for the spherical gradient $\nabla$ that

$$\widetilde{\nabla} = \mathbf{i}\mathbf{S}\mathbf{k} = \mathbf{R}^1(\mathbf{i}\mathbf{k})$$

and hence

$$\widetilde{\nabla^1 \mathbf{f}} = \widetilde{(\nabla \bullet_{\ell+1} \mathbf{f})} = \widetilde{\nabla} \bullet_{\ell+1} \widetilde{\mathbf{f}} = \mathbf{R}^1(\mathbf{i}\mathbf{k}) \bullet_{\ell+1} \widetilde{\mathbf{f}}$$

which was to show. Proceed similar for the down-derivative.

In the following we want to use as a short-hand notation for multiple STDss

$$\nabla_i^\ell := \nabla_i \nabla^\ell := \underbrace{\nabla_1 \dots \nabla_1}_{i-\text{times}} \underbrace{\nabla^1 \dots \nabla^1}_{\ell-\text{times}}.$$

Note that if when this operator is applied to a scalar field one can show that $\nabla_i^\ell = \Delta^i \nabla^{n-i}$, where $\Delta^i$ is the Laplace operator applied $i$-times.

**Proposition 5.3 (Commuting Property for Convolutions)** *Let $\mathbf{A} \in \mathscr{T}_k$ and $\mathbf{B} \in \mathscr{T}_j$ be arbitrary spherical tensor fields then*

$$(\nabla^\ell \mathbf{A}) \,\widetilde{\bullet}_J\, \mathbf{B} = \mathbf{A} \,\widetilde{\bullet}_J (\nabla_\ell \mathbf{B}) \tag{14}$$

$$(\nabla^\ell \mathbf{A}) \,\widetilde{\bullet}_L\, \mathbf{B} = \mathbf{A} \,\widetilde{\bullet}_L (\nabla^\ell \mathbf{B}) \tag{15}$$

*where $J = j - (\ell + k)$ and $L = j + \ell + k$.*

*Proof.* Both assertions originate from the associativity of the spherical product. Consider the first statement in the Fourier domain by using equation (12) and then apply the associativity given in equation (2):

$$\begin{aligned}(\widetilde{\nabla^\ell \mathbf{A}}) \bullet_J \widetilde{\mathbf{B}} &= (\mathbf{R}^1 \bullet_{k+\ell} (\widetilde{\nabla^{\ell-1} \mathbf{A}})) \bullet_J \widetilde{\mathbf{B}} \\ &= (\widetilde{\nabla^{\ell-1} \mathbf{A}}) \bullet_j (\mathbf{R}^1 \bullet_{j-1} \widetilde{\mathbf{B}}) = (\widetilde{\nabla^{\ell-1} \mathbf{A}}) \bullet_j (\widetilde{\nabla}_1 \widetilde{\mathbf{B}})\end{aligned}$$

where we abbreviated $\mathbf{R}^1 = \mathbf{R}^1(\mathbf{ik})$. A repeated application of this proves the first assertion. For the second statement it is similar but using the associativity as given in Eq. (1).

Note, that the operator $\nabla^\ell$ is equivalent to the spherical gradient $\mathbf{R}^\ell(\nabla)$ defined by Weniger in [17]. Actually, the work of Weniger and ours are equivalent. But, our work focuses more on the fact that spherical derivatives connect spherical tensor fields of different ranks quite naturally. Another emphasize of our work in contrast to Weiniger's is that higher order spherical gradients $\mathbf{R}^\ell(\nabla)$ can be obtained by repeated applications of lower order ones, which is, from a computational perspective, quite important.

## *5.1 Spherical Gaussian Derivatives*

Our goal is to represent filter kernels as linear combinations of STDss of radial symmetric functions. Suppose that $g$ is an arbitrary radial functions, i.e. $g(\mathbf{r}) = g(\|\mathbf{r}\|)$. In fact, it holds in general that the angular part of STDs of the form $\nabla_i^n g$ are spherical harmonics of degree $n-i$. In particular we are interested in a very important radial function, the Gaussian function. In this section we show that the STDs of a Gaussian are just the Gaussian-windowed solid harmonics.

**Proposition 5.4** *We define the Gaussian windowed harmonic of width $\sigma$ as*

$$\mathbf{G}_\sigma^\ell(\mathbf{r}) := \frac{1}{(\sqrt{2\pi}\sigma)^3}\left(\frac{-1}{\sigma^2}\right)^\ell \mathbf{R}^\ell(\mathbf{r})e^{-\frac{r^2}{2\sigma^2}},$$

*then, the Fourier transformation of $\mathbf{G}^\ell(\mathbf{r})$ is given by*

$$\widetilde{\mathbf{G}}_\sigma^\ell(\mathbf{k}) = \langle e^{\mathbf{ik}^\top \mathbf{r}}, \mathbf{G}^\ell(\mathbf{r})\rangle_{L_2} = (\mathbf{i}k)^\ell \mathbf{Y}^\ell(\mathbf{k})e^{-\frac{(\sigma k)^2}{2}}.$$

*Proof.* We start with the definition of the Fourier transform and plug in the spherical harmonic expansion of the plane wave in terms of spherical Bessel function $j_n$ (see e.g [13], p. 136). Then, we integrate out the angular dependend part:

$$\begin{aligned}\int_{\mathbb{R}^3} \mathbf{G}_\sigma^\ell(\mathbf{r})e^{-\mathbf{ik}^\top\mathbf{r}}d\mathbf{r} &= \int_{\mathbb{R}^3} \mathbf{G}_\sigma^\ell(\mathbf{r})\sum_n (2n+1)(-\mathbf{i})^n\ j_n(kr)\ \mathbf{Y}^n(\mathbf{r}) \bullet_0 \mathbf{Y}^n(\mathbf{k}) \\ &= \sqrt{\frac{2}{\pi}}\frac{(\mathbf{i})^\ell}{\sigma^{2\ell+3}}\mathbf{Y}^\ell(\mathbf{k})\int_0^\infty j_\ell(kr)e^{-\frac{r^2}{2\sigma^2}}r^{\ell+2}dr\end{aligned}$$

The residual radius dependend part is integrated by using the series expansion of the spherical Bessel function:

$$\int_0^\infty j_\ell(kr)e^{-\frac{r^2}{2\sigma^2}}r^{2+\ell}dr = \sum_{n=0}^{\infty}\frac{(-1)^n k^{2n+\ell}}{2^n n!(2(n+\ell)+1))!!}\underbrace{\int_0^\infty r^{2n+2\ell+2}e^{-\frac{r^2}{2\sigma^2}}dr}_{\sqrt{\frac{\pi}{2}}\sigma^{2(n+\ell)+3}(2(n+\ell)+1)!!}$$
$$= \sqrt{\frac{\pi}{2}}\sum_{n=0}^{\infty}\frac{(-1)^n k^{2n+\ell}}{2^n n!\sigma^{-2(n+\ell)-3}} = \sqrt{\frac{\pi}{2}}\sigma^{2\ell+3}k^\ell e^{-\frac{(\sigma k)^2}{2}}$$

which proves the assertion.

In fact, for $\sigma = 1$ the $\mathbf{G}^\ell$s are eigenfunctions of the Fourier transformation with eigenvalue $(-\mathbf{i})^\ell$. Using the above proposition it is also easy to show that the $\mathbf{G}^\ell$ are just the $\ell$th order spherical derivatives of a Gaussian.

**Proposition 5.5 (Spherical Gaussian Derivative)** *The homogeneous spherical derivative $\nabla^\ell$ of a Gaussian computes to*

$$\nabla^\ell e^{-\frac{r^2}{2\sigma^2}} = (\sqrt{2\pi}\sigma)^3\mathbf{G}_\sigma^\ell(\mathbf{r}) = \left(-\frac{1}{\sigma^2}\right)^\ell \mathbf{R}^\ell(\mathbf{r})\, e^{-\frac{r^2}{2\sigma^2}}$$

*Proof.* An immediate consequence of the fact that $\widetilde{\nabla^\ell g}(\mathbf{k}) = \mathbf{R}^\ell(\mathbf{ik})\widetilde{g}(\mathbf{k})$ and Proposition 5.4.

## 6 local adaptive filtering with STDs

The basic idea of the following approach is to represent the filter kernel by a linear superposition of spherical Gaussian derivatives. This will enable us to formulate the filtering process by repeated applications of spherical derivatives which is much more efficient than the explicit convolutions used in the previous section. We have seen that the Gaussian derivatives are just Gaussian-windowed harmonic polynomials, so the resulting kernels will also be Gaussian windowed harmonics. There are many possibilities to construct such filter kernels. We present a kernel which can be imagined as a squeezed or stretched Gaussian. But actually, we restrict the expansion to derivatives of the form $\nabla^j g$. Due to the symmetry properties of the Gaussian the expansion will only contain even degree derivatives $\nabla^{2j} g$. We propose to use the following filter kernel

$$\mathbf{V}^{\mathbf{n}}(\mathbf{r}) = \sum_{j=0}^{\infty}\frac{\lambda^j}{(2j-1)!!}\,\mathbf{R}^{2j+\ell}(\mathbf{n})\bullet_\ell \nabla^{2j}g(\mathbf{r}) \tag{16}$$

where $\mathbf{n}$ is the squeezing/stretching direction. The expression $(2j-1)!!$ denotes the double factorial given by $(2n-1)(2n-3)\ldots 3$. The parameter $\ell \geq 0$ determines the rank of the filter kernel. The parameter $\lambda$ controls the shape. For $\lambda < 0$ the function has a tablet-like shape, for $\lambda > 0$ the shape is stick-like. Note that if the orientation parameter $\mathbf{n}$ is not normalized the magnitude $\|\mathbf{n}\|$ has the same effect on the shape

of the filter kernel like the magnitude of $\lambda$. So, we can control the orientation as well as the shape of the filter kernel by the single parameter $\mathbf{n}$. In Figure 2 we show surface plots of the filter kernel for $\ell = 0$ for different $\lambda$.

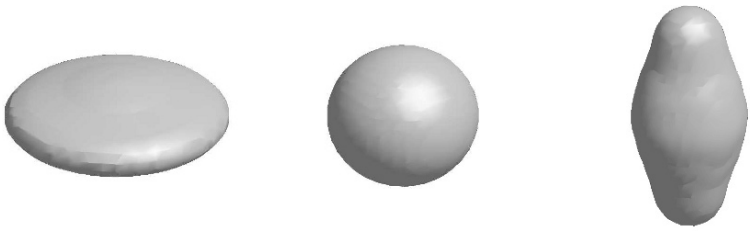


**Fig. 2** Surface-Plots for $\ell = 0$ with $\lambda = -0.2, 0, 0.2$.

## 6.1 The Convolution Integral

Again we have to compute the convolution integral as given in equation (8). Inserting the filter kernel as given in equation (16) into (8) yields

$$\begin{aligned}\mathbf{U}_{\text{conv}}(\mathbf{r}) &= \int_{\mathbb{R}^3} \sum_{j=0}^{\infty} \frac{\lambda^j}{(2j-1)!!} \underbrace{m(\mathbf{r}')\mathbf{R}^{2j+\ell}(\mathbf{n}(\mathbf{r}'))}_{\mathbf{E}^j(\mathbf{r}')} \bullet_\ell (\nabla^{2j} g)(\mathbf{r}-\mathbf{r}')d\mathbf{r}' \\ &= \sum_{j=0}^{\infty} \frac{\lambda^j}{(2j-1)!!} (\nabla^{2j} g) \widetilde{\bullet}_\ell \mathbf{E}^j \\ &= g * \sum_{j=0}^{\infty} \frac{\lambda^j}{(2j-1)!!} \nabla_{2j} \mathbf{E}^j\end{aligned}$$

where we used equation (14) to get from the second to the third line. The resulting approach is similar to the algorithm using the tensorial harmonics, but the convolutions with the basis functions are replaced by repeated differentiations.

## 6.2 The Correlation Integral

On the other hand consider the correlation integral. Starting with equation (9) and inserting expression (16) yields:

**Algorithm 3** Convolution Algorithm with STDs

**Input:** $m \in \mathscr{T}_0$, $\mathbf{n}(\mathbf{r}) \in \mathscr{T}_1$
**Output:** $\mathbf{U} \in \mathscr{T}_\ell$

Let $\mathbf{E}^0 := m \bullet_\ell \mathbf{R}^\ell(\mathbf{n})$
**for** $j = 1 : j_{\max}$ **do**
  $\mathbf{E}^j := \mathbf{E}^{j-1} \bullet_{2j+\ell} \mathbf{R}^2(\mathbf{n})$
**end for**
Let $\mathbf{U} := 0$
**for** $j = j_{\max} : -1 : 1$ **do**
  $\mathbf{U} := \frac{\lambda^j}{(2j-1)!!} \nabla_2 (\mathbf{U} + \mathbf{E}^j)$
**end for**
$\mathbf{U} := \mathbf{U} + \mathbf{E}^0$
$\mathbf{U} := g * \mathbf{U}$

$$\begin{aligned}\mathbf{U}_{\mathrm{corr}}(\mathbf{r}) &= \int_{\mathbb{R}^3} \sum_{j=0}^{\infty} \frac{\lambda^j}{(2j-1)!!} \underbrace{\mathbf{R}^{2j+\ell}(\mathbf{n}(\mathbf{r}))}_{\mathbf{N}^j(\mathbf{r})} \bullet_\ell (\nabla^{2j} g)(\mathbf{r}' - \mathbf{r}) m(\mathbf{r}')\, d\mathbf{r}' \\ &= \sum_{j=0}^{\infty} \frac{\lambda^j}{(2j-1)!!} \mathbf{N}^j(\mathbf{r}) \bullet_\ell \nabla^{2j} (m * g)\end{aligned}$$

where we used equation (15) to pull the differentiation outward. In Algorithm 4 we depict the computation process. In contrast to the convolution integral everything can be computed in place. We just need one loop for the whole process, hence, the memory consumption is much lower as for the convolution algorithm.

**Algorithm 4** Correlation Algorithm with STDs

**Input:** $m \in \mathscr{T}_0$, $\mathbf{n}(\mathbf{r}) \in \mathscr{T}_1$
**Output:** $\mathbf{U} \in \mathscr{T}_\ell$

Let $\mathbf{N} := \mathbf{R}^\ell(\mathbf{n})$
Let $\mathbf{M} := m * g$
Let $\mathbf{U} := \mathbf{N} \bullet_\ell \mathbf{M}$
**for** $j = 1 : j_{\max}$ **do**
  $\mathbf{N} := \mathbf{N} \bullet_{2j+\ell} \mathbf{R}^2(\mathbf{n})$
  $\mathbf{M} := \nabla^2 \mathbf{M}$
  $\mathbf{U} := \mathbf{U} + \frac{\lambda^j}{(2j-1)!!} \mathbf{N} \bullet_\ell \mathbf{M}$
**end for**

### 6.3 Application to Anisotropic Blurring

Finally we want to use the algorithm proposed in the last section to denoise scalar MRI data while preserving edges and surfaces, that is, we apply Algorithm 4 with $\ell = 0$. The idea is to perform a blurring operation isotropically in isotropic regions and anisotropically in anisotropic regions. As a measure anisotropy we use the gradient normalized with the local standard deviation. We choose $\lambda < 0$ such that the filter kernel has a tablet-like shape. This tablet shape is for each voxel oriented orthogonal to the observed gradient such that the smoothing is not performed across the edges. In conclusion we choose the orientation/shape parameter $\mathbf{n}$ as

$$\mathbf{n} = \frac{\nabla^1(m * g)}{\varepsilon + \sqrt{m^2 * g - (m * g)^2}}$$

where $\varepsilon > 0$ is a small regularization parameter avoiding zero divisions. In Figure 3 we show an example applied to MRI-data of a human head of size $256^3$. Obviously, the algorithm works, the isotropic regions are smoothed well and the edges are kept. The run time on a standard PC (Intel Pentium 2.2 Ghz) is about 15 seconds.

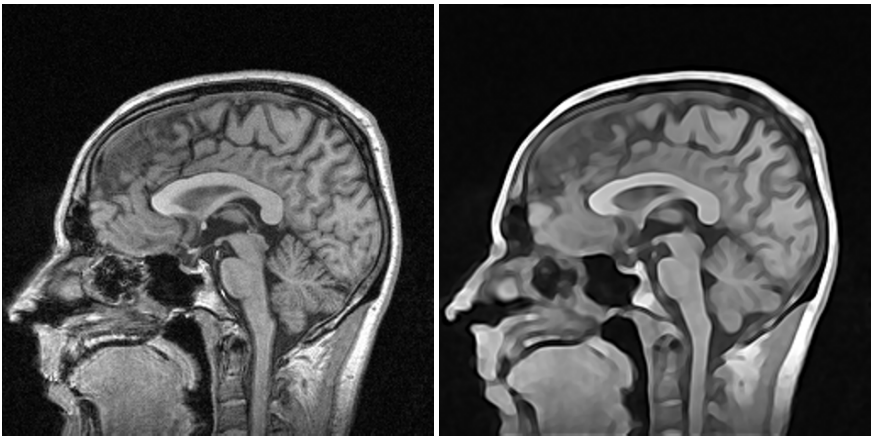


**Fig. 3** Example of anisotropic blurring filter on MRI-data.

## 7 Conclusion

In this chapter we proposed a mathematical framework for the efficient handling of nonlinear filters in 3D. The framework is based on the notion of the spherical tensor that is characterized by its transformation behavior with respect the 3D rotation

group. The advantage of this representation in contrast to the usual cartesian one is that even high rank tensors can be rotated in a quite efficient way.

Based on this framework we introduced so called tensorial harmonics that provide an orthogonal and complete basis of spherical tensor fields of any rank. Secondly, we proposed a differentiation scheme that connects spherical tensor fields of different degrees. These tools enabled us to implement local adaptive filters, e.g. anisotropic smoothing filters, in an efficient way.

## Appendix

### *Spherical Harmonics*

We always use Racah-normalized spherical harmonics. In terms of Legendre polynomials they are written as

$$Y_m^\ell(\phi,\theta) = \sqrt{\frac{(l-m)!}{(l+m)!}} P_m^\ell(\cos(\theta)) e^{\mathbf{i}m\phi}$$

We always write $\mathbf{r} \in S^2$ instead of $(\phi,\theta)$. The Racah-normalized solid harmonics can be written as

$$R_m^\ell(\mathbf{r}) = \sqrt{(\ell+m)!(\ell-m)!} \sum_{i,j,k} \frac{\delta_{i+j+k,\ell}\delta_{i-j,m}}{i!j!k!2^i 2^j} (x-\mathbf{i}y)^j (-x-\mathbf{i}y)^i z^k,$$

where $\mathbf{r} = (x,y,z)$. They are related to spherical harmonics by $R_m^\ell(\mathbf{r})/r^\ell = Y_m^\ell(\mathbf{r})$

### *Clebsch-Gordan Coefficients*

For the computation of the Clebsch-Gordan (CG) coefficients recursive formulas are applied (see e.g. [1]). The important orthogonality-relations of the CG-coefficients are

$$\sum_{j,m} \langle jm|j_1m_1, j_2m_2\rangle \langle jm|j_1m_1', j_2m_2'\rangle = \delta_{m_1,m_1'}\delta_{m_2,m_2'} \tag{17}$$

$$\sum_{m=m_1+m_2} \langle jm|j_1m_1, j_2m_2\rangle \langle j'm'|j_1m_1, j_2m_2\rangle = \delta_{j,j'}\delta_{m,m'} \tag{18}$$

$$\sum_{m_1,m} \langle jm|j_1m_1, j_2m_2\rangle \langle jm|j_1m_1, j_2'm_2'\rangle = \frac{2j+1}{2j_2'+1}\delta_{j_2,j_2'}\delta_{m_2,m_2'} \tag{19}$$

For two special cases there are explicit formulas:

$$\langle \ell m|(\ell-\lambda)(m-\mu),\lambda\mu\rangle = \binom{\ell+m}{\lambda+\mu}^{1/2}\binom{\ell-m}{\lambda-\mu}^{1/2}\binom{2\ell}{2\lambda}^{-1/2} \tag{20}$$

$$\begin{aligned}\langle \ell m|(\ell+\lambda)(m-\mu),\lambda\mu\rangle = (-1)^{\lambda+\mu}&\binom{\ell+\lambda-m+\mu}{\lambda+\mu}^{1/2}\\ &\binom{\ell+\lambda+m-\mu}{\lambda-\mu}^{1/2}\binom{2\ell+2\lambda+1}{2\lambda}^{-1/2}\end{aligned} \tag{21}$$

The CG-coefficients fulfill certain symmetry relations

$$\langle jm|j_1m_1,j_2m_2\rangle = \langle j_1m_1,j_2m_2|jm\rangle \tag{22}$$
$$\langle jm|j_1m_1,j_2m_2\rangle = (-1)^{j+j_1+j_2}\langle jm|j_2m_2,j_1m_1\rangle \tag{23}$$
$$\langle jm|j_1m_1,j_2m_2\rangle = (-1)^{j+j_1+j_2}\langle j(-m)|j_1(-m_1),j_2(-m_2)\rangle \tag{24}$$

## *Wigner D-Matrix*

The components of $\mathbf{D}_g^\ell$ are written $D_{mn}^\ell$. In Euler angles $\phi,\theta,\psi$ in ZYZ-convention we have

$$D_{mn}^\ell(\phi,\theta,\psi) = e^{\mathbf{i}m\phi}d_{mn}^\ell(\theta)e^{\mathbf{i}n\psi},$$

where $d_{mn}^\ell(\theta)$ is the Wigner d-matrix which is real-valued. Explicit formulas for the $d_{mn}^\ell(\theta)$ involve the Jacobi-polynomials (see e.g. [13]) The important relations to the Clebsch-Gordan coefficients are:

$$D_{mn}^\ell = \sum_{\substack{m_1+m_2=m\\ n_1+n_2=n}} D_{m_1n_1}^{\ell_1}D_{m_2n_2}^{\ell_2}\langle lm|l_1m_1,l_2m_2\rangle\langle ln|l_1n_1,l_2n_2\rangle \tag{25}$$

and

$$D_{m_1n_1}^{\ell_1}D_{m_2n_2}^{\ell_2} = \sum_{l,m,n} D_{mn}^\ell\langle lm|l_1m_1,l_2m_2\rangle\langle ln|l_1n_1,l_2n_2\rangle. \tag{26}$$

## References


1. M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1972.
2. D.M. Brink and G.R. Satchler. *Angular Momentum*. Oxford Science Publications, 1993.
3. Erik Franken, Markus van Almsick, Peter Rongen, Luc Florack, and Bart ter Haar Romeny. An efficient method for tensor voting using steerable filters. In *Proceedings of the ECCV 2006*, pages 228–240. Lecture Notes in Computer Science, Springer, 2006.
4. W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(9):891–906, 1991.

5. J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Trans. Image Processing*, 12(8):938–943, 2003.
6. G. Guy and G. Medioni. Inferring global perceptual contours from local features. *International Journal of Computer Vision*, 20(1):113–133, 1996.
7. W. Miller, R. Blahut, and C. Wilcox. Topics in harmonic analysis with applications to radar and sonar. *IMA Volumes in Mathematics and its Applications, Springer-Verlag, New York*, 1991.
8. Philippos Mordohai. *Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning*. Morgan and Claypool, ISBN-10: 1598291009, 2006.
9. P. Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(5):488 – 499, 1995.
10. M. Reisert and H. Burkhardt. Complex derivative filters. *IEEE Trans. Image Processing, (to appear)*, 2008.
11. M. Reisert and H. Burkhardt. Equivariant holomorphic filters for contour denoising and rapid object detection. *IEEE Trans. Image Processing*, 17(2), 2008.
12. M. Reisert and H.Burkhardt. Tensorial harmonics for efficient tensor voting in 3d. In *Proceedings of the CVPR'08*. IEEE Computer Society, 2008.
13. M.E. Rose. *Elementary Theory of Angular Momentum*. Dover Publications, 1995.
14. M. Tinkham. *Group Theory in Quantum Mechanics*. Dover Publications, 2004.
15. J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, Universität Kaiserslautern, January 1996.
16. U. Weinert. Spherical tensor representation. *Journal Archive for Rational Mechanics and Analysis, Physics and Astronomy*, pages 165–196, 1980.
17. Ernst Joachim Weniger. The spherical tensor gradient operator. *arXiv.org:math-ph/0505018*, 2005.
18. P. Wormer. Angular momentum theory. *Lecture Notes - University of Nijmegen Toernooiveld, 6525 ED Nijmegen, The Netherlands*.
19. G.Z. Yang, P.Burger, D.N. Firmin, and S.R. Underwood. Structure adaptive anisotropic image filtering. *Image and Vision Computing*, 14:135–145, 1996.

# On Geometric Transformations of Local Structure Tensors

Björn Svensson, Anders Brun,
Mats Andersson, and Hans Knutsson

## 1 Introduction

The structure of images has been studied for decades and the use of local structure tensor fields appeared during the eighties [3, 14, 6, 9, 11]. Since then numerous varieties of tensors and estimation schemes have been developed. Tensors have for instance been used to represent orientation [7], velocity, curvature [2] and diffusion [19] with applications to adaptive filtering [8], motion analysis [10] and segmentation [17].

Even though sampling in non-Cartesian coordinate system are common, analysis and processing of local structure tensor fields in such systems is less developed. Previous work on local structure in non-Cartesian coordinate systems include [21, 16, 1, 18]. A possible explanation is that it sometimes is prohibitively difficult or not necessary to take into account the geometry, e.g. in the following rather common cases:

- The geometry of the image is not known or very complicated. A photograph is for instance a projection onto an image-plane. For a complete description of the geometry it would then be necessary to know the distance to the image plane and every object visible in the scene.
- The application does not require knowledge about the correct geometry or the correct geometry does not contribute in a useful way to the observer.

Björn Svensson, e-mail: bjosv@imt.liu.se
Department of Biomedical Engineering, CMIV, LiU, Linköping, Sweden

Anders Brun, e-mail: anders@cb.uu.se
Centre for Image Analysis, SLU, Uppsala, Sweden.

Mats Andersson, e-mail: matsa@imt.liu.se
Department of Biomedical Engineering, CMIV, LiU, Linköping, Sweden

Hans Knutsson, e-mail: knutte@imt.liu.se
Department of Biomedical Engineering, CMIV, LiU, Linköping, Sweden

- The image has already a correct geometry or resampling to a correct geometry can be performed with negligible effects on the result.

We address cases where none of the above statements apply. In for instance medical imaging, geometry is very important and abnormalities can be detected by size, shape and location. It is also common that samples are acquired in a non-Cartesian coordinate system. Non-cubic voxels are perhaps the most common example, but more complex geometries are also utilized such as oblique sampling or samples acquired in a polar coordinate system. Signals acquired from such an imaging device are generally resampled in order to present a geometrically correct image. Computerized analysis of the signal such as localization of objects or estimation of shape and size does not, however, require resampling. In fact, resampling often complicates subsequent processing, since signal and noise characteristics are deteriorated. This reasoning applies to local structure in particular, since it is a feature which relates to geometry.

The research presented here reviews the concept of local structure and orientation using basic differential geometry. This is relevant, since most existing theory on this topic implicitly assumes an underlying Cartesian geometry. The concept of local structure can be generalized to arbitrary dimension. But since this work concerns geometric transformations, it is not meaningful to increase the dimensionality beyond the spatial dimensions.

## 2 Background

In non-Cartesian coordinate systems, it is important to distinguish between contravariant and covariant tensors, i.e. if the tensor is expressed in meter or per meter. To introduce covariant and contravariant tensors let us start out by studying vectors, i.e. tensors of order 1. A vector $v$ is an element in a $d$-dimensional vector space $V$, which is spanned by its natural basis $\frac{\partial}{\partial x^i}$. More compactly we write $v \in V$ and denote its coordinates by $v^i$ with an upper index, where $i = 1,\ldots,d$. The vector is said to be a tensor of order $(1,0)$, i.e. a contravariant first order tensor. As the order increases it is convenient to use the Einstein summation convention, which means that indices occurring more than once are implicitly summed over, i.e. the vector $v$ is written as

$$v = v^1 \frac{\partial}{\partial x^1} + v^2 \frac{\partial}{\partial x^2} + \cdots + v^d \frac{\partial}{\partial x^d} = v^i \frac{\partial}{\partial x^i}. \tag{1}$$

A dual vector $w$ belongs to the $d$-dimensional space $V^*$, which is the dual space of $V$. The dual space $V^*$ is then spanned by the basis $dx^i$. Thus, $w$ is a covariant tensor of order $(0,1)$ and its coordinates $w_i$ is written with a lower index. Using the same convention $w$ is written as

$$w = w_1 dx^1 + w_2 dx^2 + \cdots + w_d x^d = w_i dx^i. \tag{2}$$

In the same way a contravariant second order tensor $S \in V \otimes V$ has coordinates $S^{ij}$. It is said to be of order $(2,0)$ with two upper indices and we write

$$S = \sum_{i,j} S^{ij} \frac{\partial}{\partial x^i} \frac{\partial}{\partial x^j} = S^{ij} \frac{\partial}{\partial x^i} \frac{\partial}{\partial x^j}. \tag{3}$$

Moreover, a covariant tensor $T \in V^* \otimes V^*$ has coordinates $T_{ij}$ and is of order $(0,2)$ with two lower indices. $T$ is written as

$$T = \sum_{i,j} T_{ij} dx^i dx^j = T_{ij} dx^i dx^j. \tag{4}$$

The metric tensor $g$ defines an inner product in $V$ and is a covariant second order tensor, with components $g_{ij}$, i.e.

$$g(u,v) = \langle u,v \rangle = g_{ij} u^i v^j. \tag{5}$$

The metric makes it possible to identify elements in $V^*$ with elements in $V$ as follows

$$w = g(\cdot,v) = \langle \cdot,v \rangle = w_i dx^i. \tag{6}$$

Such operations are called index gymastics and allows us to move between covariant and contravariant tensors. The components $w_i$ of a covariant vector is obtained by lowering an index, i.e.

$$w_i = g_{ij} v^j, \tag{7}$$

where $v^i$ are components of a contravariant vector. Since the metric tensor in the Euclidean space is the Kronecker delta, i.e. $g_{ij} = \delta_{ij}$, covariant and contravariant tensors are equivalent in a Cartesian coordinate system. Such tensors are called Cartesian tensors.

## 3 Tensor Transformations

Before deriving how different tensors transform let us study how we can draw tensors of different types in the *same* space. Although a contravariant vector $v \in V$ and covariant vector $w \in V^*$ belongs to different spaces it is of interest to interpret them in terms of glyphs in $V$.

A contravariant vector are drawn as an arrow in $V$ as we are used to. Without knowledge about the metric we can draw covariant tensors of the first and second order as the iso-curves defined formed by the ensemble of contravariant vectors which satisfy the equations:

$$w_i v^i = 1 \tag{8}$$

$$T_{ij} v^i v^j = 1 \tag{9}$$

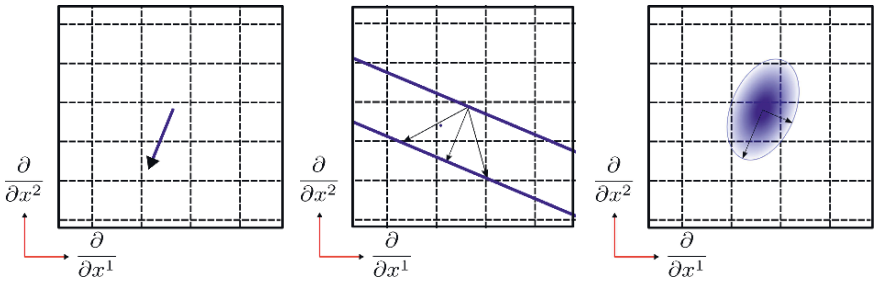



**Fig. 1** *The contravariant vector (***left***) is drawn as we are used to. Suppose that its components are $v^i$ are expressed in meter. A dual vector (***middle*** ) with components $w_1$, $w_2$ are then expressed per meter. Subsequently, the measure $w_i v^i$ has no unit and we can form the iso-curve of all covariant vectors v such that $w_i v^i = r$, where r is a constant. The components $T_{ij}$ of a second order covariant tensor (***right***) are expressed per square meter, i.e. the iso-curve formed by the vectors v satisfying $T_{ij} v^i v^j = r$ then represent $T$.*

Note that the glyphs in Fig. 1 are obtained without knowledge about the metric, i.e. they are by construction invariant to a change of coordinate system. The glyph used for the second order tensor presumes $T$ to be symmetric and positive semidefinite. In the same fundamental way as a contravariant vector of length 1 meter does not change direction or length if it is expressed in inches, a tensor $S$ must be preserved even if the coordinate system is changed. Let us study the tensor

$$S = \tilde{S}^{ij} \frac{\partial}{\partial \tilde{x}^i} \frac{\partial}{\partial \tilde{x}^j} = S^{ij} \frac{\partial}{\partial x^i} \frac{\partial}{\partial x^j}, \tag{10}$$

where the coordinate system is changed from $x$ to $\tilde{x}$. This implies new tensor components $\tilde{S}^{ij}$ expressed in a new basis $\frac{\partial}{\partial \tilde{x}^i}$. The transformation behavior for the second order contravariant tensor $S$ is then easily derived by the chain-rule:

$$\tilde{S}^{ij} = \frac{\partial \tilde{x}^i}{\partial x^k} \frac{\partial \tilde{x}^j}{\partial x^l} S^{kl} \tag{11}$$

A covariant second order tensor $T$ expressed in the new coordinate system can then be written as

$$T = \tilde{T}_{ij} d\tilde{x}^i d\tilde{x}^j = T_{ij} dx^i dx^j. \tag{12}$$

The relation between the elements $T_{ij}$ and $\tilde{T}_{ij}$ is in the same way as before derived by the chain rule, i.e.

$$\tilde{T}_{ij} = \frac{\partial x^k}{\partial \tilde{x}^i} \frac{\partial x^l}{\partial \tilde{x}^j} T_{kl}. \tag{13}$$

Note how a uniform stretch of the coordinate system, i.e. the transformation $\tilde{x}^i = a x^k$ with $a > 1$, causes covariant tensor components $T_{ij}$ to shrink whereas contravariant components $S^{ij}$ grow. We can draw the tensors from Fig. 1 in non-Cartesian coordinate system as illustrated in Fig. 2. With the grid-lines as a guide we see that the

components $v^i$, $w_i$ and $T_{ij}$ are altered, whereas the shape of the glyphs are maintained.

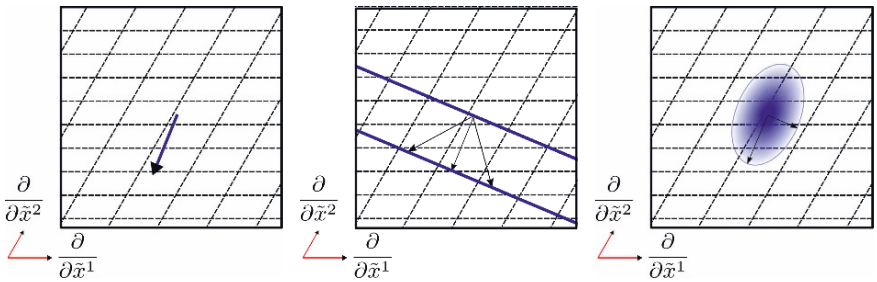



**Fig. 2** *This non-Cartesian coordinate system has a metric $\tilde{g}_{ij} \neq \delta_{ij}$. A change of coordinate system does not change the geometric objects, e.g. the contravariant vector (***left***) is preserved although its components $\tilde{v}^i$ are larger compared to $v^i$ in Fig. 1. Its length can be calculated by $\|v\| = \tilde{g}_{ij}\tilde{v}^i\tilde{v}^j$ using the proper metric. As opposed to the contravariant vector the components $\tilde{w}_i$ of the covariant vector (***middle***) are smaller, which causes the components $\tilde{v}^i$ forming the iso-curve to increase in order to satisfy $\tilde{w}_i\tilde{v}^i = r$. The same reasoning holds for the ensemble of contravariant vectors that form the glyph for the second order covariant tensor (***right***).*

A general tensor of order $(p,q)$, with $p > 0$ and $q > 0$ is called a mixed tensor and has $p$ contravariant indices and $q$ covariant indices. The components of $T$ are $T^{i_1,i_2,\ldots i_p}{}_{j_1,j_2,\ldots,j_q}$. By using the metric tensor we can move between different types of tensors, e.g. mixed second order tensors can be obtained by index gymnastics

$$S_i{}^j = \sum_k g_{ik}S^{kj} = g_{ik}S^{kj}, \tag{14}$$

$$T^i{}_j = \sum_k g^{ik}T_{kj} = g^{ik}T_{kj}, \tag{15}$$

where the contravariant metric tensor $g^{ij} = (g_{ij})^{-1}$. The transformation of a general tensor is derived by the chain rule in the same way as for the second order. Note that the components $T^i{}_j$ of a mixed second order tensor is preserved for a uniform stretch of the coordinate system. By decomposing the second order mixed tensor

$$T(w,\cdot) = T^i{}_j w^j = (u^i u_j + v^i v_j)w^j = g(u,w)u + g(v,w)v, \tag{16}$$

we see that a mixed tensor, here in 2D, can be interpreted as a projection operator.

## 4 Local Structure Tensor Fields

A local structure tensor field, as shown in Fig. 3, describes how the signal energy is locally distributed over orientation, i.e. it is a measure of signal energy per square

meter. This indicate that the tensor is a second order covariant tensor. For the sake of simplicity let us study the local structure tensor formed by the outer product of the gradient, i.e.

$$T = \nabla f \otimes \nabla f = \frac{\partial f}{\partial x^i}\frac{\partial f}{\partial x^j} dx^i dx^j, \tag{17}$$

The tensor $T$ is then a second order covariant tensor $T \in V^* \otimes V^*$ with components

$$T_{ij} = \frac{\partial f}{\partial x^i}\frac{\partial f}{\partial x^j}. \tag{18}$$

By construction this tensor has only one eigenvalue, which is a drawback of this estimate since it says nothing about the energy distribution in directions orthogonal to $\nabla f$. Moreover, interesting feature points vanish when $\nabla f = 0$. In practice spatial averaging is used to reduce these effects, i.e. the tensor components are element-wise filtered with an isotropic low-pass filter. Smoothing the tensor in this way implies that more than one eigenvalue may deviate from zero and for points where $\nabla f = 0$, the neighboring tensors will determine the estimate of $T$.

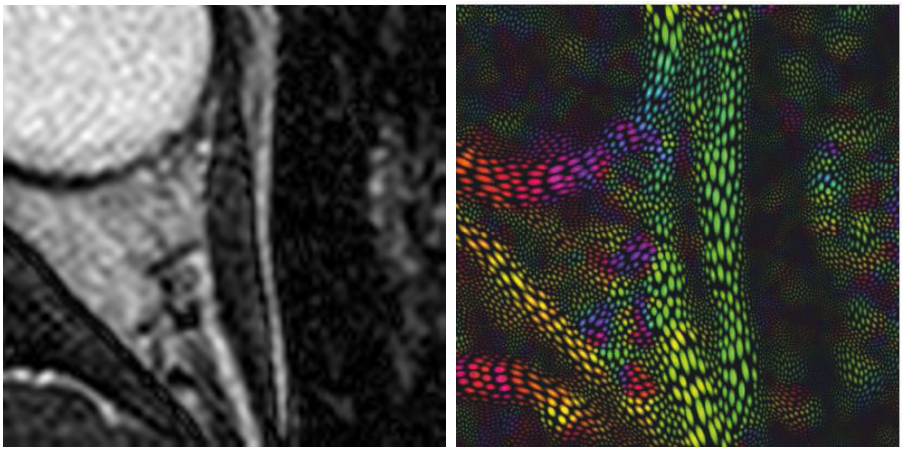


**Fig. 3** *A local structure tensor field describes the distribution of signal energy locally in an image. The close up tensor field (***right***) shown, is estimated from a T1-weighted magnetic resonance image (***left***). The glyph shown is the solution to $T_{ij}v^i v^j = r$, which means that the local energy is low along the major axis and high along the minor axis of an anisotropic glyph. To increase the visibility of tensors which describe a high signal energy the radius r is increasing with $\|T\|$, i.e. each tensor is uniformly scaled with r. The color represent the orientation of the major axis.*

There are more sophisticated ways in which local structure can be estimated based on higher order derivatives or Riesz-transforms [7, 4, 15, 5]. An example of a straightforward extension to the outer product of gradients is given in (19). Such estimates do not require spatial averaging and can be made more spatially localized [13]. This means that the resolution of the tensor field is better preserved in comparison to spatial averaging of the tensor in (17). Estimating local structure

using Riesz-transforms is based on the underlying assumption that local phase is an important feature and elegantly removes the sensitivity to vanishing feature points mentioned earlier.

$$T = \nabla f \otimes \nabla f - f\,(\nabla \otimes \nabla) f = \frac{\partial f}{\partial x^i}\frac{\partial f}{\partial x^j} dx^i dx^j - f\frac{\partial^2 f}{\partial x^i x^j} dx^i dx^j \tag{19}$$

It is easy to see that this construction also is a covariant second order tensor $T \in V^* \otimes V^*$. In fact, all local structure tensors that contain the outer product of the gradient must be of order $(0,2)$ to fulfill the fundamental requirements of a tensor [20], e.g. adding two geometric objects from different spaces is not a well-defined operation. The same argument applies also to estimates based on Riesz-transforms.

In general, local structure tensors are formed by combining responses from linear filters as in the previous chapter. Since these filters are discrete operators, they can be applied to arbitrary arrays of sampled data. But without taking into account the sampling distances, i.e. the local metric, this operation is carried out under the implicit assumption that the samples are acquired in a Cartesian coordinate system. A straightforward example is the use of central differences, which approximate the signal gradient. The components of the gradient is then approximated by

$$\begin{aligned} \frac{\partial f}{\partial x^1} &\approx \frac{f(x^1 + \Delta x^1, x^2) - f(x^1 - \Delta x^1, x^2)}{2\Delta x^1} \\ \frac{\partial f}{\partial x^2} &\approx \frac{f(x^1, x^2 + \Delta x^2) - f(x^1, x^2 - \Delta x^2)}{2\Delta x^2}, \end{aligned} \tag{20}$$

where $\Delta x^1$ and $\Delta x^2$ are the sampling distances. Obviously, if $\Delta x^1 \neq \Delta x^2$, or more generally if the metric $g_{ij} \neq \delta_{ij}$ the filters must be adapted to the underlying geometry in order to produce a geometrically correct gradient.

To achieve rotational invariance the filters which produce local structure tensors are required to be polar separable, which is not the case for (20). Such filters can be decomposed into a radial band-pass filter followed by a directional filter which is sensitive to signal orientation. In a Cartesian coordinate system, the radial band-pass function is spherically symmetric. In multi-dimensional differentiation this is often referred to as a pre-filter and can be seen as a regularization to avoid amplification of high-frequency noise. Another interpretation is that the gradient is averaged over a small signal neighborhood and by letting radius of this neighborhood tend to zero the true gradient is obtained. It is however important to realize that the shape of this neighborhood becomes anisotropic in a warped coordinate system. Also the direction might change, which in the context of local structure is less critical since this may be corrected for when combining the filter responses.

Since the filters are discrete operators, aliasing must be taken into account and it is not straightforward to reshape the spatial support of the filters in a way which correspond to the anisotropic neighborhood. The approach used here is to formulate the desired operator in the continuous Fourier domain and then design an operator in the non-Cartesian grid which has the closest fit possible to this desired frequency response [18]. For practical reasons this approach was limited for non-Cartesian co-

ordinate systems which locally are well described by an affine transformation. This is because the the Shannon sampling theorem is still applicable for such transformations.

## 5 Local Structure Analysis

Since the tensor originally was designed to represent signal orientation, this is perhaps the most obvious application. The dominant signal orientation, i.e. the direction of maximal detected signal energy is obtained by finding the vector $v$ of unit length which maximizes $T_{ij}v^i v^j$, i.e.

$$\max_v T_{ij}v^i v^j, \text{ subject to } g_{ij}v^i v^j = 1. \tag{21}$$

It is easy to see the resemblance with the direction of maximal variance, which implies that the local structure tensor can be interpreted as covariance estimate of a local signal neighborhood. The dominant orientation is then the eigenvector corresponding to the largest eigenvalue, after solving the eigenvalue equation

$$T^i{}_j v^j = g^{ik}T_{kj}v^j = \sum_{j,k} g^{ik}T_{kj}v^j = \lambda v^i. \tag{22}$$

Note that eigenvalue decomposition of a second order tensor is only defined for mixed tensors, i.e. if the tensor is not mixed, eigenvalue decomposition can not be performed without a metric. With a metric $g^{ij}$ equal to the identity operator, i.e. a Cartesian coordinate system, the standard eigenvalue equation is obtained.

Since $T$ is symmetric we see that if $v$ maximize (21) then so does $-v$, i.e. we can not distinguish between between $v$ and $-v$. This was one of the the main motivations for the representation known as the orientation tensor, which is a continuous representation of orientation that maps $v$ and $-v$ to the same geometric object [12]. This is accomplished by forming the outer product, e.g. the contravariant tensor with components

$$T^{ij} = \lambda v^i v^j. \tag{23}$$

However, since the metric is known the same tensor can equivalentely be represented as a mixed order tensor with components

$$T^i{}_j = \lambda v^i v_j, \tag{24}$$

or as the covariant tensor

$$T_{ij} = \lambda v_i v_j. \tag{25}$$

The orientation tensor is defined only for signals $f$ which are intrinsically one-dimensional. Such signals satisfy the simple signal constraint

$$f(x) = f(\langle x, v\rangle v), \tag{26}$$

i.e. signals which locally are constant in directions orthogonal to $v$. This definition is invariant to the choice of coordinate system but require knowledge of the inner product. Actually, the simple signal constraint in itself also defines the orientation tensor, since

$$f(x) = f(\langle x, v\rangle v) = f(Tx), \tag{27}$$

where $T$ is a projection operator. From the simple signal constraint the orientation tensor $T$ is then defined by

$$T = \langle \cdot, v\rangle \otimes v = w \otimes v, \tag{28}$$

where $w = \langle \cdot, v\rangle$ by definition is the dual vector of $v$. Therefore, the orientation tensor is most naturally described as a mixed second order tensor, but since it is not meaningful to talk about orientation without a metric we are free to raise or lower the indices as we like.

Now consider a signal sampled in a non-Cartesian coordinate system. Of practical reasons it is often convenient to present and look at the signal samples as if the pixels were of quadratic size. This implies that the image is warped to an incorrect geometry. Fig. 4 illustrate the same tensors that was used in the previous examples, but now the entire image is warped such that the basis $\frac{\partial}{\partial \tilde{x}^i}$ appears to be orthonormal. How data is presented is of course irrelevant for analysis, but Fig. 4 illustrates what happens if the metric, which compensates for this geometric distortion, is neglected.

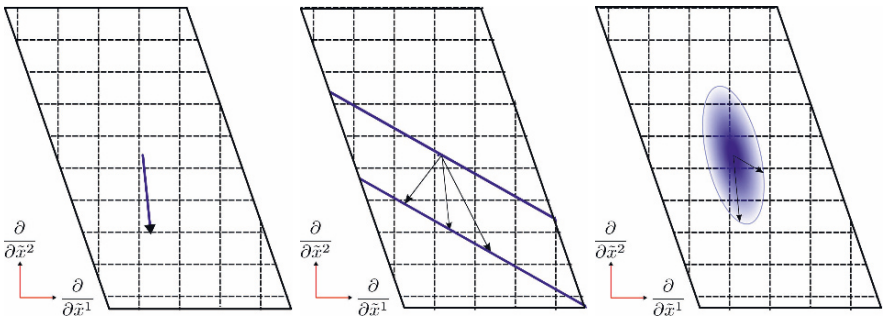



**Fig. 4** *With the data presented and the incorrect geometry shown, it is tempting to make the assumption that $\tilde{g}_{ij} = \delta_{ij}$. This implies that computations are carried out in an image with warped geometry. The length of the contravariant vector (***left***) is altered and the covariant vector (***middle***) is altered in the opposite way. Disregarding from the metric when performing eigenvalue decomposition of a second order tensor (***right***) in a non-Cartesian coordinate system implies that eigenvectors are not orthogonal and that the order of eigenvalues might change.*

From eigenvalue decomposition of the local structure tensor, various shape measures can be extracted. For instance the tensor carries information about the degree of anisotropy, how plane-like and how line-like a signal neighborhood is. Such measures are in general based on the tensor eigenvalues. The eigenvalues obtained from solving the eigenvalue equation are invariant to a change of coordinate system as

opposed to the eigenvectors. The eigenvectors obtained by solving the eigenvalue equation in (22) are contravariant vectors and will follow the transformation rules of such vectors. Consequently, shape measures based on the eigenvalues such as fractional anisotropy are invariant to a change of coordinate system. The color-code used in Fig. 3 is a good example of this.

The color-code used in Fig. 3, 7 and 8 consists of two parts. The color is a one-to-one mapping of the double angle representation, i.e. a vector $u$ with coordinates $u^i$ determined by the eigenvector $v$ corresponding to the largest eigenvalue $\lambda$. The color is then scaled by the intensity, which represent the degree of anisotropy $\alpha$ i.e.

$$\begin{aligned} \alpha &= \lambda - \mu, \\ (u^1, u^2) &= \Big( (v^1 + v^2)(v^1 - v^2), 2v^1 v^2 \Big), \end{aligned} \tag{29}$$

where $\alpha$ is the difference between the two eigenvalues $\lambda$ and $\mu$. Since the intensity only depends on the eigenvalues it is invariant to a change of coordinate system, whereas the color based on $u$ must be transformed as a contravariant vector.

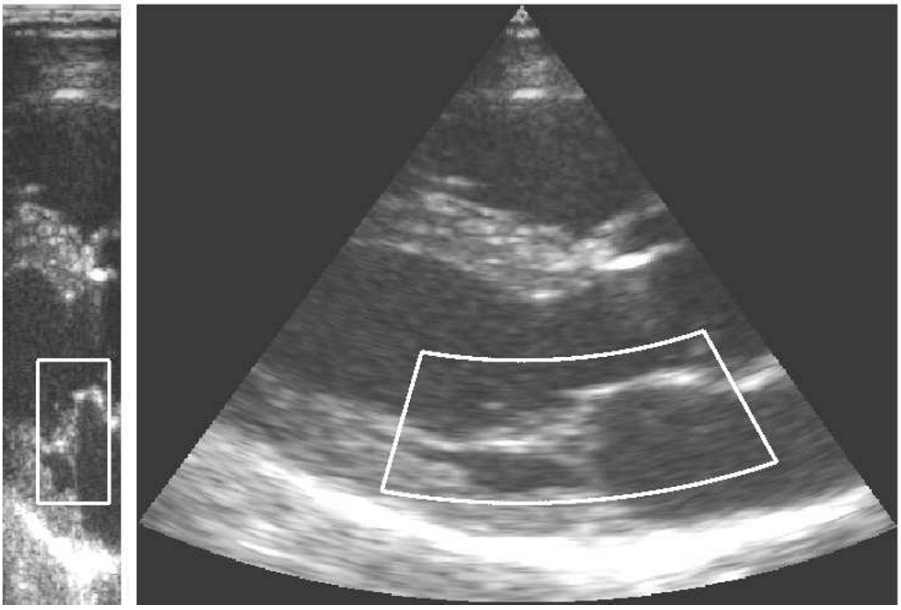


**Fig. 5 Left:** *The acquired samples of an ultrasound image displayed as quadratic pixels, the image presented is warped to an incorrect geometry.* **Right:** *The ultrasound image resampled to a correct geometry.*

## 6 Experiments

Samples from ultrasound imaging are acquired in a polar coordinate system. The example in Fig. 5 shows a human heart. The radial sampling distance is $\Delta r = 0.31$mm and the angular sampling distance is $\Delta\varphi = 0.72^\circ$. Bilinear interpolation was used to resample the acquired samples to a geometrically correct image. The size of this image is 128mm $\times$129mm with a sampling distance $\Delta x = \Delta y = 0.31$mm. Displaying the samples as an image with pixels of quadratic size implies the coordinate system in (30) is used, where the constant $c$ is the ratio between the radial and angular sampling distance $c = \frac{\Delta r}{\Delta\varphi}$.

$$\begin{aligned} x^1 &= r(x,y) = \sqrt{x^2+y^2} \\ x^2 &= c\,\varphi(x,y) = c\,\arctan(\frac{y}{x}) \\ \tilde{x}^1 &= x(r,\varphi) = r\cos\varphi \\ \tilde{x}^2 &= y(r,\varphi) = r\sin\varphi \end{aligned} \tag{30}$$

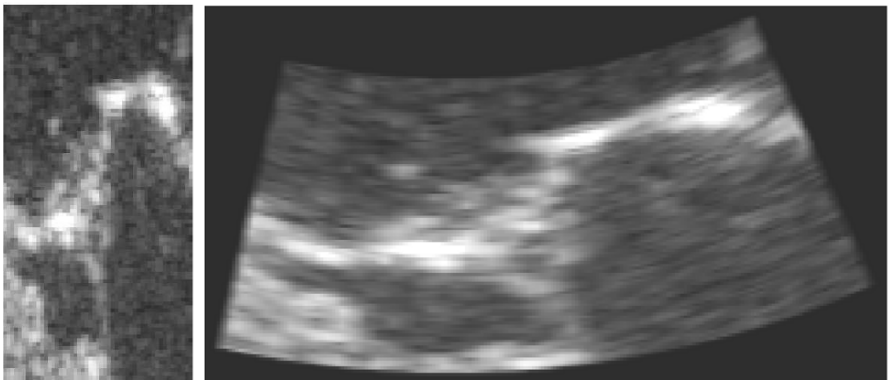


**Fig. 6** *The close-up marked in Fig. 5 in a non-Cartesian coordinate systam (***left***) and in a Cartesian coordinate system (***right***).*

In this case there are two main reasons for estimating local structure in the original grid instead of resampling the image prior to feature extraction. Firstly, the original sample grid contains five times less samples. Consequently analysis made in this coordinate system is much faster due to fewer samples. Secondly, resampling introduce a bias in to the tensor field, i.e. the degree of anisotropy in the angular direction are overestimated. By studying the close up of the resampled image in Fig. 6 it should be fairly obvious that it is difficult to obtain an unbiased estimate of local structure in the lower parts of the image.

As mentioned in section 4 and illustrated in Fig. 4 we can estimate and analyze the tensor field if we take into account the local metric. Two different tensor fields

were estimated, one field where the spatially varying metric was not taken into account, i.e. spherical filters were applied to the warped image. The other tensor field was estimated with an elliptic filter which is geometrically correct for the central parts of close-up. In order to estimate the dominant orientation the tensor field must be transformed to a Cartesian coordinate system. With the coordinate relation between the Cartesian coordinates and the polar coordinates given in (30) we compute the partial derivatives with respect to $\tilde{x}^i$ required to compute the transformation from $x^i$ to $\tilde{x}^i$ for a second order covariant tensor is given by:

$$\begin{aligned}
\frac{\partial x^1}{\partial \tilde{x}^1} &= \frac{\partial r}{\partial x} = \cos\varphi \\
\frac{\partial x^1}{\partial \tilde{x}^2} &= \frac{\partial r}{\partial y} = \sin\varphi \\
\frac{\partial x^2}{\partial \tilde{x}^1} &= c\frac{\partial \varphi}{\partial x} = -\frac{c}{r}\sin\varphi \\
\frac{\partial x^2}{\partial \tilde{x}^2} &= c\frac{\partial \varphi}{\partial y} = \frac{c}{r}\cos\varphi
\end{aligned} \tag{31}$$

From section 2 we have that the tensor field components $\tilde{T}_{ij}$ in Cartesian coordinates is obtained by the transformation $\tilde{T}_{ij} = {a_{ij}}^{kl} T_{kl}$ where $T_{kl}$ is the components of the tensor field in polar coordinates and ${a_{ij}}^{kl}$ is obtained from the partial derivatives, its components are:

$$\begin{aligned}
{a_{11}}^{kl} &= \begin{bmatrix} \cos^2\varphi & -\frac{c}{r}\cos\varphi\sin\varphi \\ -\frac{c}{r}\cos\varphi\sin\varphi & \frac{c^2}{r^2}\sin^2\varphi \end{bmatrix} \\
{a_{12}}^{kl} &= \begin{bmatrix} \cos\varphi\sin\varphi & \frac{c}{r}\cos^2\varphi \\ -\frac{c}{r}\sin^2\varphi & -\frac{c^2}{r^2}\sin\varphi\cos\varphi \end{bmatrix} \\
{a_{21}}^{kl} &= \begin{bmatrix} \cos\varphi\sin\varphi & -\frac{c}{r}\sin^2\varphi \\ \frac{c}{r}\cos^2\varphi & -\frac{c^2}{r^2}\sin\varphi\cos\varphi \end{bmatrix} \\
{a_{22}}^{kl} &= \begin{bmatrix} \sin^2\varphi & \frac{c}{r}\cos\varphi\sin\varphi \\ \frac{c}{r}\cos\varphi\sin\varphi & \frac{c^2}{r^2}\cos^2\varphi \end{bmatrix}
\end{aligned} \tag{32}$$

The tensor fields are displayed in Figs. 7 and 8 using the glyph presented in section 3 and the color-code presented in the previous section. The color-code requires eigenvalue decomposition, which is performed on both tensor fields.

## 7 Results

Fig. 7 (middle) shows the result from a tensor field estimated under the assumption that the polar grid is Cartesian, i.e. applying spherical filters to the original sam-

pling grid. By comparing this tensor field to the close-up (Fig. 6, left) we see that it describes the structure of the image quite well. The color-code used describes the anisotropy and the dominant orientation according to (29) but under the incorrect assumption that the underlying coordinate system is Cartesian. It is for instance easy to see that the curved image structure in the upper right part of (Fig. 6, left) is correctly described under this assumption, but the geometrically correct close-up (Fig. 6, right) reveals that this image structure is not curved at all. Computing the correct color-code by transforming the tensor field to a Cartesian coordinate system and then performing eigenvalue decomposition to obtain the true signal orientation reveals a strong bias of the estimated local structure (Fig. 7 right). Note however that the degree of anisotropy is not affected by this computation, since eigenvalues are invariant to a change of coordinate system.

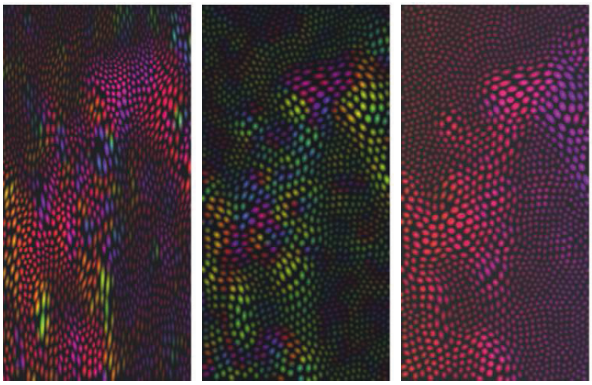


**Fig. 7 Left:** *A tensor field estimated with with elliptic filters and with correct eigenvalue decomposition. (same as in Fig. 8).* **Middle:** *A tensor field estimated with spherical filters and with incorrect eigenvalue decomposition.* **Right:** *A tensor field estimated with spherical filters and with correct eigenvalue decomposition.*

The tensor field estimated using elliptic filters to compensate for the spatially variant metric is shown in Fig. 8 (left). The same tensor field is also shown in a Cartesian coordinate system in Fig. 8 (right) and by visual inspection we can see that it reflects the structure of the image in the true geometry Fig. 6 (right). By studying the color code which represents signal orientation, it is fairly easy to see that the presented approach produce a plausible tensor field. Note also that the geometric distortion is far from being negligible. In the lower parts of the image, the pixels in the geometrically correct image are stretched approximately four times compared to the original grid.

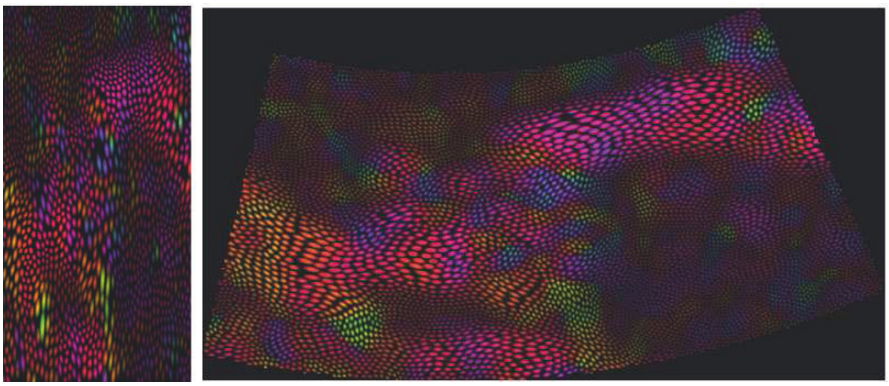


**Fig. 8** *A local structure tensor field estimated with filters which take into account the underlying geometry is presented in a non-Cartesian coordinate system (***left***) and a Cartesian coordinate system (***right***).*

## 8 Summary

Under the assumption that the underlying geometry is important, we saw that for non-Cartesian coordinate system we must distinguish between covariant and contravariant tenors. For a geometrically correct description of local structure the tensor field must be transformed in accordance to theory. However, we saw how to utilize a spatially variant metric in order to perform eigenvalue analysis in a warped image without transforming the tensor field. Geometrically correct shape estimates based on eigenvalues can then be obtained directly, whereas the eigenvectors must be transformed in order to be geometrically correct. The limitations brought on by only having access to sampled data makes it difficult to estimate a tensor, which in a strict sense is consistent with the theory presented. The presented work relies on that the transformation between the coordinate systems can be approximated by an affine transformation, at least locally. Moreover, the benefits of a iso-metric tensor glyph was demonstrated. Experiments were carried out on an ultrasound image acquired in a polar coordinate system.

## References

1. Andersson, M., Knutsson, H.: Transformation of local spatio-temporal structure tensor fields. In: Proceedings of IEEE International Conference on Acoustics, Speech, & Signal Processing (2003)
2. Bårman, H.: Hierarchical curvature estimation in computer vision. Ph.D. thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden (1991). Dissertation No 253, ISBN 91-7870-797-8
3. Bigün, J., Granlund, G.H.: Optimal orientation detection of linear symmetry. In: IEEE First International Conference on Computer Vision, pp. 433–438. London, UK (1987)

4. Farnebäck, G.: Polynomial expansion for orientation and motion estimation. Ph.D. thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden (2002). Dissertation No 790, ISBN 91-7373-475-6
5. Felsberg, M., Köthe, U.: Get: The connection between monogenic scale-space and gaussian derivatives. In: R. Kimmel, N. Sochen, J. Weickert (eds.) Scale Space and PDE Methods in Computer Vision, Hofgeismar, Germany, Springer *LNCS*, vol. 3459, pp. 192–203 (2005)
6. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points corners and centres of circular features. In: ISPRS Intercommission Workshop. Interlaken, Switzerland (1987)
7. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers (1995). ISBN 0-7923-9530-1
8. Haglund, L.: Adaptive multidimensional filtering. Ph.D. thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden (1992). Dissertation No 284, ISBN 91-7870-988-1
9. Harris, C., Stephens, M.: A combined corner and edge detection. In: Proceedings of The Fourth Alvey Vision Conference, pp. 147–15, Manchester, UK (1988)
10. Hemmendorff, M.: Motion estimation and compensation in medical imaging. Ph.D. thesis, Linköping University, Sweden, SE-581 85 Linköping, Sweden (2001). Dissertation No 703, ISBN 91-7373-060-2
11. Knutsson, H.: A tensor representation of 3-D structures. In: 5th IEEE-ASSP and EURASIP Workshop on Multidimensional Signal Processing, Noordwijkerhout, The Netherlands (1987)
12. Knutsson, H.: Representing local structure using tensors. In: The 6th Scandinavian Conference on Image Analysis, pp. 244–251. Oulu, Finland (1989). Report LiTH–ISY–I–1019, Computer Vision Laboratory, Linköping University, Sweden, 1989
13. Knutsson, H., Andersson, M.: Implications of invariance and uncertainty for local structure analysis filter sets. Signal Processing: Image Communications **20**(6), 569–581 (2005)
14. Koenderink, J.J.: The structure of images. Biological Cybernetics **50**(5), 363–370 (1984)
15. Köthe, U., Felsberg, M.: Riesz-transforms vs. derivatives: On the relationship between the boundary tensor and the energy tensor. In: R. Kimmel, N. Sochen, J. Weickert (eds.) Scale Space and PDE Mathods in Computer Vision, Hofgeismar, Germany, Springer *LNCS*, vol. 3459, pp. 179–191 (2005)
16. Ruiz-Alzola, J., Westin, C.-F., Warfield, S. K., Alberola, C., Maier, S. E., and Kikinis, R. (2002). Nonrigid registration of 3d tensor medical data. Medical Image Analysis, 6(2):143–161.
17. San Jose Estepar, R.: Local structure tensor for multidimensional signalprocesssing. applications to medical image analysis. Ph.D. thesis, University of Valladolid, Spain (2005)
18. Svensson, B., Brun, A., Andersson, M., Knutsson, H.: Estimation of non-cartesian local structure tensor fields. In: Proceedings of the 15th Scandinavian conference on image analysis (SCIA'07), pp. 948–957. Aalborg, Denmark (2007)
19. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner-Verlag, Stuttgart, Germany (1998)
20. Westin, C.F.: Feature extraction based on a tensor image description (1991). Thesis No. 288, ISBN 91-7870-815-X
21. Westin, C.F., Richolt, J., Moharir, V., Kikinis, R.: Affine adaptive filtering of CT data. Medical Image Analysis **4**(2), 161–172 (2000)

# Part III
# Tensors in Computer Vision

# Multi-View Matching Tensors from Lines for General Camera Models

Simone Gasparini and Peter Sturm

**Abstract** General camera models relax the constraint on central projection and characterize cameras as mappings between each pixel and the corresponding projection rays. This allows to describe most cameras types, including classical pinhole cameras, cameras with various optical distortions, catadioptric cameras and other acquisition devices. We deal with the structure from motion problem for such general models. We first consider an hierarchy of general cameras first introduced in [28] where the cameras are described according to the number of points and lines that have a non-empty intersection with all the projection rays. Then we propose a study of the multi-view geometry of such cameras and a new formulation of multi-view matching tensors working for projection rays crossing the same 3D line, the counterpart of the fundamental matrices and the multifocal tensors of the standard perspective cameras. We also delineate a method to estimate such tensors and recover the motion between the views.

## 1 Introduction

Tensors have been widely used in the field of computer vision as they provide frameworks that conveniently represent the multi-view geometry of cameras, help the matching of features across views and, once estimated, they allow to compute the camera motion. Many works dealt with systems of central cameras and studied the geometric and algebraic relations between correspondences of points and lines in an arbitrary number of images, thus finding multifocal tensors [8, 16, 17] useful to

Simone Gasparini
INRIA Grenoble–Rhône-Alpes, 38334 Montbonnot St. Martin, France
e-mail: simone.gasparini@inrialpes.fr

Peter Sturm
INRIA Grenoble–Rhône-Alpes, 38334 Montbonnot St. Martin, France
e-mail: peter.sturm@inrialpes.fr

compute the intrinsic and extrinsic parameters of the cameras. Other efforts have been made on non conventional cameras that can still be modeled as a central projection [1], such as central catadioptric cameras (cameras in front of a curved mirror) [3, 12, 13, 29] or the mixture of them and standard perspective cameras [27].

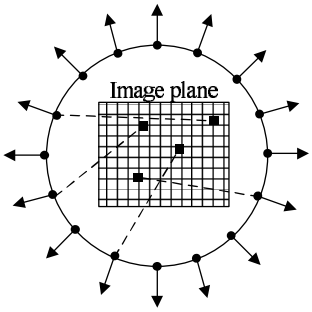
In this work we relax the constraint on central projection and we deal with a more general imaging model inspired by the one proposed by Grossberg and Nayar in [14], where the camera is modeled as a set of pixels that capture the light travelling along rays in 3D. Therefore the camera is fully described by the mapping of each pixel to the corresponding 3D ray expressed in any suitable reference frame. Such a general model can be used to represent cameras with various optical distortions (*e.g.* barrel, pincushion, wide-angle), camera clusters [24, 30], non central catadioptric cameras [2, 9, 18, 22], and other special acquisition devices such as rotating cameras [20, 23, 26] or the so-called *compound cameras* [10, 11] that try to emulate insect eyes.

Some efforts have been done in order to define the multiview geometry of such cameras. Thirthala and Pollefeys [32] proposed a study of the geometry of 1D radial cameras that allows to model some central and non central cameras, such as pinhole cameras, low distortion cameras, fish-eye cameras and catadioptric cameras. They developed a quadrifocal tensor working on uncalibrated images that allows to relate the features seen across four views.

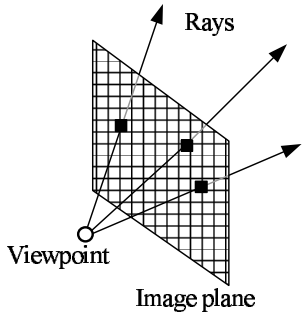
A general approach that applies to a broader set of cameras has been proposed by Sturm in [28]: it introduced a new hierarchy of camera models that allows to describe the camera according to the number of points and lines that have a non-empty intersection with all the projection rays of the camera. Then, assuming known the camera calibration, multi-view matching tensors have been developed by using corresponding projection rays in multiple views of **fully non central** cameras, in which the rays are unconstrained, **axial** cameras, in which all the rays cross a common line, **x-slit** cameras [34], in which all the rays cross two common lines, and **central** ones in which all the rays meet at a common point (see Figure 1).

In this work we use such an hierarchy and we extend the multi-view geometry of general camera models by developing novel multi-view matching tensors for line images, *i.e.* that work on projection rays that cross the same 3D line. By imposing that the projection rays associated to the images of the same 3D line across different views cross a common line in space, we propose the theoretical formulation and a complete characterization of the matching tensors for any kind of cameras described by the model. Using line correspondences can have some advantages: lines can be detected with a better accuracy than points (*e.g.* via fitting and interpolation algorithms) and, above all, they are not affected by (partial) occlusions that may occur between the views. However, it is worth noting that we follow a different approach w.r.t. other classical multifocal formulations using image line correspondences [8, 16]: once the line correspondences are available we work with the projection rays of pixels on line images, that are known by calibration.

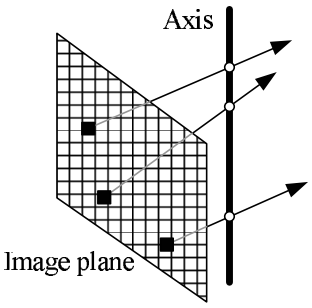
The chapter is organized as follows. In Section 2 and Section 3 we briefly recall Plücker coordinates and some useful properties of lines in space that will be used throughout the chapter; Section 4 introduces the formulation of the multifocal ten-



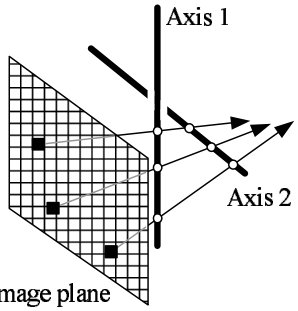

(a) Fully non central camera: the rays are totally unconstrained and, according to the model proposed in [14], each pixel is associated to a 3D direction.




(b) Central camera: all the rays meet at a common point, the *viewpoint*.




(c) Axial camera: all the rays cross a common line, the *axis* of the camera.




(d) X-Slit camera: all the rays cross two common lines, the *axes* of the camera.

**Fig. 1** The hierarchy of cameras introduced by Sturm in [28] that describes the camera according to the number of points and lines that have a non-empty intersection with all the projection rays.

sor for the general camera models which is then derived for other specific camera models in Section 5. Section 6 presents a method to estimate the tensors and some preliminary results obtained with axial cameras. Section 7 concludes the chapter.

## 2 Plücker Coordinates

For our purposes, we represent the rays and lines in 3D space via Plücker coordinates, as it is a convenient representation for lines in space, widely used in many problems involving lines in computer graphics, computational geometry and computer vision.

Given two 3D points $\mathbf{A}$ and $\mathbf{B}$, the line $\mathbf{L}$ joining them can be expressed (up to scale) via the Plücker coordinate vector of length 6

$$\mathbf{L} = \left[(\mathbf{A}-\mathbf{B})^{\mathsf{T}} \quad (\mathbf{A}\times\mathbf{B})^{\mathsf{T}}\right]^{\mathsf{T}} \tag{1}$$

Given two lines $\mathbf{L}_1$ and $\mathbf{L}_2$, their *side* product is defined as

$$side\,(\mathbf{L}_1,\mathbf{L}_2) \stackrel{\Delta}{=} \mathbf{L}_1^{\mathsf{T}} \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \mathbf{L}_2, \tag{2}$$

that is zero if they intersect or are parallel, and non-zero otherwise. Any 6-vector $\mathbf{L}$ corresponds to a "real" 3D line if and only if it satisfies the constraint $side\,(\mathbf{L},\mathbf{L}) = 0$.

Let us consider a 3D point $\mathbf{Q}$ and a rototranslation $\mathtt{P}$ defined by a rotation $\mathtt{R}$ and a translation $\mathbf{t}$ so that

$$\mathbf{Q}' \rightarrow \mathtt{P}\mathbf{Q} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \mathbf{Q};$$

the Plücker line representation is then transformed according to the transformation matrix $\hat{\mathtt{P}}$

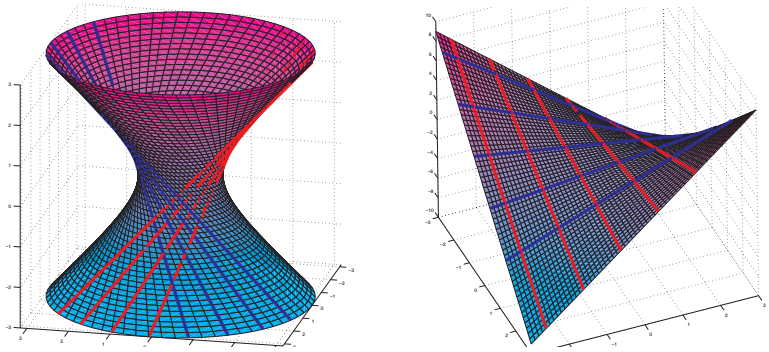
$$\mathbf{L}' \rightarrow \hat{\mathtt{P}}\mathbf{L} = \begin{bmatrix} \mathtt{R} & 0 \\ -[\mathbf{t}]_\times \mathtt{R} & \mathtt{R} \end{bmatrix} \mathbf{L}. \tag{3}$$

## 3 Lines in Space

In [28] the multi-view geometry was defined by using matching rays, *i.e*. the rays associated to the matching points, and imposing the constraint that they meet at a point in space. In this work we do not need to match the projection rays, but we use rays associated to pixels lying on matching line images. It is not necessary that pixels in different images match one another, *i.e*. that they correspond to the same point on the 3D line. We show how to build multifocal tensors by imposing the constraint that they cross a common line in space. To this end, we express the projection rays in Plücker coordinates so that such a constraint can be conveniently expressed using the *side* relation (2): since each given projection ray $\mathbf{L}_n$ crosses the common space line $\mathbf{L}$ (which is, of course, unknown), the relation $side(\mathbf{L}_i,\mathbf{L}) = 0$ must hold for all $i$, thus leading to the following linear system

$$\begin{bmatrix} \mathbf{L}_1^{\mathsf{T}} \\ \vdots \\ \mathbf{L}_n^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \mathbf{L} = \mathtt{M}\mathbf{L} = \mathbf{0}. \tag{4}$$

The problem of finding the transversal $\mathbf{L}$ of a set of rays $\mathbf{L}_i$ has been studied in [31] for computer graphics applications and then in [7, 21] to reconstruct lines from single, non central, catadioptric images. Degenerate configurations of rays may occur,



(a) Hyperboloid of one sheet (b) Hyperbolic paraboloid

**Fig. 2** Doubly-ruled quadrics are surfaces composed by two families of lines so that each line of one family (*e.g.* the blue ones) crosses all the lines of the other family (*e.g.* the red ones) and the lines of the same family are mutually skew. If the rays lie on one of the family, then there are infinitely many lines crossing them and there is no unique solution to the system (4).

in which there is no unique solution to the system (4). The six rays could lie, *e.g.*, on the same plane, so that there will be infinitely (more precisely, $\infty^2$) many lines crossing all of them; thus the system (4) is of rank 2 because only two of the rays are enough to define the plane and all the others can be seen as a linear combination.

On the other hand, the rays could also lie on a double ruled quadric, a surface composed by two families of lines so that each line of one family crosses all the lines of the other family and the lines of the same family are mutually skew. There are only two double ruled quadrics, the hyperboloid of one sheet and the hyperbolic paraboloid (see Figure 2), while the plane is also a degenerate case of ruled quadric. In this case, there will be infinitely (more precisely, $\infty^1$) many lines crossing all of them; the system (4) will be of rank 3 because any three skew lines define a ruled quadric [19].

Finally, given four lines in space in general position, there could be 0, 1, 2 or, again, an infinite number of lines intersecting them [5]. All these degenerate configurations may occur according to the camera model considered and the displacement of the line $\mathbf{L}$ in space w.r.t. the camera. As we consider rays coming from different views of general cameras, we can likely assume that the linear system has just one (non-trivial) solution $\mathbf{L}$ defined up to scale by the 1-D null space of $\mathtt{M}$. We will discuss (possible) degenerate configurations that may occur in Section 5 when we derive the multi-view tensors for specific camera models.

## 4 Multi-Focal Tensors from Lines

In this section we define multi-focal tensors for general cameras. The cameras are supposed to be calibrated, *i.e.* for each pixel of an image the associated 3D ray is

known and expressed in a local metric coordinate frame. Then we apply rotations and translations to put all cameras in a global coordinate system.

The uncalibrated case has already been treated in the case of central perspective cameras, as calibrated and uncalibrated images are linked together by 2D projective transformations. In the case of non central cameras there is not, in general, an analytical relationship among the pairs of pixels and projection rays as they could be completely independent or related by a transformation that depends on the specific camera model.

We first consider the most general model in which the pairs pixel and ray are totally unconstrained and then we derivate the results for other models than the fully non-central one, where the camera rays are constrained to meet a common point (central cameras), a line (axial cameras) or two skew lines (x-slit). As in [28], instead of projection matrices used in perspective cameras, we consider pose matrices (rototranslation matrices) associated to each view $i$ that map space points from some global reference into each camera local reference. Then, we consider a set of 6 rays coming from the views and crossing the space line $\mathbf{L}$: Table 1 reports useful choices of rays from each camera according to the number of views. From now on we call $\mathbf{a},\mathbf{b},\ldots\mathbf{f}$ the 6 rays and $\mathtt{A},\mathtt{B}\ldots\mathtt{F}$ the relevant rototranslation matrices (*cf*. (3)). We also denote with $A_i^j$ the element of the $i$-th row and $j$-th column of $\mathtt{A}$ and with $\mathbf{A}_i$ ($\mathbf{A}^j$) its $i$-th row ($j$-th column). Thus, for each ray, *e.g*. $\mathbf{a}$, combining (2) and (3) the side relation becomes:

$$\mathbf{a}^\mathsf{T}\begin{bmatrix}\mathtt{0} & \mathtt{I}\\ \mathtt{I} & \mathtt{0}\end{bmatrix}\hat{\mathtt{A}}\mathbf{L}=\mathbf{a}^\mathsf{T}\bar{\mathtt{A}}\mathbf{L}=0, \tag{5}$$

where $\bar{\mathtt{A}}$ is the remapped pose matrix. Then the linear system (4) becomes:

$$\begin{bmatrix}\mathbf{a}^\mathsf{T}\bar{\mathtt{A}}\\ \vdots\\ \mathbf{f}^\mathsf{T}\bar{\mathtt{F}}\end{bmatrix}\mathbf{L}=\mathtt{M}\,\mathbf{L}=\mathbf{0}. \tag{6}$$

The $6\times 6$ matrix $\mathtt{M}$ must be rank deficient since it has a null vector, $\mathbf{L}$: we can use the constraint $\det\mathtt{M}=0$ in order to determine a matching tensor $\mathtt{T}$. In the following we show that this constraint can be written in terms of the rays' Plücker coordinates in the form

$$\sum_{q_1=1}^{6}\sum_{q_2=1}^{6}\cdots\sum_{q_6=1}^{6}a^{q_1}b^{q_2}c^{q_3}d^{q_4}e^{q_5}f^{q_6}\,T_{q_1,\ldots q_6}=0, \tag{7}$$

where $T_{q_1,\ldots q_6}$ is the element of the $6\times6\times6\times6\times6\times6$ multifocal tensor. The constraint $\det(\mathtt{M})=0$ can be expressed in Einstein's notation as

$$\det(\mathtt{M})=\varepsilon_{ijklmn}\,M_1^i\,M_2^j\,M_3^k\,M_4^l\,M_5^m\,M_6^n=0,$$

where, e.g., $M_1^i=\mathbf{a}^\mathsf{T}\bar{\mathbf{A}}^1$ is the element in the first row of the $i$-th column of $\mathtt{M}$, so that

$$\varepsilon_{ijklmn}\;\mathbf{a}^\mathsf{T}\bar{\mathbf{A}}^i\;\mathbf{b}^\mathsf{T}\bar{\mathbf{B}}^j\;\mathbf{c}^\mathsf{T}\bar{\mathbf{C}}^k\;\mathbf{d}^\mathsf{T}\bar{\mathbf{D}}^l\;\mathbf{e}^\mathsf{T}\bar{\mathbf{E}}^m\;\mathbf{f}^\mathsf{T}\bar{\mathbf{F}}^n=0.$$

| #views | Case | Non central | Central | Axial | X-Slit |
|---|---|---|---|---|---|
| 2 | 5-1 | ✓ | | ✓ | |
| | 4-2 | ✓ | | ✓ | |
| | 3-3 | ✓ | | ✓ | ✓ |
| 3 | 4-1-1 | ✓ | | | |
| | 3-2-1 | ✓ | | ✓ | ✓ |
| | 2-2-2 | ✓ | ✓ | ✓ | ✓ |
| 4 | 3-1-1-1 | ✓ | | ✓ | ✓ |
| | 2-2-1-1 | ✓ | ✓ | ✓ | ✓ |
| 5 | 2-1-1-1-1 | ✓ | ✓ | ✓ | ✓ |
| 6 | 1-1-1-1-1-1 | ✓ | ✓ | ✓ | ✓ |

**Table 1** For each camera model, each $n$-tuple in the second column indicates the possible choices of rays from the $n$ views, that lead to a non trivial matching constraint.

Each term of the previous sum is composed in turn of a sum of $6^6$ elements; since for each ray it holds $\mathbf{a}^\top \bar{\mathbf{A}}^j = \sum_i a^i \bar{A}^j_i$ and the product of summations is the sum of products, each term can also be written as

$$\sum_{q_1=1}^{6} \dots \sum_{q_6=1}^{6} a^{q_1} b^{q_2} c^{q_3} d^{q_4} e^{q_5} f^{q_6} \bar{A}^i_{q_1} \bar{B}^j_{q_2} \bar{C}^k_{q_3} \bar{D}^l_{q_4} \bar{E}^m_{q_5} \bar{F}^n_{q_6}.$$

It can be noted that any choice of indices $ijklmn$ does not affect the first part of the product involving the Plücker coordinates of the rays and only the term involving the pose matrices $A^i_{q_1} B^j_{q_2} \dots F^n_{q_6}$ changes. So, finally, the constraint can be written as

$$\begin{aligned} \det \mathtt{M} &= \sum_{q_1=1}^{6} \dots \sum_{q_6=1}^{6} a^{q_1} \dots f^{q_6} \left( \varepsilon_{ijklmn} \bar{A}^i_{q_1} \bar{B}^j_{q_2} \dots \bar{F}^n_{q_6} \right) = \dots \\ \dots &= \sum_{q_1=1}^{6} \dots \sum_{q_6=1}^{6} a^{q_1} b^{q_2} c^{q_3} d^{q_4} e^{q_5} f^{q_6} \, T_{q_1,\dots q_6} = 0, \end{aligned}$$

where $\mathtt{T}$ is the $6 \times 6 \times 6 \times 6 \times 6 \times 6$ tensor relating the projection rays. It is worth to note that, for any given choice of indices $q_w \quad w \in \{1, \dots 6\}$, the sum $\varepsilon_{ijklmn} \bar{A}^i_{q_1} \bar{B}^j_{q_2} \dots \bar{F}^n_{q_6}$ is the determinant of the $6 \times 6$ matrix $\mathtt{S}_{q_1,\dots,q_6}$ obtained by assembling together each $q_w$-th row from the corresponding pose matrix, *i.e.*

$$T_{q_1,\dots q_6} = \det \mathtt{S}_{q_1,\dots,q_6} = \det \begin{bmatrix} \bar{\mathbf{A}}_{q_1} \\ \bar{\mathbf{B}}_{q_2} \\ \vdots \\ \bar{\mathbf{F}}_{q_6} \end{bmatrix}, \tag{8}$$

which is, in general, a sum of $6! = 720$ terms, where $\mathbf{A}_{q_1}$ is the $q_1$-th row of the matrix $\mathtt{A}$, according to the notation introduced before.

This expression allows to compute the elements of the tensor in the most general case in which each of the 6 rays belongs to a different view. It varies straightforwardly according to the number of views and the chosen case (*cf*. Table 1) where effectively the number of pose matrices involved varies.

The size of the tensor is very large, but some of its elements are zero. Consider the structure of the matrices $\bar{\mathtt{A}} \ldots \bar{\mathtt{F}}$ containing the pose information: if one of the indices $q_w > 3 \;\; w \in \{1, \ldots 5\}$, then the last three elements of the corresponding $w$-th row in $\mathtt{S}_{q_1,\ldots,q_6}$ are zero (due to the $\mathtt{0}_{3\times 3}$ submatrix in the pose matrices (3)). It follows that $T_{q_1,\ldots,q_6} = \det \mathtt{S}_{q_1,\ldots,q_6} = 0$ if the values of at least 4 of the indices $q_w$ are greater than 3: in such cases, indeed, the last 3 columns of $\mathtt{S}_{q_1,\ldots,q_6}$ are not linearly independent, hence the determinant is zero. Moreover, if there are more than one ray belonging to the same view, the indices $q_w$ that are associated to those rays, must have distinct values, otherwise the corresponding rows of $\mathtt{S}$ are identical, hence the determinant is zero.

Without loss of generality, we can also assume that the global coordinate system coincides with the first camera's local coordinate system, *i.e*. the first camera pose matrix $\mathtt{A}$ is the identity matrix. Hence, the corresponding rows in $\mathtt{S}$ contain just one non-zero element. In order to determine the exact number of non-zero elements of $\mathtt{T}$ for a given case, one should solve a combinatorial problem which takes into account all the combinations of the indices $q_w$ for which at least 4 of the indices $q_w$ are greater than 3 and the indices associated to the same view have distinct values, up to a permutation.

Finally, from the structure of the matrices $\mathtt{A} \ldots \mathtt{F}$ we can observe that they contain 9 zero elements, and only 18 of the remaining elements are unique, since the elements of the rotation matrix $\mathtt{R}$ appear twice. Therefore, different elements of $\mathtt{T}$ may be identical (up to sign), as they can be derived from $\det \mathtt{S}$ with different combinations of the pose matrices' rows.

We collect in Table 2 the number of non-zero elements and the number of unique elements (up to sign) for each useful case according to the number of the views.

## 5 Constrained Camera Models

We consider now more specific camera models than fully non central ones where the camera rays are no more unconstrained but are subject to a constraint, such as central, axial and x-slit cameras. For each case, we briefly report the parameterizations introduced in [28] and then we study the possible cases (*cf*. Table 1) and the relevant properties of the tensor $\mathtt{T}$ (*cf*. Table 2).

| Cases (*cf*. Table 1) | **Non Central** | | **Central** | | | | **Axial** | | | | **X-Slit** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | finite | | infinite | | finite | | infinite | | 2 finite | | finite+infinite | |
| 5-1 | 3240 | 18 | *n.a.* | | | | 600 | 5 | 240 | 2 | *n.a.* | | | |
| 4-2 | 7776 | 117 | *n.a.* | | | | 2256 | 46 | 1104 | 22 | *n.a.* | | | |
| 3-3 | 10152 | 200 | *n.a.* | | | | 3276 | 87 | 1620 | 41 | 576 | 16 | 432 | 12 |
| 4-1-1 | 9072 | 270 | *n.a.* | | | | 2712 | 109 | 1344 | 52 | *n.a.* | | | |
| 3-2-1 | 14796 | 900 | *n.a.* | | | | 5244 | 417 | 2724 | 207 | 1152 | 96 | 816 | 68 |
| 2-2-2 | 18360 | 1701 | 216 | 27 | 64 | 8 | 6912 | 824 | 3680 | 420 | 1728 | 216 | 1216 | 152 |
| 3-1-1-1 | 17496 | 2106 | *n.a.* | | | | 6420 | 1022 | 3396 | 518 | 1518 | 253 | 1056 | 176 |
| 2-2-1-1 | 21708 | 4050 | 324 | 81 | 96 | 24 | 8460 | 2011 | 4564 | 1037 | 2268 | 566 | 1568 | 391 |
| 2-1-1-1-1 | 25758 | 9720 | 486 | 243 | 150 | 75 | 10404 | 4946 | 5696 | 2592 | 2988 | 1491 | 2048 | 1021 |
| 1-1-1-1-1-1 | 30618 | 27702 | 729 | 729 | 233 | 233 | 12825 | 12185 | 7120 | 6480 | 3942 | 3942 | 2688 | 2678 |

**Table 2** For each camera model, the first column collects the number of non-zero elements in the tensor `T` for each possible case, while the second column collects the number of unique elements (up to sign) in the tensor `T` (the cases marked with *n.a.* give no useful constraint).

## *5.1 central cameras*

Since all rays go through a single point, *i.e*. the optical center (possibly, at an infinite point), we can consider at most two projection rays for each view: any other projection ray in the same view would be a linear combination of the previous ones, hence no useful constraint can be exploited from the matrix `M` as it is always rank deficient. Therefore, and since we need 6 rays to build the tensor, two views do not provide useful constraint and do actually not allow a projective reconstruction, as stated in [33]. Table 1 reports the useful cases according to the number of views.

A degenerate configuration occurs when the cameras share a common baseline, *i.e*. the viewpoints are collinear. Then all the projection rays cross the baseline at the relevant viewpoints. Such a configuration can be seen as a particular case of axial cameras (*cf*. Section 5.2).

We distinguish the cases of a finite and infinite optical center.

### 5.1.1 Finite Optical Center

Setting the optical center as the origin **O** of the local coordinate system, each ray is described by a Plücker vector having the last three elements equal to zero, *i.e*. according to (1) each ray is of the form

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}.$$

Hence the base size of the multi-focal tensor `T` can be reduced from 6 to 3, and (7) becomes

$$\det \mathtt{M} = \sum_{q_1=1}^{3} \dots \sum_{q_6=1}^{3} a^{q_1} b^{q_2} c^{q_3} d^{q_4} e^{q_5} f^{q_6}\, T_{q_1,\dots,q_6} = 0.$$

so that only the first three rows of each pose matrix $\bar{\mathtt{A}}, \dots, \bar{\mathtt{F}}$ can contribute to $\mathtt{S}$.

### 5.1.2 Infinite Optical Center

Choosing a suitable coordinate system where the viewpoint $\mathbf{V}$ has (homogeneous) coordinates $\mathbf{V} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\mathsf{T}$, each ray is described by a Plücker vector of the form

$$\mathbf{a} = \begin{bmatrix} 0 & 0 & a_3 & a_4 & a_5 & 0 \end{bmatrix}^\mathsf{T}.$$

Again, the base size of the multi-focal tensor $\mathtt{T}$ can be reduced from 6 to 3 so that (7) becomes

$$\det \mathtt{M} = \sum_{q_1=3}^{5} \dots \sum_{q_6=3}^{5} a^{q_1} b^{q_2} c^{q_3} d^{q_4} e^{q_5} f^{q_6}\, T_{q_1,\dots,q_6} = 0.$$

Since the indices $q_w$ run from 3 to 5, only the rows from the 3rd to the 5th of each matrix $\bar{\mathtt{A}}, \dots, \bar{\mathtt{F}}$ contribute to the matrix $\mathtt{S}$. Looking at the structure of those matrices, we can note that the tensor $\mathtt{T}$ will contain only elements from the rotation matrices $\mathtt{R}_i$ and elements from the last row of matrices $-[\mathbf{t}_i]_\times \mathtt{R}_i$.

## *5.2 Axial Cameras*

In axial cameras, all the projection rays cross a line, the camera axis, but otherwise they can be mutually skew. In such cameras, if the considered 3D line lies in an axial plane, *i.e.* a plane containing the axis, we should consider at most two rays for each view: any other ray in the same view, indeed, would be a linear combination of the previous ones, hence no useful constraint can be exploited from the matrix $\mathtt{M}$, as it would be always rank deficient.

Similarly, if the space line lies in a horizontal plane, *i.e.* a plane perpendicular to the axis, its projection rays are coplanar and, again, only two rays have to be considered from each view. On the other hand, the space line and the axis might be contained in a ruled quadric, hence at most three projection rays should be considered from each view. This is not the case for most axial-symmetric catadioptric cameras, as it has been proved in [6].

Another degenerate configuration occurs when the cameras are coaxial, *e.g.* they differ in terms of a rototranslation along the axis, then there are always two lines (the 3D line and the two coincident axes) that cross the projection rays.

We distinguish axial cameras that have a finite or an infinite axis.

### 5.2.1 Finite Axis

Assume that the camera axis is the $Z$-axis. Then, all projection rays have Plücker coordinates with $L_6 = 0$. The base size of the multi-focal tensor $\mathtt{T}$ can be reduced from 6 to 5, and the expression (7) becomes

$$\det \mathtt{M} = \sum_{q_1=1}^{5} \dots \sum_{q_6=1}^{5} a^{q_1} b^{q_2} c^{q_3} d^{q_4} e^{q_5} f^{q_6} \, T_{q_1,\dots,q_6} = 0.$$

so that only the last row of each matrix $\bar{\mathtt{A}}, \dots, \bar{\mathtt{F}}$ does not contribute to $\mathtt{S}$.

### 5.2.2 Infinite Axis

We choose a local coordinate system where the axis is the line at infinity with coordinates $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\mathsf{T}$ (line coordinates on plane at infinity), so that the camera axis' Plücker coordinates are $\alpha = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^\mathsf{T}$ and all the projection rays have the first Plücker coordinate equals to zero, *i.e.* the rays are of the form

$$\mathbf{a} = \begin{bmatrix} 0 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}^\mathsf{T}.$$

Multi-view relations for infinite axial cameras can thus be formulated via tensors of base size 5, and the expression (7) becomes

$$\det \mathtt{M} = \sum_{q_1=2}^{6} \dots \sum_{q_6=2}^{6} a^{q_1} b^{q_2} c^{q_3} d^{q_4} e^{q_5} f^{q_6} \, T_{q_1,\dots,q_6} = 0.$$

so that only the first row of each matrix $\bar{\mathtt{A}}, \dots, \bar{\mathtt{F}}$ does not contribute to $\mathtt{S}$.

## *5.3 X-Slit Cameras*

In x-slit cameras there exist two lines, *i.e.* camera axes, that cut all projection rays. Linear pushbroom cameras [15] is a special case of such cameras. The case of the two axes cutting one another, *i.e.* being coplanar, is not of interest here, so we consider two mutually skew axes. As discussed in the case of axial cameras, if the space line and one of the axes are coplanar then all the projection rays associated to the space line lie on a plane: thus, just in this case, we should consider at most two rays for each view.

Similarly, for any 3D line that is skew w.r.t. the camera axes the projection rays lie on a ruled quadric, since three skew lines completely define a ruled quadric [19]. Therefore, we should consider configurations with at most three rays from each view.

Finally, similarly to axial cameras, if the cameras share a common axis, then there are always two lines crossing all the rays associated to the space line.

Two cases are possible for x-slit cameras: (*i*) both axes are finite lines or (*ii*) one of the two axes is a line at infinity. Since at least one axis is a finite line, we adopt the same reference system used for axial cameras. As for the second axis, we have to distinguish the two cases.

### 5.3.1 Two Finite Axes

Having fixed the first axis $\alpha$, we still have the freedom to rotate about it and translate along it. Since the two axes are skew, we may thus obtain a local coordinate system, where the second axis goes through a point on the $Y$-axis, and is parallel to the $XZ$-plane. Hence, it will be defined by two points as follows:

$$Q_1^\mathsf{T} = \begin{bmatrix} 0 & Y & 0 & 1 \end{bmatrix} \qquad Q_2^\mathsf{T} = \begin{bmatrix} X & 0 & Z & 0 \end{bmatrix}.$$

The second axis' Plücker coordinates are thus given by

$$\beta^\mathsf{T} = \begin{bmatrix} X & 0 & Z & -YZ & 0 & YX \end{bmatrix}.$$

Since each projection rays **a** must cut the two axes, by imposing the side constraint with both axes, the rays must be of the form:

$$\mathbf{a}^\mathsf{T} = \begin{bmatrix} a_1 & a_2 & a_3 & (Wa_1 - Ya_3) & a_5 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ W & 0 & -Y & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_5 \end{bmatrix} = \mathtt{G} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_5 \end{bmatrix},$$

where $W = \frac{YZ}{X}$[1] and $Y$ can be seen as the intrinsic parameters of the camera. Then, after some straightforward computations, (5) becomes

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_5 \end{bmatrix} \tilde{\mathtt{A}}\,\mathbf{L} = 0,$$

where

$$\tilde{\mathtt{A}}_{4\times 6} = \mathtt{G}^\mathsf{T}\bar{\mathtt{A}}.$$

The base size of the multi-focal tensor $\mathtt{T}$ is then reduced from 6 to 4 so that the expression (7) becomes

$$\det \mathtt{M} = \sum_{q_1}^{\{1,2,3,5\}} \cdots \sum_{q_6}^{\{1,2,3,5\}} a^{q_1} b^{q_2} c^{q_3} d^{q_4} e^{q_5} f^{q_6}\, T_{q_1,\ldots,q_6} = 0.$$

[1] We divide by $X$ since it can not be zero, otherwise the second axis would be parallel to the first one, and thus coplanar, which is excluded here.

where the indices $q_w$ run on the discrete set $\{1,2,3,5\}$ and $T_{q_1,\ldots,q_6}$ can be computed as in (8) by replacing each matrix $\bar{\mathtt{A}},\ldots,\bar{\mathtt{F}}$ with the corresponding matrix $\tilde{\mathtt{A}},\ldots,\tilde{\mathtt{F}}$.

### 5.3.2 Finite + Infinite Axis

Having fixed the first axis, we still have the freedom to rotate about it and translate along it. Translation has no effect on the infinite second axis, but we may rotate about the first axis, such that the second one has coordinates $\begin{bmatrix}0 & \cos\theta & \sin\theta\end{bmatrix}^{\mathsf{T}}$ (homogeneous coordinates of a line at infinity). The second axis' Plücker coordinates are thus

$$\beta^{\mathsf{T}} = \begin{bmatrix}0 & 0 & 0 & 0 & \cos\theta & \sin\theta\end{bmatrix}.$$

Projection rays cut the two axes, so must be of the form

$$\mathbf{a}^{\mathsf{T}} = \begin{bmatrix}a_1 & Wa_3 & a_3 & a_4 & a_5 & 0\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & W & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}a_1\\ a_3\\ a_4\\ a_5\end{bmatrix} = \mathtt{H}\begin{bmatrix}a_1\\ a_3\\ a_4\\ a_5\end{bmatrix},$$

where $W = -\tan\theta$ is the intrinsic parameter of the camera. Straightforwardly, (5) becomes

$$\begin{bmatrix}a_1 & a_3 & a_4 & a_5\end{bmatrix}\tilde{\mathtt{A}}\mathbf{L} = 0,$$

where

$$\tilde{\mathtt{A}}_{4\times 6} = \mathtt{H}^{\mathsf{T}}\bar{\mathtt{A}}.$$

The base size of the multi-focal tensor $\mathtt{T}$ is then reduced from 6 to 4 so that the expression (7) becomes

$$\det\mathtt{M} = \sum_{q_1}^{\{1,3,4,5\}} \cdots \sum_{q_6}^{\{1,3,4,5\}} a^{q_1}b^{q_2}c^{q_3}d^{q_4}e^{q_5}f^{q_6}\,T_{q_1,\ldots,q_6} = 0.$$

where the indices $q_w$ run on the discrete set $\{1,3,4,5\}$ and $T_{q_1,\ldots,q_6}$ can be computed as in (8) by replacing each matrix $\bar{\mathtt{A}},\ldots,\bar{\mathtt{F}}$ with the corresponding matrix $\tilde{\mathtt{A}},\ldots,\tilde{\mathtt{F}}$.

## 6 Experimental Results

In order to estimate the tensor elements and retrieve the camera motion (*i.e.* the rototranslation), we note that (7) can be written as

$$\det\mathtt{M} = \sum_{s=1}^{Z}\alpha_s T_s = 0,$$

where $Z$ is the the number of unique terms of the tensor for the considered case, $T_s$ is the $s$-th unique element, and $\alpha_s$ is a coefficient that collects together all the contributions $a^{q_1}b^{q_2}\dots f^{q_6}$ from coordinates of rays (known from calibration) associated to the same (unique) element $T_s$. Moreover, each $T_s$ is a function of the $6(n-1)$ unknowns of the $n-1$ rototranslations relating the views). Since the explicit form of each $T_s$ is known and can be easily computed once from (8), a minimization process on the 6 unknowns can be run in order to get the motion between cameras and the elements of the tensor T as well.

In order to test the effectiveness of the proposed estimation method, we performed some tests with images taken from a standard perspective camera (*cf*. Figure 3) and a central catadioptric one (*cf*. Figure 4). In both cases, we first calibrated the cameras using two image datasets of a calibration grid. The perspective camera has been calibrated with the Camera Calibration Toolbox for perspective cameras [4], while the catadioptric one has been calibrated with the "OcamCalib" Toolbox [25]. Then, for each camera, we chose three images from the dataset and we extracted the lines from the squares of the calibration grid. Since in the case of central camera at most two rays have to be considered, for each line we considered the two projection rays associated to the two extreme points of the line lying on the outer border of the grid. We estimated the tensors with the above method and from them we extracted the motion among the views finding in both cases the same rototranslations obtained by calibration.

The main issue we experienced was that the minimization procedure tends to find the transformations that just align the three cameras: thus all the projection rays cross a common line, the common baseline, satisfying (6). To avoid such degenerate solutions, we added some constraints to the minimization process so that it avoids to align the cameras unless then the rays also cross another common line in space. In general it requires some runnings (with different initial guesses) before converging to the optimal solution, although the initial guess does not have to be close to the real solution.

## 7 Conclusions

We presented a new formulation of multifocal tensors for general camera models working on projection rays crossing the same 3D line. We extended the theoretical framework proposed in [28] and based on matching rays that meet at a point in space. We also proposed a method to estimate the tensors and retrieve the motion parameters of the camera and performed some preliminary experiments in the special case of axial cameras.

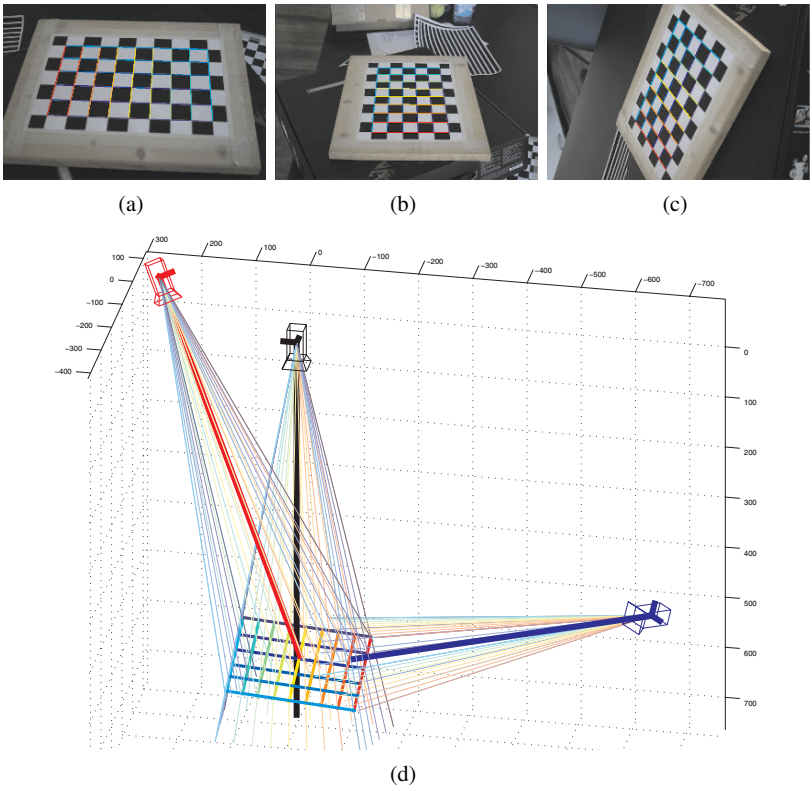



**Fig. 3** Three views of the calibration grid (a,b,c) taken with a standard perspective camera with the lines used to estimate the matching tensor and (d) the displacement among cameras computed from the tensor.

## Acknowledgements

The authors would like to thank Davide Scaramuzza for providing the catadioptric images used in the experiments.

## References

1. Baker, S., Nayar, S.K.: A theory of single-viewpoint catadioptric image formation. International Journal of Computer Vision **35**(2), 175–196 (1999). DOI 10.1023/A:1008128724364
2. Bakstein, H., Pajdla, T.: An overview of non-central cameras. In B. Likar, Editor, Proceedings of the Computer Vision Winter Workshop, Ljubljana, Slovenia, February 2001, pp. 223–233.
3. Barreto, J., Araujo, H.: Paracatadioptric camera calibration using lines. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV '03), vol. 2, pp. 1359–1365. IEEE Computer Society, Los Alamitos, CA, USA (2003). DOI 10.1109/ICCV.2003.1238648

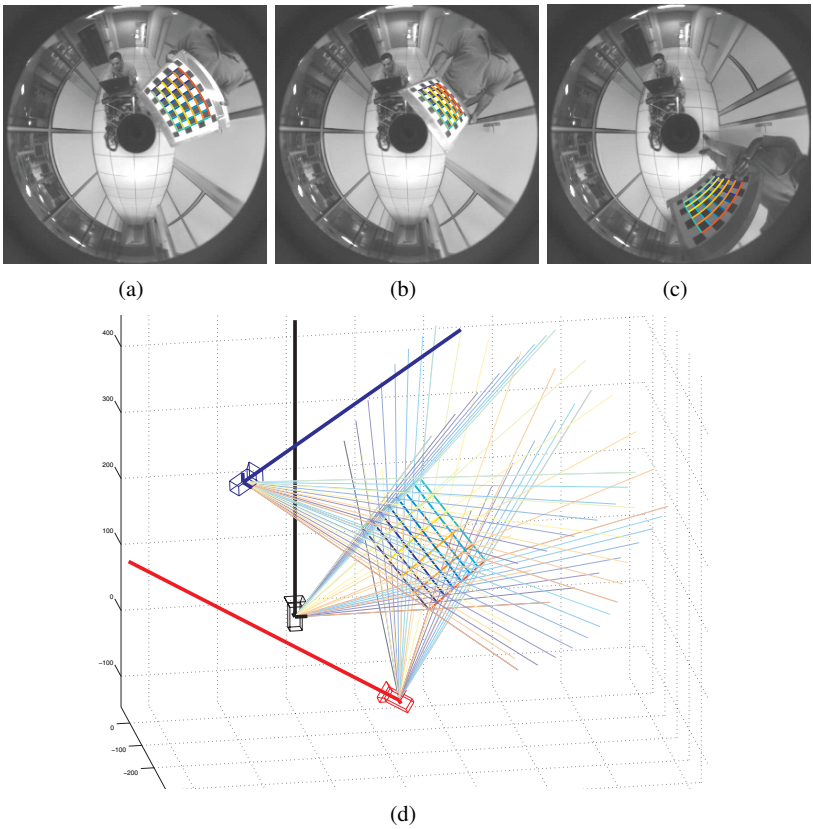



**Fig. 4** Three views of the calibration grid (a,b,c) taken with a central catadioptric camera with the lines used to estimate the matching tensor and (d) the displacement among cameras computed from the tensor. Images courtesy of Davide Scaramuzza.

4. Bouguet, J.Y.: Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj`
5. Bronnimann, H., Everett, H., Lazard, S., Sottile, F., Whitesides, S.: Transversals to line segments in three-dimensional space. Discrete Compututational Geometry **34**(3), 381–390 (2005). DOI http://dx.doi.org/10.1007/s00454-005-1183-1
6. Caglioti, V., Gasparini, S.: On the localization of straight lines in 3D space from single 2D images. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '05), vol. 1, pp. 1129–1134. IEEE Computer Society, Los Alamitos, CA, USA (2005). DOI 10.1109/CVPR.2005.257
7. Caglioti, V., Gasparini, S.: "How many planar viewing surfaces are there in noncentral catadioptric cameras?" Towards singe-image localization of space lines. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '06), vol. 1, pp. 1266–1273. IEEE Computer Society, Los Alamitos, CA, USA (2006). DOI 10.1109/CVPR.2006.1
8. Faugeras, O., Mourrain, B.: On the geometry and algebra of the point and line correspondences between n images. In: CVPR95, pp. 951–956 (1995). DOI 10.1109/ICCV.1995.466832

9. Feldman, D., Pajdla, T., Weinshall, D.: On the epipolar geometry of the crossed-slits projection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV '03), vol. 2, pp. 988–995. IEEE Computer Society, Washington, DC, USA (2003). DOI 10.1109/ICCV.2003.1238456
10. Fermuller, C., Aloimonos, Y., Baker, P., Pless, R., Neumann, J., Stuart, B.: Multi-camera networks: eyes from eyes. In: Proceedings of the 1st Workshop on Omnidirectional Vision (OMNIVIS 2000), Hilton Head, S.C., USA, 2000, pp. 11–18. DOI 10.1109/OMNVIS.2000.853797
11. Firoozfam, P., Negahdaripour, S.: Multi-camera conical imaging; calibration and robust 3D motion estimation for ROV based mapping and positioning. In: Proceedings of OCEANS MTS/IEEE Conference and Exhibition, vol. 3, pp. 1595–1602 (2002). DOI 10.1109/OCEANS.2002.1191873
12. Geyer, C., Daniilidis, K.: Paracatadioptric camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5), 687–695 (2002). DOI 10.1109/34.1000241
13. Geyer, C., Daniilidis, K.: Mirrors in motion: epipolar geometry and motion estimation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV '03), pp. 766–773. IEEE Computer Society, Los Alamitos, CA, USA (2003). DOI 10.1109/ICCV.2003.1238426
14. Grossberg, M., Nayar, S.: A general imaging model and a method for finding its parameters. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV '01), pp. 108–115. IEEE Computer Society, Los Alamitos, CA, USA (2001)
15. Gupta, R., Hartley, R.I.: Linear pushbroom cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(9), 963–975 (1997). DOI 10.1109/34.615446
16. Hartley, R.I.: Lines and points in three views and the trifocal tensor. International Journal of Computer Vision **22**(2), 125–140 (1997). DOI http://dx.doi.org/10.1023/A:1007936012022
17. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press (2004)
18. Hicks, R., Bajcsy, R.: Catadioptric sensors that approximate wide-angle perspective projections. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '00), vol. 1, pp. 545–551. IEEE Computer Society, Los Alamitos, CA, USA (2000). DOI 10.1109/CVPR.2000.855867
19. Hilbert, D., Cohn-Vossen, S.: Geometry and the Imagination. Chelsea Publishing Co., New York: Chelsea (1932)
20. Ishiguro, H., Yamamoto, M., Tsuji, S.: Omni-directional stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2), 257–262 (1992). DOI 10.1109/34.121792
21. Lanman, D., Wachs, M., Taubin, G., Cukierman, F.: Reconstructing a 3D line from a single catadioptric image. In: Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission, pp. 89–96 (2006). DOI 10.1109/3DPVT.2006.115
22. Micusik, B., Pajdla, T.: Autocalibration & 3D reconstruction with non-central catadioptric cameras. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '04), vol. 1, pp. 58–65. IEEE Computer Society, Los Alamitos, CA, USA (2004). DOI 10.1109/CVPR.2004.1315014
23. Peleg, S., Ben-Ezra, M.: Stereo panorama with a single camera. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '99), vol. 1. IEEE Computer Society, Los Alamitos, CA, USA (1999). DOI 10.1109/CVPR.1999.786969
24. Pless, R.: Using many cameras as one. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '03), vol. 2, pp. 587–93, Madison, WI, USA, 2003. DOI 10.1109/CVPR.2003.1211520
25. Scaramuzza, D., Martinelli, A., Siegwart, R.: A toolbox for easily calibrating omnidirectional cameras. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5695–5701, Beijing, China, 2006. DOI 10.1109/IROS.2006.282372
26. Shum, H.Y., Kalai, A., Seitz, S.: Omnivergent stereo. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV '99), vol. 1, pp. 22–29. IEEE Computer Society, Los Alamitos, CA, USA (1999). DOI 10.1109/ICCV.1999.791193

27. Sturm, P.: Mixing catadioptric and perspective cameras. In: Proceedings of the 3rd Workshop on Omnidirectional Vision (OMNIVIS 2002), pp. 37–44. IEEE Computer Society, Los Alamitos, CA, USA (2002). DOI 10.1109/OMNVIS.2002.1044489
28. Sturm, P.: Multi-view geometry for general camera models. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '05), vol. 1, pp. 206–212. IEEE Computer Society, Los Alamitos, CA, USA (2005). DOI 10.1109/CVPR.2005.237
29. Svoboda, T., Pajdla, T.: Epipolar geometry for central catadioptric cameras. International Journal of Computer Vision **49**(1), 23–37 (2002). DOI 10.1023/A:1019869530073
30. Swaminathan, R., Nayar, S.: Nonmetric calibration of wide-angle lenses and polycameras. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(10), 1172–1178 (2000)
31. Teller, S., Hohmeyer, M.: Determining the lines through four lines. Journal of Graphics Tools **4**(3), 11–22 (1999)
32. Thirthala, S., Pollefeys, M.: Multi-view geometry of 1D radial cameras and its application to omnidirectional camera calibration. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV '05), vol. 2, pp. 1539–1546, Beijing, China, 2005. DOI 10.1109/ICCV.2005.158
33. Weng, J., Huang, T., Ahuja, N.: Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(3), 318–336 (1992). DOI 10.1109/34.120327
34. Zomet, A., Feldman, D., Peleg, S., Weinshall, D.: Mosaicing new views: The crossed-slits projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(6), 741–754 (2003). DOI 10.1109/TPAMI.2003.1201823

# Binocular Full-Body Pose Recognition and Orientation Inference Using Multilinear Analysis

Peng and Qian

**Abstract** In this chapter, we propose an approach to full-body pose recognition and body orientation estimation using multilinear analysis. We extract low-dimensional pose and body orientation coefficient vectors by performing tensor decomposition and projection on silhouette images obtained from wide baseline binocular cameras. The coefficient vectors are then used as feature vectors in pose recognition and body orientation estimation. To do pose recognition, pose coefficient vectors obtained from synthesized pose silhouettes are used to train a family of support vector machines as pose classifiers. Using orientation coefficient vectors, a 1-D orientation manifold is learned and further used for the estimation of body orientation. Experiment results obtained using both synthetic and real image data showed that the performance of our approach is comparable to existing pose recognition approaches, and that our approach outperformed the traditional tensor-based recognition approach in the comparative test.

## 1 Introduction

Poses are important cues for inter-personal communication as well as human-computer interaction (HCI). Vision-based pose recognition has been applied to many movement-based human computer interactions (MB-HCI), such as automatic sign language analysis and interpretation [30], embodied gestural control of media [33], interactive dance performances [3, 14, 18, 31] and interactive robots [15, 22].

Bo Peng
Arts, Media & Engineering Program, and Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287 e-mail: bo.peng.1@asu.edu

Gang Qian
Arts, Media & Engineering Program, and Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287 e-mail: gang.qian@asu.edu

In an MB-HCI environment, subjects communicate with the background through their movement to control audio and visual effects. Static body poses and continuous body gestures can both be used as communication cues. A vital part in such environment is a robust movement analysis engine that accurately and time-efficiently analyzes the movement of the performers. In [14] and [31], a marker-based motion capture system is utilized to obtain body kinematics for pose recognition. This system can reliably capture the 3D coordinates of the markers placed on bony landmarks of the body. Therefore, it works very well in real life performances. However, wearing a set of markers can be cumbersome to the subjects.

To overcome such limitations, markerless video-based systems are preferred. Vision-based pose recognition has been extensively studied in the literature [40]. Existing methods can be mainly categorized into three groups according to types of features extracted, namely body kinematics [21, 29], 3D volumetric reconstruction [5] and 2-D silhouettes [10, 12].

One important feature for vision-based pose recognition required in many MB-HCI applications is view-independence. Despite the extensive research on pose recognition, view-invariant, reliable, and computationally effective pose recognition algorithms are yet to be developed. Most of the existing pose recognition methods, especially the single-view approaches, are view-based, i.e., assuming that the relative torso orientation with respect to the camera is known. While it is a valid assumption in some of the applications, such as automatic sign language interpretation in a controlled environment, having to know body orientation with respect to the camera presents an undesired constraint which hampers the flexibility and sometimes, the usability of an MB-HCI system in applications such as interactive dance and media control. In these applications, it is preferable that a pose can be recognized from any view point with respect to the cameras so that the user can freely move and orient in the space.

Another important issue related to the poses is the torso orientation of the subject. The same pose may be performed with different torso orientations (rotation about the axis perpendicular to the ground plane) relative to the camera system. Such different performances of a pose can carry totally different information (e.g. a pointing pose facing different directions). Therefore, body orientation information should also be taken into account in pose recognition tasks.

View-invariant pose recognition would be straightforward when body kinematics such as joint angles can be reliably recovered from the input images. Recently video-based motion capture has seen tremendous progress using various generative-based (e.g. [4, 13, 16]) and discriminative-based (e.g. [7, 32]) approaches. Various dynamical models have been used to represent the movement dynamics and at the same time reduce the dimensionality of the movement state space [25, 35]. Recent literature surveys can be found in [27, 28, 39]. Once body kinematics are recovered, poses can be recognized using body joint angles as the feature. In this case, body orientation angles of the performer can also be easily estimated. However, reliable recovery and tracking of poses for general movement which has not been seen in the training remains a very challenging task for video-based motion capture. For example, in a dance performance, the performer can easily go through a variety of

movements. Thus, it is unrealistic to train a video-based motion capture system that can keep tracking through such untrained movement. Hence to obtain pose recognition and body orientation estimation through pose estimation is impractical for interactive environments.

An alternative approach is to first recover the 3D volumetric reconstruction of the performer using e.g. visual-hull techniques [4, 5] and then conduct pose recognition and body orientation estimation based on the 3D voxel data. In this way, view-invariant pose recognition is readily achievable. However, the limitation of such strategy is that to recover a 3D body structure with sufficient accuracy for pose recognition, a certain amount of calibrated cameras (e.g. 6) are usually needed. In addition, matching the observed body shapes to templates possibly in different orientations remains a challenging research problem.

As another alternative, many methods have been developed for silhouette based pose recognition. Boulay, Bremond and Thonnat [2] applied a 3D human model in a pose recognition system. Recognition is achieved by comparing observed silhouette with the silhouettes projected from the 3D models of a set of poses. Similarly, Howe [10] achieved pose tracking by looking up a collection of silhouettes of known poses. Huang, Di and Xu [12] proposed a viewpoint insensitive recognition system using "envelope shape" representation of poses, and performed experiments on several simple actions. We have also performed research on dance pose recognition in [8] previously. We used Gaussian Mixture Model (GMM) for feature extraction of the silhouette and relevance vector machine (RVM) for pose recognition. One common weakness of such methods is that they can only recognize poses. Body orientation cannot be estimated.

In this chapter, we present an approach to simultaneous view-invariant pose recognition and body orientation recovery using a simple set-up of two video cameras. The cameras do not need to be calibrated. The only requirement for the camera setup is their looking directions are parallel to the ground plane and orthogonal to each other. Multilinear analysis is used to extract pose and body orientation vectors. Based on the pose vectors, support vector machine (SVM) are trained to conduct pose recognition. Using the body orientation vectors, a 1-D manifold is constructed. Using this manifold, torso orientation estimation is cast into a nonlinear least square problem.

## 2 Overview of the Proposed Approach

In this chapter, we propose a tensor-based approach to full-body pose recognition. The input for recognition is a pair of normalized silhouette image acquired from a pair of orthogonal cameras. By performing a bilinear analysis of dance pose images with a 3-mode tensor, we are able to not only effectively recognize different dance poses, but also estimate the body orientation of the performer.

To the best of our knowledge, our tensor-based method is unique compared with previous methods for pose recognition. Furthermore, we have extended the tradi-

tional tensor-based methods by overcoming limitations of such methods in two aspects. Firstly, traditional tensor methods cannot incorporate intra-class variations in their training sets. For example, only one pair of images can be included in a tensor for one pose in one orientation, while there is certainly a range of variation of joint angles when performing one pose, which yields variation in the silhouettes. To solve this problem, we generated good "representatives" of each pose in each orientation to form the tensor. Meanwhile, we incorporate more training inputs and solve their pose coefficient vectors with the decomposed tensor. The coefficient vectors are then used to train several SVM classifiers for pose recognition. In this way we applied the coefficient vectors as the low-dimensional view-invariant representation of poses. Secondly, when dealing with continuous quantities like orientation angle in this case, a traditional tensor can only perform classification into discrete ranges. However, in our approach, we performed a manifold learning method to solve the continuous value of the orientation angle.

Our proposed method worked effectively both on synthetic data and real data, and the computation efficiency is high. A block diagram of our proposed system is shown in Figure 1.

## 3 Theoretical Background

### *3.1 Background of Tensor Algebra*

As introduced in [36], a tensor, also known as n-way array or multidimensional matrix or n-mode matrix, is a higher order generalization of a vector (1-mode tensor) and a matrix (2-mode tensor). High-order tensors can represent a collection of data in a more complicated way. When a data vector is determined by a combination of $m$ factors, the collection of the data vectors can be represented as an $(m+1)$-mode tensor $\mathsf{T} \in \mathbb{R}^{N_v \times N_{f1} \times N_{f2} \cdots \times N_{fm}}$, in which $N_v$ is the dimensionality of the data vector and $N_{fi}$, $(i = 1, 2, \ldots, m)$ is the number of possible values of the $i$th contributing factor.

A tensor can be unfolded into a matrix along each mode. The mode-$j$ unfolding matrix of a tensor $\mathsf{A} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_n}$ is denoted as $\mathbf{A}_{(j)}$, and $\mathbf{A}_{(j)} \in \mathbb{R}^{N_j \times (N_1 \ldots N_{j-1} N_{j+1} \ldots N_n)}$. An illustration of the unfolding of a 3-mode tensor is shown in Figure 2.

As an analogue to matrix-matrix multiplication, an n-mode tensor can be multiplied by compatible matrices in each mode. The mode-$j$ multiplication of a n-mode tensor $\mathsf{A} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_n}$ with a matrix $\mathbf{M} \in \mathbb{R}^{N'_j \times N_j}$ can be denoted as $\mathsf{R} = \mathsf{A} \times_j \mathbf{M}$, $\mathsf{R} \in \mathbb{R}^{N_1 \times \cdots \times N_{j-1} \times N'_j \times N_{j+1} \times \cdots \times N_n}$. The entries of $\mathsf{R}$ are defined as

$$\mathsf{R}_{i_1,\ldots,i_{j-1},i,i_{j+1}\ldots,i_n} = \sum_{k=1}^{N_j} \mathsf{A}_{i_1,\ldots,i_{j-1},k,i_{j+1},\ldots,i_n} \mathbf{M}_{i,k}, \tag{1}$$

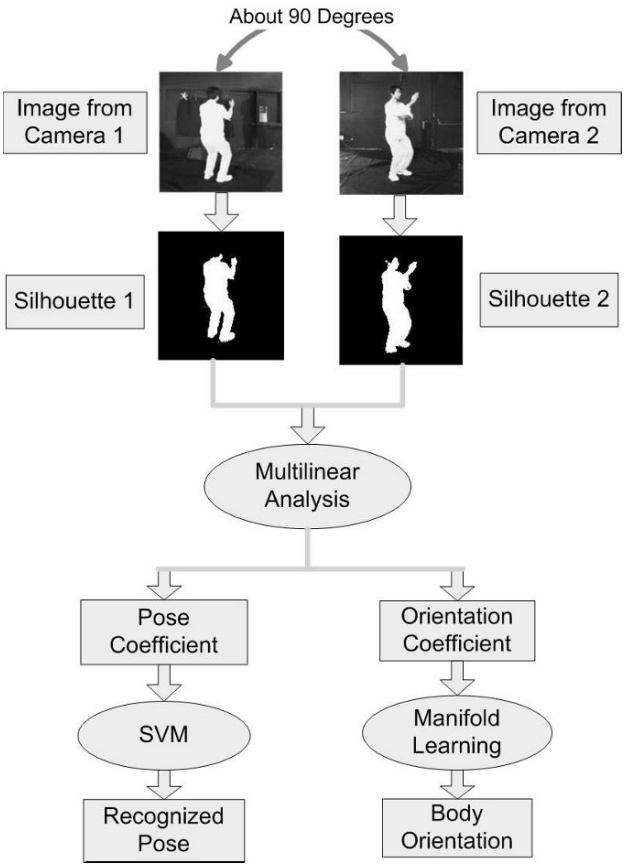



**Fig. 1** An overview of the proposed pose recognition system

in which $i = 1, 2, \ldots, N'_j$. The unfolding matrix representation of mode-$j$ multiplication of A by $\mathbf{M}$ is as follows.

$$\mathbf{R}_{(j)} = \mathbf{M}\mathbf{A}_{(j)}. \tag{2}$$

As an example, the multiplication of a 3-mode tensor and a matrix in each mode is illustrated in Figure 3. When a 3-mode tensor is multiplied by a compatible row vector, it will degenerate into a matrix, as illustrated in Figure 4.

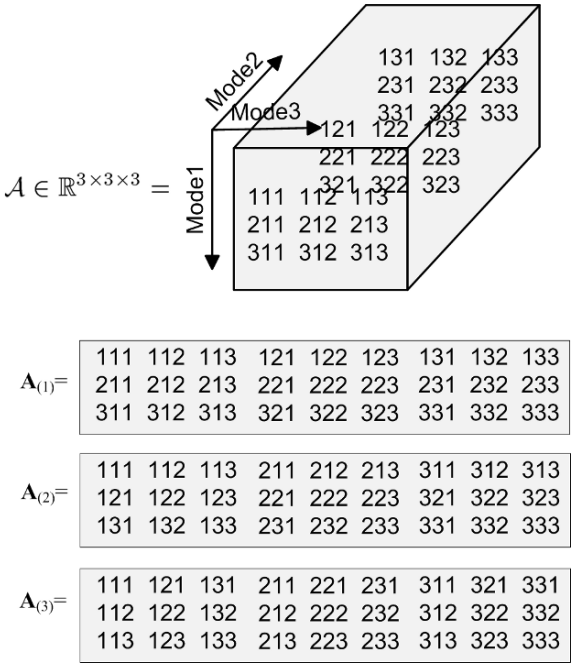



**Fig. 2** Unfolding of a 3-mode tensor

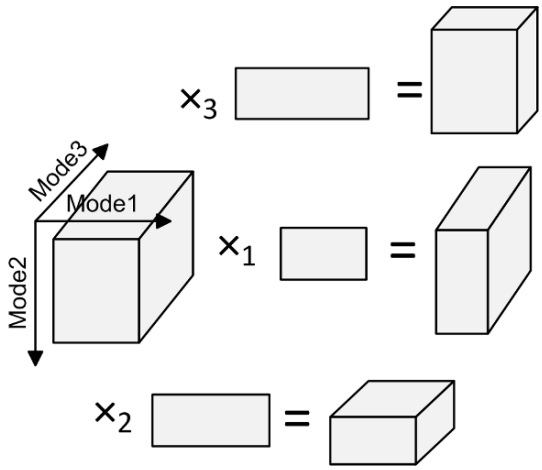



**Fig. 3** Multiplication of a 3-mode tensor with a matrix in each mode

## *3.2 High Order SVD and Multilinear Analysis*

As a generalization of singular value decomposition (SVD) on matrices, we can also perform high order singular value decomposition (HOSVD) [19] on tensors. A tensor $\mathsf{A} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_n}$ can be decomposed into

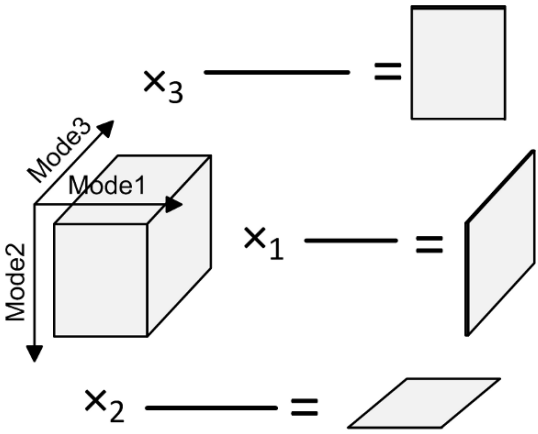



**Fig. 4** Multiplication of a 3-mode tensor with a vector in each mode

$$\mathsf{A} = \mathsf{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n, \tag{3}$$

where $\mathbf{U}_j \in \mathbb{R}^{N_j \times N'_j}$ $(N'_j \leq N_j)$ are mode matrices containing orthonormal column vectors which are analogous to the left and right matrices in SVD. $\mathsf{S} \in \mathbb{R}^{N'_1 \times N'_2 \cdots \times N'_n}$ is called the core tensor which is analogous to the diagonal matrix in SVD.

In order to calculate mode matrices $\mathbf{U}_j\,(j = 1 \ldots n)$, we can first calculate the SVD of the unfolding matrix $\mathbf{A}_{(j)}$. Then $\mathbf{U}_j$ can be obtained by taking the columns of the left matrix of the SVD of $\mathbf{A}_{(j)}$ corresponding to the $N'_j$ largest singular value. Then, the core tensor $\mathsf{S}$ can be calculated as follows.

$$\mathsf{S} = \mathsf{A} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T. \tag{4}$$

Denote $\mathbf{u}_{j,k}$ to be the $k$'th row vector of matrix $\mathbf{U}_j$, then the decomposed tensor possesses the property as follows [6].

$$\mathsf{A}_{i_1,i_2,\ldots,i_n} = \mathsf{S} \times_1 \mathbf{u}_{1,i_1} \times_2 \mathbf{u}_{2,i_2} \cdots \times_n \mathbf{u}_{n,i_n}. \tag{5}$$

Denote $\mathsf{A}(i_1, \ldots, i_{j-1}, :, i_{j+1}, \ldots i_n)$ to be the column vector containing the elements of $\mathsf{A}_{i_1,\ldots,i_j,\ldots,i_n}, i_j = 1 \ldots N_j$. Then we can also get

$$\begin{aligned} &\mathsf{A}(i_1, \ldots, i_{j-1}, :, i_{j+1}, \ldots i_n) \\ &= \mathsf{S} \times_j \mathbf{U}_j \times_1 \mathbf{u}_{1,i_1} \times_2 \mathbf{u}_{2,i_2} \cdots \times_{j-1} \mathbf{u}_{j-1,i_{j-1}} \\ &\quad \times_{j+1} \mathbf{u}_{j+1,i_{j+1}} \cdots \times_n \mathbf{u}_{n,i_n}. \end{aligned} \tag{6}$$

In this way, $\mathsf{A}(i_1, \ldots, i_{j-1}, :, i_{j+1}, \ldots i_n)$ can be represented as a multilinear combination of the column vectors $(\mathsf{S} \times_j \mathbf{U}_j)(i_1, \ldots, i_{j-1}, :, i_{j+1}, \ldots i_n), i_j = 1 \ldots N_j, j = 1 \ldots n$. The coefficients $\mathbf{u}_{k,i_k}$ in each mode can be considered as independent factors

contributing to the data vector, and the interaction of these factors are governed by the tensor $\mathsf{S} \times_j \mathbf{U}_j$.

# 4 Multilinear Analysis of Dance Pose Images

## *4.1 Previous Applications of Multilinear Analysis*

Multilinear factorization has been used successfully in decomposing ensembles of static data such as image and 3D volumetric data, into perceptually independent sources of variations. Previous successful applications include multifactor face image representation in the form of TensorFace [36], modeling of 3D face geometry [38], texture and reflectance [37], and image synthesis for articulated movement tracking [20].The TensorFace framework [36] is a well known framework, which incorporates many factors that affect the resulting face image, such as facial geometry (different person), head pose, and illumination. With mutilinear analysis by tensor decomposition, each of these affecting factors can be analyzed separately.

## *4.2 Tensor Decomposition of Pose Images*

It is the similar case in the situation of pose recognition. The images of a subject performing a pose is affected by the joint angle configuration (different poses), body orientation with respect to the camera system (rotation about the vertical axis), the texture of the subject's cloths and the illumination angle. By taking the binary silhouette of the subject, we can rule out other factors and concentrate on the first two factors affecting the resulting images. Given a new pair of images, coefficient vectors in these two modes can be extracted in a TensorPose framework.

### 4.2.1 Input Data Format

In our system, the input data for training and recognition is a pair of images taken from two cameras whose looking directions are parallel to the ground plane and orthogonal to each other. The purpose of applying two cameras instead of one is to reduce occlusion in some conditions (e.g. side view of poses that all the limbs are in the frontal plane). By setting up two cameras orthogonal to each other, the two captured images can be complementary most of the time.

These cameras are mounted at approximately half body height. Each captured silhouette is normalized to the same height and horizontally centered. The size of each normalized image is $50 \times 50$. The two normalized images are then vectorized and concatenated to form a complete input vector.

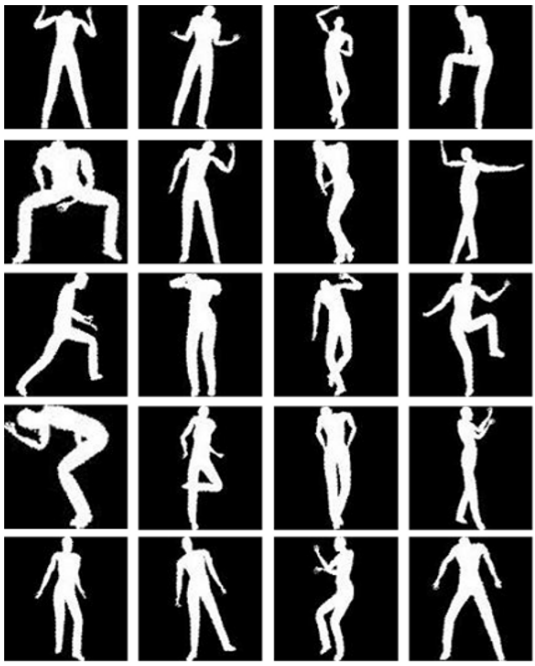


**Fig. 5** 20 dance poses in our training set

### 4.2.2 Formulation of the Tensor

In our approach, we used 20 dance poses (as shown in Figure 5) performed in 16 different orientations (evenly distributed between 0° and 360°) in the training set. The vecotrized images spanned along the orientation mode and pose mode to form a tensor. The dimension of the tensor is $5000 \times 16 \times 20$.

### 4.2.3 Tensor Decomposition using HOSVD

As described in Section 3.2, we performed high order singular value decomposition (HOSVD) on the 3-mode training tensor. The tensor $\mathsf{A} \in \mathbb{R}^{5000 \times 16 \times 20}$ can be decomposed into:

$$\mathsf{A} = \mathsf{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{7}$$

in which $\mathsf{S}$ is the core tensor of the same size as $\mathsf{A}$, $\mathbf{U}_1 \in \mathbb{R}^{5000 \times 5000}$, $\mathbf{U}_2 \in \mathbb{R}^{16 \times 16}$, $\mathbf{U}_3 \in \mathbb{R}^{20 \times 20}$ are orthogonal matrices representing respectively the image pixel mode, orientation mode and pose mode.

Denote

$$\mathsf{D} = \mathsf{S} \times_1 \mathbf{U}_1. \tag{8}$$

The decomposed tensor possesses the property as follows:

$$\mathsf{A}(:, i, j) = \mathsf{D} \times_2 \mathbf{u}_{2i} \times_3 \mathbf{u}_{3j}, \tag{9}$$

where $\mathsf{A}(:, i, j)$ stands for a vector in a tensor that contains the pixels of the image pair of pose $j$ in orientation $i$. $\mathbf{u}_{2i}$ and $\mathbf{u}_{3j}$ are respectively the $i$th row of $\mathbf{U}_2$ and the $j$th row of $\mathbf{U}_3$. Alternatively speaking, $\mathbf{u}_{2i}$ is the coefficient vector representing the $i$th orientation, and $\mathbf{u}_{3j}$ is the coefficient vector representing the $j$th pose.

### *4.3 Pose Coefficient and Body Orientation Coefficient Estimation*

When dealing with a new input vector, one important step of our procedure is to solve for the corresponding orientation coefficient and pose coefficient vectors. This bilinear problem can be defined as

$$\mathbf{z} = \mathsf{D} \times_2 \mathbf{u}_o \times_3 \mathbf{u}_p, \tag{10}$$

where $\mathbf{u}_o$ is the orientation coefficient vector and $\mathbf{u}_p$ is the pose coefficient vector.

This problem can be solved using the alternating linear least squares algorithm [17] as follows. If the orientation coefficient is known to be $\mathbf{u}_{o^*}$, we can denote

$$\mathbf{C}_o = \mathsf{D} \times_2 \mathbf{u}_{o^*}, \tag{11}$$

where $\mathsf{D}$ is degenerated into a matrix $\mathbf{C}_o$ by multiplying a row vector. Insert (11) into (10) we can get

$$\mathbf{z} = \mathbf{C}_o \mathbf{u}_p. \tag{12}$$

Then, $\mathbf{u}_p$ can be easily obtained by solving a linear equation.

Similarly, if the pose coefficient is known to be $\mathbf{u}_{p^*}$, we can get

$$\mathbf{z} = \mathbf{C}_p \mathbf{u}_o, \tag{13}$$

in which

$$\mathbf{C}_p = \mathsf{D} \times_3 \mathbf{u}_{p^*}. \tag{14}$$

Given initial values of $\mathbf{u}_p$ or $\mathbf{u}_o$, we can solve both vectors by solving (12) and (13) alternately.

When applying ALS algorithm, different initial values of $\mathbf{u}_p$ or $\mathbf{u}_o$ may converge to different solutions. Since the ground truth of body orientation angle should be close to one of the standard angles (the maximum deviation is $360/16/2 = 11.25°$), one possible method to find a stable solution is to use each row vector in $\mathbf{U}_2$ (standard orientation vectors) as the initial value of $\mathbf{u}_o$ and solve a set of candidate solu-

tions. We can choose our final solution to be the one with minimum reconstruction error.

However, applying multiple initial values is computationally expensive because the ALS procedure should be performed multiple times. In order to compromise the computational efficiency and the stability of the solution, we have developed an initialization method as follows. Firstly, we use each row vector in $\mathbf{U}_2$ as the initial value of $\mathbf{u}_o$ and solve (12) once for a candidate initial value of $\mathbf{u}_p$. Denote these initial values to be $\mathbf{u}_p^i$,$(i = 1, 2, \ldots 16)$, then the final initial value of $\mathbf{u}_p$ is chosen to be

$$\mathbf{u}_p = \arg \max_{\mathbf{u}_p^i,(i=1,2,\ldots 16)} \left( \max_{j=1,2\ldots 20} \frac{\mathbf{u}_p^i \cdot \mathbf{u}_{3j}}{\|\mathbf{u}_p^i\| \cdot \|\mathbf{u}_{3j}\|} \right). \tag{15}$$

In other words, we choose a initial value of $\mathbf{u}_p$ that is the most similar to one of the standard poses to initialize ALS. Since we choose only one initial value, the computational efficiency is much higher, and the solutions obtained using this initialization method performed stably in the experiments.

### *4.4 Traditional Methods of Tensor-based Recognition*

When both coefficient vectors $\mathbf{u}_o$ and $\mathbf{u}_p$ are solved, the traditional recognition procedure [6] is to compare the solved coefficients with the standard coefficients, namely the row vectors of $\mathbf{U}_2$ and $\mathbf{U}_3$. For example, if the difference between $\mathbf{u}_p$ and $\mathbf{U}_{3j}$ is below certain threshold, then the input can be recognized as pose $j$. To deal with torso orientation estimation, what traditional methods can do is to classify the orientation coefficient vector into one of the 16 classes and find the corresponding standard orientation angle. This classification can be done by finding the nearest neighbor in the row vectors of $\mathbf{U}_2$.

As mentioned in Section 2, the traditional procedure is limited in the sense that it cannot incorporate variations in the same class and it can only solve discrete values for a continuous quantity. Also, we can see that it's hard to find a solid rule to determine the threshold for recognition, and in turn to distinguish outliers from good poses. Dealing with orientation estimation, traditional methods offers only discrete solutions. In the following two sections, we will discuss our proposed SVM and manifold learning methods to overcome these limitations.

## 5 SVM-based Pose Recognition Using Pose Vectors

The aim of pose recognition is to classify an input into one of the 20 standard poses or identify it as an outlier. In order to achieve this task more effectively, we integrated our TensorPose framework with SVM classifiers. The input data fed into

SVM classifiers are the pose coefficient vectors, as described in Section 2.2, extracted from a pair of input image. Therefore, the 20-dimensional pose vectors actually serve as the descriptor of poses.

## *5.1 Formation of Training Data*

The silhouette images used in our training sets were synthesized using Maya animation software. Each pose was performed twice by a professional dancer, and the joint angle data of the subject was captured by a marker based mocap system. In addition to the 20 standard poses, we also captured some "trick poses", which means the poses that are somehow similar to one of the standard poses but cannot be accepted as that pose by the dancer. Using the captured skeleton (joint angle) data, we generated silhouettes of the same pose with different 3D surface models and different torso orientations in Maya.

In our standard tensor, only one pair of image can represent each pose in each orientation. In order to incorporate some individual differences at this stage, we applied 3 human surface models and generated 6 pairs of images for each pose in each orientation. The final representing data to put into the tensor is obtained by taking average of these pairs of images.

In order to train the SVM classifiers, training sets containing a reasonable number of samples are required. In the process of tensor formation, we synthesized 6 pairs of images for each pose in each of the 16 orientation. Therefore, there are in total 96 pairs of images for one pose and in turn 96 pose vectors can be obtained. We refer to these vectors as "standard pose vectors".

In practical case, a range of variation in joint angle configuration should be tolerated for each pose. To incorporate these variations, we added small Gaussian noise into the joint angle data of poses and synthesized 96 pairs of images for each pose, 32 using each human surface model. Therefore, another 96 pose vectors can be obtained for each pose. We refer to these vectors as "noisy pose vectors".

We also synthesized 36 pairs of images for the "trick poses" of each pose, so similarly 36 coefficient vectors can be obtained. We refer to these vectors as "trick pose vectors". These 3 sets of vectors composed the training sets of our SVM classifiers.

## *5.2 Pose Recognition Using SVM*

In order to accomplish the task of pose recognition, we trained 20 SVM classifiers that perform classification of data into 2 classes.

For each pose, we train a binary SVM classifier to identify whether the input "is" or "is not" the pose. The training set of the classifier consists of positive samples and negative samples. Positive samples includes the standard and noisy pose vectors of

the corresponding pose, and negative samples includes the standard pose vectors of all other 19 poses and the trick pose vectors of the corresponding pose.

When all the 20 classifiers are trained, pose recognition can be achieved by a traversal of all the classifiers and see which one accepts the input as the corresponding pose. If none of the 20 classifiers accepts an input, it will be identified as an outlier. With this algorithm, there is possibility that one input may be recognized as more than one pose. However, if the classifiers are properly trained, it can be justified by the experimental results that this possibility will be very low. Even if this situation does happen, we can solve this problem by referring to temporal and spatial context in practical situations.

## 6 Body Orientation Estimation Using Orientation Vector through Manifold Learning and Nonlinear Minimization

Once a pose has been recognized from a pair of input images, we need to estimate the body orientation angle $\theta$ of the performer in the camera coordinate system from the body orientation vector. In our approach, we first construct a 1D manifold using body orientation vectors of training data, which is essentially a cubic spline of the body orientation angle. Then, we solve the body orientation angle estimation problem using nonlinear least square techniques.

A body orientation space $\Psi$ can be obtained through the tensor decomposition and projection presented in Section 3. Since the training data was obtained from 16 body orientation angles, $\Psi$ is a 16-dimensional space. As discussed previously, this body orientation is actually describing the rotation angle $\theta$ of the performer's body about the axis perpendicular to the ground plane, which has only 1 degree-of-freedom. Hence $\Psi$ forms a 1D manifold, which can be recovered as a function of $\theta$. Recovery of such a 1D manifold has been addressed by other researchers [20]. In our approach, we follow the method described in [20]. Essentially, we fit a cubic spline as a function of $\theta$ in the 16D $\Psi$, so that a body orientation coefficient vector can be represented as $\psi_i = g(\theta_i)$. This step is implemented using the CSAPE Cubic spline interpolation function in MATLAB. Figure 6 shows the resulting $\theta$-parameterized 1D spline function embedded in the first three dimensional subspace of $\Psi$. In Figure 6, The red circles are the centers of the $\psi'$s for different views, denoted by the corresponding $\theta'$s.

Given a new body orientation coefficient vector $\psi^*$, we would like to estimate the corresponding body orientation angle $\theta^*$. We cast this problem into a nonlinear least square problem using the Mahanalobis distance as the cost function. The problem is defined as

$$\theta^* = arg\min_{\theta}(g(\theta) - \psi^*)\Sigma^{-1}(g(\theta) - \psi^*)^T, \tag{16}$$

where $\Sigma$ is the covariance matrix of the variations of the body orientation coefficient vectors for the same pose. Let $\psi_k^{(n)}$ be the body orientation coefficient vector of the

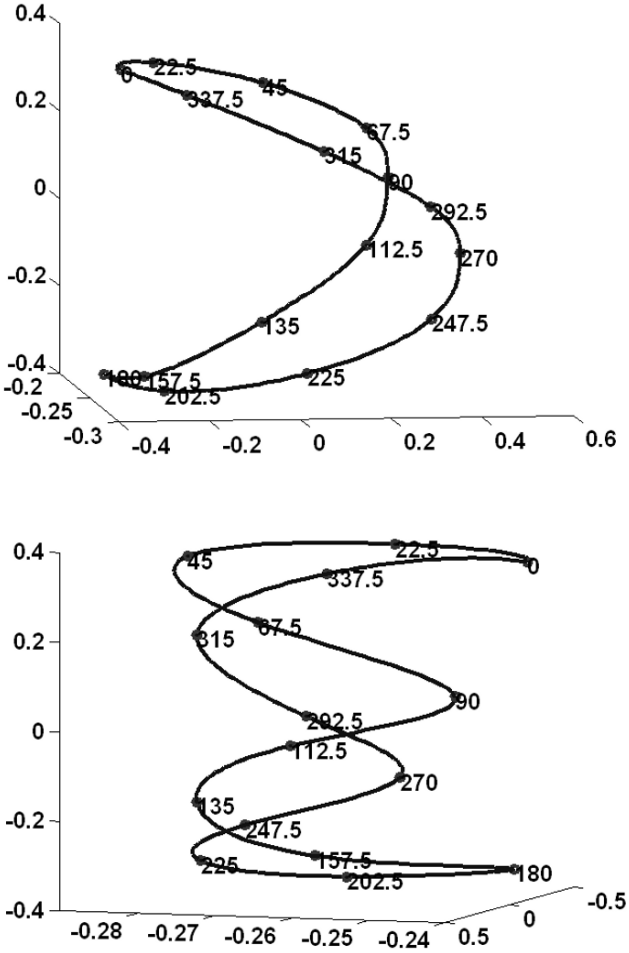



**Fig. 6** 1D spline embedding of the body orientation space $\Psi$ (the first three dimensions are shown here in 2 view angles)

$n$'th sample training data of the $k$'th view, where $n = 1,\ldots,N$, and $k = 1,\ldots,K$. $\Sigma$ is then obtained from the training data as the following:

$$\Sigma = \frac{1}{K(N-1)} \sum_{k=1}^{K} \sum_{n=1}^{N} (\psi_k^{(n)} - \bar{\psi}_k)^T (\psi_k^{(n)} - \bar{\psi}_k), \tag{17}$$

$$\bar{\psi}_k = \sum_{n=1}^{N} \psi_k^{(n)}. \tag{18}$$

We applied Levenberg-Marquardt method [23] to solve the least square problem. The initialization is performed using the result of tensor-based classification of orientation vector, as described in Section 3.

## 7 Experimental Results

We tested our proposed system with both synthetic silhouette images and silhouettes extracted from real images. The performances of our system in both cases were analyzed.

### *7.1 Criteria of Performance Analysis*

We evaluated the performance of our system in both pose recognition tasks and orientation angle estimation tasks.

For pose recognition tasks, we evaluated the performances by recognition rate, ambiguity rate and false detection rate. Assume that in a testing set, there are $N_p$ pairs of images of good poses and $N_n$ pairs of images of "trick poses". Define $N_rec$ to be the number of inputs our system has correctly recognized, and $N_a$ to be the number of inputs that are correctly recognized but with ambiguity (more than one pose is assigned to it). Then the recognition rate $R_{rec}$ can be calculated as $N_r/N_p$, and the ambiguity rate $R_{amb}$ can be calculated as $N_a/N_r$. Define $N_f$ to be the number of the inputs that are actually not the standard poses while being recognized as good poses. Then the false positive rate $R_{fp}$ can be calculated as $N_f/N_n$.

For body orientation estimation tasks, we compared the estimated angle with the ground truth data and evaluated the error.

### *7.2 Experiments on Synthetic Images*

Using Maya software, we synthesized 640 pairs of good pose images and 640 pairs of non-pose images. There are 32 pairs of each good pose, 2 in each orientation. To test the sensitivity of the system to body orientation change, we applied different orientation angles in testing images from the training images by shifting each training orientation 10 degrees. We also added a small white Gaussian noise ($\sigma^2 = 4$ ) on each joint angle, in order to better simulate the realistic situation. With these images, the recognition rate of our system is 99.53% with 0 ambiguity rate. The false positive rate is 6.09%. The performance of body orientation estimation with synthetic data is shown in Figure 7. The mean estimation error is $4.22^\circ$ with a standard deviation of $6.07^\circ$ .

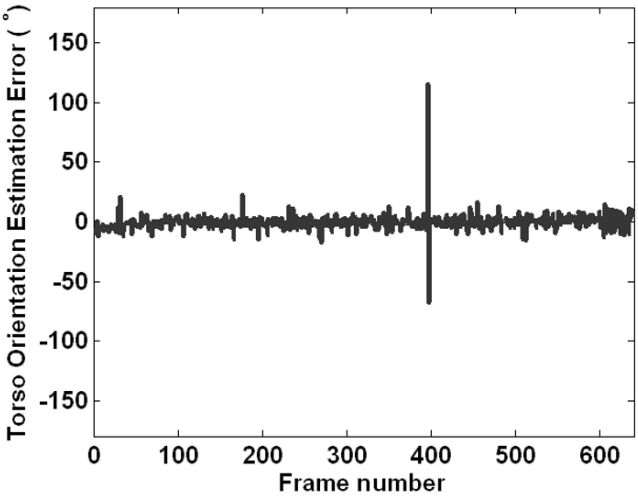



**Fig. 7** Body orientation estimation performance with synthetic data

## *7.3 Experiments on Real Images*

To evaluate how our system performs in practical situations, we had a non-dancer in our lab to perform the 20 standard poses as well as some trick poses. The images of the poses were taken from two 1394 video cameras mounted with about 90 degrees with each other. For each pose and trick pose, the subject performed 4 times in difference orientations in the space. Therefore, there are totally 80 pairs of images of standard poses and 80 pairs of trick poses.

### 7.3.1 Pose Recognition Results of the Proposed System

We have performed pose recognition on real data with our proposed method. In the process of SVM classification, we applied the RBF kernel defined as follows.

$$k(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}. \tag{19}$$

We applied different configurations of the parameter $\sigma$ and the regulation constant $C$. The results are listed in Table 1.

**Table 1** Pose recognition results of the proposed system using different parameter settings

| $\sigma$ | $C$ | $R_{rec}$ | $R_{amb}$ | $R_{fp}$ |
|---|---|---|---|---|
| 25 | 5 | 65.00% | 0% | 0% |
| 24 | 6 | 77.50% | 0% | 2.50% |
| 22 | 6 | 83.75% | 0% | 5.00% |
| **24** | **8** | **90.00%** | **0%** | **7.50%** |
| **21** | **7** | **93.75%** | **0%** | **10.00%** |
| **25** | **11** | **95.00%** | **0%** | **11.25%** |
| 17 | 20 | 96.25% | 1.3% | 15.00% |

### 7.3.2 Comparison with Existing Pose Recognition Methods

We have compared our system with some existing pose recognition systems in terms of gesture set used, method applied and recognition results. The results are listed in Table 2. The working conditions and data sets of these systems are different from ours, but it can be shown that our system is comparable to existing systems.

### 7.3.3 Comparison with Existing Tensor-based Recognition Methods

We have also compared our method with some other methods of recognition using coefficient vectors obtained by multilinear analysis. The methods we compared with were traditional Fixed-Threshold (FT) method (with different threshold of angular differences), as defined in Section 3.4, and probabilistic von Mises-Fisher (vMF) method [1] (with different probability threshold). The recognition rate versus false positive rate plot of the three methods is shown in Figure 8. It can be shown that the proposed method outperforms the existing tensor-based methods.

### 7.3.4 Comparative Test on Body Orientation Estimation

In order to obtain the ground truth data of body orientation in real images, we aligned the video camera system with the marker-based motion capture system. A set of 4 makers is imposed on the back of the performer. When each pair of images were captured, the motion capture system obtained the 3D coordinates of the markers at the same time, and afterwards the ground-truth data of body orientation can be calculated with these coordinates.

The body orientation estimation performance on real data is shown in Figure 9. The mean and standard deviation of estimation error is listed in Table 3 and compared with Nearest-Neighbor (NN) Method, as mentioned in Section 3.4.

In practice, it is possible that the orientation estimation error of some inputs are around 180 degrees. This happened to only 1 input in our testing set, as shown in

**Table 2** Pose recognition results compared with existing systems

| System | Pose Set | Feature Extraction | Recognition Method | Orientation Invariance | $R_{rec}$ | $R_{fp}$ |
|---|---|---|---|---|---|---|
| Urano, 2004 [34] | 6 daily-life poses | Outline Diameter & HLAC features | Hieratical feature matching | Yes | 89.7% | N.A. |
| Chu, 2005 [5] | 12 hand postures | Visual Hull & 3-D Shape Descriptor | Matching Posture atoms | Yes | 87.8% | N.A. |
| Li, 2006 [24] | Stand and sit | Geometric Measure-ment | Rule based | No | 94.7% | N.A. |
| Boulay, 2006 [2] | 8 daily-life poses | H.&V. projection | Comparison with 3-D model generated silhouettes | No | 74% | N.A. |
| Guo, 2006 [8] | 20 dance poses | Gaussian Mixture Model | RVM | Yes | 81.9% | 14% |
| Hu, 2007 [11] | 8 daily-life poses | Fourier descriptor of 2-D contour | Similarity measure-ment | Yes | 89.6% | N.A. |
| Guo, 2008 [9] | 9 daily-life poses | Histogram projection & universal eigenspace | Hieratical matching | No | 92.67% | N.A. |
| **The proposed system (Parameter I)** | **20 dance poses** | **Multilinear analysis** | **SVM ($\sigma$ = 24, $C$ = 8)** | **Yes** | **90.00%** | **7.5%** |
| **The proposed system (Parameter II)** | **Same as above** | **Same as above** | **SVM ($\sigma$ = 21, $C$ = 7)** | **Yes** | **93.75%** | **10.00%** |
| **The proposed system (Parameter III)** | **Same as above** | **Same as above** | **SVM ($\sigma$ = 25, $C$ = 11)** | **Yes** | **95.00%** | **11.25%** |

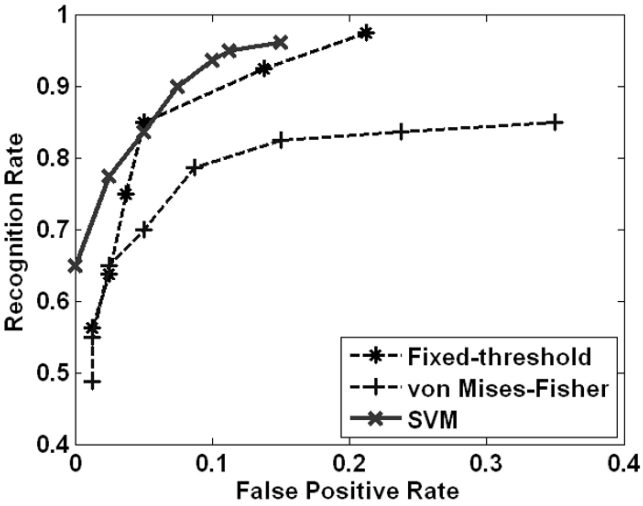



**Fig. 8** Pose recognition results compared with existing tensor-based methods

Figure 9. Such 180° ambiguity is inherent. One example of such scenario is a subject performing pose #1 facing the middle point of the two camera centers. When an ambiguous pose and torso orientation pair is obtained by the system, all the associated solutions with similar silhouettes will be returned. If 180° ambiguity is removed, the mean error of orientation estimation drops to 15.78° and the standard deviation drops to 14.45.

**Table 3** Comparison of our recognition results with Nearest Neighbor Method.

| | Mean Error | Standard Deviation |
|---|---|---|
| NN Estimation | 25.98° | 32.51 |
| **The Proposed Method** | **17.99°** | **23.16** |

## *7.4 Computational Efficiency*

The system was implemented on a PC with Intel Xeron 3.6GHz CPU using Matlab without any optimization. When running, the system takes only 35% of CPU. Under this condition, the average time to process each frame is 0.68 seconds, while under similar conditions our previous approach [8] needs 2.21 s/frame. We are developing

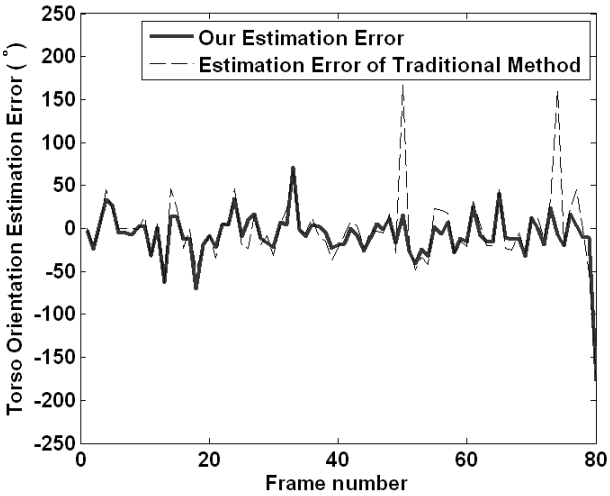



**Fig. 9** The body orientation estimation performance on real data

a C++ implementation that will greatly speed up the system, e.g. to 10 Hz or higher. Such processing rate will be sufficient for most interactive applications requiring on-line pose recognition.

## 8 Conclusions and Future Work

In this chapter, we present an effective pose recognition and body orientation system based on multilinear analysis. This system performed very well in our experiments on both synthetic and real data.

Our current system has a great potential of future improvement because of our effort to incorporate variations in the tensor-based framework. In the future we may improve the robustness of the system by incorporating probabilistic methods in the TensorPose framework.

**Acknowledgements** The authors thank Bill T. Jones for choreograph the dance poses used in this research. We also thank Stjepan Rajko for providing the data for system training and testing. This chapter is based upon work partly supported by National Science Foundation on CISE-RI No. 0403428 and IGERT No. 0504647.[1]

[1] Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

## References


1. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von mises-fisher distributions. The Journal of Machine Learning Research **6**, 1345–1382 (2005)
2. Boulay, B., Bremond, F., Thonnat, M.: Applying 3d human model in a pose recognition system. Pattern Recognition Letters **27**(15), 1788–1796 (2006)
3. Camurri, A., Hashimoto, S., Ricchetti, M., Ricci, A., Suzuki, K., Trocca, R., Volpe, G.: Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. Computer Music Journal **24**(1), 57–69 (2000)
4. Cheung, K.M., Baker, S., Kanade, T.: Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: Proc. CVPR, pp. 77–84 (2003)
5. Chu, C., Cohen, I.: Pose and gesture recognition using 3d body shapes decomposition. In: Proc. CVPR, pp. 69–78 (2005)
6. Elden, L.: Matrix Methods in Data Mining and Pattern Recognition. SIAM, Philadelphia (2007)
7. Elgammal, A., Lee, C.: Inferring 3d body pose from silhouettes using activity manifold learning. In: Proc. CVPR, vol. 2, pp. 681–688 (2004)
8. Guo, F., Qian, G.: Dance pose recognition using wide-baseline orthogonal stereo cameras. In: Proc. FGR, pp. 481–486 (2006)
9. Guo, P., Miao, Z., Yuan, Y.: Posture and activity recognition using projection histogram and pca methods. In: Proc. CISP, pp. 397–401 (2008)
10. Howe, N.R.: Silhouette lookup for monocular 3d pose tracking. Image and Vision Computing **25**(3), 331–341 (2007)
11. Hu, J.S., Su, T.M., Lin, P.C.: 3-d human posture recognition system using 2-d shape features. In: Proc. ICRA, pp. 3933–3938 (2007)
12. Huang, F., Di, H., Xu, G.: Viewpoint insensitive pose representation for action recognition. In: Proc. AMDO, pp. 143–152 (2006)
13. Mikic, I., Trivedi, M.M., Hunter, E., Cosman, P.C.: Human body model acquisition and tracking using voxel data. International Journal of Computer Vision **53**(3), 199–223 (2003)
14. James, J., Ingalls, T., Qian, G., Olsen, L., Whiteley, D., Wong, S., Rikakis., T.: Movement-based interactive dance performance. In: Proc. MULTIMEDIA, pp. 470–480 (2006)
15. Jenkins, O.C., González, G., Loper, M.M.: Tracking human motion and actions for interactive robots. In: HRI '07: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, pp. 365–372. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1228716.1228765
16. Kakadiaris, I.A., Metaxas, D.: Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In: Proc. CVPR, pp. 81–87 (1996)
17. Kiers, H.A.L.: An alternating least squares algorithms for parafac2 and three-way dedicom. Computational Statistics & Data Analysis **16**(1), 103–118 (1993)
18. Kjolberg, J.: Designing full body movement interaction using modern dance as a starting point. In: Proc. DIS, pp. 353–356 (2004)
19. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications **21**(4), 1253–1278 (2000)
20. Lee, C.S., Elgammal, A.: Modeling view and posture manifolds for tracking. In: Proc. ICCV, pp. 1–8 (2007)
21. Lee, M.W., Nevatia, R.: Integrating component cues for human pose tracking. In: Proc. Joint IEEE Int. Workshop on VS-PETS, pp. 41–48 (2005)
22. Lee, S.W.: Automatic gesture recognition for intelligent human-robot interaction. In: Proc. FGR, pp. 645–650 (2006)
23. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. The Quarterly of Applied Mathematics **2**, 164–168 (1944)
24. Li, C.C., Chen, Y.Y.: Human posture recognition by simple rules. In: Proc. SMC, pp. 3237–3240 (2006)

25. Li, R., Yang, M.H., Sclaro, S., Tian, T.P.: Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In: Proc. ECCV, pp. 137–150 (2006)
26. Mitra, S., Acharya, T.: Gesture recognition: A survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews **37**(3), 311–324 (2007)
27. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding **81**(3), 231–268 (2001)
28. Moeslund, T.B., Hilton, A., Kruger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding **104**(2), 90–126 (2006)
29. Navaratnam, R., Thayananthan, A., Torr, P., Cipolla, R.: Hierarchical part-based human body pose estimation. In: Proc. British Machine Vision Conference (2005)
30. Ong, S.C., Ranganath, S.: Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(6), 873–891 (2005). DOI 10.1109/TPAMI.2005.112
31. Qian, G., Guo, F., Ingalls, T., Olson, L., James, J., Rikakis, T.: A gesture-driven multimodal interactive dance system. In: Proc. ICME, pp. 1579–1582 (2004)
32. Rosales, R., Sclaro, S.: Learning body pose via specialized maps. In: Proc. Conference on Neural Information Processing Systems, pp. 1263–1270 (2002)
33. Sul, C., Lee, K., Wohn, K.: Virtual stage: A location-based karaoke system. IEEE Multimedia **05**(2), 42–52 (1998)
34. Urano, T., Matsui, T., Nakata, T., Mizoguchi, H.: Human pose recognition by memory-based hierarchical feature matching. In: Proc. SMC, pp. 6412–6416 (2004)
35. Urtasun, D.F.R., Fua, P.: 3d people tracking with gaussian process dynamical models. In: Proc. CVPR, pp. 238–245 (2006)
36. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In: Proc. ECCV, pp. 447–460 (2002)
37. Vasilescu, M.A.O., Terzopoulos, D.: Tensortextures: Multilinear image-based rendering. ACM Transactions on Graphics **23**(3), 334–340 (2004)
38. Vlasic, D., Brand, M., Pfister, H., Popovi, J.: Face transfer with multilinear models. In: Proc. ACM SIGGRAPH, pp. 426–433 (2005)
39. Wang, W.L., Tan, T.: Recent development in human motion analysis. Pattern Recognition **36**, 585–601 (2003)
40. Wu, Y., Huang, T.S.: Vision-based gesture recognition: A review. In: GW '99: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction, pp. 103–115. Springer-Verlag, London, UK (1999)

# Applications of Multiview Tensors in Higher Dimensions

Marina Bertolini, GianMario Besana, and Cristina Turrini

**Abstract** This chapter is devoted to applications of multiview tensors, in higher dimension, to projective recostruction of segmented or dynamic scenes. Particular emphasis is placed on the analysis of critical configurations and their loci in this context, i.e. configurations of chosen scene-points and cameras that turn out to prevent successful reconstruction or allow for multiple possible solutions giving rise to ambiguities. A general geometric set up for higher dimensional spaces ad projections is firstly recalled. Examples of segmented and dynamic scenes, interpreted as static scenes in higher dimensional projective spaces, are then considered, following Shashua and Wolf. A theoretical approach to multiview tensors in higher dimension is presented, according to Hartley and Schaffalitzky. Using techniques of multilinear algebra and proper formalized language of algebraic geometry, a complete description of the geometric structure of the loci of critical configurations in any dimension is given. Supporting examples are supplied, both for reconstruction from one view and from multiple views. In an experimental context, the following two cases are realized as static scenes in $\mathbb{P}^4$: $3D$ points lying on two bodies moving relatively to each other by pure translation and $3D$ points moving independently along parallel straight lines with constant velocities. More explicitly, algorithms to determine suitable tensors used to reconstruct a scene in $\mathbb{P}^4$ from three views are implemented with MATLAB. A number of simulated experiments are finally performed in order to prove instability of reconstruction near critical loci in both cases described above.

Marina Bertolini
Università degli Studi di Milano, Dipartimento di Matematica Via Saldini 50 20133 Milano, Italy
e-mail: marina.bertolini@unimi.it

GianMario Besana
DePaul University, College of Computing and Digital Media, 243 S Wabash, Chicago IL USA
e-mail: gbesana@depaul.edu

Cristina Turrini
Università degli Studi di Milano, Dipartimento di Matematica Via Saldini 50 20133 Milano, Italy
e-mail: cristina.turrini@unimi.it

# 1 Introduction

The analysis of certain dynamic and segmented scenes has recently brought to the fore the consideration of projective geometry in higher dimensional spaces, as for example in [7], [12], [14], [15], [21], [22], [23]. Different approaches lead to different contexts, but the process follows the same essential steps that can be summarized as follows. First, a mathematical framework is developed to encode the dynamic or segmented scenes, in which higher dimensional spaces play a key role. As a second step, within this framework, necessary tools to perform projective recostruction, given a set of marked corresponding points on multiple images, are introduced. As a third step, algorithms to concretely compute the necessary tools are designed and implemented. Finally one addresses the problem of identifying critical configurations of points for which projective recostruction is not possible in a unique way, up to projective equivalence.

As far as the first two steps are concerned, in one approach, Wolf and Shashua, [23], extended the use of classical projection matrices for projections from $\mathbb{P}^3$ to $\mathbb{P}^2$ to higher-dimensional mappings from $\mathbb{P}^k$ to $\mathbb{P}^2$, for $k = 4, 5, 6$, with the purpose of giving a static, and hence more manageable, representation in a higher dimensional space of some dynamic or segmented scenes of the usual space. In this approach, multiview tensors expressing linear constraints between multiple views are introduced and Hartley and Schaffalitzky, [14], gave a general theoretical approach for the construction of multiview tensors for projections from $\mathbb{P}^k$ to $\mathbb{P}^h$, for all $k$ and $h$, clearly describing what kind of multilinear constraints one may use for this purpose.

In another approach, Vidal et al, [7], [12], [21], [22], analyzed scenes with $n$ segmented bodies. Their approach leverages a classical technique in algebraic geometry in which the Veronese embeddings of low dimensional spaces into higher dimensional ones are introduced in order to transform higher degree constraints into linear ones. The corresponding tools introduced in this approach are the multibody fundamental matrix and multibody tensors.

Algorithms to obtain classical multifocal tensors in $\mathbb{P}^3$ are well established, see for example [13]. In higher dimensions one contribution, following guidelines given in [23], is contained in [4].

Finally, concerning the last step in the process, critical configurations for projective recostruction of various types of static scenes in 3-space have been the object of interest for several authors for quite some time. In the case of simple camera calibration (one view) the classical result of the criticality of a twisted cubic curve goes back to [6]. Quadric surfaces were shown to be critical for two views in [17], [19]. Contributions in the case of three or more views are found in [10], [16], and [20]. A comprehensive, detailed analysis both in the case of two and in the case of multiple views was conducted in [11].

In the context of projections from higher dimensional spaces to $\mathbb{P}^2$, interest in understanding critical configurations has recently grown. Critical configurations and their loci in the case of one view in any dimension were theoretically described in [5]. Experimental evidence of the instability of reconstruction near such critical loci

was given in [2]. A general framework to understand critical loci for projective recostruction from multiple view in higher dimensions was given in [3].

This chapter fits into a long term project the authors are conducting with twofold purpose: determining in detail the algebro-geometric structure of critical loci for projective reconstruction from multiple views for projections from $\mathbb{P}^k$ to $\mathbb{P}^h$; giving experimental evidence of instability of reconstruction for configurations that are close to being critical.

The structure of this chapter follows the same pattern as the process described in this introduction. More specifically, the mathematical framework for some segmented and dynamic scenes which admit a representation as static scenes in $\mathbb{P}^4$, following [23], is presented in Section 3. Multiview tensors, which are the necessary tools for projective recostruction in this framework, are presented in Section 4. Algorithms to compute multiview tensors are presented in Section 5. Section 6 is focused on the theoretical presentation of critical configurations and their loci. Examples of critical loci are then computed in Section 7. Experimental evidence of instability of the reconstruction from configuration of points that are close to being critical is presented in Section 8.

## 2 Notation and Background Material

$\mathbb{P}^k$ denotes the $k-$dimensional real (or complex) projective space. Whenever multiplication by a nonzero scalar is utilized, the scalar will be real or complex accordingly. Once a projective frame is chosen, coordinate vectors $\mathbf{X}$ of points of $\mathbb{P}^k$ are written as columns, thus their transpose are $\mathbf{X}^T = (X_1, \ldots, X_{k+1})$. A linear projective subspace $\Lambda \subseteq \mathbb{P}^k$ spanned by $m+1$ linearly independent points will be called $m$-space or subspace of dimension $m$. By convention the empty set is considered as a $(-1)$-space. Given an $m_1$-space $\Lambda_1$ and an $m_2$-space $\Lambda_2$ in $\mathbb{P}^k$, the join of $\Lambda_1$ and $\Lambda_2$ is the smallest projective linear subspace $<\Lambda_1, \Lambda_2> \subseteq \mathbb{P}^k$ containing both of them. The dimensions of $\Lambda_1, \Lambda_2, <\Lambda_1, \Lambda_2>$ and $\Lambda_1 \cap \Lambda_2$ are linked by Grassmann's formula:

$$\dim \Lambda_1 + \dim \Lambda_2 = \dim <\Lambda_1, \Lambda_2> + \dim(\Lambda_1 \cap \Lambda_2). \tag{1}$$

For the convenience of the reader, we fix our notation for cameras, centers of projection and multiple views in the context of projective recostruction. As usual, a scene is a set of $N$ points $\mathbf{X_i} \in \mathbb{P}^k$. A *camera* is represented as a central projection $P$ of points in $k$-space, from a linear center $C_P$, onto a suitable $\mathbb{P}^h$, $h < k$, where in the traditional setting of real still images are $k = 3$ and $h = 2$. We usually do not make any formal distinction between the projection *map* $P$ and one of its matrix representations, for which we use the same symbol $P$. Accordingly, if $\mathbf{X}$ is a point in $\mathbb{P}^k$, we denote its image in the projection equivalently as $P(\mathbf{X})$ or $P \cdot \mathbf{X}$. In homogeneous coordinates, the projection mapping $P : \mathbb{P}^k \setminus \{C_P\} \to \mathbb{P}^h$ is described by $\mu \mathbf{x} = P \cdot \mathbf{X}$, where $\mu$ is a non-zero constant and $P$ is a $(h+1) \times (k+1)$-matrix with maximal rank $\mathrm{rk}(P) = h+1$. The center of projection $C_P$ is the right annihilator of

$P$, hence a $(k-h-1)$-space. For a given point $\mathbf{X} \in \mathbb{P}^k$, the projecting ray, i.e. the join $< C_P, \mathbf{X} >$ is a $(k-h)$-space. While in all our applications we will consider projections from $\mathbb{P}^k$ to several spaces all of the same dimension, the general set-up allows for projections $Pj : \mathbb{P}^k \setminus \{C_P\} \rightarrow \mathbb{P}^{h_j}$ to spaces of different dimensions. In this context we will denote by $\mathbf{x_{ij}}$ the image $\mathbf{x_{ij}} = Pj(\mathbf{X_i})$ of the point $\mathbf{X_i}$ under the map $Pj$. Two different images $\mathbf{x_{ij}}$ and $\mathbf{x_{im}}$ of the same point $\mathbf{X_i}$ are *corresponding points* while $< C_{Pj}, \mathbf{X_i} >$ and $< C_{Pm}, \mathbf{X_i} >$ are *corresponding rays*. More generally, $r$ linear subspaces $S_i \subset \mathbb{P}^{h_i}$, $i = 1, \ldots, r$ are said to be *corresponding* if there exists at least one point $\mathbf{X} \in \mathbb{P}^k$ such that $Pi(\mathbf{X}) \in S_i$ for $i = 1, \ldots, r$.

Given a matrix $A = [a_{i,j}]$ with real or complex entries, $A^T$ denotes its transpose and, for a real matrix $A$, $||A|| = (\sum_{i,j} a_{ij}^2)^{1/2}$ denotes its Frobenius norm. The $i$-th row of $A$ is denoted $\mathbf{A}^i$, while the $i$-th column is denoted by $\mathbf{A}_i$. Given two matrices $A$ and $B$, with the same number of rows, $[A|B]$ denotes the matrix obtained by juxtaposing the columns of B after the columns of A. Analogously, for a $(3,3,3)$ real tensor $T = [t_{i,j,k}]$, $||T||$ denotes its norm $(\sum_{i,j,k} t_{ijk}^2)^{1/2}$. As customary, see [13], $\varepsilon_{ijk}$ denotes the $(3,3,3)$ tensor with zero elements unless $(ijk)$ is an even (respectively odd) permutation of $(123)$ in which case $\varepsilon_{ijk} = 1$, (respectively $-1$).

In the following sections we make use of terminology and basic definitions from Algebraic Geometry for which we refer the reader to [9].

## 3 Higher Dimensional Spaces As Frameworks For Some Dynamic And Segmented Scenes

As mentioned in the introduction, the first step in the analysis of certain dynamic and segmented scenes is the development of a mathematical framework.

In the context of projective recostruction of such scenes, a natural approach is to increase the dimension of the ambient space by treating several parameters (e.g. velocity, relative displacement of two segmented bodies in translation) used to describe dynamic or segmented aspects of the scene as additional coordinates. This is essentially the approach of [23] and we present here two examples with natural ambient space $\mathbb{P}^4$. Example i) could be easily generalized to cases in which velocity vectors span a 2D or 3D space, thus giving rise to models in $\mathbb{P}^5$ or $\mathbb{P}^6$. The interested reader can find details in [23] and [5].

In this approach then, successful reconstruction of the scene in $\mathbb{P}^k$ corresponds then to recovering in $\mathbb{P}^3$ both the starting position of the points and values of the other parameters utilized as additional coordinates, hence critical configurations found in $\mathbb{P}^k$ account for positions and other parameters which are critical for the dynamic scene in $\mathbb{P}^3$.

i) *Three dimensional dynamic scene* $k = 4$. A significant example leading to projections from $\mathbb{P}^4$ involves a $3D$ configuration of points $\{\mathbf{X_i}\}$ where each point moves independently along a straight-line path with constant velocities $\lambda_i$ and all trajectories are parallel to each other, i.e. the linear span of all velocity vec-

tors is one-dimensional. Following Shashua and Wolf [23], let $P(t)$ denote the projection matrix from $\mathbb{P}^3$ to $\mathbb{P}^2$, at time $t$, and assume that the common direction of the trajectories is given in $\mathbb{R}^3$ by the unit vector $(dx_0, dy_0, dz_0)^T$. If $(x,y,z)^T$ are affine coordinates in $\mathbb{R}^3$ and point $\mathbf{X_i}$ starts at $(x,y,z)^T$ at time $t=0$, then the homogeneous coordinates of point $\mathbf{X_i}$ at time $t$, in $\mathbb{P}^3$, are $\mathbf{X_i} = (x+t\lambda_i dx_0, y+t\lambda_i dy_0, z+t\lambda_i dz_0, 1)^T$. One can embed this case in $\mathbb{P}^4$, with coordinates $(X_1,\ldots,X_5)^T$ by setting $X_1 = x, X_2 = y, X_3 = z, X_4 = 1, X_5 = \lambda_i$. In this setting it is immediate to verify that considering the following $3\times 5$ projection matrix $P$ from $\mathbb{P}^4$ to $\mathbb{P}^2$:

$$\left[\mathbf{P}_1(t)|\ \ \mathbf{P}_2(t)|\ \ \mathbf{P}_3(t)|\ \ \mathbf{P}_4(t)|\ \ tdx_0\mathbf{P}_1(t)+tdy_0\mathbf{P}_2(t)+tdz_0\mathbf{P}_3(t)\right] \qquad (2)$$

one has $P\cdot(x,y,z,1,\lambda_i)^T = P(t)\cdot\mathbf{X_i}$. Thus the dynamic scene in $\mathbb{P}^3$ has been transformed into a static one in $\mathbb{P}^4$.

ii) *Two-body segmentation* $k=4$. Here we consider a $3D$ point configuration consisting of two bodies moving relatively to each other by pure translation. Let $P$ and $R$ respectively denote the $3\times 4$ camera matrices associated to the first and second body and $\mathbf{P}_j$ and $\mathbf{R}_j$ the $j-th$ column of $P$ and $R$. Recall that $\mathbf{P}_j = \mathbf{R}_j$ for $j=1,2,3$, for the pure translation assumption. In this case the coordinates of a point $(X_1,X_2,X_3,X_4,X_5)^T$ in $\mathbb{P}^4$ are $(x,y,z,1,0)^T$, for points of the first body and $(x,y,z,0,1)^T$, for points of the second body, and the projection matrix to $\mathbb{P}^2$ is the $3\times 5$ matrix $[\mathbf{P}_1|\ \ \mathbf{P}_2|\ \ \mathbf{P}_3|\ \ \mathbf{P}_4|\ \ \mathbf{R}_4]$.

## 4 Multiview Tensors in Higher Dimension

In the classical case of projections from $\mathbb{P}^3$ to $\mathbb{P}^2$, it is well known that projective reconstruction is possible from two views and that, in this context, a pair of corresponding points on both image planes imposes a bilinear constraint on their coordinates. The coefficients coming from a constraint obtained in this way are then packaged in the well known fundamental matrix. Noticing that a point in the image plane can be considered as a linear space of codimension 2, one could think of the fundamental matrix as a $3\times 3$ tensor, born out of initial data of codimensions $(2,2)$.

When one considers three views $\mathbb{P}^3 \to \mathbb{P}^2$, it is well known that to have meaningful constraints one is led to consider triplets of corresponding subspaces in the image planes where one is a point and two are lines. Again, one can package the coefficients of the trilinear relations obtained this way into the classical trifocal tensor, obtaining a $3\times 3\times 3$ tensor where the initial data this time has codimensions $(2,1,1)$ (or any of its permutations).

Going one step further to 4 views from $\mathbb{P}^3$ to $\mathbb{P}^2$, constraints are obtained from 4-ple of corresponding lines, one from each view. One then obtains a $3\times 3\times 3\times 3$ tensor, the quadrifocal tensor, coming from initial data of codimensions $(1,1,1,1)$.

Hartley and Schaffalitzky, in [14], generalized these observations and gave a comprehensive theoretical framework for the study of multiview tensors, to which

they refer as *Grassmann* tensors, in any dimension. Here we recall the basic elements of their approach. Consider a set of projections $P_i : \mathbb{P}^k$ to $\mathbb{P}^{h_i}$, $i = 1, \ldots, r$ and a partition $(\alpha_1, \alpha_2, \ldots, \alpha_r)$ of $k+1$, i.e. $\sum \alpha_i = k+1$. Let $\{S_i\}$, $i = 1, \ldots, r$, where $S_i \subset \mathbb{P}^{h_i}$, be a set of general $(h_i - \alpha_i)$-spaces. Hartley and Shaffalitzky show that imposing to $\{S_i\}$ to be a set of corresponding subspaces, leveraging the assumption that $\sum \alpha_i = k+1$, translates into a multilinear relation among the Grassmann (Plücker) coordinates of the $S_i$. The coefficients of these relations are then packaged into a multi-view tensor, called a Grassmann tensor with profile $(\alpha_1, \ldots, \alpha_r)$. In these notations then one revisits our initial examples to see that for $k = 3$ and $h = 2$ all possible profiles are: (2,2), in which case the Grassmann tensor is simply the fundamental matrix; (1,1,2) or any of its permutations, in which cases the tensor is the $3 \times 3 \times 3$ usual trifocal tensor; (1,1,1,1) which gives the usual quadrifocal tensor. Notice that Grassmann coordinates are really necessary only for target spaces of dimension 3 or higher because for spaces of maximum codimension $h_i$ (i.e. points) or minimum codimension 1 (i.e. hyperplanes) Grassmann coordinates are nothing but ordinary point coordinates or usual dual hyperplane coordinates. Under the same assumptions, Hartley and Schaffalitzky also show that from Grassmann tensors it is possible to recover projection matrices (uniquely up to projective equivalence) unless $h_i = 1$ for all $i$.

For more details on this approach we refer the reader to [14]. Although our approach is essentially the same, because all our applications will deal with projections to $\mathbb{P}^2$, we refrain from a heavy use of Grassmann coordinates.

## *4.1 Minimum Number of Views*

From now on we assume that all target spaces have the same dimension, i.e. $h_1 = h_2 = \cdots = h_r = h$. Given multiple views of a scene, from the projective reconstruction point of view, there are two kind of problems: reconstruction of the cameras and reconstruction of the scene. It is worthwhile recalling that, as in the case of projections $\mathbb{P}^3 \to \mathbb{P}^2$, the camera center is the only property of the camera which is preserved under homographies of the image space, see [5], hence reconstruction of the cameras is equivalent to reconstruction of their centers. From the approach of [14] one can quickly deduce the minimal number of views $\omega_{k,h}$ necessary to successfully accomplish projective recostruction of the cameras, although no explicit formula is given. For example, in the case of $k = 6$ and $h = 3$, starting to fill possible profiles from the left with the largest possible codimension $\alpha_1 = h = 3$, $\alpha_2 = h = 3$, one quickly sees that at least three views are necessary and a possible profile is $(3, 3, 1)$, and thus $\omega_{6,3} = 3$.

Another useful information in this context is the minimum number $\mu_{k,h}$ of views necessary for the reconstruction of the scene points, once the camera centers are known. In this subsection we are working under the assumption that the centers $C_{Pj}$ of the projections we consider are in general position.

A preliminary example can be helpful to illustrate the situation.

*Example 0.1.* Let $k=4, h=2$. Centers of projections being lines, projecting rays are 2-spaces, i.e. planes. From (1) it follows that, in general, two planes in $\mathbb{P}^4$ intersect at a point. This shows that two views are not sufficient to give necessary constraints and a third view is required. Each triple of corresponding points (or a pair of corresponding points and a corresponding line), indeed gives two (or one) constraints as described in [23] or [14]. Hence, from a large enough set of triples of correspondences, one can reconstruct the cameras. Once the cameras are known, a scene point can be uniquely reconstructed from just two of the three given views, as any pair of corresponding rays intersect at a point. Thus, for projections $\mathbb{P}^4$ to $\mathbb{P}^2$, the minimum number of views necessary for projective recostruction is 3 for the cameras and 2 for the scene, assuming the cameras are known.

The following two propositions, whose proofs rely on geometric arguments, give explicit formulas for $\omega k,h$ and $\mu_{k,h}$.

**Proposition 1** *Assume $k-1=\sigma h+\lambda$, where $\sigma$ and $\lambda$ are non negative integers and $\lambda \leq h-1$. Assuming that cameras are known (up to projective equivalence), the minimum number of views necessary to reconstruct a scene for projections from $\mathbb{P}^k$ to $\mathbb{P}^h$ is*

$$\mu_{k,h}=\sigma+1.$$

**Proposition 2** *Assume $k=sh+l$, where $s$ and $l$ are non negative integers and $l \leq h-1$. The minimum number of views necessary to reconstruct the cameras for projections from $\mathbb{P}^k$ to $\mathbb{P}^h$ is*

$$\omega_{k,h}=s+1.$$

## 5 Algorithms

After having introduced a mathematical framework and corresponding tools to perform reconstruction in the previous sections, we now turn to algorithm development. The experimental part of this chapter will present results related to situations described in Section 3 i) and ii), thus we limit our presentation here to algorithms targeted at these two examples. Nonetheless one could develop in similar ways algorithms to address other situations for different $\mathbb{P}^k$'s and different number of views. Both examples under consideration here deal with three views in $\mathbb{P}^4$. Algorithms to obtain the corresponding $3\times3\times3$ trifocal tensors are needed with two different sets of inputs: the projection matrices $Pj:\mathbb{P}^4$ to $\mathbb{P}^2$, $j=1,2,3$, and a large enough set of triplets of corresponding points $Pj(X)^T=\mathbf{x_j}^T=(x_1^j,x_2^j,x_3^j)$, $j=1,2,3$. Let $\mathbf{x_j}$, $j=1,2,3$, be a triplet of corresponding image points, and let $\ell=(l_1,l_2,l_3)$ be the row-vector of Plücker coordinates of a line in the third view-plane passing through $\mathbf{x_3}$. Each such point-point-line set imposes one multi-linear constraint on $T$, as $\sum_i\sum_j\sum_k x_j^1 x_i^2 t_{ijk} l_k=0$, as in [23, Section 2.1, Problem Definition 2 ] or [14]. These multi-linear constraints are the basis of both algorithms.

## *5.1 Trifocal Tensor from Projection Matrices*

Let $Pj, \mathbf{x_j}$ and $\ell$ be as in the introduction of this section. Here the entries of $Pj$ are given, while the coordinates of $\mathbf{x_j}$, and obviously the entries of $T$, are unknown. Let $r_j = (0, x_3^j, -x_2^j)$, $s_j = (x_3^j, 0, -x_1^j)$. Then $s_j \cdot \mathbf{x_j} = s_j \cdot Pj(X) = 0$, $r_j \cdot \mathbf{x_j} = r_j \cdot Pj(X) = 0$, and $\ell \cdot P3(X) = 0$, and thus the matrix

$$M = \begin{bmatrix} s_1 P1 \\ r_1 P1 \\ s_2 P2 \\ r_2 P2 \\ \ell P3 \end{bmatrix} = \begin{bmatrix} x_3^1 \mathbf{P1^1} - x_1^1 \mathbf{P1^3} \\ x_3^1 \mathbf{P1^2} - x_2^1 \mathbf{P1^3} \\ x_3^2 \mathbf{P2^1} - x_1^2 \mathbf{P2^3} \\ x_3^2 \mathbf{P2^2} - x_2^2 \mathbf{P2^3} \\ \ell P3 \end{bmatrix}$$

must have vanishing determinant. Because the determinant is a multi-linear function of the rows, one sees from the right hand side representation of $M$ that $\det M$ must be divisible by $x_3^1 x_3^2$. Without loss of generality here we can assume that $X$ is not on any of the hyperplanes mapped to to the lines at infinity of the three images and thus $x_3^j \neq 0$, for all $j$. Let $D = \det M/(x_3^1 x_3^2)$. Entries $t_{ijk}$ of the tensor, as functions exclusively of the entries of the projection matrices $Pj$, can now be isolated as the coefficients of $x_j^1 x_i^2 l_k$ in $D$. MATLAB®code to obtain $t_{ijk}$ with this algorithm is available from the authors.

## *5.2 Trifocal Tensor from Triplets of Image Points*

Let $\mathbf{x_j}, j = 1,2,3$ be a triplet of corresponding visible points in the three images, as in the introduction of this section. Here the coordinates of $\mathbf{x_j}$ are given and the entries of $T$ are unknown. Let $\ell_m = (l_1^m, l_2^m, l_3^m)$, $m = 1,2$, be the Plücker coordinates of two randomly generated lines through $\mathbf{x_3}$. Then each triplet $\mathbf{x_j}$ generates two point-point-line correspondences and thus gives rise to two multi-linear constraints on the entries of $T$ : $\sum_i \sum_j \sum_k x_j^1 x_i^2 t_{ijk} l_k^m = 0$, $m = 1,2$. As $T$ has 27 entries and it is defined up to a multiplicative constant, 13 triplets of corresponding points are theoretically sufficient. Let's assume we have $n$ triplets. The constraints described above give a system of $2n$ homogeneous equations in the 27 variables $t_{ijk}$. Each row of the matrix of this system consists of 27 possible products $x_j^1 x_i^2 l_k^m$ properly arranged. A solution is obtained performing SVD on such a matrix and extracting from the right hand side matrix of the decomposition the column corresponding to the smallest singular value. The 27 elements of this column are then re-packaged into a tensor. MATLAB®code to obtain $t_{ijk}$ with this algorithm is available from the authors.

# 6 Critical Configurations and Their Loci

An interesting issue that accompanies the reconstruction problem is the determination of the so called *critical configurations*: is reconstruction possible, in a suitably unique way, independent of the mutual positions of the chosen scene-points and cameras? Are there configurations of chosen scene-points and cameras that turn out to be critical in the sense that they prevent successful reconstruction or allow for multiple possible solutions giving rise to ambiguities? Which geometric object describes these critical configurations?

Answers to these questions are well known to the community of experts in several cases, as detailed in the introduction. The following definitions introduce the necessary language to address these critical situations both in case of one and of multiple views. For the sake of simplicity in the following sections we will always consider projections from $\mathbb{P}^k$ to $\mathbb{P}^2$, i.e. we always set $h = 2$.

## *6.1 Critical Configurations for Projective Reconstruction from One View: Definitions*

**Definition 1** *A set of points* $\{\mathbf{X_i}\}$, $i = 1,\dots,N$, $N \geq 5$, *in* $\mathbb{P}^k$ *is said to be a critical configuration for projective recostruction from 1-view if there exist two* $3 \times (k+1)$ *full-rank projection matrices P and Q such that the two sets of points* $\{P \cdot \mathbf{X_i}\}$ *and* $\{Q \cdot \mathbf{X_i}\}$ *are projectively equivalent, i.e. up to homography on the projection planes,* $P \cdot \mathbf{X_i} = \mu_i Q \cdot \mathbf{X_i}$, *with* $\mu_i$ *a non zero constant, for each i. Matrices P and Q are called associated matrices to the critical configuration.*

Assume now that $\{\mathbf{X_i}\}$ is a critical configuration for projective reconstruction from 1-view, as above, with associated projection matrices $P$ and $Q$. As $P \cdot \mathbf{X_i} = \mu_i Q \cdot \mathbf{X_i}$, then each $\mathbf{X_i}$ is in the right null-space of a matrix $P - \mu_i Q$. This leads one to consider matrices $P_{(\varphi,\psi)} = \varphi P + \psi Q$, for some $(\varphi,\psi) \neq (0,0)$, and their null spaces. The following definition is in the spirit of [13, Section 21.1.2].

**Definition 2** *If* $\{\mathbf{X_i}\}$ *is a critical configuration for projective reconstruction from 1-view, with associated projection matrices P and Q, the associated critical locus for projective reconstruction from 1-view in* $\mathbb{P}^k$ *is the subvariety (technically a subscheme)* $\mathfrak{C}^k_{(P,Q)}$ *of* $\mathbb{P}^k$ *defined by the parametric equations* $P_{(\varphi,\psi)} \cdot \mathbf{X} = \boldsymbol{0}$, *where* $(\varphi,\psi) \in \mathbb{P}^1$.

In this context, $\Lambda_{(\varphi,\psi)}$ will denote the right null space of $P_{(\varphi,\psi)}$. Sometimes it will be more convenient to use an affine parameter $\theta = \psi/\varphi$, and in this case $P_{(1,\theta)}$ and $\Lambda_{(1,\theta)}$ will be denoted simply by $P_\theta$ and $\Lambda_\theta$.

## 6.2 Critical Loci for Projective Reconstruction from One View: Geometric Description

This section contains basic results on critical loci $\mathfrak{C}^k_{(P,Q)}$. Only non degenerate cases (technically irreducible and reduced) are described here (without proof: for the proof see [5], Propositions 3.1 and 3.2) as they are the cases addressed by the authors' experiments in Section 7 and 8. A detailed description of the degenerate cases can be found in [13, Section 21.1.2] for $k = 3$ and [5] for $k \geq 4$.

**Proposition 3** *The critical locus for projective recostruction of a set of points in* $\mathbb{P}^k$ *from one view, if irreducible, is an algebraic variety of dimension* $k-2$ *and degree* 3 *which is the union of a 1D family of* $\mathbb{P}^{k-3}$ *parameterized by a smooth rational cubic curve in* $\mathbb{P}^3$. *If* $k = 4$ *the variety can be either a smooth surface (a linear* $\mathbb{P}^1$*-bundle over a rational cubic curve) or a cone with vertex a point over a cubic rational curve. If* $k = 5$ *the variety can be either a smooth 3D variety (a linear* $\mathbb{P}^2$*-bundle over a cubic rational curve) or a cone with vertex a point over a cubic surface* $\mathbb{P}^4$ *(the surface described above), or a cone with vertex a line over a cubic rational curve. If* $k \geq 6$ *the variety is always a cone, the vertex of which has dimension at least* $k-6$.

From the assumption, each $\mathbf{X_i}$ belongs to the null-space $\Lambda_\theta = \Lambda_{(\varphi,\psi)}$, for $\theta = \psi/\varphi$, of a matrix $P_\theta = P_{(\varphi,\psi)}$, for some $(\varphi,\psi) \neq (0,0)$. From above one easily sees that the critical locus is the union of infinitely many $(k-3)$-dimensional linear spaces $\Lambda_\theta$ parameterized by $\theta$. With the help of some standard techniques in algebraic geometry one can show that such null spaces $\Lambda_\theta$ describe a cubic curve $C_\theta$ in the Grassmannian $G(k-3,k)$, which generates a $(k-2)$ dimensional variety of degree 3 in $\mathbb{P}^k$. In cases $k = 4,5$ the variety can be either smooth, in case the null-spaces of $P$ and $Q$, i.e. the centers of the two projections, do not intersect, or cones over a cubic surface with center $V$, in case the two null spaces intersect in $V$. If $k \geq 6$ the centers of the two projections given by $P$ and $Q$ always intersect in a linear subspace $V$, hence all the null-spaces $\Lambda_\theta$ contain $V$. This implies that the variety is never smooth, but a cone with vertex $V$ over a suitable 3D variety.

## 6.3 Critical Configurations for Projective Reconstruction from Multiple Views: Definitions

Here we deal with the case of multiple views.

**Definition 3** *A set of points* $\{\mathbf{X_i}\}$, $i = 1,\ldots,N$, $N \geq k+3$, *in* $\mathbb{P}^k$ *is said to be a critical configuration for projective recostruction from n-views if there exist a non-projectively equivalent set of N points* $\mathbf{Y_i} \in \mathbb{P}^k$ *and two collections of* $3 \times (k+1)$ *full-rank projection matrices* $Pj$ *and* $Qj$ $(j = 1,\ldots,n)$ *such that, for all i and j,* $Pj \cdot \mathbf{X_i} = Qj \cdot \mathbf{Y_i}$, *up to homography in the image planes. The two sets* $\{\mathbf{X_i}\}$ *and* $\{\mathbf{Y_i}\}$ *are*

*called conjugate critical configurations, with associated conjugate matrices* $\{Pj\}$ *and* $\{Qj\}$.

The notion of conjugate critical configurations is inspired by [11].

The natural setting to study the loci of all critical configurations associated to sets of conjugate matrices is the product variety $\mathbb{P}^k \times \mathbb{P}^k$, endowed with the two standard projections $\pi_1$ and $\pi_2$ onto the two factors. Let $\{(\mathbf{X_i}, \mathbf{Y_i})\}$ be conjugate critical configurations as above, with associated conjugate matrices $\{Pj\}$ and $\{Qj\}$. For $(\mathbf{X}, \mathbf{Y}) \in \mathbb{P}^k \times \mathbb{P}^k$ (generalizing to $k \geq 3$ the approach of [20]), imposing $Pj \cdot \mathbf{X}$ and $Qj \cdot \mathbf{Y}$ to be proportional one gets a pair of equalities

$$\mathbf{X}^T \cdot Pj^T \cdot Eh \cdot Qj \cdot \mathbf{Y} = 0, \tag{3}$$

$h = 1, 2$, where $E1_{rs} = \varepsilon_{r2s}$ and $E2_{rs} = \varepsilon_{rs1}$, in other words

$$E1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \quad E2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}.$$

**Definition 4** *If* $\{(\mathbf{X_i}, \mathbf{Y_i})\}$ *in* $\mathbb{P}^k \times \mathbb{P}^k$ *are pairs of conjugate critical configurations, with associated conjugate matrices* $\{Pj\}$ *and* $\{Qj\}$*, the associated unified critical locus for projective recostruction from n-views in* $\mathbb{P}^k \times \mathbb{P}^k$ *is the locus (technically subscheme)* $\mathscr{U}^k = \mathscr{U}^k_{(\{Pj\},\{Qj\})} \subseteq \mathbb{P}^k \times \mathbb{P}^k$ *defined by* $2n$ *equations (3). Moreover the corresponding critical locus (respectively conjugate critical locus) for projective reconstruction from n-views in* $\mathbb{P}^k$ *is the locus (technically subscheme)* $\mathscr{X}^k = \mathscr{X}^k_{(\{Pj\},\{Qj\})} = \pi_1(\mathscr{U}^k)$ *(respectively* $\mathscr{Y}^k = \mathscr{Y}^k_{(\{Pj\},\{Qj\})} = \pi_2(\mathscr{U}^k)$*).*

## 6.4 Critical Loci for Projective Reconstruction from Multiple Views: Geometric Description

Definition 4 implies that explicit equations for $\mathscr{X}^k$ are obtained by imposing that the linear system of $2n$ equations (3) with $j = 1, \ldots, n$, $h = 1, 2$, in the $k+1$ unknowns $Y_1, \ldots Y_k$ has nontrivial solutions. One is lead naturally to consider the following matrix:

$$M = \mathbf{X}^T \cdot Pj^T \cdot Eh \cdot Qj = \begin{bmatrix} (-\mathbf{P1}^3 \cdot \mathbf{X})\mathbf{Q1}^2 + (\mathbf{P1}^2 \cdot \mathbf{X})\mathbf{Q1}^3 \\ (-\mathbf{P1}^3 \cdot \mathbf{X})\mathbf{Q1}^1 + (\mathbf{P1}^1 \cdot \mathbf{X})\mathbf{Q1}^3 \\ \cdots \\ (-\mathbf{Pi}^3 \cdot \mathbf{X})\mathbf{Qi}^2 + (\mathbf{Pi}^2 \cdot \mathbf{X})\mathbf{Qi}^3 \\ (-\mathbf{Pi}^3 \cdot \mathbf{X})\mathbf{Qi}^1 + (\mathbf{Pi}^1 \cdot \mathbf{X})\mathbf{Qi}^3 \\ \cdots \end{bmatrix}$$

If $2n \geq k+1$ (i.e. $n \geq \omega_{k,2}$), equations for $\mathscr{X}^k$ are given by the vanishing of maximum $(k+1)$ rank minors $G_s$, $s = 1, \ldots, \binom{2n}{k+1}$ of $M$.

**Remark 1** *If $2n < k+1$ (i.e. $n < \omega_{k,2}$) the linear system $MY = 0$ has nontrivial solutions for every choice of $X$. Thus every possible configuration of points is critical, i.e. $\mathscr{X}^k = \mathbb{P}^k$. Hence from now on we assume $2n \geq k+1$.*

Our goal is to give a geometric description of $\mathscr{X}^k$, which, from above, is the variety described by the vanishing of all minors of maximal rank $k+1$ of $M$. Such a variety is usually called a *determinantal* variety. In order to give a rigorous rationale of our description of critical loci with appropriate tools from algebraic geometry, a few technical issues need to be addressed. We do this in the following Remark 2. The reader who is interested only in the resulting description of the loci is invited to jump to Lemma 1.

**Remark 2** *Let $M(2n,k+1)$ be the projective space associated to the vector space of matrices $2n \times (k+1)$ and consider the subvariety $M_k = M_k(2n,k+1)$ of all matrices with $\mathrm{rk}(M) \leq k$. A well known formula in algebraic geometry (Porteous' formula, [1, Formula 5.1, p. 95]) computes its degree: $\deg M_k = \binom{2n}{k}$. Moreover, it is not difficult to compute the codimension of $M_k$ in $M(2n,k+1)$. Choosing a fixed $k \times k$ minor $N_k$, one can achieve the vanishing of all maximal rank minors, on the open subset where $N_k$ does not vanish, simply by annihilating the $2n-k$ minors obtained by adjoining one of the remaining rows and the remaining column to $N_k$. This implies that locally $M_k$ is described by $2n-k$ equations which are algebraically independent; therefore the codimension of $M_k$ in $M(2n,k+1)$ is $2n-k$. As entries of $M$ are linear forms in $X_1, \ldots X_{k+1}$, and because of the particular structure of the matrix $M$, it is not difficult to see that the map $\phi : \mathbb{P}^k \to M(2n,k+1)$ sending $\mathbf{X}$ to $M$, is a linear embedding, i.e. $\phi(\mathbb{P}^k)$ is a linear subspace of $M(2n,k+1)$ of dimension $k$. Hence $\mathscr{X}^k$ is the intersection of the linear subspace $\phi(\mathbb{P}^k)$ with $M_k$ and thus its degree is the same as that of $M_k$ provided that:*

*$(*)$ The linear embedding $\phi$ given by the entries of the matrix $M$ is sufficiently general, i.e. the codimension of $\phi(\mathscr{X}^k)$ in $\phi(\mathbb{P}^k)$ is the same as the codimension of $M_k$ in $M$, i.e. $2n-k$.*

*Some of the results that follows in this work are stated under this assumption which will be made explicit according to the specific context in which it will be applied, see Propositions 4 and 5.*

*To conduct detailed investigations in concrete cases, one can utilize a standard technique from algebraic geometry, relying on the semicontinuity principle for the dimension of elements of an algebraic family of varieties. In essence this principle states that given a reasonably well behaved family of varieties, parametrized in an algebraic way, the dimension of elements of the family is generically constant, with possible exceptional elements whose dimension may be larger, never smaller, than the dimension of the general element. One can then choose a specific numerical case of the situation at hand, compute with the help of a Computer Algebra package (Macaulay2, Singular, Bertini, Cocoa or Magma) the desired dimension, and infer that the dimension of the generic element of the family is less than or equal to the computed one. An instance of this method is fully described later in this work (Section 7) in the case of $k = 4$.*

The previous discussion shows that a good understanding of the factorization of minors of $M$ is crucial to obtain explicit equations for critical loci. The following Lemma will be useful in this direction.

**Lemma 1** *For $n$ views $Pi$ in $\mathbb{P}^k$, let $\Pi_i$ be the hyperplane mapped to the line at infinity of the image plane in the $i^{th}$ view. Then for all $\mathbf{X} \in \Pi_i$ the two rows $\mathbf{X}^T \cdot Pi^T \cdot E1 \cdot Qi$ and $\mathbf{X}^T \cdot Pi^T \cdot E2 \cdot Qi$ of the matrix $M$ are linearly dependent.*

*Proof.* Let $Pi$ and $Qi$ be projection matrices as above. Let's assume to have chosen homogeneous coordinates $[X_1, X_2, X_3]$ in the image plane so that the line at infinity $L_\infty$ is $X_3 = 0$. Then a point $\mathbf{X} \in \mathbb{P}^k$ is mapped in the $i$-th view to a point on $L_\infty$ if and only if $P_i\mathbf{X} = [a,b,0]$ for some $(a,b) \neq (0,0)$. Thus $\mathbf{X}^T Pi^T E1 = [0,0,b]$ and $\mathbf{X}^T Pi^T E2 = [0,0,a]$ and therefore $\mathbf{X}^T Pi^T E1 Qi$ is a scalar multiple of $\mathbf{X}^T Pi^T E2 Qi$.

In view of Remark 1, the first significant case in which critical loci can be studied is for $\omega_{k,2}$ views, and in the following of this section we will assume $n = \omega_{k,2}$.

In the case of a space of odd dimension, $k = 2r-1, n = \omega_{k,2} = \frac{k+1}{2}$ and the matrix $M$ is a $(k+1) \times (k+1)$ square matrix, thus $2n - k = 1$. Condition $(*)$ can be made explicit as follows:

$(*_1)$ $G_1 = \det(M)$ is not identically zero.

**Proposition 4** *Let $k = 2r-1$ and let $n = \omega_{k,2} = \frac{k+1}{2}$. Let $\Pi_i$ be the hyperplane mapped to the line at infinity of the image plane in the $i^{th}$ view. Then $\mathscr{X}^k = \cup_i \Pi_i \cup Z$ where $Z$ is defined by a single homogeneous polynomial of degree $r = \frac{k+1}{2}$, provided that $(*_1)$ holds.*

*Proof.* In this case $M$ is a square matrix of dimension $(k+1) \times (k+1)$. Lemma 1 implies that for all $\mathbf{X} \in \Pi_i$ it is $G_1 = \det(M) = 0$ and thus every $\Pi_i$ is an irreducible component of $\mathscr{X}^k$. As entries of $M$ are linear in the coordinates of $\mathbf{X}$, the critical locus is given by one equation of degree $k+1$, which can be factored as a product of $\frac{k+1}{2}$ linear factors and a residual polynomial of degree $(k+1) - \frac{k+1}{2} = \frac{k+1}{2}$.

An analogous result holds for $k = 2r$, even, and $n = \omega_{k,2} = r+1$. In this case the matrix $M$ is a $(k+2) \times (k+1)$ matrix, thus $2n - k = 2$ and condition $(*)$ can be made explicit as follows:

$(*_2)$ The maximal minors $G_s$, $s = 1, \ldots, \binom{2n}{k+1}$ of $M$ have no common factor.

The result is contained in Proposition 5 (which in the case of $k = 4$ is contained in [4]). The proof in the case of even dimension is significantly more involved then in the odd one.

Let $k = 2r$ and let $\Pi_i$ be the hyperplane mapped to the line at infinity of the image plane in the $i^{th}$ view. Notice that $\Pi_i$ is given by the equation $\mathbf{Pi}^3 \cdot \mathbf{X} = 0$, where, we recall, $\mathbf{Pi}^s$ is the $s$-th row of the matrix $Pi$. Let $\Lambda_{ij}$ be the $\mathbb{P}^{k-2}$ obtained as intersection of $\Pi_i$ and $\Pi_j$. The $\Lambda_{ij}$s contain the linear space $L = \Pi_1 \cap \cdots \cap \Pi_{\omega_{k,2}}$, which, if the centers of projections are in general positions, is a $(\frac{k-2}{2})$-space.

**Proposition 5** *Let $k = 2r$. The critical locus for $n = \omega_{k,2} = r+1$ views in $\mathbb{P}^k$ is:*

$$\mathscr{X}^k = (\cup_{i<j=1}^{r+1} \Lambda_{ij}) \cup (\cup_{i=1}^{r+1} Z^i) \cup W$$

*where $Z^i$ is the hyperplane section of a hypersurface of degree $\frac{k}{2}$, via the hyperplane $\Pi_i$ mapped to infinity in the i-th view, $\Lambda_{ij} = \Pi_i \bigcap \Pi_j$, and $W$ is a variety of codimension 2 and degree $\frac{(k+4)(k+2)}{8}$, provided that $(*_2)$ holds.*

*Proof.* Because of the structure of the matrices $Ei$, the $(k+2)\times(k+1)$ matrix $M$ has the following form:

$$M = \begin{bmatrix} (-\mathbf{P1}^3 \cdot \mathbf{X})\mathbf{Q1}^2 + (\mathbf{P1}^2 \cdot \mathbf{X})\mathbf{Q1}^3 \\ (-\mathbf{P1}^3 \cdot \mathbf{X})\mathbf{Q1}^1 + (\mathbf{P1}^1 \cdot \mathbf{X})\mathbf{Q1}^3 \\ \cdots \\ (-\mathbf{Pi}^3 \cdot \mathbf{X})\mathbf{Qi}^2 + (\mathbf{Pi}^2 \cdot \mathbf{X})\mathbf{Qi}^3 \\ (-\mathbf{Pi}^3 \cdot \mathbf{X})\mathbf{Qi}^1 + (\mathbf{Pi}^1 \cdot \mathbf{X})\mathbf{Qi}^3 \\ \cdots \end{bmatrix}$$

The critical locus $\mathscr{X}^k$ is defined by the $k+2$ minors $G_s = \det M_s$ where $M_s$ is obtained from $M$ by removing the $s$-th row. Let $s = 2h-1$ or $s = 2h$, according to its parity and let $t = s+2-2h$. Then the row that was removed in $M_s$ corresponds to the $h$-th view. In what follows, the symbol $\widehat{\dots}$ on top of a quantity will mean that such quantity needs to be removed. Let us also denote by $(*)$ all possible combination of indices $1 \le j_1, \dots, \hat{j_h}, \dots j_n \le 3$. As the determinant is a multi-linear function of the rows, one can write each determinant $G_s = \det M_s$ as follows:

$$G_s = (\mathbf{P1}^3 \cdot \mathbf{X}) \dots (\widehat{\mathbf{Ph}^3 \cdot \mathbf{X}}) \dots (\mathbf{Pn}^3 \cdot \mathbf{X})((\mathbf{Ph}^3 \cdot \mathbf{X})\mathfrak{Z}_t^h + (\mathbf{Ph}^t \cdot \mathbf{X})\mathfrak{Z}_3^h)$$

where,

$$\mathfrak{Z}_t^h = \sum_{(*)} (\mathbf{P1}^{j_1} \cdot \mathbf{X}) \dots (\widehat{\mathbf{Ph}^{j_h} \cdot \mathbf{X}}) \dots (\mathbf{Pn}^{j_n} \cdot \mathbf{X}) \det \mathfrak{Q}_h^t,$$

$$\mathfrak{Z}_3^h = \sum_{(*)} (\mathbf{P1}^{j_1} \cdot \mathbf{X}) \dots (\widehat{\mathbf{Ph}^{j_h} \cdot \mathbf{X}}) \dots (\mathbf{Pn}^{j_n} \cdot \mathbf{X}) \det \mathfrak{Q}_h^3$$

and where $\mathfrak{Q}_h^t$ and $\mathfrak{Q}_h^3$ denote $(k+1)\times(k+1)$ matrices with constant entries, whose rows are suitable $\mathbf{Qi}^j$. In particular $\mathfrak{Q}_h^t$ contains $\mathbf{Qh}^t$ while $\mathfrak{Q}_h^3$ contains $\mathbf{Qh}^3$.

Recall that the linear spaces $\Lambda_{ij}$ are defined by the equations $\mathbf{Pi}^3 \cdot \mathbf{X} = \mathbf{Pj}^3 \cdot \mathbf{X} = 0$ for all $i, j = 1, \dots, r+1$, $i < j$, and consider the varieties $Z^h$ defined by $\mathbf{Ph}^3 \cdot \mathbf{X} = \mathfrak{Z}_3^h = 0$. Notice that $\mathfrak{Z}_3^h$ cannot have $\mathbf{Ph}^3 \cdot \mathbf{X}$ as a factor. If this were the case, then $G_s$ would have $(\mathbf{P1}^3 \cdot \mathbf{X}), \dots, (\mathbf{Pn}^3 \cdot \mathbf{X})$ as factors for all $s$, but this contradicts assumption $(*_2)$. Thus $Z^h$ are codimension 2 varieties of degree $\deg Z^h = \deg \mathfrak{Z}^h = r$. From the expression of $G_s$ one immediately sees that each $G_s$ vanishes on all the $(k-2)$-spaces $\Lambda_{ij}$ and on all the $Z^h$, and hence all $\Lambda_{ij}$ and all the $Z^h$ are contained in the critical locus. As the total degree of $\mathscr{X}^k$ is $\frac{(k+2)(k+1)}{2}$, see Remark 2, the residual

component $W$, outside of all $\Pi_i$, has codimension 2 and degree $\frac{(k+2)(k+1)}{2} - r(r+1) - \frac{r(r+1)}{2} = \frac{(k+2)(k+1)}{2} - \frac{3k(k+2)}{8} = \frac{(k+4)(k+2)}{8}$.

Proposition 5 applies in particular to the case of $\mathbb{P}^4$, see [4]. In this case the critical locus consists of 3 planes and the solution set of 6 polynomial equations of degree 3. The latter set, for general choices of projection matrices, can be investigated with the help of Macaulay2© [8]. Outside of the planes which are mapped to infinity in each view, it turns out to be a surface of degree 6 which, when considered over $\mathbb{C}$, if irreducible and smooth, is a very well understood object in algebraic geometry, called a Bordiga surface.

## 7 Examples of Critical Loci for Projective Reconstruction

In this section we give some explicit example of critical loci, both in the case of one and multiple views, arising from dynamic or segmented scenes. In particular we refer to two situations which correspond to projections from $\mathbb{P}^4$ to $\mathbb{P}^2$, described in Section 3.

### *7.1 Three Dimensional Dynamic Scene*

We start with case i) of Section 3. In order to obtain the critical locus for one view, recall that, with notation of Proposition 3, one sees that the matrix $P_\theta$ is

$$[\mathbf{P}_1(\theta)|\ \mathbf{P}_2(\theta)|\ \mathbf{P}_3(\theta)|\ \mathbf{P}_4(\theta)|\ tdx\mathbf{P}_1(\theta)+tdy\mathbf{P}_2(\theta)+tdz\mathbf{P}_3(\theta)]$$

The critical locus is an irreducible ruled surface in $\mathbb{P}^4$, which is a cone, since the centers of projection intersect in the point $V = (-tdx, -tdy, -tdz, 0, 1)$. More explicitly one gets the following parametric equations of the critical locus:

$$x = -aD_0 - btdx, y = aD_1 - btdy,$$

$$z = -aD_2 - btdz, -1 = aD_3, \lambda = b,$$

where $D_i, i = 0, ..., 3$ is the $3 \times 3$ minor obtained by the matrix

$$[\mathbf{P}_1(\theta)|\ \mathbf{P}_2(\theta)|\ \mathbf{P}_3(\theta)|\ \mathbf{P}_4(\theta)]$$

eliminating the (i+1)-th column. Notice that when $t = 0$ or $\lambda = b = 0$ one gets the standard critical locus for projections from $\mathbb{P}^3$ to $\mathbb{P}^2$.

In order to explicitly determine the critical locus for three views (also in view of the next section in which we will investigate instability of the reconstruction in an experimental fashion) we chose a specific case: A dynamic scene obtained from a static one given in Hartley and Kahl [11, Example 8.5.1]. This example was chosen

as it represents the generic case for three views in $\mathbb{P}^3$, in which the critical locus consists of 7 distinct points. The projection matrices for the static scene, from [11, Example 8.5.1], are:

$$P0 = \begin{bmatrix} 2 & 0 & 0 & -1 \\ 0 & 3 & 0 & -1 \\ 0 & 0 & 6 & -1 \end{bmatrix} \qquad Q0 = \begin{bmatrix} -2 & 0 & 0 & 5 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 10 & 5 \end{bmatrix}$$

$$P1 = \begin{bmatrix} -2 & 0 & 0 & -1 \\ 0 & 4 & 0 & -1 \\ 0 & 0 & -8 & -1 \end{bmatrix} \qquad Q1 = \begin{bmatrix} -4 & 0 & 0 & 3 \\ 0 & -2 & 0 & 3 \\ 0 & 0 & -6 & 3 \end{bmatrix}$$

$$P2 = \begin{bmatrix} 3 & 0 & 0 & -1 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & -6 & -1 \end{bmatrix} \qquad Q2 = \begin{bmatrix} 3 & 0 & 0 & 5 \\ 0 & -1 & 0 & 5 \\ 0 & 0 & 15 & 5 \end{bmatrix}$$

The corresponding two sets of projection matrices in $\mathbb{P}^4$ are obtained from the ones above by following (2) of Section 3, where $t = t_1$ for $P0, Q0$, $t = t_2$ for $P1, Q1$, and $t = t_3$ for $P2, Q2$.

In accord with Proposition 5, the critical locus $\mathscr{X}^4$ consists of three planes given by the intersection of pairs of hyperplanes chosen from the following three:

$$\begin{aligned} 6X_3 - X_4 + (6t_1 dz)X_5 &= 0 \\ -8X_3 - X_4 - (8t_2 dz)X_5 &= 0 \\ 6X_3 - X_4 + (6t_3 dz)X_5 &= 0 \end{aligned}$$

and a residual variety defined by six polynomial equations of degree 3. Setting $t_1 = 0.1, t_2 = 0.2, t_3 = 0.3, dx = 1, dy = 0, dz = -1$ ($(dx, dy, dz)$ not being a unit vector, the velocities of the bodies are given by $\lambda_i\sqrt{2}$) the six equations can be processed by Macaulay2© [8] to reveal that beside the three planes defined above, the critical locus consists of 3 quadrics $Z^i$ and a residual Bordiga surface $W$.

## *7.2 Two-Body Segmentation*

We now consider case ii) of Section 3. In the notations of Proposition 3, the matrix $P_\theta$ is of the form $[\mathbf{P}_1(\theta)|\ \mathbf{P}_2(\theta)|\ \mathbf{P}_3(\theta)|\ \mathbf{P}_4(\theta)|\ \mathbf{R}_4(\theta)]$.

For generic choice of the projections matrices $P$ and $R$, the two centers of projection do not intersect, hence the critical locus is a smooth cubic surface in $\mathbb{P}^4$, ruled in lines. It cuts the hyperplane $X_4 = 0$ in the rational cubic curve which is the critical locus for the projection of the first body and the the hyperplane $X_3 = 0$ in the cubic curve which is the critical locus for the projection of the second one.

Also in this case, in order to study the critical locus for three views, we chose a specific case, corresponding to the following pairs of conjugate matrices $Pj$ and $Qj$:

$$P1 = \begin{bmatrix} 2 & 0 & 0 & -1 & 2 \\ 0 & 3 & 0 & -1 & 0 \\ 0 & 0 & 6 & -1 & 1 \end{bmatrix} \qquad Q1 = \begin{bmatrix} -2 & 0 & 0 & 5 & 1 \\ 0 & 1 & 0 & 5 & 2 \\ 0 & 0 & 10 & 5 & 3 \end{bmatrix}$$

$$P2 = \begin{bmatrix} -2 & 0 & 0 & -1 & 4 \\ 0 & 4 & 0 & -1 & 4 \\ 0 & 0 & -8 & -1 & 0 \end{bmatrix} \qquad Q2 = \begin{bmatrix} -4 & 0 & 0 & 3 & -4 \\ 0 & -2 & 0 & 3 & 1 \\ 0 & 0 & -6 & 3 & -1 \end{bmatrix}$$

$$P3 = \begin{bmatrix} 3 & 0 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 & 2 \\ 0 & 0 & -6 & -1 & -1 \end{bmatrix} \qquad Q3 = \begin{bmatrix} 3 & 0 & 0 & 5 & 2 \\ 0 & -1 & 0 & 5 & -1 \\ 0 & 0 & 15 & 5 & 1 \end{bmatrix}$$

In accord with Proposition 5, the critical locus $\mathscr{X}^4$ consists of three planes given by the intersection of pairs of hyperplanes chosen from the following three:

$$\begin{aligned} 6X_3 - X_4 + X_5 &= 0 \\ 8X_3 + X_4 &= 0 \\ 6X_3 + X_4 + X_5 &= 0 \end{aligned}$$

and of a residual variety consisting of three quadrics $Z^i$ and a Bordiga surface $W$.

## 8 Instability Results

The probability that, in practice, all points chosen in a scene constitute a critical configuration is extremely low. Nonetheless, as pointed out in [11] and [3], it is of practical relevance to understand the behavior of the reconstruction for configurations of points that are near the critical locus. The simulated experiments described in this section show that, in these circumstances, reconstruction solutions are extremely unstable. Although experiments were conducted in a variety of cases, we focus our attention here on the situations described in Section 3 i) and ii), both in the case of one view and of multiple views.

### *8.1 The Case of One View*

This section contains a description of simulated experiments which were performed with MATLAB®and a compilation of sample results. The philosophy of these experiments is inspired by Appendix C *Practical implications* of [11]. Instability of solutions obtained from configurations "close" to critical ones were tested. All experiments involve the reconstruction of a projection matrix, therefore an implementation of the standard algorithm to compute the reconstruction of the projection matrix given $N$ points $\mathbf{X_i}$ in $\mathbb{P}^k$ and their projections $\mathbf{x_i}$ in $\mathbb{P}^2$, was written by the authors

generalizing [13, Section 6]. For convenience of the reader we summarize here the strategy of the algorithm. Given $N$ pairs $(\mathbf{X_i}, \mathbf{x_i} = P(\mathbf{X_i}))$, from $\lambda \mathbf{x_i} = P \cdot \mathbf{X_i}$, one gets the $N$ relations $\mathbf{x_i} \wedge P \cdot \mathbf{X_i} = 0$ each of which corresponds to two equations. Since the matrix $P$ depends on $3k+2$ affine parameters, $[(3k+2)/2]+1$ pairs $(\mathbf{X_i}, \mathbf{x_i})$ are theoretically enough to get $P$. From a practical point of view, the computation is approximate; thus the more pairs are involved the more precise the result is. The resulting linear system is solved using the MATLAB®built-in Singular Value Decomposition function.

The instability phenomenon for one view is verified in various situations using a unified approach, whose steps are described below.

1 *Random generation of critical configurations*
Let's assume as usual that $\{\mathbf{X_i}\}$ is a critical configuration, with associated projection matrices $P$ and $Q$, and associated irreducible critical locus $\mathfrak{C}^k_{(P,Q)}$. As all irreducible $\mathfrak{C}^k_{(P,Q)}$ involved in Proposition 3 admit a rational parametrization, a critical configuration $\{\mathbf{X_i}\}$ of $N$ points can be generated using independent uniformly distributed sequences of parameters in assigned ranges. For all the results reported below, $N = 30$ and parameters were chosen in the range $[-10, 10]$.

2 *Perturbation of critical configurations*
Points $\{\mathbf{X_i}\}$ are then perturbed with a $k$-dimensional noise, normally distributed with zero mean and assigned standard deviation $\sigma$, obtaining a new configuration $\{\mathbf{X_i}^{pert}\}$ which is now close to being critical. This configuration is now projected via $P$ and the resulting images are again perturbed with normally distributed 2D noise with zero mean and standard deviation 0.01 to obtain $\{\mathbf{x_i}^{pert}\}$.

3 *Reconstruction*
The reconstruction algorithm quoted above is applied to pairs $(\mathbf{X_i}^{pert}, \mathbf{x_i}^{pert})$, $\mathbf{i} = 1, \ldots, N$, to obtain an estimated projection matrix $P_{rec}$.

4 *Estimating instability*
Notice that $P_{rec}$ is generated by SVD as a matrix with $||P_{rec}|| = 1$. As projection matrices are defined up to multiplication by a non-zero constant, we also normalize $P_{(\psi,\phi)}$, and identify the space of projection matrices with a quotient of the unit sphere in $\mathbb{R}^{3k+3}$, $S^{3k+2}/\simeq$, where $\simeq$ denotes antipodal identification. It is simple to account for antipodal identification when computing distances: for any pair of matrices with unit norm $A$ and $B$, we set $d(A,B) = min(||A-B||, ||A+B||)$. Using this notion of distance, the value of $\theta = \phi/\psi$ corresponding to the $P_{(\psi,\phi)}$ that is closest to $P_{rec}$ is then computed. Extra care is needed here, as the distance function to be minimized is defined over the real projective line $\mathbb{P}^1[\psi, \phi]$. The above procedure is then repeated 1000 times for a fixed value of $\sigma$.

5 *Displaying the results*
Instability of the reconstruction is best highlighted by the spread of the obtained values of $\theta$. Results are then presented in graphs showing the standard deviation of the values of $\theta$ against the values of $\sigma$ utilized. As expected, the larger the value of $\sigma$, the stabler the solution gets, with standard deviation of $\theta$ quickly approaching zero (and the average value of $\theta$ also approaching zero).

The above process is now applied to examples 7.1 and 7.2.

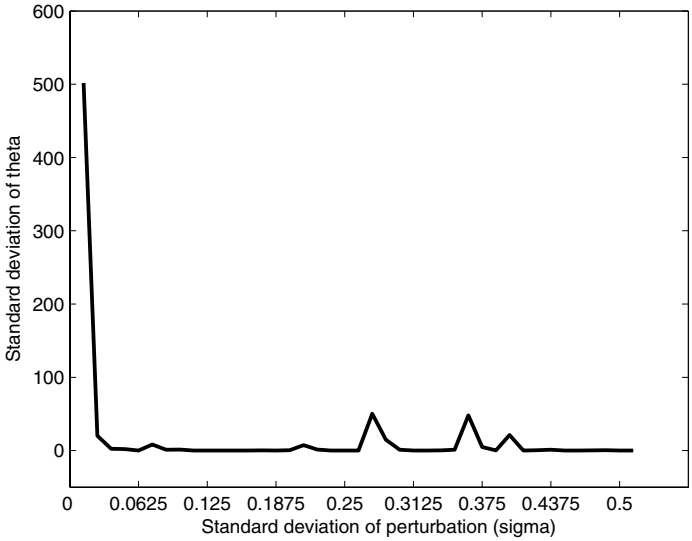



**Fig. 1** Instability of reconstruction from one view: dynamic scene as in Section 3 i) and 7.1 in $\mathbb{P}^4$. Standard deviation of $\theta$, see step 4 in 8.1, for 1000 reconstructions, as a function of a fixed value of the standard deviation of the 4D perturbation.

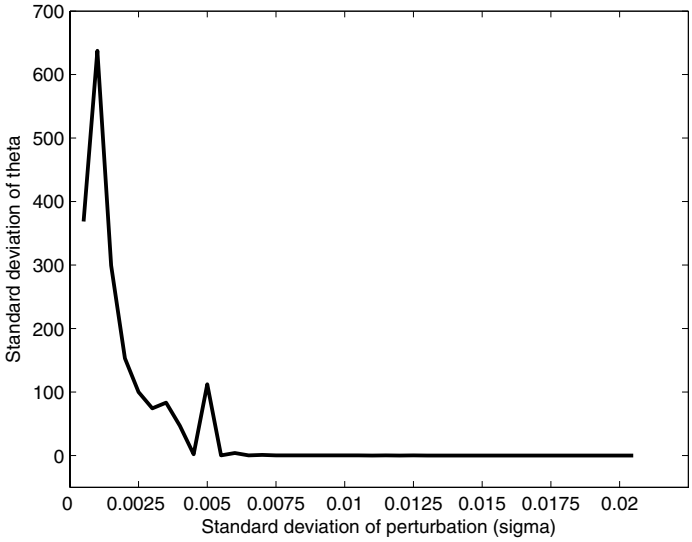



**Fig. 2** Instability of reconstruction from one view: segmented scene as in Section 3 ii) and 7.2 in $\mathbb{P}^4$. Standard deviation of $\theta$, see step 4 in 8.1, for 1000 reconstructions, as a function of a fixed value of the standard deviation of the 4D perturbation.

**Instability Of Reconstruction for 7.1**
This case refers to example 7.1, thus $k = 4$. To perform the experiment we have chosen a specific numerical case. In particular $\mathbf{v} = (0,1,0)^T$ has been chosen as the common direction of all trajectories, and the projection matrices are the following:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & t \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad Q = \begin{bmatrix} 0 & 1 & 0 & 0 & t \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

In this case the critical locus has equations $[\theta^3, -\theta^2 - bt; \theta, 1, b]$, in the notations of example 7.1. Fig. 1 shows results obtained as described in steps 1 through 5 above. Here $t$ was set to 1 for simplicity, and $\sigma \in [0, 0.5]$ with iterations every 0.0125.
**Instability Of Reconstruction for 7.2**
This case refers to example 7.2, thus $k = 4$. Also here, to perform the experiment, we have chosen a specific numerical case. In particular the projection matrices are the following:

$$P = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \qquad Q = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

In this case the critical locus has equations $[1, \frac{4-v^2}{v(1+v)} - u, -v - u, \frac{v^2-4}{v} + u, u]$. Fig. 2 shows results obtained as described in steps 1 through 5 above. Here $\sigma \in [0, 0.2]$ with iterations every 0.0005.
As Fig. 2, top graph, indicated that in this case reconstruction gets fairly stable quite quickly as $\sigma$ grows, Fig. 2, bottom graph, was obtained using a finer grid for $\sigma$ than in other previous cases. This enables one to see instability still occurring when $\sigma$ is in the range $[0, 0.005]$.

## *8.2 The Case of Multiple Views*

Results of instability of projective reconstruction from several views in situations described in Section 3 i) and ii) are contained in [3] and [4]. We reproduce here the experiments from the latter, which leverage the implementation of algorithms described in Section 5. Essential steps of the experimental process in these cases are described below.

1 *Generation of Critical Configurations*
Given two sets of projection matrices $\{P_i\}$ and $\{Q_i\}$, $i = 0, 1, 2$ of the appropriate

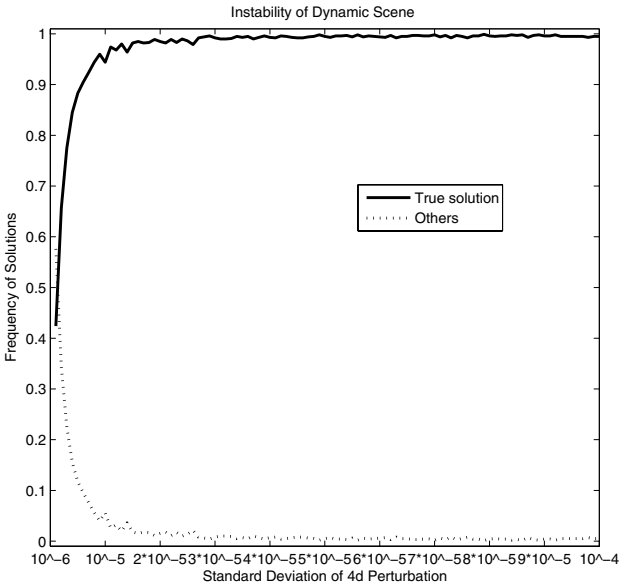



**Fig. 3** Instability of reconstruction of a dynamic scene of multiple bodies moving along straight, parallel trajectories with constant velocities, as described in Section 3 ii) and 7.1, near the critical locus: frequency with which the reconstructed solution is close or not to the true solution as function of the standard deviation of the 4D perturbation. "Close" here is defined as being within half the distance between the true solution and the known conjugate solution

type, see Subsections 7.1 and 7.2, equations of the Bordiga surface, which is the essential part of the critical locus in both cases, are obtained with the help of Maple [18]. Equations are then solved to retrieve critical configurations $\{\mathbf{X_i}\}$ of at least 13 points in $\mathbb{P}^4$. Extra care is taken in order to obtain points with rational coordinates, lying on the Bordiga surface, to avoid numerical approximations at this stage.

2 *Perturbation of critical configurations*
Points $\{\mathbf{X_i}\}$ are then perturbed with a 4D noise, normally distributed, with zero mean, and with assigned standard deviation $\sigma$, obtaining a new configuration $\{\mathbf{X_i}^{pert}\}$, which is close to being critical. This configuration is projected via $P1, P2, P3$, where the $Pj's$ are as in Subsections 7.1 and 7.2. The resulting images $\mathbf{x_{ij}} = Pj\mathbf{X_i}^{pert}$ are again perturbed with normally distributed 2D noise with zero mean and standard deviation 0.001 to obtain $\{\mathbf{x_{ij}}^{pert}\}$.

3 *Reconstruction*
The trifocal tensor corresponding to the true reconstruction, $T_P$ and its known conjugate, $T_Q$, are computed respectively from $P1, P2, P3$ and $Q1, Q2, Q3$, where again the $Pj's$ and $Qj's$ are as in Subsections 7.1 and 7.2, using algorithm 1

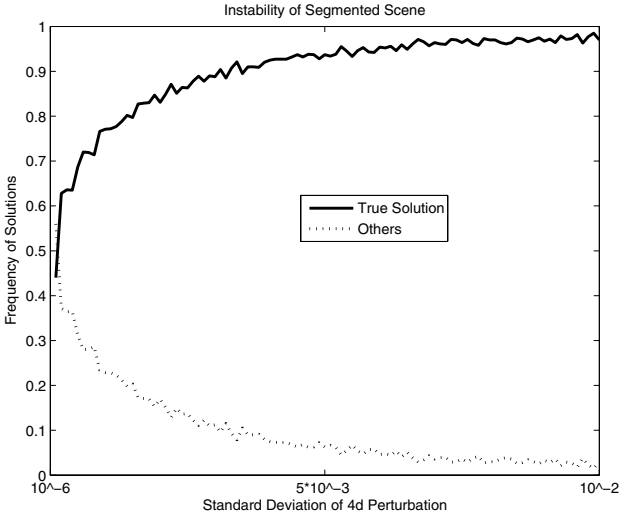



**Fig. 4** Instability of reconstruction of a 2-body segmented scene as described in Section 3 ii) and 7.2 near the critical locus: Frequency with which the reconstructed solution is close or not to the true solution as function of the standard deviation of the 4D perturbation."Close" here is defined as being within half the distance between the true solution and the known conjugate solution

described in Section 5. An estimated trifocal tensor $T$ is computed from $\{\mathbf{x_{ij}}^{pert}\}$, using algorithm 2 described in Section 5.

4 *Estimating instability*
As trifocal tensors are defined up to multiplication by a non-zero constant, $T_P, T_Q$ and $T$ are normalized and the space of trifocal tensors is then identified with a quotient of the unit sphere in $\mathbb{R}^{27}$, $S^{26}/\simeq$, where $\simeq$ denotes antipodal identification. It is simple to account for antipodal identification when computing distances: for any pair of tensors with unit norm $A$ and $B$, we set $d(A,B) = min(||A-B||, ||A+B||)$. Using this notion of distance, we estimate whether $T$ is close to $T_P$, or not, where "close" means within a hypersphere of radius $\frac{d(T_P,T_Q)}{2}$. The above procedure is then repeated 1000 times for every fixed value of $\sigma$.

5 *Displaying the results*
Results are then presented in Figs. 3 and 4, showing the frequency with which the reconstructed solution is close or far from the true solution $T_P$, against the values of $\sigma$ utilized. In both cases we ran a number of simulated experiments. Here we present two graphs for each case, the dynamic scene, described in Section 3 i) and 7.1, in Fig. 3, and the segmented scene, described in Section 3 ii) and 7.2, in Fig. 4. In Fig. 3 it is $\sigma \in (10^{-6}, 10^{-4})$, sampled every $10^{-6}$. In Fig. 4 it is $\sigma \in (10^{-6}, 10^{-2})$, sampled every $10^{-4}$.

In all graphs both the x-scale and y-scale are linear.

These results show that reconstruction in both cases is indeed unstable for configurations of points that are close to the critical locus. For example, in the segmented case, stabilization of the solution become apparent only for $\sigma > 0.01$.

## References


1. Arbarello, E., Cornalba, M., Griffiths, P., Harris, J.: Geometry of Algebraic Curves, Volume I. No. 267 in Grundlehren der Mathematischen Wissenschaften. Springer Verlag, Berlin - Heidelberg - New York - Tokyo (1985)
2. Bertolini, M., Besana, G., Turrini, C.: Instability of projective reconstruction from 1-view near critical configurations in higher dimensions. In: Algebra Geometry and their Interactions, *Contemporary Mathematics*, vol. 448, pp. 1–12 (2007)
3. Bertolini, M., Besana, G., Turrini, C.: Instability of projective reconstruction of dynamic scenes near critical configurations. In: Proceedings of the International Conference on Computer Vision, ICCV (2007)
4. Bertolini, M., Besana, G., Turrini, C.: Reconstruction of some segmented and dynamic scenes: trifocal tensors in $\mathbb{P}^4$, theoretical set up for critical loci, and instability. In: Proceedings of the International Symposium on Visual Computing, ISVC (2008)
5. Bertolini, M., Turrini, C.: Critical configurations for 1-view in projections from $\mathbb{P}^k \to \mathbb{P}^2$. Journal of Mathematical Imaging and Vision **27**, 277–287 (2007). DOI 10.1007/s10851-007-0649-6
6. Buchanan, T.: The twisted cubic and camera calibration. Comput. Vision Graphics Image Process. **42**(1), 130–132 (1988)
7. Fan, X., Vidal, R.: The space of multibody fundamental matrices: Rank, geometry and projection. In: Dynamical Vision, *Lecture Notes in Computer Science*, vol. 4358, pp. 1–17. Springer Berlin Heidelberg (2007)
8. Grayson, D.R., Stillman, M.E.: Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/
9. Harris, J.: Algebraic geometry, *Graduate Texts in Mathematics*, vol. 133. Springer-Verlag, New York (1995). A first course, Corrected reprint of the 1992 original
10. Hartley, R.: Ambiguous configurations for 3-view projective reconstruction. In: European Conference on Computer Vision, pp. I: 922–935 (2000)
11. Hartley, R., Kahl, F.: Critical configurations for projective reconstruction from multiple views. International Journal of Computer Vision **71**(1), 5–47 (2007)
12. Hartley, R., Vidal, R.: The multibody trifocal tensor: Motion segmentation from 3 perspective views. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. I, pp. 769–775 (2004)
13. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, second edn. Cambridge University Press, Cambridge (2003).
14. Hartley, R.I., Schaffalitzky, F.: Reconstruction from projections using Grassmann tensors. In: Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic, Lecture Notes of Computer Sciences 3021: 363-375 (2004)
15. Huang, K., Fossum, R., Ma, Y.: Generalized rank conditions in multiple view geometry with applications to dynamical scenes. In: European Conference on Computer Vision (ECCV) 2:201–216 (2002)
16. Kahl, F., Hartley, R., Astrom, K.: Critical configurations for n-view projective reconstruction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. II:158–163 (2001)
17. Krames, J.: Zur ermittlung eines objectes aus zwei perspectiven (ein beitrag zur theorie der gefhrlichen rter). Monatsh. Math. Phys. **49**, 327–354 (1940)
18. Maplesoft: Maple. Avaliable at http://www.maplesoft.com

19. Maybank, S.: Theory of Reconstruction from Image Motion. Springer-Verlag New York, Secaucus, NJ, USA (1992)
20. Shashua, A., Maybank, S.: Degenerate *n* point configurations of three views: Do critical surfaces exist? TR 96-19, Hebrew University, Jerusalem (1996)
21. Vidal, R., Ma, Y.: A unified algebraic approach to 2-D and 3-D motion segmentation and estimation. J. Math. Imaging Vis. **25**(3), 403–421 (2006). DOI http://dx.doi.org/10.1007/s10851-006-8286-z
22. Vidal, R., Ma, Y., Soatto, S., Sastry, S.: Two-view multibody structure from motion. Int. J. Comput. Vision **68**(1), 7–25 (2006). DOI http://dx.doi.org/10.1007/s11263-005-4839-7
23. Wolf, L., Shashua, A.: On projection matrices $\mathbb{P}^k \rightarrow \mathbb{P}^2, k =, 3, ..., 6$ and their applications in computer vision. International Journal of Computer Vision **48**(1), 53–67 (2002)

# Constraints for the Trifocal Tensor

Alberto Alzati and Alfonso Tortora

**Abstract** In this chapter we give an account of two different methods to find constraints for the trifocal tensor $T$, used in geometric computer vision. We also show how to single out a set of only eight equations that are generically complete, i.e. for a generic choice of $T$, they suffice to decide whether $T$ is indeed trifocal. Note that eight is minimum possible number of constraints.

## 1 Introduction

The *trifocal tensor* is the mathematical object relating three camera views with each other. It solves the following problem: given three cameras, whose positions and internal parameters are known, and given the images of two of them, reconstruct the image of the third one.

Mathematically, a camera is a projection $\pi : \mathbb{P}^3 \to \mathbb{P}^2$, thus it is represented, up to a nonzero factor, by a $3 \times 4$ matrix $M$ (of maximal rank). Now, given three cameras $\pi, \pi', \pi''$ and the images of $\pi'$ and $\pi''$, the recovery (from them) of the image of $\pi$ translates into a map $\mathbb{P}^2 \times \mathbb{P}^2 \to \mathbb{P}^2$, which in turn is equivalent, up to a nonzero factor, to a bilinear map $\mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$, i.e. it is a tensor $T \in \mathbb{R}^3 \otimes \mathbb{R}^3 \otimes \mathbb{R}^3$, the *trifocal tensor*; since it is defined up to a factor, $T$ is actually an element of $\mathbb{P}(\mathbb{R}^3 \otimes \mathbb{R}^3 \otimes \mathbb{R}^3) = \mathbb{P}^{26}$.

Since not all tensors in $\mathbb{R}^3 \otimes \mathbb{R}^3 \otimes \mathbb{R}^3$ are trifocal tensors of a suitable configuration of cameras, the problem arises of finding the *constraints*, i.e. equations that determine all trifocal tensors.

Alberto Alzati
Dipartimento di Matematica, Università di Milano, Italy e-mail: alberto.alzati@unimi.it

Alfonso Tortora
Dipartimento di Matematica, Università di Milano, Italy e-mail: afonso.tortora@unimi.it

Let us recall how a trifocal tensor is computed, given the three cameras (see [2] ch. 14 and 15, whose notations and conventions we broadly follow). Choosing suitable coordinates in the ambient space $\mathbb{P}^3$, we can always assume that the matrix $M$ of the camera $\pi$ is of type $M = [I_3 \mid \mathbf{0}]$, then the matrices of $\pi'$ and $\pi''$ are $M' = [A \mid \mathbf{a}_4]$ and $M'' = [B \mid \mathbf{b}_4]$ respectively, with $\det A, \det B \neq 0, A = [\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3], B = [\mathbf{b}_1\ \mathbf{b}_2\ \mathbf{b}_3]$. The $3 \times 3 \times 3$ tensor $T$ can be written as $T = [T_1 \mid T_2 \mid T_3]$, each $T_i$ being a $3 \times 3$ matrix—think of it as the i-th level "slice" of the "cube" $T$.

It is possible to prove that

$$T_i = \mathbf{a}_i \mathbf{b}_4^\top - \mathbf{a}_4 \mathbf{b}_i^\top .$$

Also, the set of all trifocal tensors (what we call the *trifocal locus*) has dimension 18. At this point, the task of finding constraints for the trifocal locus $\Theta$ is seemingly straightforward: just eliminate the parameters $a$'s and $b$'s in the relations above—explicitly, they are the 27 equations (2) of section 4—and get equations in the entries of $T$, and those are a full set of constraints. Since $\Theta$ has codimension 8 in its ambient space $\mathbb{P}^{26}$, a full set of constraints contains at least eight equations. The elimination is, alas, unworkable: too many variables and parameters, linked by relations of second degree.

Thus we find constraints using two different methods: one based on group theory, another using more down–to–earth geometric arguments. We give an account of the results, with a broad outline of the arguments leading thereto, referring to our forthcoming paper [1] for the details of the proofs.

**Remark** To use standard methods and results of group theory and algebraic geometry, it is easier to switch from $\mathbb{R}$ to $\mathbb{C}$—technically, we want the base field to be algebraically closed. Especially, a camera will be a projection $\mathbb{P}^3(\mathbb{C}) \to \mathbb{P}^2(\mathbb{C})$ and trifocal tensors will belong to $T \in \mathbb{C}^3 \otimes \mathbb{C}^3 \otimes \mathbb{C}^3$. Nevertheless, the constraints that we find, are defined over $\mathbb{R}$.

## 2 The Trifocal Tensor

Let $\mathbb{K}$ be a field, either $\mathbb{R}$ or $\mathbb{C}$; also, we denote $V := \mathbb{K}^3, W := \mathbb{K}^4$.

**Definition 2.1** *A* (projective) camera *is a projection $\pi$ from a point $C$, the center of the camera,*

$$\pi : \mathbb{P}^3 = \mathbb{P}(W^*) \to \mathbb{P}^2 = \mathbb{P}(V^*).$$

*It is represented by a $3 \times 4$ matrix $M$ having maximal rank; hence $\mathbf{x} = \pi(\mathbf{X}) \Leftrightarrow \mathbf{x} = M\mathbf{X}$*

The points $\mathbf{X} = (X^1, X^2, X^3, X^4)^\top \in \mathbb{P}^3$ and $\mathbf{x} = (x^1, x^2, x^3)^\top \in \mathbb{P}^2$ are (contravariant) column vectors; identifying $\pi$ with $M$, which is itself determined up to a nonzero factor, a camera is then an element of $\mathbb{P}(W \otimes V^*) = \mathbb{P}^{11}$.

The *trifocal tensor* describes the following geometric situation. Given three cameras $\pi, \pi', \pi''$ in general position (i.e. whose centers are not collinear), a line $\mathbf{L}$ of

the ambient space $\mathbb{P}^3$ is imaged as $\mathbf{l} = \pi(\mathbf{L}), \mathbf{l}' = \pi'(\mathbf{L})$ and $\mathbf{l}'' = \pi''(\mathbf{L})$; conversely, given two of these images, e.g. $\mathbf{l}'$ and $\mathbf{l}''$, it is possible to recover the third one, $\mathbf{l}$. So we have a map $T : (\mathbf{l}', \mathbf{l}'') \in \mathbb{P}^{2*} \times \mathbb{P}^{2*} \to \mathbf{l} \in \mathbb{P}^{2*}$, coming from a bilinear map $V \times V \to V$, hence it is represented by a tensor $T = [t_i^{jk}]_{i,j,k=1,2,3} \in V \otimes V^* \otimes V^*$, the trifocal tensor.

Recall that a line $\mathbf{l} \in \mathbb{P}^2 = \mathbb{P}(V)$ is represented by a (covariant) row vector $\mathbf{l} = (l_1, l_2, l_3)$, so the trifocal tensor is 1–covariant and 2–contravariant and the map $T$ is given by

$$T(\mathbf{l}', \mathbf{l}'') = (t_i^{jk} l'_j l''_k)_{i=1,2,3}$$

(here we observe the convention that repeated indices in covariant and contravariant positions imply summation).

A trifocal tensor $T$, being defined up to a nonzero factor, is an element of $\mathbb{P}(V \otimes V^* \otimes V^*) = \mathbb{P}^{26}$.

**Definition 2.2** *The* trifocal map

$$\mathscr{T} : (V^* \otimes W)^3 \to V \otimes V^* \otimes V^*.$$

*maps a triple of matrices $(M, M', M'')$ representing three cameras to the trifocal tensor $T = [t_i^{jk}]$ associated to them.*

Write $M = \begin{bmatrix} \mathbf{m}^1 \\ \mathbf{m}^2 \\ \mathbf{m}^3 \end{bmatrix}$, where $\mathbf{m}^i = (m_1^i, m_2^i, m_3^i)$ is the (covariant) i–th row vector, and similarly $M', M''$; also , let $\{i, r, s\} = \{1, 2, 3\}$ and let $\varepsilon(irs) = \pm 1$ be the sign of the permutation $\begin{pmatrix} 1 & 2 & 3 \\ i & r & s \end{pmatrix}$.

**Proposition 2.3** ([1]) *The entries of the trifocal tensor $T = [t_i^{jk}]$, associated to the matrices $M, M', M''$, are*

$$t_i^{jk} = \varepsilon(irs) \det \begin{pmatrix} \mathbf{m}^r \\ \mathbf{m}^s \\ \mathbf{m}'^j \\ \mathbf{m}''^k \end{pmatrix}. \tag{1}$$

Note that, although defined for any triple of matrices, $\mathscr{T}(M, M', M'')$ is actually a trifocal tensor only if the matrices represent cameras (i.e their ranks are maximal) that are in general position (i.e. their centers are not collinear).

$\mathscr{T}$ induces a map, still referred to as the trifocal map

$$\mathfrak{T} : (\mathbb{P}(V^* \otimes W))^3 \to \mathbb{P}(V \otimes V^* \otimes V^*).$$

The map $\mathfrak{T}$ is not defined everywhere, but only on an open dense set; in the language of algebraic geometry, it is a rational map. The image of $\mathfrak{T}$ contains the set of all trifocal tensors, for all configurations of three cameras in general position. It

is well known that Im$\mathfrak{T}$ has dimension 18; since $\mathfrak{T}$, as a rational map, is not defined everywhere, Im$\mathfrak{T}$ is not closed in $\mathbb{P}^{26} = \mathbb{P}(V \otimes V^* \otimes V^*)$.

**Definition 2.4** *The* trifocal locus $\Theta$ *is the (Zariski) closure of the image of the trifocal map:*

$$\Theta := \overline{Im\mathfrak{T}}.$$

A full set of *constraints* is a set of homogeneous polynomials (in the 27 variables $t_i^{jk}$) whose common zeroes in $\mathbb{P}^{26}$ are exactly the elements of the trifocal locus $\Theta$. Since $\Theta$ has codimension 8 in its ambient space, the best one can hope for is to find exactly eight polynomials that cut out $\Theta$.

## 3 Constraints via Group Theory

The group $\Gamma := \mathrm{GL}(W)$ of all linear transformations of $W$ parametrizes also all changes of coordinates in the ambient space $\mathbb{P}^3 = \mathbb{P}$. Given a configuration of three cameras $\pi, \pi', \pi''$ in general position, represented by the matrices $M, M', M''$, let $T = \mathscr{T}(M, M', M'')$ be the relative trifocal tensor; if $\mathbf{L}$ is a line in the ambient space $\mathbb{P}^3$ and $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ its images in the three cameras, then $T(\mathbf{l}', \mathbf{l}'') = \mathbf{l}$. Suppose now to change coordinates in $\mathbb{P}^3$ via $\gamma \in \Gamma$—recall that $\gamma$ is a $4 \times 4$ nonsingular matrix—but to leave coordinates unchanged in the images $\mathbb{P}^2$ of the three cameras. What happens is that the line $\mathbf{L}$ in the ambient space becomes $\gamma\mathbf{L}$, while its images $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ stay put, and $\pi, \pi', \pi''$ are represented by $M\gamma, M'\gamma, M''\gamma$.

If $\tilde{T} = \mathscr{T}(M\gamma, M'\gamma, M''\gamma)$ is the trifocal tensor relative to this "new" configuration, since it satisfies the same relation $\tilde{T}(\mathbf{l}', \mathbf{l}'') = \mathbf{l}$ as before, then $\tilde{T} = T$ up to a nonzero factor. Technically, this means that the trifocal map is invariant under the action of the group $\Gamma$, i.e.

$$\forall \gamma \in \Gamma \ \mathscr{T}(M\gamma, M'\gamma, M''\gamma) = \mathscr{T}(M, M', M'').$$

Similarly, the group $G := \mathrm{GL}(V)$ parametrizes changes of coordinates in the image $\mathbb{P}^2 = \mathbb{P}(V^*)$ of a camera. Arguing as before, it can be seen that the trifocal map is equivariant under the action of the group $G^3 := G \times G \times G$, i.e.

$$\forall g_1, g_2, g_3 \in G \ \mathscr{T}(M^{g_1}, M'^{g_2}, M''^{g_3}) = \mathscr{T}(M, M', M'')^{(g_1, g_2, g_3)}$$

(there is a technical point here: the action of $G$ on $V^*$ is the so called dual action).

The equivariance of the trifocal map implies that its image (whose closure in $\mathbb{P}^{26}$ is the trifocal locus) is invariant under the action of $G^3$ on $V \otimes V^* \otimes V^*$, thus it is an orbit of this action.

Such an orbit has long since been known to mathematicians, so we can use results of [3] to pinpoint the orbit whose elements are trifocal tensors and find the constraints.

We now give a brief summary of the results of [3]. Given a tensor $A \in V \otimes V \otimes V$, in a fixed coordinate system $A$ is represented by a $3 \times 3 \times 3$ numerical tensor $A =$

$(a_{ijk})_{i,j,k=1,2,3}$; in the notation of [3], we can identify $A$ with a trilinear form (on $V$) $F(\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{ijk} a_{ijk}x_iy_jz_k$, where $\mathbf{x} = (x_1,x_2,x_3)$, $\mathbf{y} = (y_1,y_2,y_3)$, $\mathbf{z} = (z_1,z_2,z_3)$ are three sets of three variables each.

The matrix $H_x = [\sum_i a_{ijk}x_i]_{i,j=1,2,3}$, associated to $A$ (or $F$), is a $3\times 3$ matrix whose entries are linear forms in the variables $x_i$; define $X(x) := \det(H_x)$, a cubic form in the $x_i$. Similarly, we consider the matrices $H_y = [\sum_j a_{ijk}y_j]_{i,k=1,2,3}$ and $H_z = [\sum_k a_{ijk}z_k]_{j,k=1,2,3}$, and define the corresponding cubic forms $Y(y) := \det(H_y)$ and $Z(z) := \det(H_z)$.

The main results of [3] are:

i the projective classes of the plane cubics, given by the following equations $X(x) = 0, Y(y) = 0, Z(z) = 0$ in $\mathbb{P}^2$, are invariants of the tensor $A$ with respect to the $G^3$ action, defined in [3], on the trilinear form $F$ ;
ii the projective classes of the plane cubics determine the orbit of the tensor $A$.

At this point, we only need an actual trifocal tensor, and compute the relative plane cubics $X(x) = 0, Y(y) = 0, Z(z) = 0$, in order to find the orbit whose elements are all trifocal tensors. Such a trifocal tensor $\Delta = [d_i^{jk}]$ is e.g.

$$\Delta_1 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \Delta_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \Delta_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix},$$

where we write $\Delta_i := [d_i^{jk}]_{j,k=1,2,3}$, $i = 1,2,3$. The trilinear forms associated to $\Delta$ are

$$X(x) \equiv 0,\ Y(y) = y_3^2(y_3 - y_2),\ Z(z) = z_2^2(z_3 - z_2).$$

Looking at the table of [3, p. 689], we see that there is a unique orbit with these invariants—it is the one corresponding to the entry in the second row and last column in that table. Hence, a numerical tensor $T = [t_i^{jk}]$ is a trifocal tensor if and only if the associated cubics are of the following types: $X(x)$ is identically zero, $Y(y)$ and $Z(z)$ are reducible, both being the union of a double line and another line.

To translate these conditions into equations on the entries of $T$ we recall that the coefficients of $X(x), Y(y), Z(z)$ are in turn homogeneous polynomials of degree three in the entries $t_i^{jk}$ of $T$, thus $X(x) \equiv 0$ amounts to impose that all these 10 coefficients are zero, i.e. ten cubic equations.

The requirement that $Y(y)$ is the union of a double line and another line is equivalent to the following two conditions:
(i) $Y(y)$ has a triple point (so that the curve is reducible into 3 concurrent lines) and (ii) given any three non concurrent lines, e.g. the coordinate lines (in $\mathbb{P}^2$), each of them has at least a double intersection with the curve $Y(y)$.

Condition (i) translates into imposing that the Hessian of $Y(y)$ be identically zero. The Hessian being in this case a form of degree three (in $\mathbf{y}$), whose coefficients are homogeneous polynomials of degree nine in the $t_i^{jk}$, condition (i) is eventually equivalent to ten equations of degree 9. Condition (ii) is shown (cf. [1]) to be equivalent to three equations of degree 4 in the coefficients of $Y(y)$, hence of degree 12

in the $t_i^{jk}$. Clearly the number of equations arising from conditions (i) and (ii) must be doubled, taking into account $Z(z)$; the upshot is the following

**Theorem 3.1** ([1]) *A complete set of constraints for the trifocal locus $\Theta$ is given by 10 equations of degree three, 20 equations of degree 9 and 6 equations of degree 12 on the 27 entries of a generic $3\times 3\times 3$ tensor.*

## 4 Constraints via Geometry

In the previous section we saw that the trifocal map $\mathscr{T}$ is invariant under the action of $\Gamma = \mathrm{GL}(W)$ on $\mathbb{P}(V^*\otimes W)$; since $M$, the matrix representing the camera $\pi$ has rank three, then there exists a $\gamma\in\Gamma$ such that $M\gamma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. In other terms, it is always possible to choose coordinates in the ambient space $\mathbb{P}^3 = \mathbb{P}(W^*)$ so that the matrices representing the three cameras are of the form $M = [I_3 \mid \mathbf{0}], M' = [a_i^j], M'' = [b_i^j]$. For such a choice of matrices, the relations (1) expressing the entries of the trifocal tensor $T = [t_i^{jk}]$ become

$$t_i^{jk} = a_i^j b_4^k - a_4^j b_i^k. \tag{2}$$

As pointed out in the introduction, finding the constraints of $\Theta$ by eliminating the parameters in (2) seems to be computationally too hard, due to the great number of variables and parameters (27 and 24 respectively) linked by relations of second degree. We write now a tensor $T$ as $T = [T_1 \mid T_2 \mid T_3]$, where $T_i$ is a $3\times 3$ matrix $T_i = [t_i^{jk}]_{j,k=1,2,3}$. We denote by $\mathbf{t}_i^{j\bullet}$ and $\mathbf{t}_i^{\bullet k}$ the j-th row vector and the k-th column vector of the matrix $T_i$ respectively; for example $\mathbf{t}_2^{3\bullet} = (t_2^{31}, t_2^{32}, t_2^{33})$. We denote by $R_i$ and $C_i$ the subspaces (of $\mathbb{K}^3$) generated by the row vectors and the column vectors of $T_i$ respectively, i.e. $R_i = \langle \mathbf{t}_i^{1\bullet}, \mathbf{t}_i^{2\bullet}, \mathbf{t}_i^{3\bullet}\rangle$ and $C_i = \langle \mathbf{t}_i^{\bullet 1}, \mathbf{t}_i^{\bullet 2}, \mathbf{t}_i^{\bullet 3}\rangle$.

With these notations, (2) becomes

$$T_i = \mathbf{a}_i\mathbf{b}_4^\top - \mathbf{a}_4\mathbf{b}_i^\top \quad i = 1,2,3 \tag{3}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ are the i-th column of the matrices $M'$ and $M''$ respectively; thinking of $T_i$ as an element of $V^*\otimes V^* \simeq \mathbb{K}^3\otimes\mathbb{K}^3$, we have

$$T_i = \mathbf{a}_i\otimes\mathbf{b}_4 - \mathbf{a}_4\otimes\mathbf{b}_i. \tag{4}$$

Our goal is to find conditions that a tensor $T = [t_i^{jk}]$ must satisfy in order to be of the form (3)—or equivalently (4).

To begin with, the spaces $R_i$ and $C_i$ must have dimension (at most) two: indeed, $R_i = \langle \mathbf{b}_i^\top, \mathbf{b}_4^\top\rangle$ and $C_i = \langle \mathbf{a}_i, \mathbf{a}_4\rangle$; algebraically, this implies

$$\det T_i = 0, \quad i = 1,2,3. \tag{5}$$

Now, assuming $\dim C_i = 2, i = 1,2,3$, the subspaces $C_1, C_2, C_3$ must intersect, in $\mathbb{K}^3$, along a subspace of dimension (at least) one, since $C_1 \cap C_2 \cap C_3 \supseteq \langle \mathbf{a}_4 \rangle$. This in turn means that, given vectors $\mathbf{v}_i$ normal to $C_i, i = 1,2,3$, the vectors $\mathbf{v}_i$ are coplanar, hence linearly dependant.

The vectors $\mathbf{t}_i^{\bullet r} \wedge \mathbf{t}_i^{\bullet s}, r \neq s$ are either zero or perpendicular to $C_i$ and, since $\dim C_i = 2$, at least one of them is nonzero (if $\dim C_i < 2$, all of them are zero). Thus the condition $\dim C_1 \cap C_2 \cap C_3 = 1$ implies

$$\det(\mathbf{t}_1^{\bullet r_1} \wedge \mathbf{t}_1^{\bullet s_1} \;\; \mathbf{t}_2^{\bullet r_2} \wedge \mathbf{t}_2^{\bullet s_2} \;\; \mathbf{t}_3^{\bullet r_3} \wedge \mathbf{t}_3^{\bullet s_3}) = 0; \tag{6}$$

these are relations of degree six, and there are 27 of them. Likewise, considering the spaces $R_i$ another 27 relations of degree six occur; they are

$$\det \begin{pmatrix} \mathbf{t}_1^{p_1 \bullet} \wedge \mathbf{t}_1^{q_1 \bullet} \\ \mathbf{t}_2^{p_2 \bullet} \wedge \mathbf{t}_2^{q_2 \bullet} \\ \mathbf{t}_3^{p_3 \bullet} \wedge \mathbf{t}_3^{q_3 \bullet} \end{pmatrix} = 0. \tag{7}$$

Being now in the situation: $\dim C_i = \dim R_i = 2, \dim C_1 \cap C_2 \cap C_3 = \dim R_1 \cap R_2 \cap R_3 = 1$, we want that $T_i \in C_i \otimes R_i$ be in the form (4). This is a little trickier to translate into algebraic relations; in [1] we show that it implies the following algebraic relations

$$(\mathbf{t}_k^{p \bullet} \wedge \mathbf{t}_k^{q \bullet}) \cdot T^i \cdot (\mathbf{t}_j^{\bullet r} \wedge \mathbf{t}_j^{\bullet s}) = 0, \tag{8}$$

for any $i, j, k, p, q, r, s = 1,2,3$, with $j \neq i$, $k \neq i$, $p \neq q$, $r \neq s$, where $\cdot$ represents the usual row–column product of matrices—recall that $\mathbf{t}_k^{p \bullet} \wedge \mathbf{t}_k^{q \bullet}$ is a row vector and $\mathbf{t}_j^{\bullet r} \wedge \mathbf{t}_j^{\bullet s}$ is a column vector, so we end up with a scalar. These are quintic relations and there are 108 of them.

The relations (5)—(8) just found are necessary conditions for a tensor to be trifocal, i.e. of the form (3). In general they are not sufficient. More precisely, (5)—(8) are sufficient if we assume furthermore that $\dim C_i = \dim R_i = 2$ (note that (5) implies only that $\dim C_i = \dim R_i \leq 2$) and $\dim \cap C_i = \dim \cap R_i = 1$ (while (6) and (7) implies only that $\dim \cap C_i, \dim \cap R_i \geq 1$). A careful analysis shows that the locus of (5)—(8) in $\mathbb{P}^{26}$ has several components, of which the trifocal locus $\Theta$ is one of maximal dimension. Summing up:

**Theorem 4.1** ([1]) *Let $\Omega$ be the subvariety of $\mathbb{P}^{26}$ defined by equations (5)—(8), then the trifocal locus $\Theta$ is an irreducible component, of maximal dimension, of the variety $\Omega$.*

## 5 A Minimal Set of Constraints

Both set of constraints that we found contains many more relations than the possible minimum, i.e. eight relations.

Whether it is possible to achieve such a minimal complete set of eight constraints is not yet clear (technically, this means that the trifocal locus $\Theta$ is a complete inter-

section); the best we can do is to find a set of eight constraints that is *generically* complete, i.e. they cut out $\Theta$ almost everywhere (outside of a Zariski closed subset of $\mathbb{P}^{26}$).

We do so by taking all three cubic equations (5), two sextic equations, one chosen among the relations (6) and another among (7), and three quintic equations among (8), one for each of the slices $T^i$; then we add, as inequalities, two suitably chosen $2\times 2$ minors of the matrices

$$A := \begin{bmatrix} \mathbf{t}_1^{r_1\bullet} \wedge \mathbf{t}_1^{s_1\bullet} \\ \mathbf{t}_2^{r_2\bullet} \wedge \mathbf{t}_2^{s_2\bullet} \\ \mathbf{t}_3^{r_3\bullet} \wedge \mathbf{t}_3^{s_3\bullet} \end{bmatrix}, \quad B := [\mathbf{t}_1^{\bullet p_1} \wedge \mathbf{t}_1^{\bullet q_1} \;\; \mathbf{t}_2^{\bullet p_2} \wedge \mathbf{t}_2^{\bullet q_2} \;\; \mathbf{t}_3^{\bullet p_3} \wedge \mathbf{t}_3^{\bullet q_3}].$$

For example, we take, besides (5), the following sextic equations

$$\begin{aligned} &\det(\mathbf{t}_1^{1\bullet} \wedge \mathbf{t}_1^{2\bullet} \;\; \mathbf{t}_2^{1\bullet} \wedge \mathbf{t}_2^{2\bullet} \;\; \mathbf{t}_3^{1\bullet} \wedge \mathbf{t}_3^{2\bullet}) = 0, \\ &\det(\mathbf{t}_1^{\bullet 1} \wedge \mathbf{t}_1^{\bullet 2} \;\; \mathbf{t}_2^{\bullet 1} \wedge \mathbf{t}_2^{\bullet 2} \;\; \mathbf{t}_3^{\bullet 1} \wedge \mathbf{t}_3^{\bullet 2}) = 0 \end{aligned} \tag{9}$$

and quintic equations

$$\begin{aligned} &(\mathbf{t}_2^{1\bullet} \wedge \mathbf{t}_2^{2\bullet}) \cdot T^1 \cdot (\mathbf{t}_2^{\bullet 1} \wedge \mathbf{t}_2^{\bullet 2}) = 0, \\ &(\mathbf{t}_1^{1\bullet} \wedge \mathbf{t}_1^{2\bullet}) \cdot T^2 \cdot (\mathbf{t}_3^{\bullet 1} \wedge \mathbf{t}_3^{\bullet 2}) = 0, \\ &(\mathbf{t}_1^{1\bullet} \wedge \mathbf{t}_1^{2\bullet}) \cdot T^3 \cdot (\mathbf{t}_2^{\bullet 1} \wedge \mathbf{t}_2^{\bullet 2}) = 0. \end{aligned} \tag{10}$$

We denote the quartic inequalities by

$$F(t) \neq 0, \quad G(t) \neq 0 \tag{11}$$

where: $F(t)$ is the determinant of the $2\times 2$ submatrix of the matrix $A$, determined by the first and second rows and the first and second columns; $G(t)$ is the determinant of the $2\times 2$ submatrix of the matrix $B$, determined by the first and second rows and the second and third columns.

The idea is that the inequalities (11), coupled with (5), imply that $\dim C_i = \dim R_i = 2$ and $\dim \cap C_i = \dim \cap R_i = 1$, and also that all vectors $\mathbf{t}_1^{1\bullet} \wedge \mathbf{t}_1^{2\bullet}$, $\mathbf{t}_2^{1\bullet} \wedge \mathbf{t}_2^{2\bullet}$, $\mathbf{t}_2^{\bullet 1} \wedge \mathbf{t}_2^{\bullet 2}$, $\mathbf{t}_3^{\bullet 1} \wedge \mathbf{t}_3^{\bullet 2}$ are nonzero, so the argument of the previous section, showing that (5), (9) and (10) are necessary conditions, can be reversed; what we get is the following

**Theorem 5.1** ([1]) *Let $I$ be the ideal generated by the eight polynomials (5), (9) and (10), then it defines the trifocal locus $\Theta$, outside the hypersurface $F(t)G(t) = 0$.*

## References

1. Alzati, A., Tortora, A.: A geometric approach to the trifocal tensor. *To appear*
2. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, Cambribge (2000)

3. Thrall, R. M., Chandler, J. H.: Ternary trilinear forms in the field of complex numbers. Duke Math. J. **4**, 678–690 (1938)

# Part IV
# Diffusion Tensor Imaging and Medical Applications

# Review of Techniques for Registration of Diffusion Tensor Imaging

Emma Muñoz-Moreno, Rubén Cárdenes-Almeida and Marcos Martin-Fernandez

**Abstract** Image registration is a common image processing task, and therefore, many algorithms have been proposed and described to carry it out for different image modalities. However, the application of these algorithms to diffusion tensor imaging is not straightforward due to the special features of this kind of data, where a tensor is defined at each voxel. The information provided by the diffusion tensor is related to the anatomical structures in tissues, and this relation should be preserved, even though the image has been transformed by a registration procedure. On the other hand, the registration problem can be viewed as an optimization problem, where a similarity measure has to be maximized. The appropriate definition of this similarity measure is indeed an important issue for the registration of diffusion tensor images. In this paper, we compile the different approaches for the registration of diffusion tensor images that have been published. Special attention is paid to the aforementioned topics: how to preserve the coherence between the tensor and the underlying tissue structure, and how to measure the similarity between two diffusion tensors. Methods to evaluate results are also reviewed, since a reliable validation leads to more conclusive results, specially in the comparison of different techniques. Most challenging issues for diffusion tensor images registration are underlined, and open research lines about this topic are pointed out.

## 1 Introduction

Diffusion Tensor Imaging (DTI) [8] has become a powerful imaging technique, playing an important role in neurological and neurosurgical applications, since it allows to identify and visualize fiber tracts and, thus, to analyze the connectivity inside the brain white matter. The Diffusion Tensor (DT) represents the local diffu-

Emma Muñoz-Moreno, Rubén Cárdenes-Almeida and and Marcos Martin-Fernandez
Laboratory of Image Processing, University of Valladolid, Spain, e-mail: emunmor@lpi.tel.uva.es, ruben@lpi.tel.uva.es, marcma@tel.uva.es

sion anisotropy of water molecules. The diffusion of water molecules in the brain is constrained by the myelin coat of the axons that constitute the nerve fibers, and therefore, there is high anisotropy inside the fiber tracts, where the diffusion occurs mainly in the direction along the fiber. Thus, it is possible to identify anatomical structures that cannot be visualized by means of other image modalities. One of the most extended applications of DTI is the tractography or fiber tracking, that is the computation of the nerve fiber trajectories inside the brain [30]. The algorithms that compute such trajectories are mainly based on the information given by the principal eigenvector of the tensor, that is related to the principal diffusion direction.

Image registration consists in the assessment of a geometric transformation that maps one image into another, so that objects or structures in both images are aligned [21]. This technique has a great importance in medical image processing, since it allows, for instance, diagnosis assistance, patient follow-up or surgical planning applications. The registration of DT-MRI (Diffusion Tensor Magnetic Resonance Image) is useful for the diagnosis and following-up of diseases such as epilepsy, multiple sclerosis or tumors [42]. Moreover, brain connectivity atlases can be built by spatial normalization of a set of images, what is done by applying registration algorithms [46]. Atlases based on DT-MRI can be used in the study of pathologies such as schizophrenia or Alzheimer diseases.

A large number of works dealing with the registration of traditional image modalities, whose values are scalar magnitudes, have been reported. For instance, a review of image registration techniques can be found in [54]. Nevertheless, literature about DTI registration is not yet so extensive, and there is not an exhaustive review of existing techniques. The most remarkable problem definition appears in [23], where the problem description is complemented with a brief review of the state of the art, and in [20], where a rigorous statement for the DTI registration problem is introduced. The main difference between scalar and tensor registration is that, in the latter case, not only a spatial transformation must be computed, but also the tensor data in each voxel must be transformed, to remain consistent with the underlying fiber structures. This is the main reason, but not the only, because of which the scalar registration algorithms cannot be directly applied to tensor images.

Another challenging issue in the DTI registration problem, is the choice of the appropriate similarity or error measure between tensor data. The registration algorithms are usually driven by a measure that must be optimized. Its definition has an influence on the registration result, that is not negligible. Hence, several criteria have been proposed in the scalar case in order to achieve the best registration result. Some of these measures have been extended to the tensor case and other new ones have also been proposed to manage the special features of DT. Moreover, usual similarity criteria for scalar data are also used for scalar magnitudes associated with DTI, such as anisotropy measures (see Apendix) or the tensor eigenvalues. They can be also computed for images related to the acquisition procedure such as the baseline T1-weighted or T2-weighted images.

Regarding to the quality evaluation, there is not a common method to assess the quality of a registration algorithm and to compare it with other proposals. In some cases only visual inspection is considered, whereas in others, quantitative error mea-

sures are defined. The quality assessment is an important issue, since it is necessary for achieving conclusive results in the comparison of the different techniques for DTI registration.

In this chapter, an extensive review of the state of the art of DTI registration is presented. The chapter is structured as follows: in the next section, the problem of DT-MRI registration is described, paying special attention to new challenges that arise from this kind of images. In Sect. 3, registration algorithms that have been applied to DT-MRI are compiled. Next, a review of the similarity measures defined for DTI is shown in Sect. 4, whereas Sect. 5 exposes different methods to preserve the coherence between tensors and the underlying structure. The evaluation of the registration results is treated in Sect. 6, where different techniques to validate the registration performance are described. Finally, the conclusions are summed up in Sect. 7.

## 2 Registration of DT-MRI

As previously defined, image registration is the procedure that finds the geometrical transformation that better matches two images, so that structures in both images become aligned [21]. Usually, the optimal transformation is searched within a previously defined set of allowed functions, such as rigid, affine or higher order. The two images implied in the registration procedure are named moving and fixed image. The transformation obtained by the registration algorithm is applied to the moving image, so that the structures in the registered image are supposed to be aligned with the structures in the fixed image.

Because of the special features of DT images, registration of this kind of images is an involved task. On the one hand, tensor provides more information than a simple scalar value, and, for this reason, it provides more data for finding the matching between images, mainly in areas that seem homogeneous in other modalities. On the other hand, this more complete information implies that more complex processing is required for registration of DT images. The advantages of exploiting the information provided by the DT have been proved in papers such as [4, 31, 33]. Likewise, the problems and peculiarities of DT registration have also been widely analyzed in the literature.

The two main topics that should be taken into account in DT-MRI registration are the coherence between the tensors and the transformed image structures and the computation of similarity or error measures to drive the algorithm. Other topics are related to the interpolation of the diffusion tensor field, the computational load and the evaluation of results. Below, we comment in detail these issues.

The DT orientation in a voxel is related to the diffusion direction, and therefore, to the fiber structure within the brain. If a geometrical transformation is applied to the image grid, the fiber structure is also modified. If DTs are directly arranged in their new locations, incoherence between their orientation and the fiber orientation arises. For this reason, each DT should be transformed in order to preserve the

coherence with the underlying image structure. It can be viewed in the synthetic example in Fig. 1, where tensors are represented by ellipses, whose axes length and directions are given respectively by the tensor eigenvalues and eigenvectors. The image in Fig. 1.a is rotated, and the tensors are translated to the new locations, so the image in Fig. 1.b is obtained. The tensor orientation does not coincide with the fiber orientation, since fiber has been rotated. Therefore, the same rotation should be applied to tensors in order to be aligned with the fiber, as it is shown in Fig. 1.c. Thus, the warping of DT is necessary to realign the diffusion direction with the fiber structure, and some strategies have been developed to tackle this problem. They assume that the size of a DT is related to the amount of diffusion per volume unit, and that for this reason it should remain constant, although the image size changes. Therefore, they search for a rotation matrix that should be estimated from the image transformation, to reorient the tensor without changing its size. Although in [23] it is explained that the assumption of only rotations may be not valid when fibers cross, it is considered reasonable, since it is said that effects of the fiber crossing are minor in real brain images. The methods that have been proposed for estimate the tensor warping are detailed in Sect. 5.

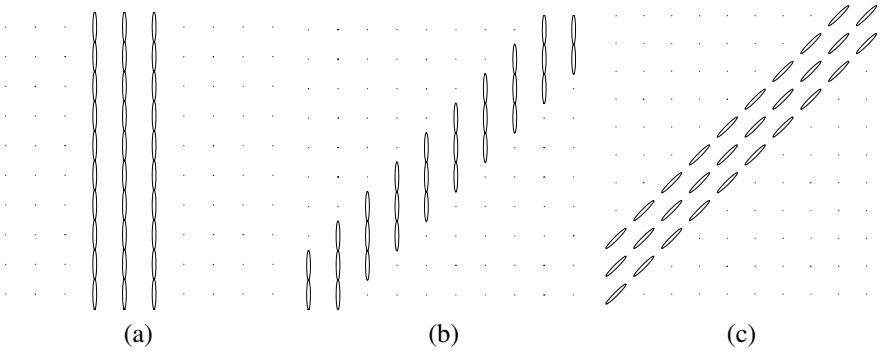



**Fig. 1** Synthetic example that shows the need for warping tensors after the transformation of the tensor field. a) Synthetic tensor field with a vertical fiber; b) rotation of (a) without tensor warping; c) rotation of (a) with tensor warping.

The second issue that must be considered in DT-MRI registration is related to the measure that drives the registration algorithm. Most registration algorithms are based in a measure of similarity between the moving and fixed images. This measure must be optimized along the registration procedure in order to achieve the best matching. Conventional scalar measures may not reflect all the tensor features, such as its orientation and shape. In order to make the most of the information provided by the tensor, specific DT similarity measures have been defined and used to drive the registration algorithms. In the literature, some scalar similarity measures have been extended to the tensor case, such as correlation index [40] or Euclidean distance based measures [4]. In other cases, new criteria based on DT features have

been considered [33, 34, 52, 53]. In Sect. 4, similarity measures that have been used for DT-MRI registration are thoroughly reviewed.

The two previous problems are the more tackled in works about DTI registration, and different approaches have been developed to correctly warp tensors or to define an appropriate cost function for the algorithms. Nevertheless, there are other important topics that are related to the registration task, but less attention has been paid to them. For instance, the tensor field interpolation method influences the registration result although it is usually not described when registration algorithms are proposed.

Usually, registration algorithms require the application of interpolation strategies. For instance, it is common the use of multiresolution schemes to obtain the geometrical transformation with lower computational burden. In these schemes, interpolation is necessary to evolve from the displacement field or the transformed image at a given level to the next higher resolution level. Also, interpolation is required when the computed transformation is applied to the moving image, since the resulting new location for a voxel does not necessarily correspond to a grid point in the fixed image. Moreover, some registration algorithms consist in an iterative scheme to optimize a similarity measure. In these cases, interpolation must be done at each iteration step. This shows the importance of interpolation inside a registration scheme. However, this topic has been left aside in the description of registration techniques for DTI. Only in few works the interpolation method applied to the tensor field is mentioned. This is the case of [31] that carries out a trilinear interpolation and [33] where Log-Euclidean metric is used to interpolate the tensor field. Nevertheless, specific works about DT interpolation have been done, which show that the specific tensor features should be taken into consideration to avoid undesired effects in the interpolation result. For instance, if Euclidean metrics are considered a swelling effect may appear, that makes the determinant of the mean tensor of a given set is bigger than the tensors of such set [5]. Thus, specific algorithms have been described for tensor field interpolation, that are usually based on non-Euclidean metrics [5, 11].

Regarding the computational load, it is obvious that tensor processing implies higher burden than scalar processing. For this reason, multiresolution schemes becomes even more important in this case. An alternative approach is the application of the registration algorithm only to areas of higher local structure and then, to interpolate the transformation in the rest of the image, as implemented in [40]. Anyway, the importance of the computation time always depends on the requirements of the application for which registration is used.

Finally, another open issue is how to evaluate the performance of a registration algorithm. The assessment of the quality of matching can be made both qualitatively and quantitatively. In the literature, several algorithms have been evaluated by visual inspection of the results [26, 40]. Regarding quantitative evaluation, the problem is closely related to the definition of similarity measures appropriate for DT, and in some papers, the same measures defined to drive the algorithm have been used to evaluate the registration result [4]. In most cases, the objective of the registration procedure is to find matching between structures, and therefore quality assessment

could require the definition of structural based measures. Thus, some authors compare the fiber structures computed from the fixed DT-MRI and from the transformed moving image in order to asses the validity of their algorithms. In other cases, voxel based measures that compare for instance the eigenvector directions, or the tensor component values have been described. A more detailed description of evaluation techniques is compiled in Sect. 6.

## 3 Review of DT-MRI registration Algorithms

Several of the well-known registration methods whose goodness have been widely proved with other image modalities have been adapted to deal with DT-MRI registration. In this section, these methods are classified according to the degree of freedom of the desired transformation: rigid, affine and elastic registration. We will focus on the algorithms themselves. The considerations about reorientation strategies and similarity measures, are detailed in further sections.

In some cases, the registration algorithm is applied to scalar images in order to compute the transformation, and this transformation is then applied to the tensor image. Such scalar images may be derived from the tensor data, such as anisotropy images (described in the Appendix), or they may be related to DT-MRI acquisition procedure, such as T1 or T2 weighted images or Diffusion Weighted Magnetic Resonance Images (DW-MRI), which are other kind of magnetic resonance images, that reflect the diffusion inside brain.

### *3.1 Rigid Registration*

The rigid registration allows to find the best rigid transformation (translation and rotation) that makes two images match. It is used to align images from different acquisitions of the same patient, or as a first registration step, to align the basic structures before applying more complex registration algorithms.

Since a rigid transformation can be expressed in a 3D space with six parameters (three for translation and three for the rigid rotation), the registration problem can be reduced to the resolution of a linear system of equations. Thus, in [25] a closed-form expression is given to compute the parameters of the rigid transformation that matches two DT-MRIs.

### *3.2 Affine Registration*

An affine transformation allows rotation, translation and scaling of the image, and therefore, for 3D images, is defined by twelve parameters. Just like rigid, affine reg-

istration can be reduced to solve a linear system of equations if the correspondence between a set of landmarks in both images is known. In this case a closed algebraic form for registration can be achieved as proved in [25]. If this correspondence between landmarks is not known, an iterative multiscale scheme can be applied to compute the coefficients of the affine transformation. These coefficients are those that minimize the difference between tensors in both images. In Sect. 4 we further describe how this difference is computed.

Additionally, some authors use affine registration due to its simplicity to better analyze other features of the registration algorithm. This is the case of [2, 3, 15, 16], where reorientation influence is analyzed or [33, 34] that are focused on the definition and performance analysis of similarity measures. The affine registration in [16], is carried out by means of an optimization method called gradient annealing, which consists in a combination of a local and a global optimization, and such implementation is used in [17] to compare different similarity measures.

As stated previously, in some cases, registration is applied not directly over the tensor field itself, but over some quantities related to the tensor. This is the case in [28], where an affine transformation is computed by means of a multichannel affine registration algorithm between DW-MRI. Then, the registered tensor field is calculated from the registered DW-MRIs.

In some cases the affine registration belongs to a more complex registration algorithm. Thus, in [18, 52, 53] a piece-wise affine registration algorithm for DT-MRI is proposed. Thus, the images are split into regions, and the affine transformation is computed for each region, whose coefficients minimize a given target function. In [52, 53] the target function is based on the diffusion profiles, whereas in [18], the objective function is based on the differences between anisotropy measures and eigenvectors in both images. It is necessary to impose continuity conditions in the interfaces between regions to ensure a smooth transformation. Moreover, a multiscale framework is defined and therefore the final transformation is not globally affine, but it allows higher degrees of deformation.

Furthermore, in [45], affine registration is carried out on the Fractional Anisotropy (FA) images, and the transformation is then refined, so higher order deformations are allowed. The refinement is based on features of the tensor at each voxel, that are computed by means of a bank of oriented Gabor filters. These features are arranged into an attribute vector, and the difference between such attribute vectors in voxels of both images is considered to obtain the best matching.

Moreover, affine registration is performed in some cases to align images before the application of higher order registration algorithms such as the ones compiled in the next section.

### *3.3 Elastic Registration*

We call elastic registration to those algorithms that search for curved transformations, that is, transformations that allow higher degree of freedom than global trans-

lation, rotation and scaling. Thus, some of the aforementioned registration algorithms, such as [18, 52, 53] could be also included in this section, since the global transformation is curved, although they are based on locally affine registration.

Elastic matching algorithms consider the image an elastic medium that is deformed by certain forces under some constraints. These constraints are represented by an energy term that should be optimized by means of the registration algorithm. Different approaches can be distinguished within this group of techniques: first, the registration can be viewed as an optimization problem, where a displacement field is searched, that minimizes the difference between the fixed and moving images. The optimum displacement field can also be found by means of a block matching algorithm, that searches for correspondences between templates in both images. In other cases, the registration is viewed as a variational problem, so that an optimum functional that makes both images to be aligned is searched from a space of continuous functionals. Such functionals can depend on every tensor components, or on other magnitudes, such as the eigenvectors or anisotropy measures. Another group of techniques are the multichannel schemes that have been applied to DT-MRI to elastically register the images. Finally, we mention the elastic algorithms that have been applied to images related to DT-MRI, instead of to the DT-MRI itself.

### Displacement Field-Based Registration Algorithms

The first approach to the DT-MRI elastic matching has been described in [4], where a multiresolution elastic matching algorithm is applied. The displacement at each voxel is computed by minimizing a given energy function, that is composed by a deformation and a similarity term. Since elastic matching has already been defined and broadly used for scalar images registration, higher efforts have been made in the definition of the similarity term [4, 24]. The tensor reorientation is included in the energy term in [24],

Another earlier approach to DT-MRI registration is the block matching algorithm described in [40]. It searches for correspondence between templates in both images. On behalf of better results and lower computation burden, the matching is only applied in highly structured areas, and then the displacement field is interpolated all over the image. The algorithm that is used to detect structured areas can be found in [38], and the displacement field is interpolated by an algorithm based on the Kriging estimator [37].

### Variational Frameworks

There are a number of works that focus the registration problem as a variational problem. An exhaustive description of the DTI registration from this point of view can be found in [20], where analytical expression of the gradient of the energy function is developed. Also in [14], a variational framework is considered to estimate the optimum transformation between two DT-MRIs. Moreover, this framework is

extensible to higher order tensors, which allows to better describe the diffusion, specially in areas where fibers are crossing. Likewise, [6] searches for the optimum transformation inside a variational framework, where a four-term energy function is defined to obtain a smooth transformation that correctly warps both the image and the tensors.

Moreover, in some cases the transformation is imposed to be diffeomorphic. Thus, the elastic registration approach in [12, 13] finds the optimum dipheomorfic transformation that matches two DT-MRIs by means of the resolution of a variational problem. A multiscale scheme is again used to achieve better matching in reduced computational time. In [13] the registration is carried out between the vector fields corresponding to the major eigenvector of the tensor, meanwhile in [12] all the components of the tensor are considered. It is shown that the tensor matching performance is better than the vector matching, which is better than the intensity matching. Therefore, it is shown than the more information taken into account to match the images, the better matching is achieved. Diffeomorphism is also searched in [51], that applies to DTI the diffeomorphic version of the demons algorithms proposed in [44].

### Multichannel Approaches

In several cases, the registration of DT-MRI has been viewed as a multichannel registration problem, where scalar channels out of the DT have been defined, for instance, DT trace, its components or some anisotropy images. In Fig. 2 some examples of these magnitudes are shown, namely, Fractional Anisotropy (FA), Mean Diffusivity (MD) and T2-weighted images, three tensor components and the three tensor eigenvalues. FA and MD are defined in Eqs. (17, 20) in the Appendix. A multichannel elastic registration based on radial basis function is proposed in [35, 36], where different channel configurations are defined. Specifically, in [35] combinations of the following three channels are considered: trace of the tensor, Relative Anisotropy (RA) index and skewness measure (see Eqs. (18, 25) in the Appendix for definition), whereas in [36] two other multichannel configurations are taken into account: the former consists of two channels, the trace and the RA; and the latter consists of six channels, each of them corresponding to one of the diffusion tensor components. Better results are achieved with the second approach, which exploits all the information that the tensor provides.

A multichannel approach is also introduced by [26], that implements a multichannel demons algorithm [43]. A multiresolution scheme is also defined in order to achieve better matching with lower computational times. Three channel configurations are compared: only the T2 channel, the T2 channel and the three tensor eigenvalues; and the T2 channel and the six tensor components. Opposite to the conclusions in other works, where the importance of including orientation information is highlighted, [26] argues that the use of the tensor components or the tensor eigenvalues leads to similar results, though the seven channel configuration is computationally more expensive. Multichannel demons algorithm is also used in [31]

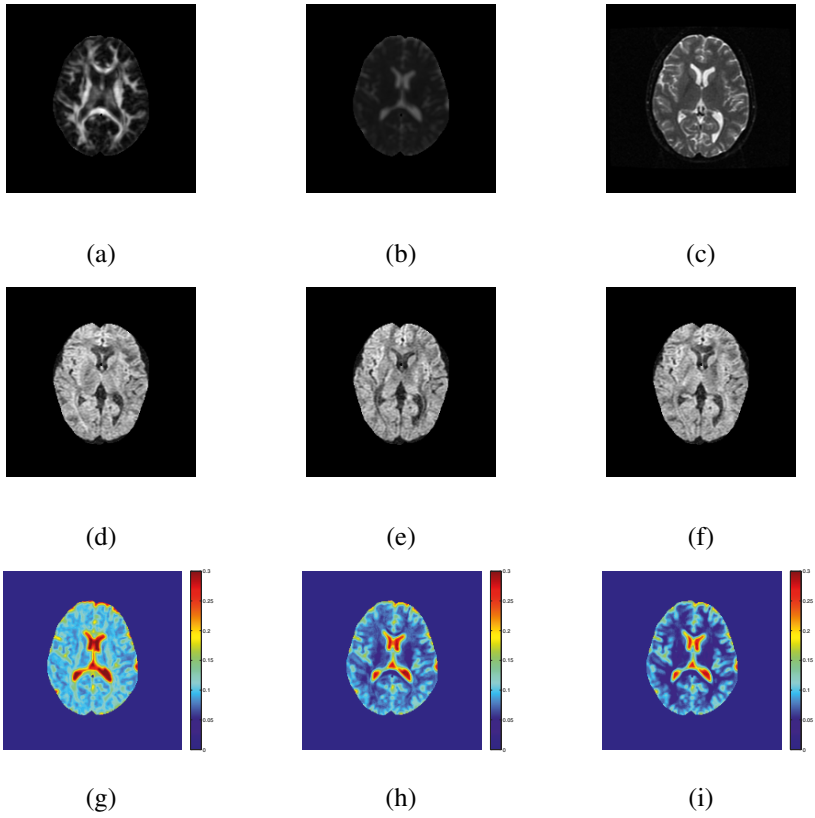



**Fig. 2** Scalar magnitudes associated with tensor data: (a) FA, (b) MD, (c) T2-weighted, (d), (e), (f) DWI in three different gradient directions, and (h), (i), (j) eigenvalues (from higher to lower) represented in a RGB color map.

for spatial normalization of DT-MRI. Among the six combinations of channels that are compared, the one that uses all the tensor components achieves the more reliable results.

## Elastic Registration of Scalar Images

As explained above, there are some works that compute the transformation from scalar images related to the DT-MRI, instead of directly from the tensor image. In the approach proposed by [50] the HAMMER (Hierarchical Attribute Matching Mechanism for Elastic Registration) [41] algorithm is applied to the T1-weighted image, and then the transformation obtained by the registration procedure is applied to the DT images. This chapter focuses more on the reorientation of DT after the

transformation, instead of on the registration algorithm itself, and for this reason a specific registration method has not been developed.

## 4 Similarity Measures

One of the most important challenges in DT-MRI registration is the definition of an appropriate measure to quantify similarity between DTs. Since registration algorithms have been applied to DTI, similarity measures for this kind of data have become necessary. In some cases, cost functions are defined from scalar magnitudes computed from the DT (anisotropy measures, trace...) or other images related to the DT-MRI such as T1 or T2-weighted images, see Fig. 2. Thus, in [50] T1-weighted images are used for the registration and in [27], registration is applied to the FA images. Moreover, measures based on T1-weighted images, and based on the trace and the Volume Ratio (defined in the Appendix, Eq. (19)) are compared with other criteria based on the whole DT in [4]. Usual criteria for the assessment of similarity between scalar images can be applied to these magnitudes, such as squared differences (SD) or correlation [4].

Scalar based similarity measures have also been used in multichannel algorithms: the demons registration defined in [26, 31, 51] are based on the SD of each of the image channels, whereas the multichannel affine registration described in [28] is based on mutual information [47] computed for each channel. Furthermore, a definition for multivariated mutual information can be found in [35, 36].

A different approach consists in the computation of measures regarding all the information provided by the tensor. In most cases, the extension of the SD to the tensor case, that is the Frobenius norm of the difference between two DT, is computed. This measure can be viewed as the Euclidean distance (ED) between two tensors. The formulation of the ED between two DTs, $\mathbf{A}$ and $\mathbf{B}$, whose components are $A_{i,j}$ and $B_{i,j}$, is given by:

$$ED(\mathbf{A},\mathbf{B}) = \sqrt{\sum_{i=1}^{3}\sum_{j=1}^{3}(A_{i,j}-B_{i,j})^2} = \sqrt{trace\,[(\mathbf{A}-\mathbf{B})^2]}. \tag{1}$$

The registration algorithms compiled in [4, 6, 12, 16, 24, 25, 40] are driven by a cost function based on this measure. In [39] a formal definition of this metric for tensors can be found, whereas in [24] the difference is computed between the reoriented and the fixed tensor.

A set of pointwise similarity measures are defined in [1]. In addition to the ED between tensors, they propose a comparison term based on the scalar product of tensors:

$$\mathbf{A}\cdot\mathbf{B} = \sum_{i=1}^{3}\sum_{j=1}^{3} A_{i,j}B_{i,j}, \tag{2}$$

and other measure, that has been previously defined in [32] to quantify the similarity between the diffusion in two media:

$$\mathbf{A} : \mathbf{B} = \sum_{i=1}^{3} \sum_{j=1}^{3} \lambda_i^A \lambda_j^B \left( \mathbf{e}_i^A \cdot \mathbf{e}_j^B \right)^2, \tag{3}$$

where $\lambda_i^A$ and $\lambda_i^B$ stands for the $i$-th eigenvalue of the tensors $\mathbf{A}$ and $\mathbf{B}$ respectively, and $\mathbf{e}_i^A$ and $\mathbf{e}_i^B$ are the corresponding eigenvectors.

Among these three measures proposed in [1], the ED was found to be the one that produces a better matching, and therefore it was used in further works such as [4]. Moreover, the same three measures in Eqs. (1, 2, 3) are applied to the deviatoric (see Appendix, Eq. (21) for definition) that is a measure of the anisotropic part of the DT. Once again, the ED produces the best results, and for this reason is used in further works.

Also the angle between the major eigenvector of tensors is defined as similarity measure in [1, 4]. This angle is weighted by some anisotropy measure, since the more isotropic the diffusion, the less significant the direction difference is. Thus, this measure, that we will denote as orientation difference (OD), is given by:

$$OD(\mathbf{A}, \mathbf{B}) = \sqrt{\nu(\mathbf{A})\nu(\mathbf{B})} \mathbf{e}_1^A \cdot \mathbf{e}_1^B, \tag{4}$$

where $\nu(\mathbf{A})$ stands for a measure of the anisotropy, such as VR [4], RA [17] or lattice anisotropy [1] (see Eq. (26) in the Appendix). The difference between eigenvectors is also used in [13] since their algorithm is applied directly over the vector field. Likewise, the registration algorithm in [18] is also based on the difference between eigenvectors weighted by the anisotropy, although in this case the cost function also includes the square differences between FA images. The orientation difference is also considered in [19], where the similarity measure consists of two terms: the former takes into account the tensor eigenvalues and the latter is a region-based measure for orientation matching.

In the Gabor-features based algorithm proposed in [45] the similarity measure is based on the computation of the distance between the attribute vectors. These vectors are composed by features detected by means of a bank of Gabor filters. They also contains the value of the FA and the angle that forms the major eigenvector with the dominant orientation in a neighborhood of the tensor. This dominant orientation is computed by Principal Component Analysis of the main eigenvector of every tensors in the neighborhood. The registration minimizes the square difference between such vectors, that is defined by two terms: the difference between angles, and the Euclidean distance between the vector composed by the values obtained by the Gabor filters.

Furthermore, the correlation coefficient has also been extended to diffusion tensor. A rigorous definition can be found in [40]. Since this is a costly measure, the block matching based on this measure is only applied to a set of highly structured areas of the image.

On the other hand, there are a set of measures that take into account most specifically the nature of the data in their definition. For instance, two similarity measures based on the comparison of diffusion profiles, that are assumed to be Gaussian, are proposed in [53]. These metrics, that we denote as diffusion profile based distance (DPBD), are defined as:

$$DPBD_1(\mathbf{A},\mathbf{B}) = \sqrt{\frac{8\pi}{15}\left(||\mathbf{A}-\mathbf{B}||^2 + \frac{1}{2}trace^2(\mathbf{A}-\mathbf{B})\right)} \tag{5}$$

$$DPBD_2(\mathbf{A},\mathbf{B}) = \sqrt{\frac{8\pi}{15}\left(||\mathbf{A}-\mathbf{B}||^2 - \frac{1}{3}trace^2(\mathbf{A}-\mathbf{B})\right)}, \tag{6}$$

where $||\cdot||$ stands for the Frobenius norm of the matrices. The latter measure is more focused on the comparison of the anisotropic components of the tensor. Other approach based on the Gaussian model of the diffusion can be found in [14], where the tensor is modeled as the covariance matrix of the Gaussian probability density function. Thus, the symmetrized Kullback-Leibler divergence, also known as $J$-divergence is minimized by the registration algorithm. It is argued that this measure can be extended to higher order diffusion models, besides the conventional $3\times 3$ tensors. It is computed as:

$$sKL(\mathbf{A},\mathbf{B}) = \frac{1}{4}\left(trace(\mathbf{A}\cdot\mathbf{B}^{-1}) + trace(\mathbf{B}\cdot\mathbf{A}^{-1}) - 6\right). \tag{7}$$

Also a specific measure is defined in [33, 34], where the physical meaning of the DT is taken into account in the definition of the Mode Based Similarity (MBS). It classifies tensors in three types according to its shape, described by three geometric coefficients [48]: the lineal $c_l$, planal $c_p$ and spherical $c_s$ coefficients (see Eqs. (22, 23, 24) in the Appendix). Then, the main features for each type of diffusion are compared. Thus three similarity terms are defined, the linear similarity:

$$S_l(\mathbf{A},\mathbf{B}) = |\mathbf{e}_1^A \cdot \mathbf{e}_1^B|, \tag{8}$$

the planar similarity:

$$S_p(\mathbf{A},\mathbf{B}) = |\mathbf{e}_3^A \cdot \mathbf{e}_3^B|, \tag{9}$$

and the spherical similarity:

$$S_s = \frac{1}{2}\left(2 - \frac{|trace(\mathbf{A}) - trace(\mathbf{B})|}{\max(trace(\mathbf{A}), trace\mathbf{B}))} - \frac{|b_0^A - b_0^B|}{\max(b_0^A, b_0^B)}\right). \tag{10}$$

$\mathbf{e}_1^A$ and $\mathbf{e}_1^B$ stands for the major eigenvector of the tensors $\mathbf{A}$ and $\mathbf{B}$ respectively, and $\mathbf{e}_3^A$ and $\mathbf{e}_3^B$ are their minor eigenvectors. Moreover, the spherical comparison term includes comparison between MRIs without diffusion that are acquired to obtain the DTI. These images are known as baselines, and are denoted as $b_0$. These three terms are combined to define the MBS measure, that is given by:

$$MBS(\mathbf{A},\mathbf{B}) = c_l^A c_l^B S_l + c_p^A c_p^B S_p + c_s^A c_s^B S_s \tag{11}$$

A comparison of this measure with other similarity measures can be found in [33].

Finally, since DTI are often used to identify fiber tracts, in [29] the registration algorithm is based on comparison between such structures. It defines two parameters to describe the fiber tracts, namely curvarture and torsion. The mean square difference of these parameters between two fiber tracts is minimized to find the best matching between DTIs.

## 5 Diffusion Tensor Reorientation

As mentioned, not only the image grid should be transformed by means of the registration procedure. The tensors must also be transformed in order to preserve the coherence with the underlying fiber structure. It has been assumed that the only transformation that should be applied to tensors is rotation, and therefore, algorithms that estimates a rotation matrix from the image transformation have been proposed. The only work that applies a shear term besides the rotation matrix is the approach in [6], although it assumes that tensor shear should be small. In this work, the warping matrix is included in the energy term that is optimized by the registration. Thus, the rotation angle and shear parameters are computing by the optimization procedure. A compatibility term is defined to ensure the relation between image transformation and tensor warping.

In the other works, the algorithms proposed to preserve the coherence between tensors and structures aim to compute the appropriate rotation matrix $\mathbf{R}$. Thus, the tensor value in a given voxel is computed as $\mathbf{R}^T\mathbf{D}\mathbf{R}$, where $\mathbf{D}$ is the original DT. Depending on the freedom degree of the transformation, the same rotation matrix can be applied to every tensors in the image or a different matrix is required to describe the transformation at each voxel. The first option is valid for rigid registration, such as the rigid algorithm in [25]. In this case, the global image transformation can be split into a translation and a rotation part, and therefore, the same rotation applied to the image is applied to tensors. For more complex transformations, some strategies to compute the rotation matrix have been proposed. The first approach to this topic is found in [2, 3], where three methods are defined: small strain (SS), finite strain (FS) and preservation of the principal direction (PPD).

The SS strategy is defined for very small deformations or rigid models. It consists in the decomposition of the gradient of the deformation field into a rotation and a pure deformation component [2]. It is supposed that the gradient of this deformation field can be decomposed into a symmetric and a skew-symmetric matrices: $\mathbf{J} = \mathbf{E} + \Omega$, where the first term $\mathbf{E}$ is the symmetric part and it corresponds to the deformation and the second one, the skew-symmetric matrix $\Omega$ is related to the rotation. Therefore, the rotation matrix $\mathbf{R}$ is computed from $\Omega$. The same rotation matrix will be applied to every tensor in the image. Since this strategy is only valid

in case of rigid or globally affine transformations, it has not been further applied in the literature.

The second strategy in [2, 3], FS, was first described for affin transformations, that are given by a matrix $\mathbf{F}$, that can be decomposed into a strain and a pure rotation component $\mathbf{F} = \mathbf{UR}$, according to the Polar Decomposition Theorem. Thus, the rotation matrix can be computed as:

$$\mathbf{R} = (\mathbf{F}\mathbf{F}^T)^{-1/2}\mathbf{F} \tag{12}$$

If the transformation is not affine, the FS strategy works on the assumption that it can be considered locally affine. In this case, the jacobian matrix $\mathbf{J}$ of the transformation is computed, and the rotation matrix is computed by substituting in Eq. (12) $\mathbf{F}$ by $\mathbf{J}$. The same strategy based on the Polar Decomposition Theorem is also described in [40], and it has been widely used in the literature, for instance in [16, 26, 36, 40, 51, 53].

However, the tensor orientation may be changed by the shear effect that is described in the deformation term $\mathbf{U}$. This effect is discarded by the FS strategy, but its influence in the tensor field is not negligible, as is shown in Fig. 3. The original tensor fields in Fig. 3.a and Fig. 3.c are sheared by the same transformation. However, the tensors in Fig. 3.b remain aligned with the fiber structure, whereas tensors in Fig. 3.d require to be rotated in order to be aligned with the fiber, as in Fig. 3.e. For this reason in [3] a third approach, named PPD, is proposed. Let notice that the shear effect depends on the original tensor orientation, and for thus, the PPD reorientation strategy is based on the major eigenvector in order to ensure that it is correctly reoriented taking into account both the pure rotation and the rotational effect of shear. Thus, the proposed method consists of the next steps:

1. The transformation matrix $\mathbf{F}$, if it is affine, or the Jacobian matrix of the transformation $\mathbf{J}$, if it is not affine, is applied to the major eigenvector $\mathbf{e_1}$, so the new direction is obtained. Then, the rotation matrix $\mathbf{R}_1$ is computed as the matrix that converts the original eigenvector $\mathbf{e_1}$ into the unitary vector in the new direction:

$$\mathbf{e_1'} = \frac{\mathbf{Fe_1}}{||\mathbf{Fe_1}||}$$

.

2. Since the second eigenvector must be orthogonal to the first, the projection of the transformed second eigenvector $\mathbf{Fe_2}$ or $\mathbf{Je_2}$ on the plane defined by its normal vector $\mathbf{e_1'}$ is computed. The projection vector is then normalized, so the reoriented second eigenvector $\mathbf{e_2'}$ is obtained. A rotation matrix $\mathbf{R}_2$ is computed, that converts the original eigenvector $\mathbf{e_2}$ into the new $\mathbf{e_2'}$.
3. The rotation matrix $\mathbf{R}$, that is applied to the tensor, is computed as $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2$

Besides the description of this method in [3], another formulation of it can be found in [28]. The analysis of this method and its comparison with other reorientation strategies is detailed in [3, 16], where it is proved that better results are obtained by means of PPD, specially in the case of high order deformations, where

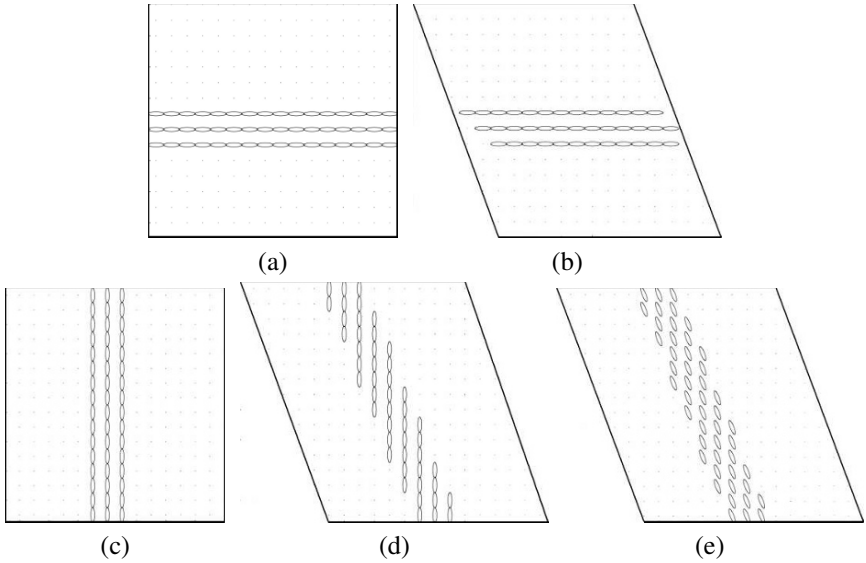



**Fig. 3** Synthetic example showing the rotation effect of shearing: a) Horizontal fiber; b) Shear of the synthetic field in (a); c) Vertical fiber; d) Shear of the synthetic field in (c) without tensor reorientation; e) Shear of the synthetic field in (c) with tensor reorientation.

shear effect is higher. However, it should be taken into account that this algorithm is computationally more expensive than the FS strategy, and for this reason the FS algorithm is preferred in some cases. Nevertheless, PPD has been chosen by many authors, such as [14, 17, 27, 31, 33, 52].

In the literature, most efforts have been made in the analysis of registration algorithms or the definition of new similarity measures, meanwhile research about new reorientation techniques has been left aside and the previous reorientation strategies FS and PPD have been applied. In the earlier approaches, the rotation matrix is computed once the final image transformation is assessed. However, it have been proved that better results can be achieved if reorientation is explicitly included in the cost function [24, 51, 52]. Furthermore, in [6] the reorientation and shear parameters are computed together with the transformation parameters by the optimization algorithm.

Moreover, the work in [49, 50] analyzes the drawbacks of the FS and PPD methods and it proposes a new method to compute the rotation matrix, based on statistics of the fiber direction and procrustean estimation. It considers that the assumption made for the definition of the PPD strategy is not valid in presence of noise. Due to acquisition noise, the diffusion tensor could not be exactly aligned with the fiber structure. For this reason, it computes the fiber orientation taking into account the mean values of the eigenvectors in a neighborhood of the voxel. The neighborhood assessment is based on the average of tensors inside it and it is computed by an iterative algorithm. The major eigenvectors of tensors in the neighborhood are con-

sidered as random samples of the fiber direction, and a matrix $\mathbf{B}$ is composed with such samples. Another matrix $\mathbf{A}$ is built by the eigenvectors obtained when the displacement field is applied, so that they can be viewed as random samples of the transformed fiber. The rotation matrix will be the one that minimizes $||\mathbf{A} - \mathbf{R} \cdot \mathbf{B}||^2$, and it can be computed by means of a Procrustean estimation as $\mathbf{R} = \mathbf{V} \cdot \mathbf{W}^T$, where $\mathbf{V}$ and $\mathbf{W}$ are the matrices resulting from the singular value decomposition of $\mathbf{A} \cdot \mathbf{B}^T = \mathbf{V} \cdot \Sigma \cdot \mathbf{W}^T$. Moreover, the random samples could be weighted in order to give more importance to those closer to the voxel.

## 6 Performance Evaluation

The evaluation of the performance of a registration algorithm is not a trivial task. The computation of the difference between images may not be a valid measure, since the best matching of structures is not necessarily the best intensity matching. For this reason, indexes based on structural information have been defined to assess the performance of scalar registration algorithms. When evaluating the registration of DT-MRI, not only this question should be taken into account, but also the aspect related with the comparison between tensors. Thus, several methods to analyze the performance of the registration algorithms have been described in the literature.

Although the visual inspection [26] and the difference between images [4, 12, 16, 45] have been used to establish the goodness of the registration, other more objective validation techniques have been also proposed. In some cases, the error measures are based on the similarity measures defined to drive the registration algorithm. This is the case in [4] where a set of similarity criteria are defined and then their average value in the image is considered as a performance measure. Also in [6] the same measure defined to assess the similarity between the target and fixed images is considered to evaluate the error. Likewise, the angle between the major eigenvectors has been used as similarity criterion and its average value has also been used as a quality measure, specially to compare the reorientation strategies performance [3, 12, 13, 16]. Since orientation matching is more interesting in areas of higher anisotropy values, which contain fiber tracts, the angle between eigenvectors is usually weighted by some anisotropy measures [16, 52]. Another similar option is to compute the angle only in the white matter, that is, the region where anisotropy measures are higher than a given threshold, as proposed in [12] .

Also quality measures are computed over scalar images related to the DT-MRI. For instance, in [3], the average difference is computed between T1-weigthed images, the trace of the tensor or the anisotropy images to give a measure of the registration error, and in [34] the normalized mutual information is computed for both T2-weighted images and FA images. The difference between the FA images also appears in [12].

As mentioned previously, registration allows for the construction of anatomical atlases. They are built by spatial normalization of a set of brain images acquired from a certain group of control subject. In this case, the registration algorithm is

applied to the set of images, and its performance can be evaluated by means of statistics computed in the normalized set. For instance, in [35], the variance of the data in each voxel is computed for each scalar channel, whereas, in [36] the variance of the trace and the anisotropy images in the normalized set is computed. Comparison between the mean value of the FA in the white matter region is also used to evaluate performance in [31]. Notice that, since these scalar magnitudes do not provide information about the orientation and shape of the diffusion, they do not allow to evaluate orientation matching.

The variability of tensors in a set of normalized images is also computed in [27] by measuring dispersion about the mean and the median. Thus, definitions for the mean and median tensors are provided, that are based on the Fréchet definitions [22]. Then the normalized standard deviation is defined as the dispersion about the mean, normalized by the norm of the mean tensor. Moreover, the dispersion about the median is also defined, as well as its normalized form, that in this case is obtained normalizing by the median tensor. The normalized standard deviation is also used in [53] to quantify the variability of tensors in a set of images.

Moreover, in [27], the quality of the registration is also analyzed by means of scattergrams of the FA and the dyadic coherence. This last measure was formulated in [9], and it is based on the dyadic tensor, that is formed by the components of the principal eigenvector in a voxel. Thus, if a set of $N$ subjects is considered, the dyadic tensor $\langle \mathbf{e}_i \mathbf{e}_i^T \rangle$ is defined as:

$$\langle \mathbf{e}_i \mathbf{e}_i^T \rangle = \frac{1}{N} \sum_{j=1}^{N} \mathbf{e}_i^j \mathbf{e}_i^{jT}, \tag{13}$$

where $\mathbf{e}_i^j$ stands for the $i$-th component of the major eigenvector of the DT of the $j$ subject. The dyadic coherence $\kappa$ is defined in function of the three eigenvalues $\beta_1, \beta_2$ and $\beta_3$ of the dyadic tensor:

$$\kappa = 1 - \sqrt{\frac{\beta_2 + \beta_3}{2\beta_1}} \tag{14}$$

Therefore, it is a method to quantify the alignment of the principal eigenvectors among a set of DT-MRI. It was afterwards used by [14, 53]. Other quality indexes based on the measures proposed in [9] have been defined to evaluate registration algorithms: the overlap of eigenvalue-eigenvector pairs [14, 31, 52]; and the mean diffusion index [31]. The former measure is computed between pairs of images, where the white matter has been segmented. Let be $\mathscr{W}$ the set of voxels that belong to the white matter region, and $W$ the number of elements of such set, the overlap is defined as:

$$OVL = \frac{1}{W} \sum_{x \in \mathscr{W}} \frac{\sum_i^3 \lambda_i(\mathbf{x}) \lambda_i^R(\mathbf{x}) (\mathbf{e}_i(\mathbf{x}) \cdot \mathbf{e}_i^R(\mathbf{x}))^2}{\sum_i^3 \lambda_i(\mathbf{x}) \lambda_i^R(\mathbf{x})}, \tag{15}$$

where $\lambda_i(\mathbf{x})$ and $\mathbf{e}_i(\mathbf{x})$ are the $i$-th eigenvalue and eigenvector of the tensor in the $\mathbf{x}$ voxel of the fixed image, and $\lambda_i^k(\mathbf{x})$ and $\mathbf{e}_i^k(\mathbf{x})$ are the $i$-th eigenvalue and eigenvector of the tensor in in the $\mathbf{x}$ voxel of the registered images.

The mean dispersion index is also computed in the voxels that are segmented as white matter. It is based on the dyadic tensor and is defined as:

$$MDI = \frac{1}{W} \sum_{x \in \mathscr{W}} \sqrt{\frac{\beta_2(\mathbf{x}) + \beta_3(\mathbf{x})}{2\beta_1(\mathbf{x})}} \tag{16}$$

On the other hand, one of the most common applications of DT-MRI is fiber tracking, that estimates the fiber structures in the brain, following the diffusion direction given by the tensor. For this reason, a way to evaluate the registration procedure is to compare the fiber tracts computed from the fixed and registered DT-MRI. Thus, [31] computes the mean dissimilarity between fiber tracts in the set of normalized images. The dissimilarity is based on two measures: the mean square error between tracts, that is the mean distance between fiber tracts normalized by the tract length; and the endpoint divergence, that computes the difference between the endpoints of two tracts computed from the same seed point. The distance between fiber tracts is also considered in [53], where the mean of the closest distance for every fiber in two bundles is computed. As well, the Euclidean distance between fibers is computed in [18], where the registration algorithm is focused on obtaining good results for fiber tracking applications.

Furthermore, synthetic data or transformations have also been used to better understand and analyze the performance of the registration algorithms [3, 25, 28, 40, 45]. For instance, in [25] a synthetic transformation is applied to the data, and then the ability to recover this transformation in presence of noise is studied. Synthetic transformations are also applied in [35, 36], where the recovered parameters are compared to the known parameters.

## 7 Discussion and Conclusions

Although the registration of DT-MRI is a relatively new problem, several approaches to deal with it can be found in the literature. In general, these approaches consist in the adaptation of existing registration algorithms for other image modalities to the DTI data. When adapting these algorithms, the special features of DT-MRI must be borne in mind especially regarding two questions: the importance of the orientation information that provides the tensor, that should remain coherent with the underlying fiber structure; and the definition of similarity measures that exploit the information given by the tensor.

Although the warping of the tensors after the image transformation has been accepted as a necessary task, few research has been done to find the best method to estimate the transformation. Rotation has been assumed to be the only allowed transformation, and most authors have adopted the PPD and the FS strategies proposed

by [3]. Only in [49, 50] the drawbacks of such algorithms have been analyzed and a new approach to the problem is proposed. This new technique seems to be more rigorous because it takes into account the noise that appears in real DT-MRI, but it has higher computational requirements than the PPD strategy. In addition, the only approach that applies shear to the tensor data is [6], which supposes that tensors can be both rotated and sheared. However, undesired effects can occur if tensors are sheared, for instance, isotropic tensors become anisotropic due to the shear effects, what implies that gray matter regions can be transformed into white matter. Anyway, since only small shears are allowed, this problem may be negligible. Nevertheless, to the best of our knowledge, there are no conclusive comparison studies between the previous methods, and therefore it is not possible to conclude what is the best approach to warp tensors. Moreover, it should be taken into account that the warping is defined from the transformation that was estimated by the registration algorithm, and therefore the orientation matching depends on the quality of such estimation. It is supposed that, if registration gives more reliable transformation, more reliable warping would be computed. Moreover, when the Jacobian matrix of the transformation is used to assess the warping, the numerical method used to compute the gradient of the displacement field also influences in the tensor matching. Due to the influence of these factors, the evaluation of the reorientation strategies for real data is not a simple issue.

On the other hand, the influence in the final matching of the similarity measure defined to drive the registration algorithm has been proved. Most works conclude that the more information is taken into account, the better matching is achieved. Thus, the inclusion of orientation information, such as the major eigenvectors, improves the results obtained when using rotational invariant characteristics. A set of similarity measures based on all the tensor components have been proposed. Some of them are based only on mathematical considerations, such as the Euclidean distance between tensors, and others exploit features related to the data acquisition or the physical interpretation of the data, such as diffusion profiles based metrics or measures based on the diffusion shape. There are some comparisons between purely mathematical measures and DT-specific measures that point out the advantages of the use of DT-specific measures. However, more work should be done to get more conclusive results. Moreover, since more rigorous frameworks for tensor calculus have been developed, similarity measures that takes into account these computational frameworks could lead to better matching results. This also should be borne in mind when interpolation of tensor fields is required by the registration algorithm. It has been proved that the consideration of Euclidean metrics for the DT calculus is not correct, since it can produce effects of swelling when interpolation is carried out. Therefore, the advances in interpolation of diffusion tensor fields should be included in the implementation of registration algorithms.

Finally, an important topic, especially to compare different registration approaches, is the evaluation of the results. In some cases, evaluation is based on the difference of voxel values, what does not necessary reflects the quality of the structural matching. Also structural measures, such as mutual information, have been used but only applied between scalar images related to the tensor, and therefore

they do not indicate the goodness of the tensor matching. In some cases, the orientation matching is directly evaluated by computing the angle between eigenvectors in the same location in both images. However, pointwise comparison loose the information about the quality of the structural matching. More specific ways to evaluate the performance are based on the fiber tracts computed from the DT-MRI. In such a way, both the structural and orientation matching would be taken into account in the results evaluation. Nevertheless, the need for fiber tracts computation introduces a new factor in the problem, and the tractography algorithm performance has a non negligible influence in the quality measure. For this reason, more work should be done in the definition of quality measures that comprise both the underlying structure and the orientation matching information. If a standard quality evaluation protocol would be available, more conclusive comparisons between different algorithms could be performed.

**Acknowledgements** The authors acknowledge the Comisiòn Interministerial de Ciencia y Tecnología for research grant TEC 2007–67073/TCM. The first author is funded by the Junta de Castilla y Leòn and the European Social Fund (ESF).

## Appendix. Scalar Measures of the Diffusion

Several scalar measures can be computed to describe the diffusion from the tensor data. In this appendix, a review of such measures is presented. Some of them describe the anisotropy of the diffusion, and other are more related to the amount of diffusion or its shape. Moreover, a structural measure of the diffusion in a neighborhood is defined.

### Anisotropy Measures

Let be $\mathbf{D}$ a diffusion tensor, whose ordered eigenvalues, from highest to lowest, are $\lambda_1, \lambda_2$ and $\lambda_3$, and whose eigenvectors are $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$. The Fractional Anisotropy (FA) is defined as [10]:

$$FA(\mathbf{D}) = \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1-\lambda_2)^2+(\lambda_1-\lambda_3)^2+(\lambda_2-\lambda_3)^2}}{\sqrt{\lambda_1^2+\lambda_2^2+\lambda_3^2}} \tag{17}$$

The Relative Anisotropy (RA) is given by [10]:

$$RA(\mathbf{D}) = \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1-\lambda_2)^2+(\lambda_1-\lambda_3)^2+(\lambda_2-\lambda_3)^2}}{\lambda_1^2+\lambda_2^2+\lambda_3^2} \tag{18}$$

Both measures take values between 0, for isotropic diffusion, and 1, when diffusion occurs in only one direction.

Moreover, the Volume Ratio is other anisotropy measure defined as [4]:

$$VR(\mathbf{D}) = 1 - \frac{27 \cdot \lambda_1 \cdot \lambda_2 \cdot \lambda_3}{(trace(\mathbf{D})^3} \tag{19}$$

Measures of the Amount of Diffusion

The Mean Diffusivity (MD) provides an idea of the amount of diffusivity [10] and is defined as:

$$MD(\mathbf{D}) = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \tag{20}$$

The deviatoric of a tensor $dev(\mathbf{D})$ quantifies the anisotropic part of a tensor, and it is defined as:

$$dev(\mathbf{D}) = \mathbf{D} - \frac{trace(\mathbf{D})}{3} \tag{21}$$

Diffusion Shape Measures

The shape of the diffusion can be described by the linear $c_l$, planar $c_p$ and spherical $c_s$ coefficients defined in [48], whose values are comprised between 0 and 1. They are computed by:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{22}$$

$$c_p = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{23}$$

$$c_s = \frac{\lambda_3}{\lambda_1} \tag{24}$$

In addition, the skewness defined in [7] allows to distinguish between prolate ($\lambda_1 >> \lambda_2, \lambda_3$) and oblate ($\lambda_1, \lambda_2 >> \lambda_3$) tensors. It is defined as:

$$Sk(\mathbf{D}) = \frac{(\lambda_1 - MD(\mathbf{D}))^3 + (\lambda_2 - MD(\mathbf{D}))^3 + (\lambda_3 - MD(\mathbf{D}))^3}{3} \tag{25}$$

Structural Measure

The lattice index is a structural measure that is computed at each voxel $\mathbf{x}$ in the image taking into account the tensors in a neightborhood $\mathcal{N}(\mathbf{x})$. It is defined as:

$$LI(\mathbf{D}) = \sum_k a_k \left( \frac{3}{8} \frac{\sqrt{dev(\mathbf{D}) \cdot dev(\mathbf{D_k})}}{\sqrt{\mathbf{D} \cdot \mathbf{D_k}}} + \frac{3}{4} \frac{dev(\mathbf{D}) \cdot dev(\mathbf{D_k})}{||\mathbf{D}|| \cdot ||\mathbf{D_k}||} \right), \tag{26}$$

where $\mathbf{D_k}$ are the tensor at the voxel $\mathbf{x}_k \in \mathcal{N}(\mathbf{x})$ and $a_k$ is a spatial mask whose coefficient sum is equal to one. The "$\cdot$" stands for the scalar product between tensors.

## References


1. Alexander, D., Gee, J., Bajcsy, R.: Similarity measures for matching diffusion tensor images. In: T.P. Pridmore, D. Elliman (eds.) British Machine Vision Conference, pp. 93–102. Nottingham, UK (1999)
2. Alexander, D., Pierpaoli, C., Basser, P., Gee, J.: Techniques for spatial normalisation of diffusion tensor images. In: SPIE Medical Imaging, vol. 3979, pp. 470–479. San Diego, CA, USA (2000)
3. Alexander, D., Pierpaoli, C., Basser, P., Gee, J.: Spatial transformations of diffusion tensor magnetic resonance images. IEEE Transactions on Medical Imaging **20**(11), 1131–1139 (2001)
4. Alexander, D.C., Gee, J.C.: Elastic matching of diffusion tensor images. Computer Vision and Image Understanding **77**(2), 233–250 (2000)
5. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance in Medicine **56**(2), 411–421 (2006)
6. Barbieri, S., Welk, M., Weickert, J.: Variational registration of tensor-valued images. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Workshops. Anchorage, Alaska, USA (2008)
7. Basser, P.: New histological and physiological stains derived from diffusion-tensor MR images. Annals of the New York Academy of Science **820**, 123–137 (1997)
8. Basser, P., Mattiello, J., Le Bihan, D.: MR diffusion tensor spectroscopy and imaging. Biophysical Journal **66**, 259–267 (1994)
9. Basser, P., Pajevic, S.: Statistical artifacts in diffusion tensor MRI (DT-MRI) caused by background noise. Magnetic Resonance in Medicine **44**, 41–50 (2000)
10. Basser, P., Pierpaoli, C.: Microstructural and physiological features of tissues elucidated by quantitative diffusion tensor MRI. Journal of Magnetic Resonance. Ser. B **111**, 209–219 (1996)
11. Batchelor, P.G., Moakher, M., Atkinson, D., Calamante, F., Connelly, A.: A rigorous framework for diffusion tensor calculus. Magnetic Resonance in Medicine **53**, 221–225 (2005)
12. Cao, Y., Miller, M.I., Mori, S., Winslow, R.L., Younes, L.: Diffeomorphic matching of diffusion tensor images. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), pp. 67–74. New York, NY, USA (2006)
13. Cao, Y., Miller, M.I., Winslow, R.L., Younes, L.: Large deformation diffeomorphic metric mapping of vector fields. IEEE Transactions on Medical Imaging **24**(9), 1216–1230 (2005)
14. Chiang, M.C., Leow, A.D., Klunder, A.D., Dutton, R.A., Barysheva, M., Rose, S.E., McMahon, K.L., de Zubicaray, G.I., Toga, A.W., Thompson, P.M.: Fluid registration of diffusion tensor images using information theory. IEEE Transactions on Medical Imaging **27**(4), 442–456 (2008)
15. Curran, K., Alexander, D.C.: Diffusion tensor orientation matching for image registration. In: J.M.F. Milan Sonka (ed.) SPIE Medical Imaging 2003: Image Processing, vol. 5032, pp. 149–156. San Diego, CA, USA (2003)
16. Curran, K., Alexander, D.C.: Orientation coherence optimisation in tensor image registration. In: Medical Image Understanding and Analysis Conferences (MIUA), pp. 259–272. London, UK (2004)
17. Curran, K., Alexander, D.C.: Comparison of similarity measures for driving diffusion tensor registration. In: Proceedings of International Society for Magnetic Resonance in Medicine, vol. 13, p. 226 (2005)

18. Davoodi-Bojd, E., Soltanian-Zadeh, H.: Atlas based segmentation of white matter fiber bundles in DTMRI using fractional anisotropy and principal eigenvectors. In: Proceedings of 2008 International Symposium on Biomedical Imaging, pp. 879–882. Paris, France (2008)
19. Duda, J., Rivera, M., Alexander, D.C., Gee, J.C.: A method for nonrigid registration of diffusion tensor magnetic resonance images. In: J.M.F. Milan Sonka (ed.) SPIE Medical Imaging 2003: Image Processing, vol. 5032, pp. 1186–1196. San Diego, CA, USA (2003)
20. Faugeras, O., Lenglet, C., Papadopoulo, T., Deriche, R.: Non-rigid registration of diffusion tensor images. Tech. Rep. 6104, Institu National de Recherche en Informatique et en Automatique (INRIA), France (2007)
21. Fitzpatrick, J., Hill, D.L.M., Maurer Jr., C.: Handbook of Medical Image Registration, vol. 2. Medical Image Processing and Analysis, chap. 8. Image Registration, pp. 447–513. SPIE Press (2000)
22. Fréchet, M.: Les elements aleatoires de nature quelconque dans un espace distancié. Annales de l'Institute Henri Poincaré **10**(4), 215–310 (1948)
23. Gee, J.C., Alexander, D.C.: Visualization and Processing of Tensor Fields, chap. 20. Diffusion-Tensor Image Registration, pp. 327–342. Springer-Verlag Berlin Heidelberg, Germany (2006)
24. Gee, J.C., Alexander, D.C., Rivera, M., Duda, J.T.: Non-rigid registration of diffusion tensor MR images. In: IEEE International Symposium on Biomedical Imaging, pp. 477–480. Washington, DC, USA (2002)
25. Goh, A., Vidal, R.: Algebraic methods for direct and feature based registration of diffusion tensor images. In: European Conference on Computer Vision, pp. 514–525. Graz, Austria (2006)
26. Guimond, A., Guttman, C.R.G., Warfield, S.K., Westin, C.F.: Deformable registration of DT-MRI data based on transformation invariant tensor characteristics. In: IEEE International Symposium on Biomedical Imaging, pp. 761–764. Washington, DC, USA (2002)
27. Jones, D.K., Griffin, L.D., Alexander, D.C., Catani, M., Horsfiled, M.A., Howard, R., Williams, S.C.R.: Spatial normalization and averaging of diffusion tensor MRI data sets. NeuroImage **17**(2):592–617 (2002)
28. Leemans, A., Sijbers, J., De Backer, S., V., E., Parizel, P.M.: Affine coregistration of diffusion tensor magnetic resonance images using mutual information. In: Proceedings of 2005 Advanced Concepts for Intelligent Vision Systems 2005, pp. 523–530. Antwerp, Belgium (2005)
29. Leemans, A., Sijbers, J., Vandervliet, E., Parizel, P.M.: Multiscale white matter fiber tract coregistration: A new feature-based approach to align diffusion tensor data. Magnetic Resonance in Medicine **55**, 1414–1423 (2006)
30. Mori, S., van Zijl, P.C.M.: Fiber tracking: principles and strategies-a technical review. NMR in biomedicine **15**, 468–480 (2002)
31. Park, H., Kubicki, M., Shenton, M.E., Guimond, A., McCarley, R.W., Maier, S.E., Kikinis, R., Jolesz, F.A., Westin, C.F.: Spatial normalization of diffusion tensor MRI using multiple channels. NeuroImage **20**, 1995–2009 (2003)
32. Pierpaoli, C., Basser, P.: Toward a quantitative assessment of diffusion anisotropy. Magnetic Resonance in Medicine **36**, 893–906 (1996)
33. Pollari, M., Neuvonen, T., Lilja, M., Lötjönen, J.: Comparative evaluation of voxel similarity measures for affine registration of diffusion tensor MR images. In: IEEE International Symposium on Biomedical Imaging, ISBI, pp. 768–771. Washington, DC., USA (2007)
34. Pollari, M., Neuvonen, T., Lötjönen, J.: Affine registration of diffusion tensor MR images. In: R. Larsen, M. Nielsen, J. Sporring (eds.) Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006, vol. 4191, pp. 629–636. Springer-Verlag Berlin Heidelberg, Copenhagen, Denmark (2006)
35. Rohde, G., Pajevic, S., Pierpaoli, C., Basser, P.: A comprehensive approach for multichannel image registration. In: International Workshop on Biomedical Image Registration, *Lecture Notes on Computer Science*, vol. 2717, pp. 214–223. Pennsylvania, PA, USA (2003)
36. Rohde, G.K., Pajevic, S., Pierpaoli, C.: Multi-channel registration of diffusion tensor images using directional information. In: 2004 IEEE International Symposium on Biomedical Imaging: from Nano to Macro, vol. 1, pp. 712–715. Arlington, VA, USA (2004)

37. Ruiz-Alzola, J., Alberola-López, C., Westin, C.F.: Kriging filters for multidimensional signal processing. Signal Processing **85**(2), 413–439 (2005)
38. Ruiz-Alzola, J., Kikinis, R., Westin, C.F.: Detection of point landmarks in multidimensional tensor data. Signal Processing **81**, 2243–2247 (2001)
39. Ruiz-Alzola, J., Westin, C.F., Warfield, S., Nabavi, A., Kikinis, R.: Nonrigid registration of 3D scalar vector and tensor medical data. In: Medical Imaging Computing and Computer-Assisted Intervention, Lecture Notes on Computer Science, pp. 541–550. Springer-Verlag (2000)
40. Ruiz-Alzola, J., Westin, C.F., Warfield, S.K., Alberola, C., Maier, S., Kikinis, R.: Nonrigid registration of 3D tensor medical data. Medical Image Analysis **6**, 143–161 (2002)
41. Shen, D., Davatzikos, C.: HAMMER: Hierarchical Attribute Matching Mechanism for Elastic Registration. IEEE Transactions on Medical Imaging **21**, 1421–1439 (2002)
42. Sundgren, P.C., Dong, Q., Gómez-Hassan, D., Mukherji, S.K., Maly, P., Welsh, R.: Diffusion tensor imaging of the brain: review of clinical applications. Neuroradiology **46**, 339–350 (2004)
43. Thirion, J.P.: Image matching as a diffusion process: an analogy with Maxwell's demons. Medical Image Analysis **2**(3), 243–260 (1998)
44. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Non-parametric diffeomorphic image registration with the demons algorithm. In: Medical Imaging Computing and Computer-Assisted Intervention, vol. 4792, pp. 319–326 (2007)
45. Verma, R., Davatzikos, C.: Matching of diffusion tensor images using Gabor features. In: IEEE International Symposium on Biomedical Imaging, pp. 396–399. Arlington, VA, USA (2004)
46. Wakana, S., Jiang, H., Nagae-Poetscher, L.M., van Zijl, P.C.M., Mori, S.: Fiber tract-based atlas of human white matter anatomy. Radiology **230**, 77–87 (2004)
47. Wells, W.M., Viola, P., Atsumi, H., Nakajima, S., Kikinis, R.: Multi-modal volume registration by maximization of mutual information. Medical Image Analysis **1**, 25–51 (1996)
48. Westin, C.F., Maier, S.E., Mamata, H., Nabavi, A., Jolesz, F.A., Kikinis, R.: Processing and visualization for diffusion tensor MRI. Medical Image Analysis **6**, 93–108 (2002)
49. Xu, D., Hao, X., Bansal, R., Plessen, K.J., Peterson, B.S.: Seamless warping of diffusion tensor fields. IEEE Transactions on Medical Imaging **27**(3), 285–299 (2008)
50. Xu, D., Mori, S., Shen, D., van Zijl, P.C.M., Davatzikos, C.: Spatial normalization of diffusion tensor fields. Magnetic Resonance in Medicine **50**, 157–182 (2003)
51. Yeo, B.T., Vercauteren, T., Fillard, P., Pennec, X., Golland, P., Ayache, N., Clatz, O.: DTI registration with exact finite-strain differential. In: Proceedings of 2008 International Symposium on Biomedical Imaging, pp. 700–703. Paris, France (2008)
52. Zhang, H., Yushjevich, P.A., Gee, J.C.: Deformable registration of diffusion tensor MR images with explicit orientation optimization. In: J. Duncan, G. Gerig (eds.) Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005, vol. 3749, pp. 172–179. Springer-Verlag Berlin Heidelberg, Palm Springs, CA, USA (2005)
53. Zhang, H., Yushkevich, P.A., Alexander, D.C., Gee, J.C.: Deformable registration of diffusion tensor MR images with explicit orientation optimization. Medical Image Analysis **10**, 764–785 (2006)
54. Zitová, B., Fluser, J.: Image registration methods: a survey. Image and Vision Computing **21**, 977–1000 (2003)

# Practical and Intuitive Basis for Tensor Field Processing with Invariant Gradients and Rotation Tangents

Gordon L. Kindlmann and Carl-Fredrik Westin

**Abstract** Recent work has outlined a framework for analyzing diffusion tensor gradient and covariance tensors in terms of invariant gradient and rotation tangents, which span local variations in tensor shape and orientation, respectively. This chapter hopes to increase the adoption of this framework by giving it a more intuitive conceptual description, as well as providing practical advice for its numeric implementation. Example applications are described, with an emphasis on decomposing the third-order gradient of a diffusion tensor field.

## 1 Introduction

Diffusion tensor imaging (DTI) analysis aims to describe the complex and subtle architecture of white matter in the central nervous system based on multi-variate MRI measurements [3]. One of the challenges in tensor-valued image processing is determining how to handle the multiple degrees of freedom in each tensor sample. Many algorithms treat the coefficients of the tensor (as measured in the laboratory coordinate frame of the scanner) as channels in a multi-scalar image, similar to the independent color channels of an RGB image. The mathematics of tensor analysis, however, provide ingredients for designing tensor processing algorithms in a way that respects the biologically meaningful shape and orientation properties associated with the diffusion tensor.

The basic idea of this approach is to treat diffusion tensors as elements of a six-dimensional vector space, and to build at each tensor value $\mathbf{D}$ a coordinate system with three basis tensors (the *invariant gradients*) that span local variations in tensor shape, and three basis tensors (the *rotation tangents*) that span local variations in tensor orientation. This allows, for example, an edge detector to be sensitive to changes in anisotropy (part of tensor shape) but not fiber direction. Recent work [22]

Laboratory of Mathematics in Imaging, Brigham and Women's Hospital, Boston, Massachusetts, e-mail: gk|westin@bwh.harvard.edu

gives a detailed account of invariant gradients and rotation tangents, and the current chapter will not reproduce all the derivations. Rather, we hope here to give a better intuitive description of the method, to give more concrete information about how to implement it, and to provide an additional demonstration of its value for edge detection in tensor fields.

## 2 Mathematical Background

Describing our framework is easier with coordinate-free tensor expression. While it is always more concrete to represent tensors with their $3 \times 3$ matrix of coefficients, coordinate-free expressions can permit more concise derivations, and can also build on existing intuition about vector spaces and their bases. The notation reviewed here respects the difference between coordinate-free and coordinate-based expressions.

Our notation is summarized in Table 1, much of which is based on conventions of tensor analysis [10, 19]. A coordinate-free vector $\mathbf{v}$ has a representation in basis $\mathscr{B}$ as three coefficients $[v_1\, v_2\, v_3]^{\mathrm{t}} = v = [\mathbf{v}]_{\mathscr{B}}$ or just $[\mathbf{v}]$ where $\mathscr{B}$ is assumed. Each of the $v_i$ coefficients is determined by $v_i = \mathbf{v} \cdot \mathbf{b}_i$. We use Einstein notation: a repeated index within a term implies summation over that index, *e.g.*, $[\mathbf{D}\mathbf{v}]_i = D_{ij} v_j = \sum_{j=1}^{3} D_{ij} v_j$; and $\mathbf{D} = D_{ij} \mathbf{b}_i \otimes \mathbf{b}_j = \sum_{i=1}^{3} \sum_{j=1}^{3} D_{ij} \mathbf{b}_i \otimes \mathbf{b}_j$.

This work starts with the recognition that tensors are linear transforms, and that linear transforms constitute a vector space. We stress these points because they appear infrequently in the tensor analysis commonly used for DTI. The tensor product $\mathbf{u} \otimes \mathbf{v}$ is a linear transform defined by $(\mathbf{u} \otimes \mathbf{v})\mathbf{w} = \mathbf{u}(\mathbf{v} \cdot \mathbf{w})$ for all vectors $\mathbf{w}$ [19]. Any linear transform $\mathbf{T}$ can be expressed as a linear combination of tensor products of orthonormal basis vectors $\mathbf{b}_i$, according to $\mathbf{T} = T_{ij} \mathbf{b}_i \otimes \mathbf{b}_j$ and $T_{ij} = \mathbf{b}_i \cdot \mathbf{T}\mathbf{b}_j$. Tensor contraction $\mathbf{A} : \mathbf{B}$ is an inner product on tensors. A *principal frame* $\mathscr{E} = \{\mathbf{e}_i\}$ is an orthonormal basis of eigenvectors of $\mathbf{D}$, which diagonalizes the matrix representation $[\mathbf{D}]_{\mathscr{E}} = diag(\lambda_1, \lambda_2, \lambda_3)$. The *spectral decomposition* $\mathbf{D} = \lambda_i \mathbf{e}_i \otimes \mathbf{e}_i$ is a coordinate-free expression of a tensor $\mathbf{D}$ in terms of its eigensystem.

While it is common to think of $\mathbf{D}$ as the covariance matrix of molecular displacements due to diffusion, it also revealing to recognize that a diffusion tensor $\mathbf{D}$ is a symmetric linear transform that maps (by Fick's first law) from concentration gradient vector $\nabla c$ to diffusive flux vector $\mathbf{j} = -\mathbf{D}\nabla c$ [12]. This fundamental property was pointed out by Basser in his original DTI work [3]. Diffusion tensors are also positive-definite [3], the significance of which for diffusion tensor image processing is discussed in Section 8.

$\mathrm{Sym}_3$ is a six-dimensional vector space, as seen by forming an orthonormal basis $\mathscr{B} = \{\mathbf{B}_i\}_{i=1..6}$ for $\mathrm{Sym}_3$ from an orthonormal basis $\mathscr{B} = \{\mathbf{b}_i\}_{i=1,2,3}$ for $W$.

**Table 1** Mathematical Conventions and Notation

| | |
|---|---|
| $W$ | three-dimensional space |
| $W \otimes W$ | three-dimensional second-order tensors |
| $\mathrm{Sym}_3$ | symmetric tensors in $W \otimes W$ |
| $\mathrm{SO}_3$ | three-dimensional rotations |
| $\mathscr{B} = \{\mathbf{b}_i\}_{i=1,2,3}$ | orthonormal basis for $W$ |
| $\delta_{ij}$ | $\delta_{ij} = 1$ if $i = j$, 0 otherwise |
| $\mathbf{v}$ | vector in $W$ |
| $v = [\mathbf{v}]_{\mathscr{B}}$ | matrix representation of $\mathbf{v}$ in $\mathscr{B}$; $v_i = \mathbf{v} \cdot \mathbf{b}_i$ |
| $\mathbf{D}$ | second-order tensor in $\mathrm{Sym}_3$ |
| $D = [\mathbf{D}]_{\mathscr{B}}$ | matrix representation of $\mathbf{D}$ in $\mathscr{B}$; $D_{ij} = \mathbf{b}_i \cdot \mathbf{D}\mathbf{b}_j$ |
| $\mathbf{I}$ | identity tensor |
| $\mathbf{u} \otimes \mathbf{v}$ | $\in W \otimes W$, tensor product of $\mathbf{u}$ and $\mathbf{v}$; $[\mathbf{u} \otimes \mathbf{v}]_{ij} = u_i v_j$ |
| $\mathbf{A} : \mathbf{B}$ | $= \mathrm{tr}(\mathbf{A}\mathbf{B}^{\mathrm{t}}) = A_{ij}B_{ij} \in \mathbb{R}$; contraction of $\mathbf{A}$ and $\mathbf{B}$ |
| $\lvert\mathbf{A}\rvert$ | $= \sqrt{\mathbf{A} : \mathbf{A}}$, tensor norm of $\mathbf{A}$, the Frobenius norm of matrix $[\mathbf{A}]$ |
| $\overline{\mathbf{A}}$ | $= \mathrm{tr}(\mathbf{A})\mathbf{I}/3$, isotropic part of $\mathbf{A}$ |
| $\widetilde{\mathbf{A}}$ | $= \mathbf{A} - \overline{\mathbf{A}}$, deviatoric part of $\mathbf{A}$ |
| $\{\lambda_i\}, \{\mathbf{e}_i\}$ | eigenvalues, eigenvectors of $\mathbf{D} = \lambda_i \mathbf{e}_i \otimes \mathbf{e}_i$ ; $\lambda_1 \geq \lambda_2 \geq \lambda_3$ |
| $\{K_i\}, \{R_i\}$ | orthogonal invariant sets; $K_1 =$ trace; $R_2 =$ FA; $K_3 = R_3 =$ mode |
| $\{\widehat{\boldsymbol{\nabla}_D K_i(\mathbf{D})}\}$ | cylindrical invariant gradients, basis for shape variation around $\mathbf{D}$ |
| $\{\widehat{\boldsymbol{\nabla}_D R_i(\mathbf{D})}\}$ | spherical invariant gradients, basis for shape variation around $\mathbf{D}$ |
| $\{\widehat{\boldsymbol{\Phi}_i(\mathbf{D})}\}$ | rotation tangents, basis for orientation variation around $\mathbf{D}$ |
| $\mathscr{G}$ | third-order tensor in $\mathrm{Sym}_3 \otimes W$; matrix representation $G = [\mathscr{G}]$ |
| $\mathbf{F}(\mathbf{x})$ | tensor field; the DTI volume |
| $\mathbf{D} : \mathscr{G}$ | $= D_{ij}G_{ijk}\mathbf{b}_k \in W$ contraction of $\mathscr{G}$ with $\mathbf{D}$ |
| $\nabla J$ | $= \boldsymbol{\nabla}_D J : \nabla\mathbf{F}$, spatial gradient of $J$ in $\mathbf{F}$ |
| $\nabla\widehat{J}$ | $= \widehat{\boldsymbol{\nabla}_D J} : \nabla\mathbf{F}$, projected gradient of $J$ |
| $\nabla\widehat{\phi_i}$ | $= \widehat{\boldsymbol{\Phi}_i} : \nabla\mathbf{F}$, spatial "gradient" of $\mathbf{e}_i$ rotation |

$$\left.\begin{array}{rcl} \mathbf{B_1} &=& \mathbf{b}_1 \otimes \mathbf{b}_1 \\ \mathbf{B_2} &=& (\mathbf{b}_1 \otimes \mathbf{b}_2 + \mathbf{b}_2 \otimes \mathbf{b}_1)/\sqrt{2} \\ \mathbf{B_3} &=& (\mathbf{b}_1 \otimes \mathbf{b}_3 + \mathbf{b}_3 \otimes \mathbf{b}_1)/\sqrt{2} \\ \mathbf{B_4} &=& \mathbf{b}_2 \otimes \mathbf{b}_2 \\ \mathbf{B_5} &=& (\mathbf{b}_2 \otimes \mathbf{b}_3 + \mathbf{b}_3 \otimes \mathbf{b}_2)/\sqrt{2} \\ \mathbf{B_6} &=& \mathbf{b}_3 \otimes \mathbf{b}_3 \end{array}\right\} \tag{1}$$

Tensors in $\mathrm{Sym}_3$ can be decomposed into vector components by

$$\mathbf{D} = D_{\boldsymbol{i}}\mathbf{B}_{\boldsymbol{i}} = (\mathbf{D} : \mathbf{B}_{\boldsymbol{i}})\mathbf{B}_{\boldsymbol{i}}. \tag{2}$$

We use bold subscripts $\boldsymbol{i}$ to index components of $\mathrm{Sym}_3$ considered as vectors rather than tensors. In addition to the coordinate-free description by (1), the treatment of

symmetric tensors as vectors can also be shown with matrix representations:

$$[\mathbf{D}]_{\mathscr{B}} = \begin{bmatrix} D_{\mathbf{1}} \\ D_{\mathbf{2}} \\ D_{\mathbf{3}} \\ D_{\mathbf{4}} \\ D_{\mathbf{5}} \\ D_{\mathbf{6}} \end{bmatrix} = \begin{bmatrix} D_{11} \\ \sqrt{2}D_{12} \\ \sqrt{2}D_{13} \\ D_{22} \\ \sqrt{2}D_{23} \\ D_{33} \end{bmatrix} \tag{3}$$

$$[\mathbf{D}]_{\mathscr{B}} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ & D_{22} & D_{12} \\ \text{sym} & & D_{33} \end{bmatrix} = \begin{bmatrix} D_{\mathbf{1}} & D_{\mathbf{2}}/\sqrt{2} & D_{\mathbf{3}}/\sqrt{2} \\ & D_{\mathbf{4}} & D_{\mathbf{5}}/\sqrt{2} \\ \text{sym} & & D_{\mathbf{6}} \end{bmatrix}. \tag{4}$$

## 3 Conceptual Overview

As described in [22], a tensor $\mathbf{D}$ is rotated by $\mathbf{R}$ in $\mathrm{SO}_3$ by

$$\psi(\mathbf{R},\mathbf{D}) = \mathbf{R}\mathbf{D}\mathbf{R}^{\mathrm{t}} \tag{5}$$

which changes the eigenvectors but not the eigenvalues (and hence not the *shape*) of $\mathbf{D}$. The *orbit* $\mathrm{SO}_3(\mathbf{D})$ of $\mathbf{D}$ is the set of all possible values of $\psi(\mathbf{R},\mathbf{D})$, that is, all reorientations of $\mathbf{D}$. Two tensors have the same shape if and only if they are on the same orbit. An *invariant* $J$ is a scalar-valued function of tensors that is constant on orbits: $\psi$: $\mathrm{SO}_3(\mathbf{D}_0) = \mathrm{SO}_3(\mathbf{D}_1) \Rightarrow J(\mathbf{D}_0) = J(\mathbf{D}_1)$. Trace $\mathrm{tr}()$ and determinant $\det()$ are invariants, as are the eigenvalues, and any function of the eigenvalues.

We create *at each tensor* $\mathbf{D}$ a local orthonormal $\mathrm{Sym}_3$ basis, with basis vectors (or "basis tensors") aligned with biologically meaningful degrees of freedom, namely *shape* and *orientation*. The tensor-valued *invariant gradients* are perpendicular to the orbits, and thus span local variations in tensor shape. The tangents to orbits, which we term *rotation tangents*, span local variations in tensor orientation, see Figure 1. In the following we develop expressions for the invariant gradients and

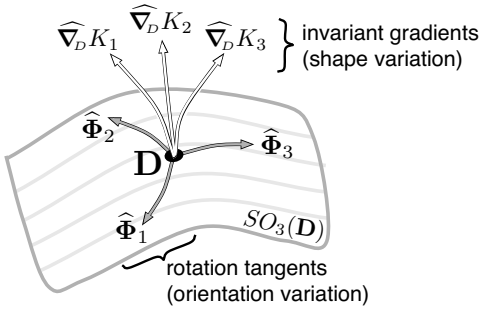



**Fig. 1** Schematic view of degrees of freedom of shape and orientation variation around a given tensor $\mathbf{D}$. The tangents to the orbit $SO_3(\mathbf{D})$ (the rotation tangents, notated $\widehat{\mathbf{\Phi}}_i$) span variation in orientation around $\mathbf{D}$. The gradients of invariants (notated either $\widehat{\nabla}_D K_i$ or $\widehat{\nabla}_D R_i$ depending on the invariant set chosen) span variations in shape around $\mathbf{D}$.

rotation tangents that can be applied for practical operations. The full explanation for these derivations can be found in [22].

## 4 Invariant Gradients

Just as a scalar function defined over three-dimensional space has a vector-valued gradient that points in the direction of greatest increase, the gradient of a tensor invariant is a tensor-valued direction of fastest increase in the invariant. In terms of a first-order Taylor expansion [19],

$$J(\mathbf{D}_0 + \mathrm{d}\mathbf{D}) = J(\mathbf{D}_0) + \frac{\partial J}{\partial \mathbf{D}}(\mathbf{D}_0) : \mathrm{d}\mathbf{D} + O(\mathrm{d}\mathbf{D}^2) \tag{6}$$

$$\nabla_D J = \frac{\partial J}{\partial \mathbf{D}}. \tag{7}$$

We use $\nabla_D$ to denote the gradient of a function with respect to its tensor-valued argument (while gradients with respect to position in $W$ are denoted by the usual $\nabla$). Two invariants $J_1$ and $J_2$ are orthogonal if $\nabla_D J_1(\mathbf{D}) : \nabla_D J_2(\mathbf{D}) = 0$ for all $\mathbf{D}$, which intuitively means their level-sets are everywhere perpendicular.

Previous work has advocated two particular sets of three orthogonal invariants, notated $K_i$ and $R_i$ [15]

$$\begin{array}{ll} K_1(\mathbf{D}) = \mathrm{tr}(\mathbf{D}) & R_1(\mathbf{D}) = |\mathbf{D}| \\ K_2(\mathbf{D}) = |\widetilde{\mathbf{D}}| & R_2(\mathbf{D}) = \mathrm{FA}(\mathbf{D}) \\ K_3(\mathbf{D}) = R_3(\mathbf{D}) = \mathrm{mode}(\mathbf{D}) & . \end{array} \tag{8}$$

The mode invariant is [13]

$$\mathrm{mode}(\mathbf{D}) = 3\sqrt{6}\det(\widetilde{\mathbf{D}}/|\widetilde{\mathbf{D}}|). \tag{9}$$

The $K_i$ and $R_i$ invariant sets can be understood as cylindrical ($K_i$) or spherical ($R_i$) coordinate systems on the three-dimensional space of diagonal matrices ($D_{12} = D_{13} = D_{23} = 0$), centered on a central axis where $D_{11} = D_{22} = D_{33}$ [15]. We adopt these invariant sets because they naturally isolate biologically significant tensor attributes of size (tensor trace $K_1$ [28, 34] or norm $R_1$), amount of anisotropy (eigenvalue standard deviation $K_2$ or fractional anisotropy FA $= R_2$ [5, 24, 25, 33, 35]), and type of anisotropy (mode $K_3 = R_3$) [22]. Note that Trace ($K_1$) and FA ($R_2$) are not part of the same invariant set and are therefore not orthogonal, despite their common paired use as two complementary shape measures [20, 27].

The third invariant in both sets is *mode* [13]. Mode is a dimensionless parameter of anisotropy *type*, varying between -1 and +1, proportional to eigenvalue skewness [15]. Negative mode indicates planar anisotropy (oblateness, two large eigenvalues and one small eigenvalue); positive mode indicates linear anisotropy (prolateness, one large eigenvalue and two small). Fig. 2 illustrates the space spanned by tensor

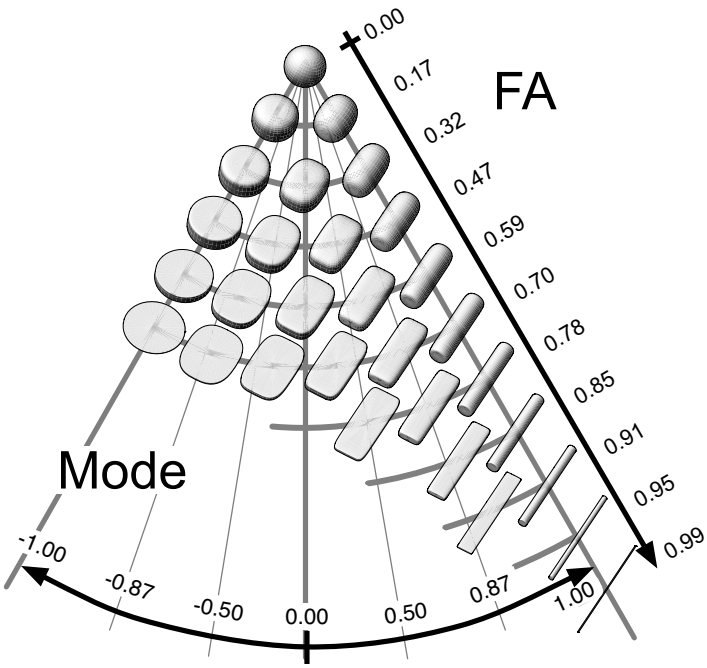



**Fig. 2** Illustration of the bivariate space of FA $= R_2$ and Mode $= R_3 = K_3$ for tensors of fixed norm $R_1$. Tensors not shown (at high FA and negative mode) have negative eigenvalues.

mode and FA, using superquadric tensor glyphs [16]. Mode becomes less meaningful when $K_2$ or $R_2$ is low.

The tensor-valued gradients of $K_i$ and $R_i$ span local shape variations [15]:

$$\begin{aligned} \nabla_D K_1(\mathbf{D}) &= \mathbf{I} & \nabla_D R_1(\mathbf{D}) &= \mathbf{D}/|\mathbf{D}| \\ \nabla_D K_2(\mathbf{D}) &= \theta(\mathbf{D}) & \nabla_D R_2(\mathbf{D}) &= \sqrt{\tfrac{3}{2}}\left(\tfrac{\theta(\mathbf{D})}{|\mathbf{D}|} - \tfrac{|\widetilde{\mathbf{D}}|\mathbf{D}}{|\mathbf{D}|^3}\right) \\ \nabla_D K_3(\mathbf{D}) &= \nabla_D R_3(\mathbf{D}) = \tfrac{3\sqrt{6}\theta(\mathbf{D})^2 - 3K_3(\mathbf{D})\theta(\mathbf{D}) - \sqrt{6}\mathbf{I}}{K_2(\mathbf{D})} \end{aligned} \tag{10}$$

where $\theta(\mathbf{D}) = \widetilde{\mathbf{D}}/|\widetilde{\mathbf{D}}|$. To create elements of an orthonormal $\mathrm{Sym}_3$ basis, we normalize the invariant gradients. $\widehat{\nabla_D} J$ denotes the unit-norm tensor-valued gradient of invariant $J$:

$$\widehat{\nabla_D} J(\mathbf{D}) = \nabla_D J(\mathbf{D})/|\nabla_D J(\mathbf{D})|\,. \tag{11}$$

These formulae obscure the fact that the numerical implementation of the invariant gradients is fairly simple. Finding the coefficients for the matrix representation of $\widehat{\nabla_D} K_1 = \nabla_D K_1$ and $\widehat{\nabla_D} R_1 = \nabla_D R_1$ is straightforward. With

$$[\widetilde{\mathbf{D}}]_{\mathscr{B}} = \begin{bmatrix} \frac{1}{3}(2D_{11} - D_{22} - D_{33}) & D_{12} & D_{13} \\ & \frac{1}{3}(2D_{22} - D_{11} - D_{33}) & D_{23} \\ \text{sym} & & \frac{1}{3}(2D_{33} - D_{11} - D_{22}) \end{bmatrix}, \tag{12}$$

$\widehat{\nabla_D} K_2 = \nabla_D K_2 = \widetilde{\mathbf{D}}/|\widetilde{\mathbf{D}}|$ is easily computed. Knowing that the magnitude of $\nabla_D R_2$ will be normalized to compute $\widehat{\nabla_D} R_1$, we can easily compute

$$\mathbf{E} = \sqrt{\frac{2}{3}}|\mathbf{D}|^2 \nabla_D R_2 = \frac{|\mathbf{D}|}{|\widetilde{\mathbf{D}}|}\widetilde{\mathbf{D}} - \frac{|\widetilde{\mathbf{D}}|}{|\mathbf{D}|}\mathbf{D} \tag{13}$$

and then $\widehat{\nabla_D} R_2 = \widehat{\nabla_D}\mathrm{FA} = \mathbf{E}/|\mathbf{E}|$. The formula for the remaining gradient $\nabla_D K_3 = \nabla_D R_3 = \nabla_D \mathrm{mode}$ is unwieldy, but it can be found by starting with the gradient of the determinant, which is is known from tensor analysis as [19]

$$\nabla_D \det(\mathbf{D}) = \det(\mathbf{D})\mathbf{D}^{-1}. \tag{14}$$

Subtracting out the components of $\nabla_D \det(\mathbf{D})$ parallel to $\widehat{\nabla_D} K_1$ and $\widehat{\nabla_D} K_2$ gives

$$\mathbf{G} = \nabla_D \det - (\nabla_D \det : \widehat{\nabla_D} K_1)\widehat{\nabla_D} K_1 - (\nabla_D \det : \widehat{\nabla_D} K_2)\widehat{\nabla_D} K_2 \tag{15}$$

which is simple to compute as a matrix given the equations above. It can be shown [23] that $\mathbf{G}$ vanishes only at the extremum of mode (where the gradient of mode vanishes anyway), so without loss of generality we can then define

$$\widehat{\nabla_D} K_3 = \mathbf{G}/\|\mathbf{G}\|. \tag{16}$$

which completes the numerical implementation of the normalized invariant gradients.

## 5 Rotation Tangents

The rotation tangents are defined in terms of the tensor eigenvectors $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, which are important for DTI applications. In nervous tissue, the principal eigenvector $\mathbf{e}_1$ is aligned with the direction of the white matter fiber tracts [9, 14, 32], which is the basis of most deterministic fiber tracking algorithms [7, 11].

Let $\mathbf{R}_\mathbf{v}(\phi) \in \mathrm{SO}_3$ be rotation by angle $\phi$ around $\mathbf{v}$. We define the *rotation tangent* $\boldsymbol{\Phi}_i(\mathbf{D})$ associated with eigenvector $\mathbf{e}_i$ of $\mathbf{D}$ as the change in tensor value due to infinitesimal rotations (5) around $\mathbf{e}_i$ [22]:

$$\boldsymbol{\Phi}_i(\mathbf{D}) = \left.\frac{\partial \psi(\mathbf{R}_{\mathbf{e}_i}(\phi), \mathbf{D})}{\partial \phi}\right|_{\phi=0} \Rightarrow \tag{17}$$

$$\boldsymbol{\Phi}_1(\mathbf{D}) = (\lambda_2 - \lambda_3)(\mathbf{e}_2 \otimes \mathbf{e}_3 + \mathbf{e}_3 \otimes \mathbf{e}_2)\,, \tag{18}$$

$$\boldsymbol{\Phi}_2(\mathbf{D}) = (\lambda_1 - \lambda_3)(\mathbf{e}_1 \otimes \mathbf{e}_3 + \mathbf{e}_3 \otimes \mathbf{e}_1)\,, \tag{19}$$

$$\boldsymbol{\Phi}_3(\mathbf{D}) = (\lambda_1 - \lambda_2)(\mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1)\,. \tag{20}$$

The rotation tangents $\boldsymbol{\Phi}_i(\mathbf{D})$ are mutually orthogonal, and all $\boldsymbol{\Phi}_i(\mathbf{D})$ are orthogonal to all invariant gradients [22].

The eigenvalue differences that scale the magnitude of the $\boldsymbol{\Phi}_i$ correspond to the intuition that if two eigenvalues are equal, then the tensor is rotationally symmetric, and there is no effect of rotating around its symmetry axis. To strengthen this intuition, for a tensor $\mathbf{D} = \lambda_i(\mathbf{e}_i \otimes \mathbf{e}_i)$ (spectral decomposition), we define $\mathbf{P} = \|\boldsymbol{\Phi}_i\|(\mathbf{e}_i \otimes \mathbf{e}_i)$ which has the same eigenvectors as $\mathbf{D}$, but has eigenvalues that reflect the magnitudes of the corresponding rotation tangents. These tensors are visualized in Figure 3(e). When $\mathbf{D}$ has rotational symmetry, either with linear or planar anisotropy (where mode $K_3$ is extremal), the orientation space of $\mathbf{D}$ is only two-dimensional (instead of three-dimensional), thus one eigenvalue of $\mathbf{P}$ is zero, and the $\mathbf{P}$ glyph is a flat disc. Note also that the overall size of $\mathbf{P}$ varies with anisotropy, as measured by $K_2$. Interestingly, although the space of orientation variation is in general three-dimensional, it is never isotropic: the $\mathbf{P}$ glyphs are never spheres (except at $\mathbf{P} = \mathbf{0}$). Thus, the $\mathbf{P}$ glyphs provide an general sense of how the space of orientation variation depends on tensor anisotropy and mode. The total number of degrees of freedom in the tensor is always six, but at rotational symmetries, the distinction between directions of shape variation and directions of orientation variation becomes blurred, as described in [22].

Unit-norm rotation tangents are defined as

$$\widehat{\boldsymbol{\Phi}}_1(\mathbf{D}) = (\mathbf{e}_2 \otimes \mathbf{e}_3 + \mathbf{e}_3 \otimes \mathbf{e}_2)/\sqrt{2} \tag{21}$$

$$\widehat{\boldsymbol{\Phi}}_2(\mathbf{D}) = (\mathbf{e}_3 \otimes \mathbf{e}_1 + \mathbf{e}_1 \otimes \mathbf{e}_3)/\sqrt{2} \tag{22}$$

$$\widehat{\boldsymbol{\Phi}}_3(\mathbf{D}) = (\mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1)/\sqrt{2}\,. \tag{23}$$

There are no simple matrix expressions for the $\widehat{\boldsymbol{\Phi}}_i$ in the laboratory frame because they depend on the tensor eigenvectors, but they are simple to create once the eigenvectors are found. Our approach for tensor analysis is the combination of normalized invariant gradients (either $\{\widehat{\nabla_D K_i}\}$ or $\{\widehat{\nabla_D R_i}\}$) and rotation tangents $\{\widehat{\boldsymbol{\Phi}}_i\}$, which together constitute an orthonormal $\mathrm{Sym}_3$ basis, designed around a given tensor, to span the sub-spaces of shape and orientation variation.

## 6 Example Application: Edge Detection

With the machinery of invariant gradients and rotation tangents in place, one of the simplest applications is edge detection, based on measuring the spatial gradient

within the tensor data. Let $\mathbf{F}$ be a smooth tensor-valued image, or tensor field that represents DTI data:

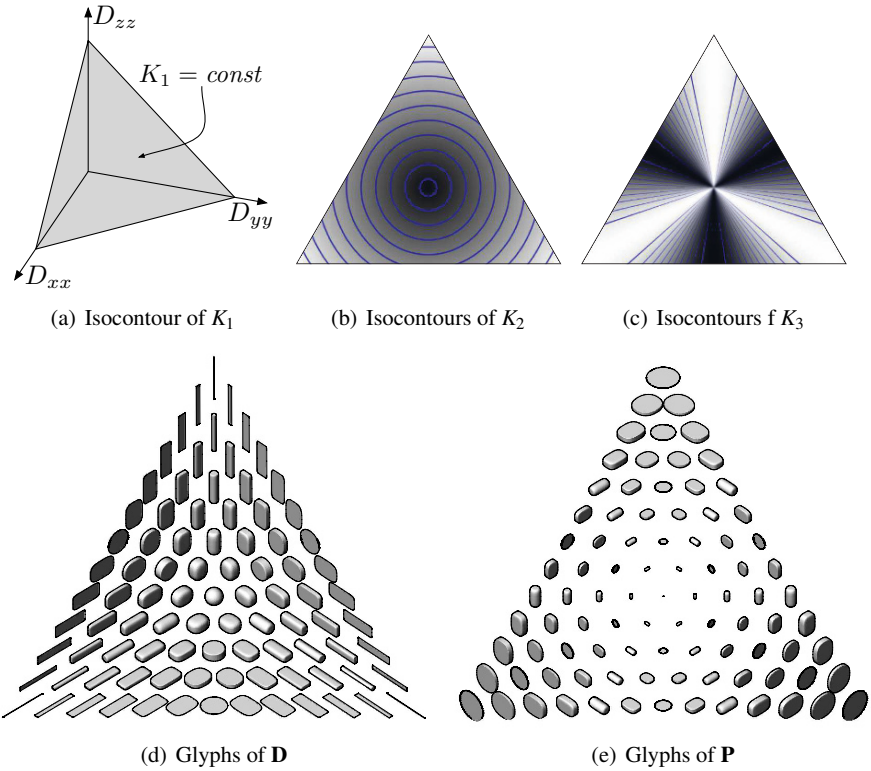
$$\mathbf{F} : W \mapsto \mathrm{Sym}_3 \tag{24}$$

$$\mathbf{F}(\mathbf{x}) = \mathbf{D}. \tag{25}$$

The spatial gradient of $\mathbf{F}$ is a third-order tensor [19], previously described by Pajevic *et al.*, as part of their spline-based reconstruction [29]:

$$\begin{aligned} \nabla\mathbf{F} : W &\mapsto \mathrm{Sym}_3 \otimes W \\ \nabla\mathbf{F}(\mathbf{x}) &= \mathscr{G} \\ [\mathscr{G}]_{ijk} &= G_{ijk} = [\nabla\mathbf{F}(\mathbf{x})]_{ijk} = \left[\frac{\partial \mathbf{F}(\mathbf{x})}{\partial x_k}\right]_{ij} \end{aligned} \tag{26}$$

(26) describes how to compute the third-order gradient tensor; each $D_{ij}$ in the matrix representation of the tensor is replaced by the spatial gradient of $D_{ij}$ in the field.




(a) Isocontour of $K_1$ (b) Isocontours of $K_2$ (c) Isocontours f $K_3$

(d) Glyphs of $\mathbf{D}$ (e) Glyphs of $\mathbf{P}$

**Fig. 3** The space of effective orientation change in $\mathbf{D} = \lambda_i(\mathbf{e}_i \otimes \mathbf{e}_i)$, visualized by $\mathbf{P} = \|\boldsymbol{\Phi}_i\|(\mathbf{e}_i \otimes \mathbf{e}_i)$. Symmetries in $\mathbf{D}$ correspond to zero eigenvalues in $\mathbf{P}$.

Considering the coordinate-free representation, however, we see that the contraction of the full tensor gradient $\mathscr{G}$ with a fixed second-order tensor $\mathbf{T}$

$$\mathbf{T} : \mathscr{G} = \mathbf{T} : \nabla \mathbf{F}(\mathbf{x}) = \nabla (\mathbf{T} : \mathbf{F}(\mathbf{x})). \tag{27}$$

is the vector-valued gradient of the scalar $\mathbf{T} : \mathbf{F}(\mathbf{x})$. Thus, contractions of the gradient tensor $\nabla \mathbf{F}$ can access the differential structure of attributes of $\mathbf{F}$. Invariant gradients and rotation tangents provide the tensors with which we contract $\nabla \mathbf{F}$, generating three spatial gradient vectors of tensor shape, and three spatial gradients of tensor orientation.

Using the normalized invariant gradients, we define the *projected gradient* of invariant $J$ in tensor field $\mathbf{F}$ by contracting $\nabla \mathbf{F}$ with the unit-norm $\widehat{\nabla_D J}$

$$\begin{aligned} \nabla \widehat{J} : W &\mapsto W \\ \nabla \widehat{J}(\mathbf{x}) &= \widehat{\nabla_D J}(\mathbf{F}(\mathbf{x})) : \nabla \mathbf{F}(\mathbf{x}) && (28) \\ &= \nabla_D J(\mathbf{F}(\mathbf{x})) : \nabla \mathbf{F}(\mathbf{x}) / |\nabla_D J(\mathbf{F}(\mathbf{x}))| && (29) \end{aligned}$$

$\nabla \widehat{J}$ is an abuse of notation to indicate normalization by tensor norm $|\nabla_D J|$, rather than vector length $|\nabla J|$; *i.e.*, $\nabla \widehat{J} \neq \widehat{\nabla} J$. $\widehat{\nabla_D J}$ differs from the regular spatial gradient of the invariant $\nabla J$ by a scaling factor that depends on the parameterization of $J$. By using normalized invariant gradients, the specifics of parameterization are removed. To numerically compute the projected invariant gradient of an invariant $J$, one first computes the $3 \times 3$ matrix $[\widehat{\nabla_D J}]$ (Sect. 4 describes this for $\widehat{\nabla_D K_i}$ and $\widehat{\nabla_D R_i}$), then computes the $3 \times 3 \times 3$ matrix $[\nabla \mathbf{F}]$ of the tensor field spatial derivative by finding the spatial gradient of each tensor coefficient $D_{ij}$ (26). The project gradient is then found by contraction:

$$\begin{aligned} [\nabla \widehat{J}(\mathbf{x})]_k &= \widehat{\nabla_D J}(\mathbf{F}(\mathbf{x})) : \nabla \mathbf{F}(\mathbf{x}) \\ &= \sum_{i=1,2,3} \sum_{j=1,2,3} [\widehat{\nabla_D J}(\mathbf{F}(\mathbf{x}))]_{ij} [\nabla D_{ij}]_k && (30) \end{aligned}$$

Using the rotation tangents, we define three spatial gradients of orientation, one for each of the tensor eigenvectors

$$\begin{aligned} \nabla \widehat{\phi_i} : W &\mapsto W \\ \nabla \widehat{\phi_i}(\mathbf{x}) &= \widehat{\boldsymbol{\Phi}}_i(\mathbf{F}(\mathbf{x})) : \nabla \mathbf{F}(\mathbf{x}) \,. && (31) \end{aligned}$$

$\nabla \widehat{\phi_i}$ is also clearly an abuse of notation: there is no scalar field $\phi_i$ in which we can measure the spatial gradient. Rather, $\nabla \widehat{\phi_i}$ indicates the direction (in $W$) along which the tensor orientation "$\phi_i$" around eigenvector $\mathbf{e}_i$ varies fastest. Indeed, this shows the utility of the rotation tangents: it allows the rate of rotation around eigenvector $\mathbf{e}_i$ to be isolated and measured, even though there is no global $\phi_i$ scalar quantity that represents the orientation around $\mathbf{e}_i$. The numerical computation of the orientation gradients is similar to that in (30).

To demonstrate this machinery for edge detection in a slice of a brain scan, Fig. 4 illustrates the tensor field gradient $|\nabla \mathbf{F}|$ and its decomposition along the invariant gradients and rotation tangents. Note that most of the gradient $\nabla \mathbf{F}$ is aligned along $\widehat{\nabla_D R_1}$, variation of tensor norm $R_1$, because of the large difference in diffusivity between the parenchyma (white and gray matter) and CSF. Previous work which

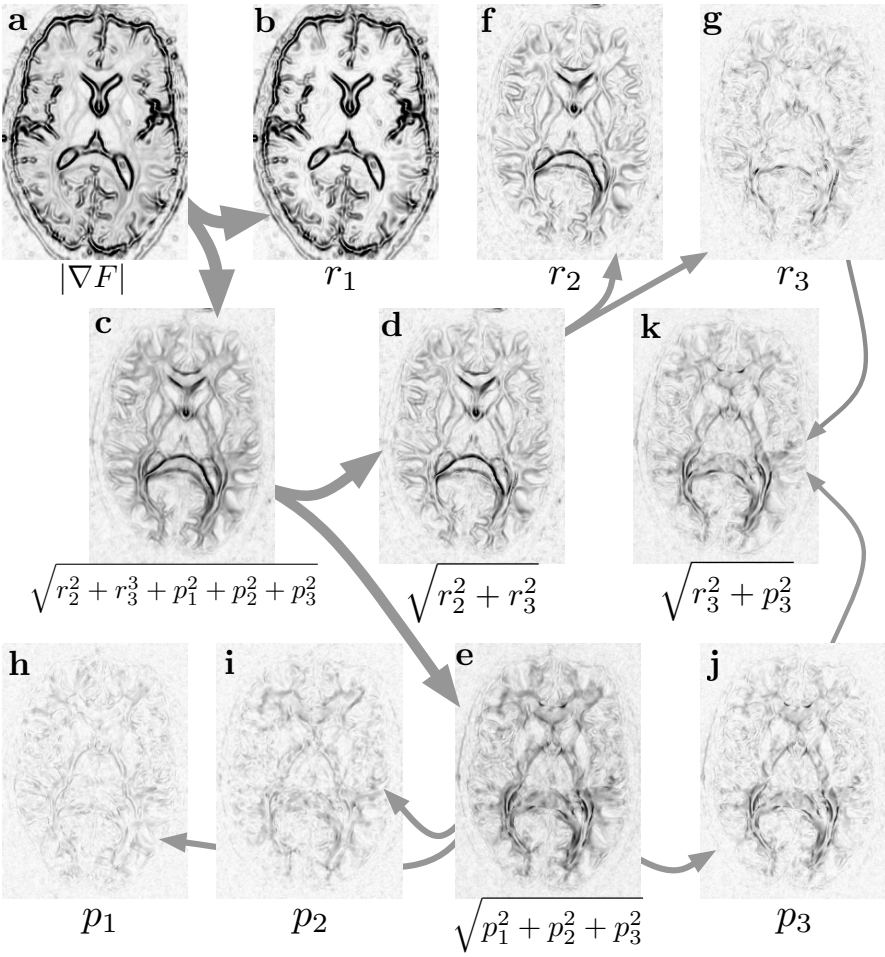



**Fig. 4** Decomposition of tensor field gradient the magnitudes of the projected $R_i$ invariant gradients $|\nabla \mathbf{F}|$ (a) with $r_i = |\nabla \widehat{R}_i|$, and the magnitudes of the orientation gradients $p_i = |\nabla \widehat{\phi}_i|$. Splitting $r_1$ (b) from the rest of the gradient (c) shows how most of the gradient is aligned with variation in tensor norm $R_1$, isolating the parenchyma-CSF boundary. Splitting (c) further, the remaining invariant gradients (d) capture tissue boundaries better than rotation tangents (e). The component along $R_2 = \text{FA}$ (f) in particular clearly shows the white matter boundary. The combination of $|\nabla \widehat{\phi}_3|$ (j) and $|\nabla \widehat{R}_3|$ (g) into (k) delineates white matter tracts that are adjacent yet distinctly oriented, such as between the cingulum bundle and corpus callosum, and between the tapetum, posterior corona radiata, and superior longitudinal fasciculus. Previous work terms (k) Adjacent Orthogonality [22].

introduced the study of tensor field gradients in DTI decomposed $\nabla\mathbf{F}$ into the gradient of the isotropic $\nabla\overline{\mathbf{F}}$ and deviatoric $\nabla\widetilde{\mathbf{F}}$ components, which is equivalent in our framework to separating out the component of $\nabla\mathbf{F}$ along $\widehat{\nabla_D K_1}$. Both trace $K_1$ and norm $R_1$ measure over-all size, so either will capture the CSF boundary. Our framework offers a more fine-grained decomposition of the gradient, enabling for example the Adjacent Orthogonality (AO) measure (Fig. 4(k)) that indicates locations where distinctly-oriented white matter pathways touch [22].

For comparison, Fig. 5 shows the decomposition of the tensor gradient in terms of the gradients of the individual tensor components, as measured in the $\mathscr{B}$ basis defined in (1).

The gradient components associated with the on-diagonal tensor coefficients emphasize the CSF boundary, while the gradients of off-diagonal coefficients do not, but the decomposition is not as specific as the decomposition along invariant gradients and rotation tangents, nor is it rotationally invariant. Comparing Figs. 4 and 5 supports our claim that the invariant gradients and rotation tangents offer a more biologically meaningful basis for tensor analysis.

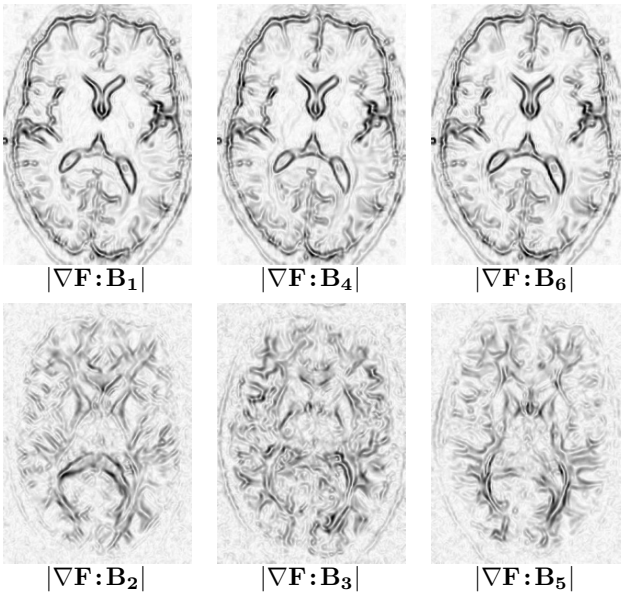



**Fig. 5** Decomposition of tensor field gradient $|\nabla\mathbf{F}|$ with the $\mathscr{B}$ basis (1) corresponding to variation in the individual tensor components $D_{ij}$; compare to (4).

## 7 Other Applications

With the ability to selectively detect different kinds of tissue boundaries with the specialized edge detectors described above, one interesting application of invariant gradients and rotation tangents is to non-linear filtering of tensor fields. Just as Perona-Malik filtering [31] and most subsequent approaches to anisotropic (inhomogeneous) diffusion are based on a conductance function of a gradient magnitude, one can imagine a PDE-based tensor filtering algorithm that is based on a conductance function of some of the select edge components illustrated in Fig. 4. It might also be possible to restrict the updates to the per-sample tensor value according to variation in shape or orientation alone. Note also that the gradients in Fig. 4 are shown only as gradient *magnitudes*, but the gradient *vectors* also have spatial direction that is likely useful for describing the orientation of tissue interfaces.

Another application of the invariant gradients and rotation tangents is for the task of tensor interpolation. Previous work has described *geodesic-loxodromes* that are geodesics along tensor orientation orbits (Sect. 3), and loxodromes through variation of tensor shape [21]. These paths have the property that the angles between the interpolation path $\mathbf{P}(s)$ and three orthogonal invariant gradients (for example $\widehat{\nabla_D R_i}$) are constants $c_i$:

$$\frac{d\mathbf{P}}{ds} : \widehat{\nabla_D R_i} = c_i \ \forall i = 1,2,3. \tag{32}$$

One can also imagine, however, an interpolation path that is also a loxodrome in orientation, which obeys (32) as well as

$$\frac{d\mathbf{P}}{ds} : \widehat{\mathbf{\Phi}}_i = k_i \ \forall i = 1,2,3. \tag{33}$$

Along such a path, the rate of rotation around each eigenvector is constant. Investigations of such interpolation paths and their properties is ongoing.

## 8 Discussion

Our framework is "Euclidean" in that we consider diffusion tensors as elements of a vector space, even though this overlooks the positive-definiteness of diffusion. This simplifying assumption has established precedent in the DTI literature [4, 6, 7], even in the context of reconstructing tensors from discrete samples [1, 29]. In some approaches DTI analysis, tensors are located on a Riemannian manifold endowed with a metric that effectively creates an infinite distance between valid tensors and those with zero determinant [2, 8, 17, 18, 26, 30]. While these methods can leverage a wide range of Riemannian formalisms, we do not feel that the positive-definite constraint is a necessary ingredient for effective DTI processing, because we view the larger goals of image analysis (such as generating geometric models of anatomical features) as more important than maintaining the physical plausibility of the sam-

ple values at every stage in the analysis. By analogy, much work with traditional scalar-valued MR images use interpolation methods (*e.g.* windowed sinc) that do not necessarily preserve the positiveness of each pixel value, even though, as the magnitude of a complex-valued MR signal, it must be positive. We feel that the immediate practical utility of our tensor gradient decomposition (shown in Fig. 4) is adequate justification for our Euclidean approach to tensor analysis; the Riemannian methods have not produced a decomposition of the same specificity. We hope that invariant gradients and rotation tangents will provide a means of extending and creating a wide variety of image processing methods to tensor fields.

**Acknowledgements** This work supported by NIH grants NIBIB T32-EB002177, U41-RR019703, R01-MH074794, and NCRR P41-RR13218.

## References

1. Akram Aldroubi and Peter Basser. Reconstruction of vector and tensor fields from sampled discrete data. *Contemporary Mathematics*, 247:1–15, 1999.
2. Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, August 2006.
3. P J Basser, J Mattiello, and D LeBihan. Estimation of the effective self-diffusion tensor from the NMR spin-echo. *Journal of Magnetic Resonance, Series B*, 103(3):247–254, 1994.
4. P J Basser and S Pajevic. A normal distribution for tensor-valued random variables: Applications to diffusion tensor MRI. *IEEE Trans. on Medical Imaging*, 22(7):785–794, July 2003.
5. Peter J. Basser. Inferring microstructural features and the physiological state of tissues from diffusion-weighted images. *Nuclear Magnetic Resonance in Biomedicine*, 8:333–344, 1995.
6. Peter J. Basser and Sinisa Pajevic. Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI. *Signal Processing*, 87:220–236, 2007.
7. Peter J. Basser, Sinisa Pajevic, Carlo Pierpaoli, Jeffrey Duda, and Akram Aldroubi. In vivo fiber tractograpy using DT-MRI data. *Magnetic Resonance in Medicine*, 44:625–632, 2000.
8. P. G. Batchelor, M. Moakher, D. Atkinson, F. Calamante, and A. Connelly. A rigorous framework for diffusion tensor calculus. *Magnetic Resonance in Medicine*, 53(1):221–225, January 2005.
9. C Beaulieu. The basis of anisotropic water diffusion in the nervous system-A technical review. *Nuclear Magnetic Resonance in Biomedicine*, 15:435–455, 2002.
10. D E Bourne and P C Kendall. *Vector Analysis and Cartesian Tensors*. CRC Press, 3rd edition, 1992.
11. Thomas E. Conturo, Nicolas F. Lori, Thomas S. Cull, Erbil Akbudak, Abraham Z. Snyder, Joshua S. Shimony, Robert C. McKinstry, Harold Burton, and Marcus E. Raichle. Tracking neuronal fiber pathways in the living human brain. *Proc. National Academy of Sciences*, 96:10422–10427, August 1999.
12. J Crank. *The Mathematics of Diffusion*. Oxford University Press, Oxford, England, 1975.
13. J C Criscione, J D Humphrey, A S Douglas, and W C Hunter. An invariant basis for natural strain which yields orthogonal stress response terms in isotropic hyperelasticity. *Journal of Mechanics and Physics of Solids*, 48:2445–2465, 2000.
14. Julien Dauguet, Sharon Peled, Vladimir Berezovskii, Thierry Delzescaux, Simon K. Warfield, Richard Born, and Carl-Fredrik Westin. 3D histological reconstruction of fiber tracts and direct comparison with diffusion tensor MRI tractography. In *Proceedings MICCAI 2006*, Lecture Notes in Computer Science 4190, pages 109–116, Copenhagen, Denmark, October 2006.

15. D B Ennis and G Kindlmann. Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images. *Magnetic Resonance in Medicine*, 55(1):136–146, 2006.
16. Daniel B. Ennis, Gordon Kindlman, Ignacio Rodriguez, Patrick A. Helm, and Elliot R. McVeigh. Visualization of tensor fields using superquadric glyphs. *Magnetic Resonance in Medicine*, 53:169–176, January 2005.
17. P Fillard, V Arsigny, X Pennec, and N Ayache. Clinical DT-MRI estimation, smoothing and fiber tracking with Log-Euclidean metrics. In *Proceedings ISBI 2006*, LNCS, pages 786–789, Arlington, Virginia, USA, April 2006.
18. P. Thomas Fletcher and Sarang Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *Proceedings ECCV 2004 Workshop on Computer Vision Approaches to Medical Image Analysis (CVAMIA)*, volume 31107 of *LNCS*, pages 87–98. Springer-Verlag, 2004.
19. G A Holzapfel. *Nonlinear Solid Mechanics*. John Wiley and Sons, Ltd, England, 2000.
20. Derek K. Jones, David Lythgoe, Mark A Horsfield, Andrew Simmons, Steve C. R. Williams, and Hugh S. Markus. Characterization of white matter damage in ischemic leukoaraiosis with diffusion tensor MRI. *Stroke*, 30:393–397, 1999.
21. G Kindlmann, RSJ Estépar, M Niethammer, S Haker, and C-F Westin. Geodesic-loxodromes for diffusion tensor interpolation and difference measurement. In *Proceedings MICCAI 2007*, Lecture Notes in Computer Science 4792, pages 1–9, Brisbane, Australia, October–November 2007.
22. Gordon Kindlmann, Daniel B. Ennis, Ross T. Whitaker, and Carl-Fredrik Westin. Diffusion tensor analysis with invariant gradients and rotation tangents. *IEEE Trans. on Medical Imaging*, 26(11):1483–1499, November 2007.
23. Gordon L Kindlmann. *Visualization and Analysis of Diffusion Tensor Fields*. PhD thesis, University of Utah, September 2004. `http://www.cs.utah.edu/$\sim$gk/PhD`.
24. T Klingberg, M Hedehus, E Temple, T Salz, J D E Gabrielli, M E Moseley, and R A Poldrack. Microstructure of temporo-parietal white matter as a basis for reading ability: Evidence from diffusion tensor magnetic resonance imaging. *Neuron*, 25:493–500, 2000.
25. M Kubicki, C-F Westin, S E Maier, H Mamata, M Frumin, H Ernst-Hirshefeld, R Kikinis, F A Jolesz, R W McCarley, and M E Shenton. Cingulate fasciculus integrity disruption in schizophrenia: A magnetic resonance diffusion tensor imaging study. *Biological Psychiatry*, 54:1171–1180, 2003.
26. C Lenglet, M Rousson, and R Deriche. DTI segmentation by statistical surface evolution. *IEEE Trans. on Medical Imaging*, 25:685–700, June 2006.
27. Stanley Lu, Daniel Ahn, Glyn Johnson, Meng Law, David Zagzag, and Robert I. Grossman. Diffusion-tensor MR imaging of intracranial neoplasia and associated peritumoral edema: Introduction of the tumor infiltration index. *Neuroradiology*, 232(1):221–228, July 2004.
28. M E Moseley, Y Cohen, J Mintorovitch, L Chileuitt, H Shimizu, J Kucharczyk, M F Wendland, and P R Weinstein. Early detection of regional cerebral ischemia in cats: Comparison of diffusion- and T2-weighted MRI and spectroscopy. *Magnetic Resonance in Medicine*, 14(2):330–346, May 1990.
29. S Pajevic, A Aldroubi, and P J Basser. A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue. *Journal of Magnetic Resonance*, 154:85–100, 2002.
30. Xavier Pennec. Probabilities and statistics on Riemannian manifolds: A geometric approach. Technical Report 5093, INRIA, Sophia Antipolis, January 2004.
31. Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropy diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
32. C Pierpaoli, P Jezzard, P J Basser, A Barnett, and G DiChiro. Diffusion tensor MR imaging of the human brain. *Radiology*, 201(3):637–648, Dec 1996.
33. S M Smith, M Jenkinson, H Johansen-Berg, D Rueckert, T E Nichols, C E Mackay, K E Watkins, O Ciccarelli, M Z Cader, P M Matthews, and T E J Behrens. Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data. *NeuroImage*, 31:1487–1505, 2006.

34. C H Sotak. The role of diffusion tensor imaging in the evaluation of ischemic brain injury-A review. *Nuclear Magnetic Resonance in Biomedicine*, 15:561–569, 2002.
35. D S Tuch, D H Salat, J J Wisco, A K Zaleta, N D Hevelone, and H D Rosas. Choice reaction time performance correlates with diffusion anisotropy in white matter pathways supporting visuospatial attention. *Proc. National Academy of Sciences*, 102(34):12212–12217, 2005.

# From Second to Higher Order Tensors in Diffusion-MRI

Aurobrata Ghosh and Rachid Deriche

**Abstract** Diffusion MRI, which is sensitive to the Brownian motion of molecules, has become today an excellent medical tool for probing the tissue micro-structure of cerebral white matter *in vivo* and *non-invasively*. It makes it possible to reconstruct fiber pathways and segment major fiber bundles that reflect the structures in the brain which are not visible to other non-invasive imaging modalities. Since this is possible without operating on the subject, but by integrating partial information from Diffusion Weighted Images into a reconstructed 'complete' image of diffusion, Diffusion MRI opens a whole new domain of image processing. Here we shall explore the role that tensors play in the mathematical model. We shall primarily deal with Cartesian tensors and begin with 2nd order tensors, since these are at the core of Diffusion Tensor Imaging. We shall then explore higher and even ordered symmetric tensors, that can take into account more complex micro-geometries of biological tissues such as axonal crossings in the white matter.

## 1 Introduction

The Brownian motion or diffusion of particles observed by Robert Brown in 1828 was first modelled independently by Albert Einstein in 1905 while trying to provide an experimentally testable hypothesis for the kinetic-molecular theory of matter [12]. Nuclear Magnetic Resonance (NMR) was discovered by Felix Bloch [7] and Edward M. Purcell [32] in 1946. In 1950 Erwin L. Hahn published a paper [16] where he noted that the amplitude of the observed NMR spin echo in the presence of a magnetic field inhomogeneity would be attenuated due to the inherent Brownian motion of the spins. Shortly after, in 1954 Herman Y. Carr and Purcell developed a set of equations [10] to describe this attenuation as a function of discrete motion of the spins. H. C. Torrey subsequently developed the continuum description in 1956

Project Team Odyssée, INRIA Sophia Antipolis – Méditerranée
Aurobrata.Ghosh@sophia.inria.fr

[35]. And about a decade later in 1965 E. O. Stejskal and J. E. Tanner designed the classical pulsed gradient spin echo (PGSE) experiment that made it possible to measure the coefficients of molecular diffusion from Diffusion NMR [33].

Magnetic Resonance Imaging (MRI) was developed by Paul C. Lauterbur in 1973 [19] making it possible to generate two and three dimensional images using NMR principles. Peter Mansfield developed the magnetic gradient scheme called Echo Planar Imaging (EPI) in [25]. This facilitated the development of Diffusion MRI (D-MRI), which saw its development in [20, 26, 34].

Since then, D-MRI has come a long way today to become a state-of-the-art medical tool for probing the tissue micro-structure of cerebral white matter *in vivo* and *non-invasively*. This became possible due to the concept of Diffusion Tensor Imaging (DTI) introduced in [5, 6] and due to the development of the diffusion propagator formalism. DTI and the diffusion propagator formalism provide ways to infer the geometry of the underlying medium.

Here we shall take a look at the role played by Cartesian tensors in the mathematical model of the diffusion propagator and the aquired Diffusion Weighted Images (DWI) or signal. Second order diffusion tensors were first introduced by Peter J. Basser in 1994 [5, 6] to accomodate anisotropic diffusion phenomenon in biological tissues, and it became the corner stone in medical D-MRI as DTI. Higher Order and symmetric Tensors (HOT) were recent additions to the propagator and signal model and were introduced to take into account more complex micro-geometries of the underlying tissue such as axonal crossings in the white matter.

We shall begin with the fundamental equations that describe the phenomenon of D-MRI. DTI shall be presented in this framework. DTI with the 2nd order tensor describes Gaussian diffusion or free or unrestricted diffusion. However, the anisotropy that the 2nd order diffusion tensor can describe can only arise from restriction, which would seem to present a contradiction [36]. This shall be resolved by exploring Generalized DTI (GDTI) which will consider the Gaussian model to be a low spatial frequency approximation of the propagator. Two distinct models of GDTI were developed and are often known by the names GDTI-1 and GDTI-2. Both shall be presented, and both employ HOTs to formulate the propagator and signal model. Finally we shall take a look at recent attempts to apply the positivity constraint on the diffusivity function while estimating 2nd and 4th order diffusion tensors from the signal. This is motivated by the fact that diffusion is a positive quantity and negative diffusion has no physical meaning. However, since the DWIs contain noisy signal the estimated tensorial diffusivity functions can often have negative values.

## 2 Principles of Diffusion MRI

We shall begin by examining the fundamental equations governing the principles of D-MRI, namely the Bloch-Torrey equation and the Stejskal-Tanner equation. These will provide the basic framework for the 2nd order diffusion tensor and its role in DTI to describe diffusion anisotropy. We shall then go back to the physics of

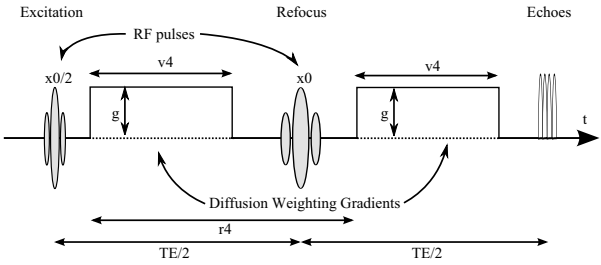



**Fig. 1** The PGSE sequence. For the idealised case of rectangular gradient pulses, $g$ represents gradient intensity, $\delta$ gradient duration, and $\Delta$ gradient spacing. Courtesy [27].

the problem and present the propagator formulation which will make it possible to describe more complex anisotropies by introducing HOTs in later sections.

## *2.1 The Bloch-Torrey Equations*

Torrey proposed the addition of two terms to the Bloch equation to account for flow and diffusion, which gives the Bloch-Torrey equation[9, 35] in the rotating frame and in the absence of radio frequency (RF) field to describe diffusion NMR as

$$\frac{\partial M_+}{\partial t} = -i\gamma \mathbf{r}.\mathbf{g}(t)M_+ - \frac{M_+}{T_2} + \nabla.(D\nabla M_+) - \nabla.\mathbf{v}M_+ \tag{1}$$

where $M_+$ is the complex magnetization vector, $\mathbf{r}$ is the spin position vector, $\mathbf{g}(t)$ is the applied magnetic gradient as a function of time, $\gamma$ is the gyromagnetic ratio, $T_2$ is the spin-spin or the transverse relaxation time, $\mathbf{v}$ is the flow velocity, $D$ is the diffusion coefficient, and $\nabla.(D\nabla M_+)$ is Fick's first law of diffusion. We shall consider the case of pure diffusion where the net flow term $\nabla.\mathbf{v}M_+$ is zero. The following development follows closely [9, 36].

The PGSE experiment was designed by Stejskal and Tanner to quantitatively measure the diffusion coefficient (Fig-1). Essentially the spin system is excited with a $\pi/2$ RF pulse and a magnetic gradient field constant over time $\delta$ (which is not always achievable in practice) is applied to encode the spin positions. Then the spins are flipped around by a $\pi$ RF pulse and the same time constant magnetic gradient is re-applied after a time $\Delta$. The echo signal is finally aquired after a time TE. If a spin diffuses during the time $\Delta$, then its phase will not return to its original orientation after the second gradient is applied. This increase of disorder in the spin phase distribution will cause the spin echo signal to attenuate proportionally to the mean phase difference.

Equation (1) can be solved by realizing that $M_+$ is a function of both $\mathbf{r}$ and $t$ and by making the substitution

$$M_+(\mathbf{r},t) = M(t)\exp(-i\mathbf{r}.\mathbf{G}(t))\exp(-t/T_2), \tag{2}$$

with

$$\mathbf{G}(t) = \gamma \int_0^t \mathbf{g}(t')dt', \tag{3}$$

in (1), where we now consider **g** to be the 'effective gradient' incorporating also the effect of the phase inversion RF pulses. Solving from there for only the real part of $M_+(\mathbf{r},t)$ which corresponds to attenuation due to diffusion, we obtain

$$M(t) = M(0)\exp\left(-D\int_0^t \mathbf{G}(t')^T\mathbf{G}(t')dt'\right), \tag{4}$$

which is the well known Stejskal-Tanner equation [33].

Equation (4), can be rewritten in the following manner by introducing the $b$-factor for $t = TE$

$$b = \int_0^{TE} \mathbf{G}(t')^T\mathbf{G}(t')dt', \tag{5}$$

and **u** a unit vector along the gradient direction **g**, giving us

$$M = M(0)\exp\left(-bD\mathbf{u}^T\mathbf{u}\right). \tag{6}$$

It can be shown that $b = \gamma^2 g^2 \delta^2 (\Delta - \frac{\delta}{3})$ [20, 9].

Equation (4) can also be modified to introduce the **B**-matrix which can take into account greater variations in the gradient term to account for imaging and spoiling gradients

$$\mathbf{B} = \int_0^{TE} \mathbf{G}(t')\mathbf{G}(t')^T dt', \tag{7}$$

then equation (4) becomes

$$M = M(0)\exp\left(-D\mathrm{tr}(\mathbf{B})\right). \tag{8}$$

The **B**-matrix was introduced by Basser [5] and is more generic than the scalar $b$-factor. This can be seen when the diffusion is anistropic and the diffusion coefficient $D$ is replaced by the 2nd order diffusion tensor **D**. We will see the 2nd order diffusion tensor shortly. In that case equation (8) becomes

$$M = M(0)\exp\left(-\mathrm{tr}(\mathbf{BD})\right). \tag{9}$$

This formulation improves the accuracy by allowing the off-diagonal terms of the **B**-matrix to couple imaging and diffusion gradient pulses in orthogonal directions over and above only the diagonal terms which account for interactions between diffusion and imaging gradient pulses in the same direction [5]. The orthogonal effects are, however, often overlooked, since their strength is typically small in comparison to the diffusion-weighting gradients, and using the $b$-factor ($b = B_{11} + B_{22} + B_{33}$) greatly simplifies calculations (reference 44 in [27]).

## 2.2 Diffusion Tensor Imaging (2nd Order)

The diffusion phenomenon is called isotropic when the apparent diffusion is independent of the diffusion direction. Anisotropic diffusion occurs when the apparent diffusion varies for different directions. It is generally caused by strongly aligned micro-structures in the medium. Diffusion can therefore occur more freely along the aligned micro-structure, while it is restricted in the perpendicular direction. Anisotropic diffusion provides a convenient way to infer the alignment in the structure of the medium.

To describe Gaussian anisotropy, Basser introduced the 2nd order diffusion tensor, a 3*imes*3 symmetric positive definite matrix $\mathbf{D}$ to replace the scalar diffusion coefficient $D$ [5, 6]. Its effect was to modify equation (8) to (9) and (6) to

$$M = M(0)\exp\left(-b\mathbf{u}^T\mathbf{D}\mathbf{u}\right). \tag{10}$$

The $\mathbf{u}^T\mathbf{D}\mathbf{u}$ is known as the diffusivity function and is often written as $D(\mathbf{g}) = \mathbf{g}^T\mathbf{D}\mathbf{g}$ with the letter $\mathbf{g}$ replacing the letter $\mathbf{u}$.

DTI provided quantitative framework for describing anisotropic diffusion. It made it possible to identify the alignment of the medium's micro-structure by diagonalizing $\mathbf{D}$ and associating its major eigenvector to this dominant alignment direction. This in turn made it possible to trace fibers in the brain's white matter.

The six independent coefficients $\{D_{ij}\}$ of $\mathbf{D}$ and the unweighted MR image $M(0)$ can be estimated from a set of seven or more DWIs. The estimation is linear in its simplest form and can be computed by linearizing equation (10) by taking the logarithms on both sides

$$\ln(M) = \ln(M(0)) - b\mathbf{u}^T\mathbf{D}\mathbf{u}. \tag{11}$$

By creating a vector out of the unknowns $\mathbf{d} = [D_{11}, D_{12}, D_{13}, D_{22}, D_{23}, D_{33}, \ln(M(0))]^T$, by computing an $N$x7 matrix $\mathbf{Y}$ for the $N$ gradient directions $\{\mathbf{u}\}_{i=1}^N$ and $b$-values from (11) such that the $i$-th row is $b_i.[u_{i1}u_{i1}, u_{i1}u_{i2}, u_{i1}u_{i3}, u_{i2}u_{i2}, u_{i2}u_{i3}, u_{i3}u_{i3}, 1/b_i]$, and by storing the logarithms of the $N$ observed diffusion signals along the corresponding gradient directions $\{\ln(M_i)\}_{i=1}^N$ in an $N$x1 vector $\mathbf{S}$, the unknowns can be estimated as

$$\mathbf{d}_{opt} = \arg\min_{\mathbf{d}} ||(\mathbf{Y}\mathbf{d} - \mathbf{S})||^2, \tag{12}$$

which gives

$$\mathbf{d}_{opt} = ((\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T)\mathbf{S}, \tag{13}$$

which is the Moore-Penrose pseudoinverse or the linear Least Square (LS) approach.

More sophisticated methods exist [39], many also estimate the diffusion tensor $\mathbf{D}$ in the presence of noisy signal. Basser in [5] takes into account a Gaussian noise model, while other approaches constrain the problem further to guarantee positive diffusivity or to apply spatial smoothing [11, 13, 22, 28, 31, 37].

## 2.3 The Diffusion Propagator

It is important to understand that the signal from the PGSE experiment measures the diffusion of spin bearing particles, and that it is an average of the superimposed signals of a large number of such particles. Therefore to understand and model the signal generation one has to be able to both describe the diffusion motion of a spin bearing particle and also be able to compute something of an 'ensemble average' quatity.

Since the Brownian motion of a particle is erratic and random it is convenient to describe it using a conditional probability density function (PDF) $P(\mathbf{r}'|\mathbf{r},t)$ which is the probability of finding a particle at position $\mathbf{r}'$ at time $t$ given that it was at the position $\mathbf{r}$ at time 0. This PDF or propagator has to then satisfy the diffusion process described by Fick's first law

$$\mathbf{J}(\mathbf{r}) = -D\nabla_{\mathbf{r}} P(\mathbf{r}'|\mathbf{r},t), \tag{14}$$

with the initial condition

$$P(\mathbf{r}'|\mathbf{r},0) = \delta(\mathbf{r}-\mathbf{r}'), \tag{15}$$

where $\mathbf{J}$ is the particle probability density flux and $D$ is the diffusion coefficient. Conservation of total conditional probability implies the continuity theorem

$$\nabla_{\mathbf{r}}\mathbf{J} = -\partial P/\partial t \tag{16}$$

which states that the rate of change of particle displacement probability is equal to the loss of probability due to the particle flux. Combining (14) and (16) gives us Fick's second law

$$\partial P/\partial t = D\nabla_{\mathbf{r}}.(\nabla_{\mathbf{r}} P)\ . \tag{17}$$

The differential equation (17) can be solved for unrestricted or free diffusion which has the special boundary condition $P \to 0$ as $\mathbf{r}' \to \infty$ and with the initial condition (15) [9], yielding

$$P(\mathbf{r}'|\mathbf{r},t) = (4\pi Dt)^{(-3/2)} \exp\left\{-(\mathbf{r}-\mathbf{r}')^2/4Dt\right\}. \tag{18}$$

Notice in this equation that $P$ only depends on $\mathbf{r}-\mathbf{r}'$ and not on the initial position $\mathbf{r}$.

The Ensemble Average Propagator (EAP) defined as

$$\overline{P}(\mathbf{R},t) = \int P(\mathbf{r}+\mathbf{R}|\mathbf{r},t)\rho(\mathbf{r})d\mathbf{r}, \tag{19}$$

can then be computed, where $\rho(\mathbf{r})$ is the particle density. It is a useful quantity and gives the average probability of any particle to have a displacement $\mathbf{R}$ over time $t$. For the free diffusion propagator (18), since it is independent of the starting position, the EAP is the same for all the particles and can be written as

$$P(\mathbf{R},t) = (4\pi Dt)^{(-3/2)} \exp\left\{-\mathbf{R}^2/4Dt\right\}, \tag{20}$$

by dropping the overhead bar.

This Gaussian propagator describing free or unrestricted diffusion is defined for isotropic diffusion by the diffusion coefficient $D$. It can be generalized to the case of anisotropic diffusion by introducing the 2nd order diffusion tensor $\mathbf{D}$ in the equation (14) to yield

$$P(\mathbf{R},t) = (|\mathbf{D}|(4\pi t)^3)^{(-1/2)} \exp\left\{-\mathbf{R}^T\mathbf{D}^{-1}\mathbf{R}/4t\right\}. \tag{21}$$

DTI, therefore, not only provides a framework for describing anisotropic diffusion, for identifying the major micro-structural alignment direction of the medium from the eigen-decomposition of $\mathbf{D}$, but also for connecting the signal (10) to the Gaussian propagator formulation (21) using the tensor $\mathbf{D}$.

The simplicity of the Gaussian or free diffusion both in terms of a model and computational load make DTI today the preferred approach in D-MRI. The decomposition and easy geometric interpretation of the 2nd order tensor $\mathbf{D}$ also plays in its favour. However, one would notice the paradox that though free diffusion is described by a Gaussian EAP, anisotropy can occur only in the presence of restriction. This seeming contradiction is the motivation to move on to a more generic propagator formulation. It will help us understand the Gaussian propagator as a low spatial frequency approximation of the EAP. It will also provide us with a framework to introduce HOTs to be able to describe more complex geometries such as fiber crossings.

## *2.4 The Fourier Relationship*

Let us assume that $\delta$ is infinitesimally short, which we shall denote as the short gradient pulse (SGP) condition. For the sake of simplicity let us also assume at first without loss of generality that the gradients are applied along the x-axis and that $g$ denotes only the magnitude of the gradient vector (Fig-1) [27]. The dephasing resulting from the two gradient pulses is then

$$\phi_1 = \gamma \int_0^{\delta} g x_1 dt = \gamma g \delta x_1, \tag{22}$$

$$\phi_2 = \gamma \int_{\Delta}^{\Delta+\delta} g x_2 dt = \gamma g \delta x_2, \tag{23}$$

where $x_1 = x(t=0)$ and $x_2 = x(t=\Delta)$ are the position vectors. The SGP condition allows for $x(t)$ to be constant over the integrals. The sign of $\phi_1$ and $\phi_2$ are opposite due to the $\pi$ RF pulse. Therefore the total dephasing due to both the gradient pulses of one particle is

$$\phi = \phi_2 - \phi_1 = \gamma g \delta (x_2 - x_1). \tag{24}$$

And so the complex signal from one particle is

$$c = e^{i\phi} = e^{i\gamma\delta g(x_2 - x_1)} = e^{i\gamma\delta gX}, \tag{25}$$

where $X$ is the net displacement due to diffusion along the x-axis. In the general case without assumptions on the gradient, which would then have the value $\mathbf{g}$ and the net displacement vector $\mathbf{R}$ the complex signal is given by

$$c = e^{i\gamma\delta\mathbf{g}.\mathbf{R}}. \tag{26}$$

To compute the ensemble average complex signal, we have to again take into account that the signal is the superimposed signals from a large number of spin bearing particles whose displacement probability is the EAP $P(\mathbf{R},t)$, therefore

$$C(\mathbf{q},t) = \left\langle e^{i\mathbf{q}.\mathbf{R}} \right\rangle = \int_{-\infty}^{\infty} e^{i\mathbf{q}.\mathbf{R}} P(\mathbf{R},t) d\mathbf{R} \tag{27}$$

$$= \mathscr{F}[P(\mathbf{R},t)], \tag{28}$$

where $\mathbf{q} = \gamma\delta\mathbf{g}$, and $\langle ... \rangle$ denotes the ensemble average. The ensemble average complex signal is, therefore, the Fourier transform of the EAP. The propagator can therefore be estimated by taking the inverse Fourier transform of the complex signal

$$P(\mathbf{R},t) = \mathscr{F}^{-1}[C(\mathbf{q},t)]. \tag{29}$$

What is interesting, however, is that it can be shown that for a pure diffusion process, the inverse Fourier transform of the complex signal is equal to the inverse Fourier transform of the modulus of the signal [36]

$$P(\mathbf{R},t) = \mathscr{F}^{-1}[|C(\mathbf{q},t)|] = \mathscr{F}^{-1}[S(\mathbf{q},t)], \tag{30}$$

which we shall call the modulus Fourier transform. Hence the diffusion signal $C(\mathbf{q},t)$ is equal to its modulus $C(\mathbf{q},t) = |C(\mathbf{q},t)| = S(\mathbf{q},t)$.

This establishes the Fourier relation between the signal and the EAP. Diffusion Spectrum Imaging (DSI) uses this relation to compute the propagator from the signal [38]. However, it requires aquiring the signal from a detailed sampling of $\mathbf{q}$-space and therefore suffers from lengthy aquisition times.

## *2.5 Cumulant Expansion of the Propagator*

Equation (27) can also be used to interpret the signal as the characteristic function or the complex cumulant generating function of the EAP. This realization permits us to rewrite the logarithm of the signal in function of the cumulants of the EAP. For simplicity of notation we do this for the one dimensional case along, say, the x-axis

$$\ln(C) = \sum_{n=1}^{\infty} \kappa_n \frac{(i\gamma\delta g)^n}{n!}, \tag{31}$$

where $\kappa_n$ are the cumulants. We shall see in the three dimensional case that these cumulants are HOTs. Assuming a pure diffusion process it can be shown that all odd order cumulants are zero or that the EAP is an even function with respect to the displacement variable. The signal is therefore the modulus signal $S$.

Truncating the cumulant expansion (31) after the second term yields a signal from a Gaussian EAP with $\kappa_1 = \mu$ the mean and $\kappa_2 = \sigma^2$ the variance. Comparing to the free Gaussian propagator from equation (20) and considering it in the one dimensional case gives $\kappa_1 = 0$ and $\kappa_2 = 2Dt$. This implies that the logarithm of the diffusion signal is

$$\ln(S) = -\kappa_2 \frac{(\gamma g \delta)^2}{2} = -\gamma^2 g^2 \delta^2 \Delta D \tag{32}$$

$$\approx -bD, \tag{33}$$

Equation (33) is essentially equation (6). The SGP condition therefore yields a signal that differs from the finite pulse experiment (33) by $\delta/3$.

Hence, assuming the SGP condition, the Gaussian propagator or the free diffusion can be viewed as the low spatial frequency approximation of the EAP.

## *2.6 Diffusion Kurtosis Imaging*

The cumulant expansion allows us to naturally characterize the deviation of the diffusion from the Gaussian behaviour due to restriction by considering the higher order cumulants [18]. For example truncating the expansion after the fourth order [27] yields

$$\ln(S) = -\kappa_2 \frac{(\gamma g \delta)^2}{2} + \kappa_4 \frac{(\gamma g \delta)^4}{4!}. \tag{34}$$

By defining the *excess kurtosis*, or the fourth standardized moment as

$$K = \frac{\kappa_4}{\kappa_2^2}, \tag{35}$$

and again substituting the value of $\kappa_2$ from (20) gives

$$\kappa_4 = K\kappa_2^2 = 4KD^2\Delta^2, \tag{36}$$

which therefore yields

$$\ln(S) = -bD + \frac{4KD^2\Delta^2(\gamma g\delta)^4}{4!} \quad (37)$$

$$= -bD + \frac{1}{6}b^2D^2K. \quad (38)$$

This makes it possible to directly estimate the diffusion kurtosis by taking three DWI measurements for different $b$-values along any gradient direction (including the $b = 0$ unweighted image). This approach is, therefore, known as Diffusion Kurtosis Imaging (DKI) [18].

For anisotropic diffusion in three dimensions the diffusion coefficient $D$ is a 2nd order Cartesian tensor $\mathbf{D}$ and the kurtosis coefficient $K$ becomes a 4th order Cartesian tensor $\mathbf{K}^{(4)}$.

The kurtosis is a measure of the peakedness of the EAP and tells us if the diffusion is more sharply peaked or less sharply peaked than a Gaussian or free diffusion. Gaussian diffusion corresponds to $K = 0$. The cumulant expansion has, however, a finite radius of convergence centered around $b = 0$, and, therefore, DKI is useful at intermediate and low $b$-values.

## 3 Higher Order Tensors in D-MRI

Tensors in the form of 2nd order diffusion tensors at the core of the DTI framework allow for the inference of the medium's micro-structure analytically and quantitatively. The EAP formulation, however, generalizes the anisotropy model of the diffusion tensor by considering the Gaussian propagator as a low spatial frequency approximation. This is done from the cumulant expansion of the EAP. In DKI we see higher order cumulants being used, especially the 4th order cumulant which is a 4th order Cartesian tensor to examine the deviation of the diffusion from Gaussian or free diffusion.

The EAP formulation offers the possibility of extracting more complex geometric information of the medium's micro-structure, such as fiber crossings, by admitting more general anisotropy models. The anisotropy is often modelled using HOTs, such as in GDTI, which lend greater geometric details to the propagator with their increased multi-linearity. However, it must be noted that the GDTI models are phenomenoligical or that the order of the tensors are increased to fit the diffusion signal data more closely.

At this point it is relevant to mention D-MRI reconstruction schemes that attempt to be model-free or model-independent. The classical approach along this line is DSI. Q-Ball Imaging (QBI) is another [36]. QBI computes a function known as the Orientation Distribution Function (ODF) which is the radial integration of the EAP. No models are set for the propagator however.

### *3.1 Generalizing Fick's Laws*

Generalized Diffusion Tensor Imaging tries to model the diffusion signal containing anisotropies suspected to be more complex than Gaussian anisotropy using higher order tensors. Two GDTI models have been proposed. Both modify Fick's laws of diffusion to accomodate non-Gaussian diffusion.

Let us go back to equations (14) and (17) and rewrite them for $J$ the flux, $D$ the diffusion coefficient, and $C$ the concentration as:

$$J = -D\nabla C \tag{39}$$

$$\frac{\partial C}{\partial t} = D\nabla^2 C. \tag{40}$$

And in the case of Gaussian-anisotropic three dimensional diffusion the equations become

$$J = -D_{ij}\nabla_j C \tag{41}$$

$$\frac{\partial C}{\partial t} = \mathrm{tr}(D_{ij}\nabla^2_{ij} C), \tag{42}$$

where $D_{ij}$ are the coefficients of the 2nd order tensor $\mathbf{D}$. In equations (41) and (42), we use Einstein's notation convention and a repetition of indices, such as $D_{ij}\nabla_j C$, implies a summation over the repeated index over all its possible values $\sum_j D_{ij}\nabla_j C$.

Two generalizations to Fick's laws were proposed independently by the authors in [24] and [30]. They are sometimes referred to as GDTI-1 and GDTI-2 respectively [27].

In GDTI-1, Fick's laws in three dimensions are written as an infinite sum of products of tensors of increasing orders with partial derivatives of $C$ of increasing orders

$$\begin{aligned} J_{i_1} &= -\sum_{k=2}^{\infty}\left[D^{(k)}_{i_1 i_2 \ldots i_k}\frac{\partial^{(k-1)}C}{\partial x_{i_2}\partial x_{i_3}\ldots\partial x_{i_k}}\right] \\ &= -\sum_{k=2}^{\infty}\left[D^{(k)}_{i_1 i_2 \ldots i_k}\nabla^{(k-1)}_{i_2\ldots i_k}C\right] \end{aligned} \tag{43}$$

$$\frac{\partial C}{\partial t} = \sum_{k=2}^{\infty}\left[D^{(k)}_{i_1 i_2 \ldots i_k}\nabla^{(k)}_{i_1 i_2 \ldots i_k}C\right], \tag{44}$$

where $D^{(k)}_{i_1 i_2 \ldots i_k}$ are the coefficients of the $k$-th order three dimensional Cartesian diffusion tensor $\mathbf{D}^{(k)}$, by an abuse of notation $\nabla^{(k)}_{i_1 i_2 \ldots i_k}C$ denotes the $k$-th order partial derivatives of $C$, and we again employ Einstein's notation convention of a repetitive index to represent summation. GDTI-1 modifies the fundamental relation between the flux and the concentration in Fick's laws.

In GDTI-2, Fick's laws in three dimensions for spin bearing particles under the influence of a magnetic gradient $\mathbf{g}$ are written for a given order $k$ as

$$J = -D_{i_1 i_2 \ldots i_k} g_{i_1} g_{i_2} \ldots g_{i_k} \nabla C \tag{45}$$

$$\frac{\partial C}{\partial t} = D_{i_1 i_2 \ldots i_k} g_{i_1} g_{i_2} \ldots g_{i_k} \nabla^2 C, \tag{46}$$

where $D_{i_1 i_2 \ldots i_k}$ are the coefficients of $\mathbf{D}^{(k)}$, $g_{i_i} \ldots g_{i_k}$ are components of the gradient $\mathbf{g}$, and we use Einstein's notation convention for the indices. GDTI-2 modifies Fick's law by replacing the diffusion coefficient by a generalized diffusion function.

## *3.2 GDTI-1*

In [24] the author uses the modified Fick's law (44) in the Bloch-Torrey equation (1). Solving this yields the diffusion function

$$D(\mathbf{g}) = \sum_{k=1}^{\infty} \left[ i^{2k} D^{(2k)}_{l_1 l_2 \ldots l_{2k}} B^{(2k)}_{l_1 l_2 \ldots l_{2k}} \right], \tag{47}$$

where $i$ is the imaginary number, $B^{(k)}_{l_1 l_2 \ldots l_k}$ are the coefficients of the $k$-th order Cartesian tensor $\mathbf{B}^{(k)}$ that is the generalization of the $\mathbf{B}$-matrix (7), and we again use Einstein's summation notation. We only retain the even ordered diffusion tensors here, unlike in the authors presentation [24]. This is justified by the explanation that the even ordered tensors account for the magnitude of the signal while the odd ordered tensors (on the imaginary axis) account for the phase, and as we have seen earlier, for a pure diffusion process the signal is equal to the modulus Fourier transform of the propagator.

What makes GDTI-1 attractive is that with this form of the diffusion function and given the Fourier relation between the propagator and the signal, it is possible to reconstruct the propagator as a function of the estimated higher order diffusion tensors. This can be seen by replacing the diffusion function (47) in the linearized model of the signal, given by taking logarithms on both sides of equation (9), which yields

$$\ln\left(\frac{M}{M(0)}\right) = \sum_{k=1}^{\infty} \left[ i^{2k} D^{(2k)}_{l_1 l_2 \ldots l_{2k}} B^{(2k)}_{l_1 l_2 \ldots l_{2k}} \right]. \tag{48}$$

The cumulant expansion of the signal (31) written in the three dimensional case using Einstein's summation notation is

$$\ln(S) = \sum_{j=1}^{\infty} \left[ i^{2j} \frac{K^{(2j)}_{l_1 l_2 \ldots l_{2j}} q_{l_1} q_{l_2} \ldots q_{l_{2j}}}{2j!} \right], \tag{49}$$

where only the even ordered cumulants are non-zero, $K^{(j)}_{l_1 l_2 \ldots l_j}$ are the coefficients of the $j$-th order cumulant $\mathbf{K}^{(j)}$ which is now a $j$-th order Cartesian tensor, and

$q_{l_1} q_{l_2} \dots q_{l_{2j}}$ are the coefficients of $\mathbf{q} = \gamma\delta g\mathbf{u}$ with $\mathbf{u}$ the unit vector in the direction of the gradient pulse satisfying the SGP condition.

Comparing equations (48) and (49) reveals [24]

$$\begin{aligned} K^{(n)}_{l_1 l_2 \dots l_n} &= (-1)^n n! D^{(n)}_{l_1 l_2 \dots l_n} \left( \Delta - \frac{n-1}{n+1}\delta \right) \\ &\approx (-1)^n n! D^{(n)}_{l_1 l_2 \dots l_n} \Delta, \end{aligned} \tag{50}$$

or in other words the higher order cumulants $\mathbf{K}^{(n)}$ can be computed from the diffusion tensors of the same order $\mathbf{D}^{(n)}$ that parameterize the generalized Fick's law (46), while the diffusion tensors can be estimated from the signal or DWIs using the linear LS approach [23].

It is then possible to reconstruct the propagator as a function of the cumulants using the Gram-Charlier series [24]

$$P(\mathbf{r}) = N(0, K^{(2)}_{l_1 l_2}) \times \left( 1 + \frac{K^{(4)}_{l_1 l_2 l_3 l_4}}{4!} H_{l_1 l_2 l_3 l_4}(\mathbf{r}) + \dots \right), \tag{51}$$

where $N(0, K^{(2)}_{l_1 l_2})$ is the normal distribution with zero mean and covariance matrix $K^{(2)}_{l_1 l_2}$, and $H_{l_1 l_2 \dots l_n}(\mathbf{r})$ is the $n$-th order Hermite tensor.

If $p_{l_1 l_2}$ be the components of $N(0, K^{(2)}_{l_1 l_2})^{-1}$, and if $w_{l_1} = p_{l_1 l_2} r_{l_2}$, then [24]

$$H_{l_1 l_2 l_3 l_4}(\mathbf{r}) = w_{l_1} w_{l_2} w_{l_3} w_{l_4} - 6 w_{(l_1} w_{l_1} p_{l_3 l_4)} + 3 p_{(l_1 l_2} p_{l_3 l_4)}, \tag{52}$$

where indices within parantheses designate that the term is to be averaged over all permutations of those indices that produce different terms, remembering that $p_{l_1 l_2} = p_{l_2 l_1}$ and $w_{l_1} w_{l_2} = w_{l_2} w_{l_1}$. This gives the components of the fourth order Hermite tensor for completeness.

### *3.3 GDTI-2*

In [30] the author modifies the Bloch-Torrey equation (1) by replacing the diffusion term described by Fick's classical law by the modified diffusion law of equation (46). Solving this modified Bloch-Torrey equation yields the diffusion function of order $k$

$$D(\mathbf{g}) = \sum_{i_1=1}^{3} \sum_{i_2=1}^{3} \dots \sum_{i_k=1}^{3} D_{i_1 i_2 \dots i_k} g_{i_1} g_{i_2} \dots g_{i_k}. \tag{53}$$

which can be seen as a extension of the Gaussian diffusion function defined for the 2nd order tensor

$$D(\mathbf{g}) = \mathbf{g}^T \mathbf{D} \mathbf{g} = \sum_{i=1}^{3} \sum_{j=1}^{3} D_{ij} g_i g_j. \tag{54}$$

The $k$-th order diffusion tensor $\mathbf{D}^{(k)}$ has to be of even order and symmetric. The symmetry constraint implies that the coefficients $D_{i_1 i_2 \ldots i_k}$ are equal under any permutation of the indices, yielding

$$N(k) = (k+1)(k+2)/2 \tag{55}$$

independent coefficients. These unknowns can be estimated from the diffusion signal or DWIs in exactly the same fashion as the unknowns of DTI were estimated – by constructing the vectors $\mathbf{d}$, $\mathbf{S}$ and the matrix $\mathbf{Y}$ appropriately. One can then again use the linear Least Squares approach (13).

The estimated diffusion function $D(\mathbf{g})$ allows to approximate the diffusion signal, by extrapolating, everywhere in $\mathbf{q}$-space. The EAP has to be, therefore, computed from the diffusion function by extrapolating the diffusion signal and then by numerically computing its inverse Fourier transform (29). This does not give a closed form for the propagator. However, since the diffusion function is more complex, it allows the propagator to model complex anisotropies such as fiber crossings.

## 4 Positivity Constraints

Diffusion is a positive quantity and negative diffusion does not correspond to anything physical. This is the reason behind introducing a *positive definite* 2nd order diffusion tensor by Basser in DTI [6]. However, since the signal is often noisy it is common to estimate non-positive definite tensors using the linear LS approach. Tackling this problem revealed that $3 \times 3$ symmetric positive definite diffusion tensors belong to a Riemannian space, with a Riemannian metric which assigns an inner product to each point of this space. In fact two affine invariant metrics were proposed that rendered the space of symmetric positive definite matrices $S^+$ complete, allowing various tasks like interpolation and geodesic computation to be naturally confined to this space. These were the Riemannian metric [13, 28, 31, 22] and the Log-Euclidean metric [1, 2]. The linear LS approach uses the Euclidean metric of $\mathbb{R}^{3\text{x}3}$.

It is well known, and as we have seen, DTI, using 2nd order tensors, cannot model complex anisotropic diffusion phenomenon like fiber crossings. It is limited to a single dominant alignment direction (one major eigenvector), and in the case of crossing fibers the tensors become oblate or spherical. GDTI overcomes this shortcoming by estimating the diffusion function with HOTs. However, HOTs in GDTI are also estimated using the linearized LS approach which doesn't guarantee a positive diffusion function. The problem of guaranteeing a positive diffusion using HOTs is rendered hard by the increased multi-linearity of the tensors. Two ap-

proaches were proposed recently to tackle 4th order diffusion tensors in the GDTI-2 model. These shall be presented here.

## *4.1 Riemannian Approach*

In [15] the authors propose to extend the Riemannian framework from 2nd order tensors to the space of 4th order tensors by mapping a 4th order 3D tensor to a 2nd order 6D tensor which is a $6 \times 6$ matrix. Then they proceed to use the Riemannian framework for $S^+$ in the space $S^+(6)$ to guarantee a positive diffusion function.

A 4th order tensor is defined to be a linear transformation $\mathbf{A}^{(4)} : Lin(V) \rightarrow Lin(V)$, where $V$ is a vector space over $\mathbb{R}^n$ [29]. The double-dot-product is introduced as $\mathbf{A}^{(4)} : \mathbf{D}^{(2)} = A_{ijkl} D_{kl}$, using Einstein's summation notation, where $\mathbf{D}^{(2)}$ is a 2nd order tensor. Transpose $\mathbf{A}^{(4)T}$ is then defined as $\langle \mathbf{A}^{(4)} : \mathbf{D}^{(2)} \mid \mathbf{C}^{(2)} \rangle = \langle \mathbf{D}^{(2)} \mid \mathbf{A}^{(4)T} : \mathbf{C}^{(2)} \rangle$ using the inner-product $\langle . \mid . \rangle$ in the space of 2nd order tensors. And the Euclidean inner-product in the space of 4th order tensors is defined to be $\langle \mathbf{A}^{(4)} \mid \mathbf{B}^{(4)} \rangle = \mathrm{tr}(\mathbf{A}^{(4)T} \mathbf{B}^{(4)})$.

A 4th order tensor satisfying major and minor symmetries has 21 independent coefficients, in three dimensions, and has an eigen decomposition. If it satisfies *total* symmetry it has 15 independent coefficients. This symmetry corresponds to the symmetric HOTs in GDTI, and by replacing $k = 4$ in (55) one can arrive at the same number of independent coefficients.

A proposition [29] states that

$$\langle \mathbf{A}_s^{(4)} \mid \mathbf{B}_a^{(4)} \rangle = \mathrm{tr}(\mathbf{A}_s^{(4)} \mathbf{B}_a^{(4)}) = 0, \tag{56}$$

where $\mathbf{B}_a^{(4)}$ is the remainder or anti-symmetric part that remains when the totally symmetric part $\mathbf{B}_s^{(4)}$ of a tensor $\mathbf{B}^{(4)}$ is subtracted from itself.

When a 4th order tensor in three dimensions $\mathbf{A}^{(4,3)}$, satisfies major and minor symmetries it can be mapped to a symmetric 2nd order tensor in six dimensions $\mathbf{A}^{(2,6)}$ [4, 29]. The double-dot-product, for a symmetric 2nd order tensor $\mathbf{D}^{(2,3)}$, can be rewritten as a matrix vector product $\mathbf{A}^{(4,3)} : \mathbf{D}^{(2,3)} = \mathbf{A}^{(2,6)} \mathbf{d}^{(1,6)}$, where $\mathbf{d}^{(1,6)} = [D_{11}, D_{22}, D_{33}, \sqrt{2}D_{12}, \sqrt{2}D_{13}, \sqrt{2}D_{23}]^T$, where $D_{ij}$ are the six independent coefficients of $\mathbf{D}^{(2,3)}$. The diffusion function of GDTI-2 (53) for order 4 can then be written as

$$\begin{aligned} D(\mathbf{g}) &= \mathbf{D}^{(2,3)} : \mathbf{A}^{(4,3)} : \mathbf{D}^{(2,3)} \\ &= \mathrm{tr}(\mathbf{A}^{(4,3)} \mathbf{G}^{(4,3)}), \end{aligned} \tag{57}$$

where $\mathbf{D}^{(2,3)} = \mathbf{g} \otimes \mathbf{g}$, with $\mathbf{g}$ the gradient, $\otimes$ the outer-product, and $\mathbf{G}^{(4,3)} = \mathbf{g} \otimes \mathbf{g} \otimes \mathbf{g} \otimes \mathbf{g}$, a totally symmetric 4th order tensor. For computations the equivalent matrix formulation can be used instead

$$D(\mathbf{g}) = \mathbf{d}^{(1,6)^T} \mathbf{A}^{(2,6)} \mathbf{d}^{(1,6)}. \tag{58}$$

The 4th order diffusion tensor $\mathbf{A}^{(4,3)}$ can be estimated in $S^+(6)$ by using the Riemannian metric and an M-estimator $\Psi$, to account for outlier data, along the lines of [21]. The error energy functional that has to be minimized is

$$E(\mathbf{A}^{(2,6)}) = \sum_{i=1}^{N} \Psi \left( \frac{1}{b_i} \ln \left( \frac{M}{M(0)} \right) + \mathbf{d}_i^{(1,6)T} \mathbf{A}^{(2,6)} \mathbf{d}_i^{(1,6)} \right), \tag{59}$$

where $N$ is the number of DWIs aquired, as a non-linear gradient descent problem. The gradient of this functional using the Riemannian metric in $S^+(6)$ is

$$\nabla E = \sum_{i=1}^{N} \Psi'(r_i(\mathbf{A}^{(2,6)})) \mathbf{A}^{(2,6)} \mathbf{d}_i^{(1,6)} \left( \mathbf{A}^{(2,6)} \mathbf{d}_i^{(1,6)} \right)^T, \tag{60}$$

where $r_i(\mathbf{A}^{(2,6)}) = \frac{1}{b_i} \ln \left( \frac{M}{M(0)} \right) + \mathbf{d}_i^{(1,6)T} \mathbf{A}^{(2,6)} \mathbf{d}_i^{(1,6)}$. Since $\mathbf{A}^{(2,6)}$, is estimated in $S^+(6)$, the diffusion function (58) is guaranteed to be positive for any $\mathbf{g}$. However, since $\mathbf{A}^{(2,6)}$ is estimated in $S^+(6)$, it has 21 independent coefficients, while a 4th order diffusion tensor is totally symmetric and can have only 15. This indeterminacy can be overcome by noticing that $\mathbf{G}^{(4,3)}$ is totally symmetric, therefore

$$\begin{aligned} D(\mathbf{g}) &= \text{tr}(\mathbf{A}^{(4,3)} \mathbf{G}^{(4,3)}) \\ &= \text{tr}((\mathbf{A}_s^{(4,3)} + \mathbf{A}_a^{(4,3)}) \mathbf{G}^{(4,3)}) \\ &= \text{tr}((\mathbf{A}_s^{(4,3)} \mathbf{G}^{(4,3)}), \end{aligned} \tag{61}$$

where the last equality comes from equation (56), $\mathbf{A}_s^{(4,3)}$ contains the coefficients of the 4th order diffusion tensor and $\mathbf{A}_a^{(4,3)}$, the residue, contains the excess parameters. The symmetry constraint of $||\mathbf{A}_a^{(4,3)}|| = 0$ can, therefore, be applied by projecting $\mathbf{A}^{(4,3)}$ to its symmetric part $\mathbf{A}_s^{(4,3)}$.

## *4.2 Ternary Quartics Approach*

The authors in [3] were the first to attempt a positivity constraint on the 4th order diffusion tensor. The diffusion function of GDTI-2 (53) for order 4 was rewritten as

$$D(\mathbf{g}) = \sum_{i+j+k=4} D_{ijk} g_1^i g_2^j g_3^k, \tag{62}$$

due to the bijection that exists between a symmetric tensor of order $k$ and a homogeneous polynomial of degree $k$. In this form, the diffusion function can be seen to be a function of the three variables $g_1, g_2, g_3$, and of total degree four. It is a Ternary

Quatic. The positivity constraint on the diffusion function requires the ternary quartic to be positve for any values of the three variables. Hilbert proved in 1888 [17] that

**Theorem 1** *Every non-negative real ternary quartic form is a sum of three squares of quadratic forms.*

Using this theorem, the diffusion function (62) can be parameterized as

$$\begin{aligned} D(\mathbf{g}) &= (\mathbf{v}^T\mathbf{q}_1)^2 + (\mathbf{v}^T\mathbf{q}_2)^2 + (\mathbf{v}^T\mathbf{q}_3)^2 \\ &= \mathbf{v}^T\mathbf{Q}\mathbf{Q}^T\mathbf{v} = \mathbf{v}^T\mathbf{G}\mathbf{v}, \end{aligned} \tag{63}$$

where $\mathbf{v}$ contains the monomials consisting of the gradient coefficients $[g_1^2, g_2^2, g_3^2, g_1g_2, g_1g_3, g_2g_3]^T$, $\mathbf{Q} = [\mathbf{q}_1|\mathbf{q}_2|\mathbf{q}_3]$ is a $6\times 3$ matrix, and $\mathbf{G} = \mathbf{Q}\mathbf{Q}^T$ is the $6\times 6$ Gram Matrix which contains 18 independent coefficients, of which 15 are of the 4th order diffusion tensor. The coefficients of the diffusion tensor can be extracted from $\mathbf{G}$ using a map described in [3], and which closely resembles the map used in [4, 29].

Parameterized in this fashion, with $\mathbf{Q}$ estimated from the DWIs, $\mathbf{G}$ computed from $\mathbf{Q}$, to finally extract the coefficients of the 4th order diffusion tensor from $\mathbf{G}$, ensures that the diffusion function, a ternary quartic, is always non-negative. The energy functional to estimate $\mathbf{Q}$ from $N$ DWIs is

$$E(\mathbf{Q}) = \sum_{i=1}^{N}\left(M_i - M(0)e^{-b_i\mathbf{v}_i^T\mathbf{Q}\mathbf{Q}^T\mathbf{v}_i}\right)^2. \tag{64}$$

For any given $\mathbf{Q}$, however, it is possible to compute a whole family of $\{\mathbf{Q}'\}$s such that $\mathbf{Q}'\mathbf{Q}'^T = \mathbf{Q}\mathbf{Q}^T = \mathbf{G}$, from the group of rotation matrices $\mathbf{R}$, since they are orthogonal and, therefore, $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. This can be seen by constructing $\mathbf{Q}' = \mathbf{Q}\mathbf{R}$, since $\mathbf{Q}$ is 6*imes*3 and $\mathbf{R}$ is 3*imes*3, and computing $\mathbf{Q}'\mathbf{Q}'^T = (\mathbf{Q}\mathbf{R})(\mathbf{Q}\mathbf{R})^T = \mathbf{Q}\mathbf{Q}^T$. To reduce this infinite solution space to a finite set of solutions, $\mathbf{Q}$ is separated into its upper and lower 3*imes*3 blocks $\mathbf{A}$ and $\mathbf{B}$ and re-parameterized via a QR-decomposition of its upper block $\mathbf{A}$ as

$$\mathbf{Q} = \begin{pmatrix}\mathbf{T}\mathbf{R}\\ \mathbf{B}\end{pmatrix} = \begin{pmatrix}\mathbf{T}\\ \mathbf{C}\end{pmatrix}\mathbf{R}, \tag{65}$$

where $\mathbf{T}\mathbf{R}$ is the QR decomposition of $\mathbf{A}$, $\mathbf{T}$ is a lower trangular matrix, $\mathbf{R}$ is an orthogonal matrix, and $\mathbf{C} = \mathbf{B}\mathbf{R}^T$. Re-parameterized in this fashion $\mathbf{Q}$ now has exactly 15 independent coefficients which corresponds to the number of unknowns of the 4th order diffusion tensor, and $\mathbf{R}$ is simplified when computing $\mathbf{Q}\mathbf{Q}^T$.

The authors also proposed a distance function between two 4th order diffusion tensors $\mathbf{A}_1$ and $\mathbf{A}_2$ for spatial regularization. It is defined as

$$dist(\mathbf{A}_1, \mathbf{A}_2) = \frac{1}{4\pi}\int_{S^2}\left(D_1(\mathbf{g}) - D_2(\mathbf{g})\right)^2 d\mathbf{g}, \tag{66}$$

where $D_1(\mathbf{g})$ and $D_2(\mathbf{g})$ are the diffusion functions that correspond to $\mathbf{A}_1$ and $\mathbf{A}_2$, and the integral is over the unit sphere $S^2$. This metric is invariant to rotations and can be computed analytically as a function of the coefficients of the 4th order difference tensor $\boldsymbol{\Delta} = \mathbf{A}_1 - \mathbf{A}_2$.

## 5 Conclusion

Since its inception in the mid 1900s, Diffusion MRI has today become a state-of-the-art medical tool for probing cerebral white matter. Its strength lies in being able to infer the micro-structure of the biological tissue *non-invasively* and *in vivo* by reconstructing a 'complete' diffusion image by integrating the partial information from DWIs. This is possible due to the anisotropic phenomenon of diffusion in mediums with strongly alligned micro-structures.

Starting from the fundamental equations that describe the diffusion phenomenon, we have presented here the role played by Cartesian tensors in modelling this anisotropic diffusion and from there extracting the geometric information of the underlying tissue. DTI, using 2nd order diffusion tensors was able to model Gaussian-anisotropy and from there extract a single major diffusion direction or structural alignment from its eigen-decomposition. This allowed to trace fibers in the white matter.

However, we saw how the propagator formulation generalized on DTI by considering the Gaussian propagator to be a low order spatial approximation of the actual EAP. It was then possible to model the general EAP by employing higher order Cartesian diffusion tensors which were capable of accounting for more complex anisotropic diffusion phenomena such as fiber crossings.

In the last section we tackled the problem of the positivity constraint on the diffusion function. Diffusion being a physical phenomenon, negative diffusion does not correspond to anything. However, in the presence of noisy data straightforward linear Least Squares approximation can result in a non-positive diffusion function. This had been tackled by the Riemannian metric formulation in the case of DTI. We saw two recent methods which attempted to tackle this problem in the case of 4th order diffusion tensors using the GDTI-2 model.

Tensors play an important role in Diffusion MRI. Higher order tensors are a recent addition, but the 2nd order diffusion tensor in DTI is a cornerstone technique in the clinical scenario today. Recent work in [8, 14] have shown how it is possible to use higher order Cartesian tensors to extract the maxima from relevant spherical functions such as the Orientation Distribution Function (ODF) with great precision. Since the maxima of the ODF indicate fiber directions, extracting them correctly is of fundamental importance in tracing fiber tracts in regions where complex fiber configurations like crossings are possible.

*Acknowledgements:* We would like to thank Peter Basser, Evren Ozarslan and Maxime Descoteaux for their valuable discussions.

## References


1. V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Fast and simple calculus on tensors in the log-euclidean framework. In *Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pages 115–122, 2005.
2. Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, August 2006. PMID: 16788917.
3. Angelos Barmpoutis, Bing Jian, and Baba C. Vemuri. Symmetric positive 4th order tensors & their estimation from diffusion weighted MRI. In *Information Processing in Medical Imaging (IPMI 2007)*,Kerkrade, The Netherlands 2007.
4. Peter J. Basser and Sinisa Pajevic. Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI. *Signal Processing*, 87:220–236, 2007.
5. P.J. Basser, J. Mattiello, and D. LeBihan. Estimation of the effective self-diffusion tensor from the NMR spin echo. *Journal of Magnetic Resonance*, B(103):247–254, 1994.
6. P.J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66(1):259–267, 1994.
7. F. Bloch. Nuclear induction. *Physical Review*, 70(7-8):460–474, October 1946.
8. Luke Bloy and Ragini Verma. On computing the underlying fiber directions from the diffusion orientation distribution function. In *MICCAI (1)*, pages 1–8, 2008.
9. P. T. Callaghan. *Principles of nuclear magnetic resonance microscopy*. Oxford University Press, Oxford, 1993.
10. H. Y. Carr and E. M. Purcell. Effects of diffusion on free precession in nuclear magnetic resonance experiments. *Physical Review*, 94:630–638, May 1954.
11. C. Chefd'hotel, D. Tschumperlé, R. Deriche, and O. Faugeras. Constrained flows on matrix-valued functions : application to diffusion tensor regularization. In *Proceedings of ECCV'02*, Copenhagen, Denmark, June 2002.
12. A. Einstein. Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. *Annalen der Physik*, 322(8):549–560, 1905.
13. P.T. Fletcher and S. Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *Proc. Computer Vision Approaches to Medical Image Analysis*, Prague, May 2004.
14. A. Ghosh, E.Tsigaridas, M. Descoteaux, P. Comon, B.Mourrain, and R. Deriche. A polynomial based approach to extract the maxima of an antipodally symmetric spherical function and its application to extract fiber directions from the orientation distribution function in diffusion MRI. New York, USA, 10/09/2008 2008.
15. Aurobrata Ghosh, Maxime Descoteaux, and Rachid Deriche. Riemannian framework for estimating symmetric positive definite 4th order diffusion tensors. In *MICCAI (1)*, pages 858–865, 2008.
16. E. L. Hahn. Spin echoes. *Physical Review*, 80(4):580594, November 1950.
17. D. Hilbert. Uber die darstellung definiter formen als summe von formenquadraten. *Mathematische Annalen*, 32:342–350, 1888.
18. Jens H. Jensen, Joseph A. Helpern, Anita Ramani, Hanzhang Lu, and Kyle Kaczynski. Diffusional kurtosis imaging: The quantification of non-gaussian water diffusion by means of magnetic resonance imaging. *Magnetic Resonance in Medicine*, 53:1432–1440, 2005.
19. P.C. Lauterbur. Image formation by induced local interactions: examples employing nuclear magnetic resonance. *Nature*, 242:190–191, 1973.
20. D LeBihan and E. Breton. Imagerie de diffusion *in vivo* par résonnance magnétique nucléaire. *CR Académie des Sciences*, (301):1109–1112, 1985.
21. C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor MRI processing. *Journal of Mathematical Imaging and Vision*, 25(3):423–444, 2006.

22. Christophe Lenglet. *Geometric and Variational Methods for Diffusion Tensor MRI Processing*. PhD thesis, Universite de Nice–Sophia Antipolis, 2006.
23. C. Liu, R. Bammer, B. Acar, and M. E. Moseley. Characterizing non-gaussian diffusion by using generalized diffusion tensors. *Magnetic Resonance in Medicine*, 51:924–937, 2004.
24. Chunlei Liu, Roland Bammer, and Michael E. Moseley. Generalized diffusion tensor imaging (GDTI): A method for characterizing and imaging diffusion anisotropy caused by non-gaussian diffusion. *Israel Journal of Chemistry*, 43:145–154, 2003.
25. P. Mansfield. Multi-planar image formation using nmr spin echoes. *Journal of Physics C*, 10:55–58, 1977.
26. K.D. Merboldt, W. Hanicke, and J. Frahm. Self-diffusion nmr imaging using stimulated echoes. *J. Magn. Reson.*, 64:479–486, 1985.
27. Ludovico Minati and Wladyslaw P. Weglarz. Physical foundations, models, and methods of diffusion magnetic resonance imaging of the brain: A review. *Concepts in Magnetic Resonance Part A*, 30A(5):278–307, 2007.
28. M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.*, 26(3):735–747, April 2005.
29. Maher Moakher. The algebra and geometry of fourth-order tensors with application to diffusion MRI. In Joachim Weickert and Hans Hagen, editors, *Perspectives Workshop: Visualization and Image Processing of Tensor Fields*, number 04172, 2006.
30. Evren Ozarslan and Thomas H Mareci. Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magn Reson Med*, 50(5):955–965, Nov 2003.
31. X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. Technical Report 5255, INRIA, Sophia Antipolis, July 2004.
32. E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance absorption by nuclear magnetic moments in a solid. *Physical Review*, 69(1-2):37+, January 1946.
33. E. O. Stejskal and J. E. Tanner. Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient. *Journal of Chemical Physics*, 42(1):288–292, 1965.
34. D. G. Taylor and M. C. Bushell. The spatial mapping of translational diffusion coefficients by the nmr imaging technique. *Physics in Medicine and Biology*, 30(4):345–349, 1985.
35. H. C. Torrey. Bloch equations with diffusion terms. *Phys. Rev.*, 104(3):563–565, Nov 1956.
36. D.S. Tuch. *Diffusion MRI of complex tissue structure*. PhD thesis, Division of Health Sciences and Technology, Massachusetts Institute of Technoloy, Boston, MA, USA, 2002.
37. Z. Wang, B.C. Vemuri, Y. Chen, and T.H. Mareci. A constrained variational principle for direct estimation and smoothing of the diffusion tensor field from complex DWI. *IEEE Transactions on Medical Imaging*, 23(8):930–939, 2004.
38. V. J. Wedeen, T. G. Reese, and D.S. et al. Tuch. Mapping fiber orientation spectra in cerebral white matter with fourier-transform diffusion mr. *8th International Society for Magnetic Resonance in Medicine*, page 82, 2000.
39. C.F. Westin, S.E. Maier, H. Mamata, A. Nabavi, F.A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis*, 6(2):93–108, June 2002.

# DT-MRI Connectivity and/or Tractography?: Two New Algorithms

Burak Acar and Erdem Yörük

**Abstract** Diffusion Tensor MRI (DTI) is a special MR imaging technique where the second order symmetric diffusion tensors that are correlated with the underlying fibrous structure (eg. the nerves in brain), are computed based on Diffusion Weighted MR Images (DWI). DTI is the only in vivo imaging technique that provides information about the network of nerves in brain. The computed tensors describe the local diffusion pattern of water molecules via a 3D Gaussian distribution in space. The most common analysis and visualization technique is tractography, which is a numerical integration of the principal diffusion direction (PDD) that attempts to reconstruct fibers as streamlines. Despite its simplicity and ease of interpretation, tractography algorithms suffer from several drawbacks mainly due to ignoring the information in the underlying spatial distribution but using the PDD only. An alternative to tractography is connectivity which aims at computing probabilistic connectivity maps based on the above mentioned 3D Gaussian distribution as described by the DTI data. However, the computational cost is high and the resulting maps are usually hard to visualize and interpret. This chapter discusses these two approaches and introduces two new tractography techniques, namely the *Lattice-of-Springs (LoS)* method that exploits the connectivity approach and the *Split & Merge Tractography (SMT)* that attempts to combine the advantages of tractography and connectivity.

Burak Acar
Boğaziçi University, Electrical & Electronics Eng. Dept., Bebek, Istanbul, Turkey, e-mail: acarbu@boun.edu.tr

Erdem Yörük
Johns Hopkins University, Department of Applied Mathematics and Statistics, Baltimore, MD, USA, e-mail: yoruk@ams.jhu.edu

# 1 Introduction

Proton self-diffusion provides inherent tissue contrast in MRI, which is utilized in diffusion-weighted imaging (DWI) [1, 2]. Since the early 1990s DWI has been used successfully for the diagnostic work up of acute stroke patients [3, 4] and later also for abscesses [5] and Creutzfeld Jacob disease (CJD) [6]. In biological tissue, such as brain white matter (WM), diffusion is normally anisotropic [7]; that is, the mobility of diffusing protons is different if measured along different directions. While WM appears homogeneous on conventional imaging, its underlying tissue structure is truly more complex and comprises a complex network of axonal bundles routing electrophysiological signals from and to cortical regions, as well as basal ganglia, the cerebellum, and the periphery. A large number of pathological processes target WM structures and can have critical impact on the well-being of a patient. It has been shown that certain WM abnormalities, such as multiple sclerosis (MS) [8] or amyotrophic lateral sclerosis (ALS) [9], as well as genetically defined psychiatric disorders, such as fragile X syndrome [10, 11], or velocardiofacial syndrome [12] are paralleled by reduced diffusion anisotropy. In fact, in some of these diseases, diffusion anisotropy changes occur much earlier than any other abnormality seen in conventional imaging. Altered diffusion might therefore be a helpful factor that needs to be considered for earlier diagnosis that in turn would provide access to more advanced treatment options. In addition, the understanding how different areas of the brain connect to each other is not only of great interest for basic neuroscientists but also for clinicians to further their knowledge on how diseases and injuries might alter functional and anatomical connectivity, which ultimately could improve diagnosis and treatment.

In the past, the interrogation of WM fiber tracts could only be performed invasively and normally destroyed the tissue [13, 14, 15, 16]. Various methods have been tested in animal studies, but clinically there exists no test to image axonal fiber paths. Several years ago diffusion tensor imaging (DTI) [17, 18, 19], an improved variant of DWI, has been introduced to quantify diffusion anisotropy. In DTI, it is widely assumed that the significant diffusion anisotropy is due mostly to the restriction the myelin sheaths impose on water protons as they diffuse across axons. Conversely, diffusion along the axons demonstrates more or less no restriction. This highly anisotropic diffusion property lays the foundation of DT-MRI analysis and visualization techniques, specifically of fiber tracking [20, 21, 22]. However, it is of utmost importance to understand what the DT-MRI signal represents in order to develop adequate analysis and visualization methods and understand the shortcomings of tractography. DT-MRI measures the average signal attenuation within a small subvolume (i.e. a voxel) due to water molecules spinning out-of-phase.

The basis of MRI is to *perturb* the water molecules (the dipoles) that were aligned in a constant magnetic field ($B_0 \approx 1-7Tesla$) and let them re-orient themselves with $B_0$ during which the dipoles rotate around $B_0$ according to the Bloch's Equation. This rotation causes a temporal change in total magnetic field which induces a time-varying current at the receiving coils of the MR scanner. The time it takes for the dipoles to fully relax depends on the environment (i.e. the tissue). Thus, the amount

of current induced is proportional to the number of dipoles (i.e water concentration) and the tissue type. These current measurements are transformed to monochromatic images in which each pixel value is also a function of water concentration and tissue type [1].

In DT-MRI, extra spatially varying magnetic fields, the so called Diffusion Weighting Gradients (DWG), $G$, are applied together with $B_0$. Due to this added $G$ field, the water molecules under continuous Brownian motion experience different total magnetic field at different locations. This causes then to rotate at different frequencies, i.e. to be out-of-phase. The effect of out-of-phase dipoles on the induced current is attenuation. So, the amount of attenuation in the received signal (equivalently in the diffusion weighted MR images) is a function of the Brownian motion (i.e diffusion) of water molecules and the applied $G$ field.

This phenomenon is described by the superposition of the Bloch's Equation and the Diffusion Equation as follows:

$$\frac{\partial M}{\partial t} = \gamma M \times (B_0 + G) - \left( M - \begin{bmatrix} 0 \\ 0 \\ M_z \end{bmatrix} \right) \begin{bmatrix} \frac{1}{T_2} & 0 & 0 \\ 0 & \frac{1}{T_2} & 0 \\ 0 & 0 & \frac{1}{T_1} \end{bmatrix} - \nabla \cdot D \nabla M \qquad (1)$$

where the first term is the rotation, the second term is the relaxation and the third term is the diffusion components of the temporal change of the magnetic field, consequently of the induced current which is related to the MR image. The solution to the transverse component of $M$, i.e. $m = M_x + jM_y$, just before the imaging sequence, is given as,

$$m(TE) = m^0 exp(-\sum_{i,j} b_{ij} D_{ij}) \ , \ (i,j) \in \{x,y,z\} \qquad (2)$$

The six independent components of the symmetric diffusion tensor $D$ can be computed using at least 6 linearly independent equations of the form of Equation 2. When the diffusing water molecules are in a restricted media, such as nerve fibers, then the physical barriers (the membrane) limit the space into which the particles diffuse, thus the apparent diffusion pattern is correlated with these barriers, i.e. the structure. The underlying assumption of DT-MRI tractography is that the principal diffusion direction is tangent to the fiber.

The diffusion component of Equation 1 is the well known diffusion PDE based on Fick's Law which establishes the relation between concentration gradient and flux as $F_k = -D_{kl} \frac{\partial C}{\partial x_l}$ for $(k,l) \in \{1,2,3\}$ in 3D. However, as suggested by Liu et al. in [23], if the gaussianity assumption of the underlying diffusion process is removed, then the physical process can be characterized by using higher order diffusion tensors. This corresponds to generalizing the Fick's Law as

[1] In order to compute this for each voxel in 3D, spatial coding magnetic fields must be applied, which is equivalent to saying that $B_0$ must be a function of space. For the sake of clarity, we ignored spatially varying $B_0$ and assumed that our space consists of a single voxel.

$$F_k = -D^{(2)}_{kl}\frac{\partial C}{\partial x_l} - D^{(3)}_{klm}\frac{\partial^2 C}{\partial x_l \partial x_m} - D^{(4)}_{klmn}\frac{\partial^3 C}{\partial x_l \partial x_m \partial x_n} - \ldots \quad (3)$$

The details of Generalized Diffusion Tensor can be found in [23]. The points we would like to emphasize regarding the nature of the DT-MRI data are,

- The second order diffusion tensor is a second order approximation to the diffusion process which assumes a Gaussian distribution of diffusing particles
- The measured diffusion tensor in DT-MRI is computed using the average MR signal acquired from a subvolume (a voxel)

Consequently, the DT-MRI data provides a coarse picture of the underlying fiber structure. Not only its spatial resolution is insufficient to identify the individual fibers, but also its modeling of the diffusion process relies on the gaussianity assumption. Nevertheless, DT-MRI data does provide useful clinical information as has been shown by numerous studies in literature.

## 2 Connectivity or Tractography?

There are two basic approaches in utilizing the information DT-MRI provides: Tractography and Connectivity Mapping. The former approach relies on either the deterministic pathway tracing or probabilistic pathway sampling with a likelihood estimation. The basic tool used for deterministic tractography is numerical integration of the principal diffusion direction (PDD, the major eigenvector of the diffusion tensor) among which the most popular method is the 4th order Runge-Kutta integration [24]. Deterministic tracking is prone to cumulative errors, can not overcome the partial volume effect and disregards large part of the information embedded in the diffusion tensor (which itself is an approximation based on the gaussianity assumption of the diffusion process). Yet, these pathways are easy to compute, visualize and interpret. The probabilistic tractography methods, on the other hand, use a pathway sampling strategy and a likelihood model to score and order the sampled pathways [25, 26, 27, 28, 29, 30]. However, low probability estimations for known to exist pathways have been reported, which makes the selection of the most probable pathway questionable [31, 32]. Recently, a new approach to probabilistic tracking has been proposed [33]. Different than the previous approaches, the method has been proposed to assess the *known* pathways rather than estimating the existence of a pathway. It is composed of three steps, the likelihood scoring of pathways to select anatomically correct ones, pathway sampling and selecting the most likely pathways. Typically these methods generate a set of pathways, ordered with respect to their likelihood. This approach addresses the above mentioned problems of deterministic tractography but still fails to communicate the stochastic nature of the underlying DT-MRI data. The Connectivity Mapping approach, on the other hand, attempts to utilize the true nature of the DT-MRI data, i.e. the Gaussian diffusion process, by estimating connectivity maps rather than single pathways. They consider each and every possible connection with weights set by the dataset. Several

approaches in this group are based on some sort of Monte-Carlo simulations of the random walk model [34, 35, 36]. Related to probabilistic tracking, we should also mention the approaches based on recasting the problem to the Riemannian differential geometry framework by defining local metrics using the diffusion tensors and solving for the geodesics and/or performing DT-MRI segmentation based on tensor similarities [37, 38].

The output of deterministic or probabilistic tractography methods is composed of single pathways pretending to follow a single fiber. At points of high uncertainty (*crossing* and *kissing* fibers), the deterministic approaches choose a direction to proceed or stops tracking while the probabilistic methods explore a large space of possible pathways to select the one with the highest likelihood. On the other hand, the connectivity mapping based methods provide a scalar field of connectivity probabilities that, unlike tractography, allows branching. Although anatomically branching is not valid, presenting the DT-MRI data in this way is more loyal to the nature of the acquired data (localized Gaussian maps of diffusing particles). Thus, we can say that connectivity mapping is a more direct way of communicating the information embedded in DT-MRI data. However, it is not trivial to visualize and interpret the connectivity maps.

We will focus on two methods recently proposed for DT-MRI analysis: *The Lattice-Of-Springs (LoS)* [41] and *The Split & Merge Tractography (SMT)* [39]. The LoS method is based on a lattice-of-springs system that models the local connectivity based on diffusion tensors and allows us to query for connectivity between multiple seed points simultaneously. It is essentially an efficient diffusion simulation governed by the defined tensor similarity (local connectivity). As such, LoS provides a framework that can be used with different tensor similarity/connectivity definitions. The SMT, on the other hand, is an attempt to combine the advantages of tractography and connectivity. The pathways are defined as groups of local and deterministically computed short tracks. A critical property of SMT is that it allows for branching of pathways at points of uncertainty, thus representing the true information in DT-MRI data.

## 3 Lattice-of-Springs (LoS)

### *3.1 Method*

Before going into details of LoS method, let us first layout, how we interpret connectivity. We will consider connectivity in white matter, as a positive function defined on the volume, that is always associated to a point of reference, which we call the “seed” or “source”. Intuitively, connectivity should decrease with distance from the seed while revealing the neural pattern underneath, that is, it should decay less along the fibers than across them. With these design objectives at hand, we propose

a physical "Lattice of Springs" model, as an intuitive tool to obtain white matter connectivity maps from DTI data, with respect to user defined seeds.

To better explain our model and our motivation, we will make use of graph notation. As in Markov Random Fields applied on images, we will consider an undirected graph $G = (V,E)$ such that the vertex set $V = \{1,2,...,n\}$ corresponds to the lattice of $n$ voxels (where we will use words vertex, node and voxel interchangeably), and the edge set $E$ is composed of links $(ij)$ between adjacent vertices $i$ and $j \in N_i$, where $N_i$ is the 6-neighborhood of $i$.

As in Figure 1, we imagine a mechanical system embedded in graph $G$, where at each node $i \in V$, there is a virtual object, which can physically move within its virtual framework, perpendicular to some fixed reference, the so called ground. We assume that the object at $i$ is connected to the ground with a ground spring of stiffness $K_i$, and to each neighboring object of $j \in N_i$ with a corresponding neighbor spring of stiffness $K_{ij}$. The overall system is arranged such that, depending on the vertical positions $x_i$ and $x_j$ of adjacent objects at $i$ and $j$, we have elongations $x_i$ and $|x_i - x_j|$, at the ground spring of $i$ and at the neighbor spring between $i$ and $j$, respectively.

Given the configuration $\mathbf{x}$, an $n$-dimensional vector formed by $x_i$'s ($i \in V$), the total energy stored in the lattice of springs is given by

$$U(\mathbf{x}) = \frac{1}{2}\left[\sum_{i\in V} K_i x_i^2 + \sum_{(ij)\in E} K_{ij}(x_i - x_j)^2\right].$$

Now, suppose $s \in V$ is the seed node, we assume $x_s \equiv 1$, which is kept constant. Then, with appropriate choices for spring constants, we will take

$$\mathbf{x}_s = \arg\min U(\mathbf{x})\big|_{x_s=1}$$

as the connectivity of the volume with respect to seed $s$. That is, the $i^{th}$ entry ($i \in V$) in the $U$'s minimizer subject to the seed constraint $x_s = 1$, will give how strong voxel $i$ is connected to seed $s$. The configuration $\mathbf{x}_s$ will be the state where there is equilibrium in the system and to avoid possible confusion with other definitions of connectivity in the literature, we will call it *xMap*, in the following sections.

Note that, with $U$ containing an energy term for each possible clique $c \in \{\{i\},\{i,j\} : i \in V,(ij) \in E\}$ in $G$, $\mathbf{x}_s$ will also correspond to the most likely configuration given the seed constraint, if one models the conditional random vector $(\mathbf{X}|X_s = 1)$ to be a Gibbs random field with $U$'s clique potentials.

To solve this constrained minimization problem, we introduce a Lagrange multiplier $\lambda$, to write

$$\widetilde{U}(\mathbf{x},\lambda) = \frac{1}{2}\left[\sum_{i\in V} K_i x_i^2 + \sum_{(ij)\in E} K_{ij}(x_i - x_j)^2\right] + \lambda(x_s - 1).$$

Then we get

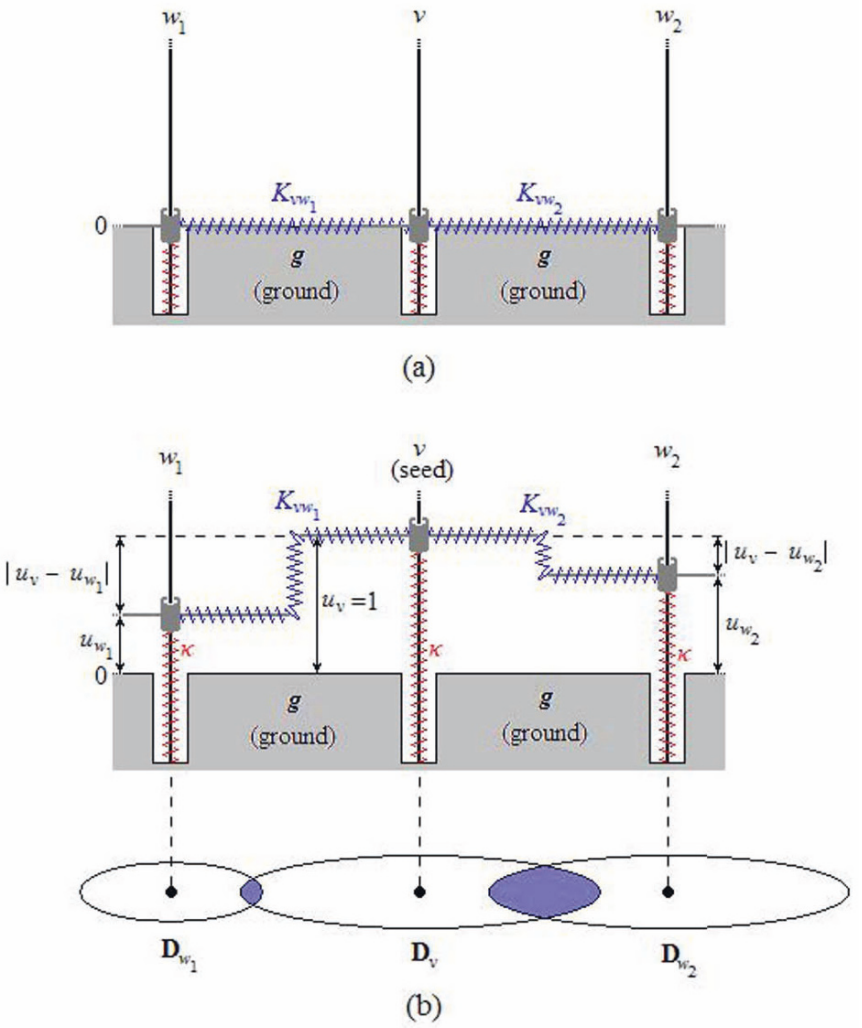



**Fig. 1** 1D model of the proposed system. a) The undisturbed system with all node potentials being zero. b) The seed node's potential is fixed at 1, which is distributed to other nodes through the interconnecting springs. Higher spring constants are assigned to higher local connectivity links, which is proportional to the local interaction of two diffusion tensors (the shaded regions). More stiff springs transport energy more effectively, which causes the nodes that are connected to the seed(s) via stiff springs, have higher node potentials (eg. $w_2$ vs. $w_1$). The node potentials are interpreted as the connectivity maps.

$$(\mathbf{x}_s, \lambda_s)^T = \arg\min \widetilde{U}(\mathbf{x}, \lambda) = \mathbf{A}_s^{-1}\mathbf{b}$$

where $(n+1)\times(n+1)$ symmetric matrix $\mathbf{A}_s$ has entries

$$a_{ij} = \begin{cases} K_i + \sum_{k \in N_i} K_{ik}, & \text{if } i = j \leq n; \\ -K_{ij}, & \text{if } (ij) \in E; \\ 1, & \text{if } (i,j) = (s, n+1) \text{ or } (i,j) = (n+1, s); \\ 0, & \text{otherwise.} \end{cases}$$

and $\mathbf{b}$ is $n+1$-dimensional column vector with

$$b_i = \begin{cases} 1, & \text{if } i = n+1; \\ 0, & \text{otherwise.} \end{cases}$$

However, For DTI data, number of voxels $n$ can be very large, so that one would prefer numerical approximation instead of a huge matrix inversion. As a positive sum of convex functions, $U$ is convex, so subject to seed condition it will have a unique global minimum which can be achieved by gradient descent, namely by iterating $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} - \alpha^{(t)} \nabla U|_{\mathbf{x}^{(t-1)}}$ with an appropriate step size $\alpha^{(t)}$ and $x_s^{(t)} = 1$ for all steps $t$. Moreover, since the minimum is global, one can replace $\alpha \nabla U$ by any vector $\mathbf{v}$ of sufficiently small norm, provided that $\mathbf{v}^T \nabla U > 0$. In other words, one always decreases $U$, if one moves from the current point in a direction which makes a negative dot product with the current gradient. If we take $\mathbf{v}$ such that

$$v_i = \frac{\frac{\partial U}{\partial x_i}\big|_{\mathbf{x}^{(t-1)}}}{K_i + \sum_{j \in N_i} K_{ij}},$$

then one descent iteration at voxel $i \in V$ becomes

$$x_i^{(t)} = \begin{cases} 1, & \text{if } i = s; \\ \frac{\sum_{j \in N_i} K_{ij} x_j^{(t-1)}}{K_i + \sum_{j \in N_i} K_{ij}}, & \text{otherwise.} \end{cases}$$

which locally minimizes $U$ in the neighborhood of $i$, where the total spring force $K_i x_i + \sum_{j \in N_i} K_{ij}(x_i - x_j)$ vanishes.

In order to obtain anatomically meaningful connectivity maps, one should define spring constants in accordance with DTI data. Especially, for neighbor springs, we want constants $K_{ij}$ ($(ij) \in E$) to be proportional to the amount of neural interaction between adjacent points $\mu_i$ and $\mu_j$ of white matter. We will quantify this interaction by inducing it from diffusion tensors and what they measure.

Recall that, the diffusion tensor $D_i \in \mathbb{R}^{3\times 3}$ at voxel $i \in V$, with its center $\mu_i \in \mathbb{R}^3$, identifies the probability density

$$f_i(y) = \frac{1}{(2\pi)^{\frac{3}{2}} |D_i|} \exp\left\{ -\frac{1}{2}(y - \mu_i)^T D_i^{-1} (y - \mu_i) \right\}$$

of the position $Y_i \in \mathbb{R}^3$ of a Brownian particle, which has been traveling from $\mu_i$ for a small period of time that is implicit in $D_i$. Then, $f_{ij}(\delta) = \int_{\mathbb{R}^3} f_i(y) f_j(y+\delta) dy$ will be the probability density of the spatial difference $\Delta = Y_i - Y_j$ between such two particles, simultaneously and independently diffusing from points $\mu_i$ and $\mu_j$

with respective tensors $D_i$ and $D_j$. We will simply set $K_{ij} = f_{ij}(0)$ ($(ij) \in E$), as a measure of how likely it is, that these two particles end up in the same vicinity to interact with each other.

According to this definition, $K_{ij}$ is the overlap integral between the densities $f_i$ and $f_j$, and can be evaluated by Laplace's method. Note that, the integrand $h(y) = f_i(y)f_j(y)$ has a local maximum at $\mu_i$ (as well as at $\mu_j$), and so does its logarithm. Thus, $\log h(y)$ has a vanishing gradient at $\mu_i$, and moreover, with Gaussian densities $f_i$ and $f_j$, it is a quadratic polynomial in $y$'s components, such that all of its third or higher order partial derivatives are zero, as well. Therefore, $\log h(y)$'s Taylor expansion around $\mu_i$, will only have zero and second order terms, which after taking back the exponential, yields a much simpler integrand, namely multiple of another Gaussian, that can now be integrated exactly. In this way, we define the spring constant for edge $(ij) \in E$ to be

$$K_{ij} = \frac{1}{(2\pi)^{\frac{3}{2}}|D_i + D_j|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(\mu_i - \mu_j)^T (D_i + D_j)^{-1}(\mu_i - \mu_j) \right\}$$

as a function of $D_i$ and $D_i$ of neighboring voxels $i$ and $j$, located at respective volume points $\mu_i$ and $\mu_j$. On the other hand, for ground springs we set the stiffness $K_i = \kappa$ to be constant for all voxels $i \in V$.

Now, with these spring definitions, spring energy $U$ is more concentrated at edges $(ij) \in E$, which are aligned with principal directions of anisotropic tensors, i.e. with fiber bundles, than those between isotropic tensors, especially with small overlap. Hence, after minimizing $U$, we will observe less decay in the $xMap$ along fibrous structures, so that given the seed, its anatomic connections are revealed.

Suppose, we want to find connectivity with respect to a region rather than a single point, then we can specify the seed to be a set of voxels, and not just one of them. In this general case, the only change in the method will be specifying $s$ as a set, introducing multiple seed constraints $x_k = 1$ $k \in s$, and hence multiple Lagrange multipliers during constrained minimization, whereas the descent approach remains the same.

## *3.2 Results*

The experiments were conducted on DT-MRI data of a healthy individual and brain tumor patient to demonstrate its correlation with known anatomy and tumor. The real patient data set were acquired by single-shot EPI scans with diffusion encoding along 12 non-collinear directions plus one reference without diffusion-weighting. The FOV was 25-26cm, TE was minimum with partial k-space acquisition. TR was $\sim 10s$ and b-value was $\sim 800 - 850s/mm^2$. The computed connectivity maps ($xMaps$) are all shown with a color scale ranging from 0 (blue) to 1 (red). The tensor fields were linearly interpolated along the vertical direction for isotropic sampling and smoothed with a Gaussian kernel of variance 1 voxel for noise suppression.

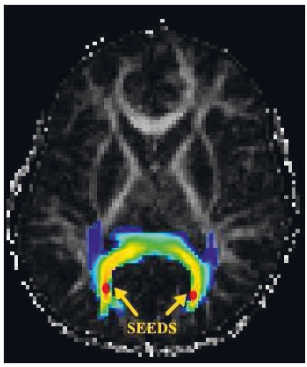
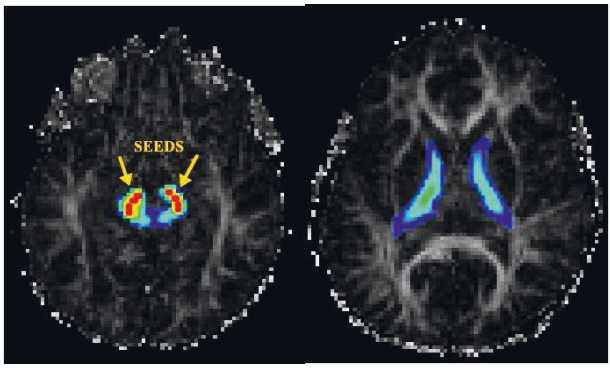



**Fig. 2** The connectivity map (*xMap*) associated with two seed points, marked by red dots, on the corpus callosum is overlaid with the FA map. The *xMap* clearly marks the most likely connection, which follows the ridge of the *xMap* and it agrees with the known anatomy.

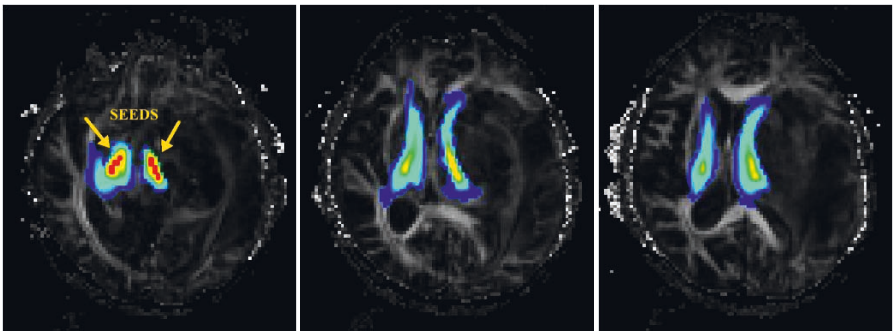



(a) (b)

**Fig. 3** The connectivity map (*xMap*) associated with two seed regions, marked by red dots, at the inferior part of the cortico-spinal tracks demonstrates a spread of connectivity as well as marking a single path (the ridge of *xMap*) as the most likely path. These are in agreement with the known anatomy of pyramidal tracks.

Figure 2 shows the *xMap* for two seed points on left and right sides of the corpus callosum / optic radiation of a healthy individual, on the same axial slice. The *xMap* follows the known anatomy and clearly marks the connection between these two seed points. The ridge of the *xMap* (i.e. the path of the least change on *xMap*) marks the path between the seed points. This experiment also demonstrates the use of multiple seed points in LoS framework.

Figure 3 shows the computed *xMap* for two seed regions selected on the same axial slice at the inferior part of the cortico-spinal tracks of the same healthy individual, both on the left and right sides. The cortico-spinal tracks are known to spread




(a) (b) (c)

**Fig. 4** (a) The axial slice with two seed regions on the cortico-spinal tracks, marked by red dots, (b,c) Axial slices superior to the seed positions, marking the cortical-spinal tracks. The mass effect of the tumor in the left hemisphere causes the visible fiber displacement.

as they extend to the superior regions. The *xMap* clearly shows this spread in the superior axial slices.

Figure 4 shows the *xMap* for two seed regions selected on the cortico-spinal tracks on the same axial slice. The patient has a brain tumor in the left hemisphere. The mass effect of the tumor has caused a midline shift and markedly displaces the left cortico-spinal track bundle. The right cortico-spinal track bundle, on the other hand, is only moderately affected by this mass effect. The fiber displacement caused by the mass effect is clearly reflected by the *xMap*. It also shows that the tumor, at least on a macroscopic level, is only displacing the cortico-spinal track rather than diffusely infiltrating it.

## 4 Split and Merge Tractography (SMT)

Split & Merge Tractography (SMT) aims at combining the ease of interpretation of the tractography approach with the theoretical soundness (and loyalty to data) of the connectivity approach [39, 40, 42]. The fiber tracks are modeled as clusters of short, local tracks and the Markov Chain Monte Carlo (the MCMC) approach is used to estimate the unknown distribution of the clusters of short fiber tracks, where clusters represent full pathways. Unlike previously proposed methods that exploit the stochastic nature of DT-MRI data for tractography [28, 33, 43], the output of SMT is not the full tracks but rather clusters of short tracks. Reconstruction of the full tracks is left to the users' visual interpretation. Thus, SMT provides a tool to explore the DT-MRI data. The underlying rationale behind using short tracks is to avoid the error accumulation during tracking and the MCMC approach is used to exploit the stochastic nature of the data itself.

### 4.1 Method

The short tracks are computed by the numerical integration of the PDD field using the fourth order Runge-Kutta method [24]. SMT avoids such error accumulation in PDD tracking by using short tracks. PDD tracking is started from each voxel unless that voxel is on a previously computed short tract. The maximum length of the short tracks is set to $2.8mm$, the tracking is terminated when the Fractional Anisotropy is below 0.25 or the curvature exceeds $20^\circ$ per step. This is the *splitting* step where the whole brain is populated with short tracks. Since these fiber tracks are of limited length, the error does not accumulate during their computation. The merging part of SMT groups these short tracks into clusters based on their interconnectivity. The groupings are dynamic and varies with a confidence threshold.

The *merging* step is composed of estimating a co-occurrence matrix, $M$, for this abundant set of short tracks. The element $M_{ij}$ of $M$, represents the probability of having the short tracks $S_i$ and $S_j$ connected. $M$ matrix is estimated using the Metropolis-Hastings algorithm (MHA) [44].

Let $\Gamma_i$ be a cluster of short tracks that includes $S_i$. $\Gamma_i$ represents a full fiber. Then, SMT aims at estimating

$$M_{ji} = P(S_j \in \Gamma_i | S_i \in \Gamma_i) \; , M_{ij} = M_{ji} \; , i, j = 1, \cdots, N \tag{4}$$

where $N$ is the total number of short tracks that populates the complete brain.

Let $S_i^{(k)}; i = 1, \cdots, N; k = 1, 2$ represent the $k^{th}$ endpoint of $S_i$, without any specific ordering of endpoints. For a short track $S_i$, a bridge is built between $S_i^{(e)}$ and the $S_j^{(s)}$ with the highest probability of being connected to $S_i^{(e)}$. If we denote the position of $S_i^{(s)}$ with $r_i^{(s)}$ and the diffusion tensor at that position with $D_i^{(e)}$, then the probability of bridging $r_i^{(e)}$ and $r_j^{(s)}$ can be approximated by

$$\begin{aligned} c_{i \to j} = P(r_i^{(e)}, r_j^{(s)}) \;\; &= \;\; (\frac{1}{4\pi\tilde{D}})^{\frac{3}{2}} exp\left( \frac{-\|r_i^{(e)} - r_j^{(s)}\|^2}{4\pi\tilde{D}} \right) \\ \tilde{D} &= \frac{(r_i^{(e)} - r_j^{(s)})^T D_i^{(e)} (r_i^{(e)} - r_j^{(s)})}{\|r_i^{(e)} - r_j^{(s)}\|^2} \end{aligned} \tag{5}$$

This is the Gaussian distribution as represented by $D_i^{(e)}$. Without loss of generality, let all bridges originate from the first end-point ($k = e$) and terminate in the second end point ($k = s$). We repeat the whole process starting from $r_j^{(e)}$, until no bridge with high enough (an arbitrarily small threshold, $\varepsilon$) probability can be built. This completes forward clustering. The whole process is repeated for backward clustering starting from the other end-point of $S_i$. Thus, we get the initial cluster for a given $S_i$. Figure 5 depicts bridging from $S_i$ to $S_j$.

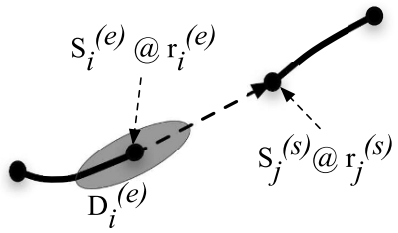



**Fig. 5** A bridge is built from the current short track $S_i$ (initially a seed tract) to $S_j$, which is selected based on the Gaussian PDF described by $D_i^{(e)}$.

This initial cluster is a sample from the distribution of all clusters that include $S_i$. Let us denote this cluster by $\Gamma_i^0$. $M_{ij}$ and $M_{ji}$ are incremented by one for all $j$ such that $S_j \in \Gamma_i^0$. The whole process is iterated $K$ times, generating $\{\Gamma_i^0, ..., \Gamma_i^{K-1}\}$. Consecutive iterations are performed by breaking the weakest bridge of the current cluster, building a new one originating from the end-point of the retained section of the broken cluster and completing the rest of forward clustering as explained above. SMT targets to estimate the distribution of these clusters for each $S_i$ as the seed tract. In other words, SMT approximates the probability distribution function (PDF) of the connectivity of $S_i$ to all other short tracks. Connectivity between $S_i$ and $S_j$ is conjectured to be proportional to the probability of the existence of a cluster that includes them both. The $i^{th}$ row of $M$ corresponds to the histogram approximating this PDF. The complete $M$ matrix is thus the collection of histograms representing the connectivity of each one of the short tracks. During the iterations, each new cluster is accepted/rejected using the Metropolis-Hastings algorithm (MHA)[44].

The components of MHA are *i)* a sampling strategy, *ii)* a sample fitness function, $f(.)$, *iii)* a candidate generating density, $q(.,.)$, which is the probability of generating a new sample from a given sample. Let $\Gamma_i^{(m)}$ denote the $m^{th}$ sample selected from the space of clusters that include $S_i$. The corresponding SMT components are as follows:

1. Sampling Strategy: Given a cluster of short tracks, $\Gamma_i^{(m)}$, the weakest bridge between two short tracks is identified. The strength of a bridge between $r_p^{(e)}$ and $r_q^{(s)}$ is represented by the *Fractional Anisotropy* (FA) of $D_p^{(e)}$ because the reliability of PDD tracking decreases with decreasing FA. Recall that forward and backward clustering is based on bridging the neighbouring short tracks with the highest connectedness probability which is analogous to conventional PDD tracking. Let us denote the FA at $D_p^{(e)}$ with $F_p^{(e)}$. Removing the weakest bridge, the section of $\Gamma_i^{(m)}$ that includes $S_i$, is retained. A new bridge between $r_p^{(e)}$ and one of its neighbours is built at random and a new cluster is formed beyond the new bridge. The new bridge is constrained to have a curvature angle less than 90°. Let the new bridge be built between $r_p^{(e)}$ and $r_w^{(s)}$.
2. Sample (Short Tract Cluster) Fitness: The fitness of a sample $\Gamma_i^{(m)}$, i.e. $f(\Gamma_i^{(m)})$ is chosen to be the minimum of the strengths of its bridges because a cluster's reliability is dominated by its weakest bridge.

3. Candidate Generating Density: Probability of generating a new sample candidate cluster from a given one is formulated as the product of the probability of removing the weakest bridge and building a new one. It is given as,

$$q(\Gamma_i^{(m)},\Gamma_i^{(m+1)}) = \underbrace{\frac{1/F_p^{(e)}}{\sum_{j\in A} 1/F_j^{(e)}}}_{\text{Prob. of removing a bridge}} \times \underbrace{\frac{c_{p\to w}}{\sum_{z\in B} c_{p\to z}}}_{\text{Prob. of building a bridge}} \tag{6}$$

where $F_p^{(e)}$ is the fitness of the removed bridge, $c_{p\to w}$ is the probability of the newly built bridge originating from $r_p^{(e)}$, $A$ is the set of short track indices that belong to $\Gamma_i^{(m)}$ and $B$ is the set of plausible short track indices that are in the neighbourhood of $r_p^{(e)}$.

For a given seed track $S_i$, the MHA is iterated. The newly generated sample at each iteration is accepted with a probability given as [44],

$$\alpha(\Gamma_i^{(m)},\Gamma_i^{(m+1)}) = \min\left(1, \frac{f(\Gamma_i^{(m+1)})q(\Gamma_i^{(m)},\Gamma_i^{(m+1)}))}{f(\Gamma_i^{(m)})q(\Gamma_i^{(m+1)},\Gamma_i^{(m)})}\right) \tag{7}$$

If $\Gamma_i^{(m+1)}$ is accepted, then we increment $M_{in}, M_{ni} \quad \forall S_n \in \Gamma_i^{(m+1)}$, otherwise, we increment $M_{in}, M_{ni} \quad \forall S_n \in \Gamma_i^{(m)}$ by one. The number is iterations, $K$, is arbitrarily set to be 100. The whole process repeated to build $M$ by taking each short track as the seed tract.

The co-occurrence matrix $M$ is computed and saved off-line. It represents the whole brain connectivity. The user is required to select a volume of interest to mark a set of *seed tracks* and a confidence threshold, $\tau$. For each seed track $S_i$ in the volume interest, all $S_j$'s with $M_{ij} \geq \tau \times max(M_{ij})$ , $0 \leq \tau \leq 1$, are selected and displayed. The interface is similar to the dynamic queries interface proposed in [45].

The downside of the above scheme is that when the weakest bridge is broken and rebuilt at random, its strength remains unaffected because the strength of a bridge is modeled by the FA of the diffusion tensor at the retained end of the broken cluster. A better approach would be to consider the diffusion probability between the connected endpoints, ie $c_{i\to j}$. Then the probability of removing a bridge would be modeled as

$$\text{Prob. of removing the bridge between } S_i \& S_j = \frac{c_{i\to j}}{\sum_{k\in A} c_{k\to k+1}} \tag{8}$$

where $A$ is the set of short track indices in the current cluster indexed by $k$ and ordered along the underlying fiber.

The SMT method proposes a framework to find a compromise between the connectivity analysis approach and the fiber tractography approach, both of which have their complementary advantages and disadvantages. The models proposed for the current SMT implementation shows the feasibility of the approach. Research on

different models for MHA, such as the model for bridge removal mentioned above, is in progress.

### *4.2 Results*

Real patient DT-MRI data, explained in Section 3.2 was used for the initial validation of the SMT method. The method was implemented within the *DTInteract*[2] together with the conventional fiber tractography using fourth order Runge-Kutta (RK4) [24]. The initial evaluation is based on visual comparison of SMT tracks and the RK4 tracks for identical volume of interests (VOIs) as shown in Figure 6.

Seed tracks are selected with a cubic volume of interest (VOI) on the left side of the corpus callosum / optic radiation of a healthy individual, as marked in Figures 6a and 6b. VOIs are identical for both images on the same line. low and high confidence thresholds, $\tau$, are used in Figures 6a and 6b respectively. Note the decrease of the number of short tracks with increasing confidence.

A second set of seed tracks are selected with a cubic VOI at the inferior part of the cortico-spinal tracks of the same healthy individual as shown in Figures 6*c* and 6d. The VOI covers both the left and the right sides. The cortico-spinal tracks are known to spread as they extend to the superior regions. The confidence threshold, $\tau$, is lower in Figure 6c. In addition to the interactive exploration opportunity provided by $\tau$, we can also observe the branching that SMT allows in these figures.

The final set of seed tracks are selected with a cubic VOI in the inferior longitudinal fasciculus region, close to the uncinate fasciculus, as marked in Figures 6e and 6f. Again, the confidence threshold, $\tau$, is lower for Figure 6e.

Computation of the short tracks (approximately 38000 short tracks for the current dataset) throughout the brain and the co-occurrence matrix $M$ takes approximately 8 minutes on a PC with Intel Core 2 Quad CPU ($2.66GHz$) and $3.25GB$ RAM. This computation is performed once for each dataset in batch mode and $M$ is saved. Visualization and analysis of the data based on the computed $M$ is real-time.

## 5 Concluding Remarks

DT-MRI is the widely used medical imaging modality that allows researchers and clinicians to study the fiber network in brain in vivo. As such, it is of great importance in studies human brain pathways in vivo and does not have an alternative. However, the processing, visualization and interpretation of the 3D tensor field generated by DT-MRI is not trivial and may easily lead to wrong interpretations. Specif-

[2] *DTInteract* is a DTI analysis and visualization platform developed at Boğaziçi University, Electrical & Electronics Engineering Department, VAVlab, Istanbul, Turkey. *DTInteract* is developed using the VAVlab's C++ visualization and analysis platform *VAVframe*. For more information contact http://www.vavlab.ee.boun.edu.tr.

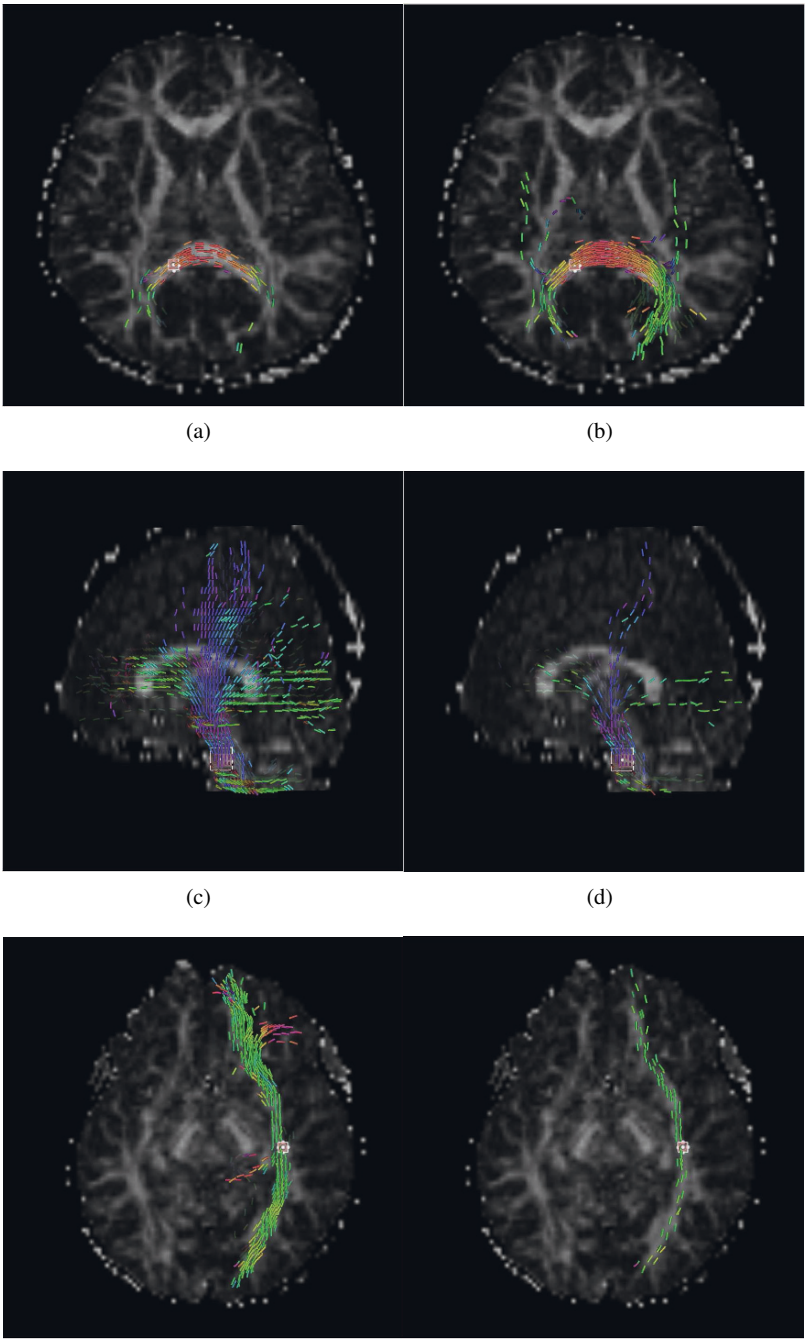



**Fig. 6** Three sets of seed tracks, one for each row, in different regions of the brain of a healthy human are selected and the corresponding short track clusters with different confidence levels (low for the left images, high for the right images) are displayed. Color codes the orientation of short tracks.

ically, the users of DT-MRI should keep in mind that the technique does not image the fibers per se but rather records the stochastic anisotropic (restricted media) diffusion process. Consequently, the data represents the spatial distribution of diffusing particles in a finite volume (the voxels), which should be larger than the fiber thickness. This restriction leads to a fundamental resolution problem and makes the popular fiber tractography techniques unreliable especially at problematic points such as crossing/kissing fibers.

Our major point is that a preferable DT-MRI analysis and visualization technique should be loyal to the data and be able to communicate the information embedded, directly to the researchers/clinicians. This approach calls for probabilistic tractography and/or connectivity approaches. The LoS method presented proposes a physical system of springs to model the connectivity and generates connectivity maps for a given set of seed points simultaneously. The SMT method, on the other hand, combines the advantages of conventional tractography and probabilistic connectivity analysis, that is the easy interpretation and visualization of tractography with the stochastic modeling. SMT allows tractography with branching that represents the stochastic nature of data and thus can communicate the information content directly to the researchers/clinicians.

**Acknowledgements** This research was in part supported by EU FP6 Programme, SIMILAR NoE and by TÜBİTAK KARİYER-DRESS (104E035) project research grants. We would like to thank Dr. Roland Bammer from Stanford University for his support in data acquisition and contributions to the application areas of DT-MRI and to Sıla Girgin for her contributions in the development of *DTInteract* platform with the *VAVframe* software framework and in DT-MRI visualization.

## References

1. Stejskal E.O., Tanner J.E. (1965) Spin diffusion measurements:spin echoes in the presence of a time dependent field gradient. J Chem Phys 42:288-292.
2. Le Bihan D. (1995) Diffusion, perfusion and functional magnetic resonance imaging. J Mal Vasc 20:203-214.
3. Moseley M.E., Cohen Y., Mintorovitch J. et al. (1990) Early detection of regional cerebral ischemia in cats: comparison of diffusion- and T2-weighted MRI and spectroscopy. Magn Reson Med 14:330-346.
4. Lansberg M.G., Norbash A.M., Marks M.P. et al. (2000) Advantages of adding diffusion-weighted magnetic resonance imaging to conventional magnetic resonance imaging for evaluating acute stroke. Arch Neurol 57:1311-1316.
5. Guo A.C., Provenzale J.M., Cruz L.C. et al. (2001) Cerebral abscesses: investigation using apparent diffusion coefficient maps. Neuroradiology 43:370-374.
6. Young G.S., Geschwind M.D., Fischbein N.J. et al. (2005) Diffusion-weighted and fluid-attenuated inversion recovery imaging in Creutzfeldt-Jakob disease: high sensitivity and specificity for diagnosis. Am J Neuroradiol 26:1551-1562.
7. Moseley M.E., Kucharczyk J., Asgari H.S. et al. (1991) Anisotropy in diffusion-weighted MRI. Magn Reson Med 19:321-326.
8. Bammer R., Fazekas F. (2002) Diffusion imaging in multiple sclerosi. Neuroimaging Clin N Am 12:71-106.
9. Ellis C.M., Simmons A., Jones D.K. et al. (1999) Diffusion tensor MRI assesses corticospinal tract damage in ALS. Neurology 53:1051-1058

10. Barnea-Goraly N., Eliez S., Hedeus M. et al. (2003) White matter tract alterations in fragile X syndrome: preliminary evidence from diffusion tensor imaging. American Journal of Medical Genetics. Part B, Neuropsychiatric genetics 118:81-88.
11. Barnea-Goraly N., Kwon H., Menon V. et al. (2004) White matter structure in autism: preliminary evidence from diffusion tensor imaging. Biological psychiatry 55:323-326.
12. Barnea-Goraly N., Eliez S., Menon V. et al. (2005) Arithmetic ability and parietal alterations: a diffusion tensor imaging study in velocardiofacial syndrome. Brain Res Cogn Brain Res. 25:735-740.
13. Pribam K., MacLean P. (1953) Neuronographic analysis of medial and basal cerebral corte. J Neurophysiol 16:324-340.
14. Whitlock D.G., Nauta W.J.H. (1956) Subcortical projections from temporal neocortex in Macaca mulatto. J Comp Neurol 106:183-212.
15. Turner B.H., Mishkin M., Knapp M. (1980) Organization of the amygdalopetal projections from modality-specific cortical association areas in the monkey. J Comp Neurol 191:515-543.
16. Yagishita A., Nakano I., Oda M. et al. (1994) Location of the corticospinal tract in the internal capsule at MR imaging. Radiology 191:455-460.
17. Basser P.J. (1995) Inferring microstructural features and the physiological state of tissues from diffusion-weighted images. NMR Biomed 8:333-344.
18. Basser P.J., Mattiello J., LeBihan D. (1994) Estimation of the effective self-diffusion tensor from the NMR spin echo. J Magn Reson B 103:247-254.
19. Basser P.J., Pierpaoli C. (1996) Microstructural and physiological features of tissues elucidated by quantitative diffusion tensor MRI. J Magn Reson B 111:209-219.
20. Basser P.J., Pajevic S., Pierpaoli C. et al. (2000) In vivo fiber tractography using DT-MRI data. Magn Reson Med 44:625-632.
21. Conturo T.E., Lori N.F., Cull T.S. et al. (1999) Tracking neuronal fiber pathways in the living human brain. Proc Natl Acad Sci USA 96:10422-10427.
22. Jones D.K., Simmons A., Williams S.C. et al. (1999) Non-invasive assessment of axonal fiber connectivity in the human brain via diffusion tensor MRI. Magn Reson Med 42:37-41
23. Liu C., Bammmer R., Acar B. et al (2004) Characterizing non-Gaussian diffusion by using generalized diffusion tensors. Magnetic Resonance in Medicine 51:924-937
24. Tench C.R., Morgan P.S., Wilson M. et al. (2002) White matter mapping using diffusion tensor MRI. Magnetic Resonance in Medicine 47:967-972
25. Behrens T.E., Johansen-Berg H., Woolrich M.W., et al (2003) Non-invasive mapping of connections between human Thalamus and Cortex using diffusion imaging. Nature Neuroscience 6:750-757
26. Bjornemo M., Brun A., Kikinis R. et al (2002) Regularized stochastic white matter tractography using diffusion tensor MRI. In: Lecture Notes in Computer Science, 2488:435-442, Springer-Verlag (Proceedings of MICCAI 2002, Tokyo, Japan)
27. Friman O., Westin C.F. (2005) Uncertainty in white matter fiber tractography In: Lecture Notes in Computer Science, 3749:107-114, Springer-Verlag (Proceedings of MICCAI 2005, Palm Springs, CA, USA)
28. Hagmann P., Thiran J.P., Jonasson L. et al (2003) DTI mapping of human brain connectivity: statistical fibre tracking and virtual dissection. Neuroimage 19:545-554
29. Lazar M, Alexander A.L. (2002) White matter tractography using random vector (RAVE) perturbation. In: Proceedings of ISMRM Annual Meeting, Honolulu, HI, USA.
30. Parker G.J., Haroon H.A., Wheeler-Kingshott C.A. (2003) A Framework for a streamline-based probabilistic index of connectivity (PICo) using a structural interpretation of MRI diffusion measurements. Journal of Magnetic Resonance Imaging 18:242-254.
31. Behrens T.E., Berg H.J., Jbabdi S. et al (2007) Probabilistic diffusion tractography with multiple fibre orientations: What Can We Gain? Neuroimage 34:144-155
32. Jbabdi S., Woolrich M.W., Andersson J.L. et al (2007) A bayesian framework for global tractography. Neuroimage 37:116-129
33. Sherbondy, A., Dougherty R. et al. (2008). ConTrack: Finding the most likely pathways between brain regions using diffusion tractography. Journal Of Vision 8:1-16

34. Koch M.A., Norris D.G., Hund-Georgiadis M. (2002) An investigation of functional and anatomical connectivity using magnetic resonance imaging. Neuroimage 16:241-250
35. Hagmann P., Thiran J.P., Vandergheynst P. et al (2000) Statistical fiber Tracking on DT-MRI Data as a Potential Tool for Morphological Brain Studies. ISMRM Workshop on Diffusion MRI: Biophysical Issues
36. Chung M.K., Lazar M., Alexander A.L. et al (2003) Probabilistic connectivity measure in diffusion tensor imaging via anisotropic kernel smoothing. Technical Report No:1081, University of Wisconsin
37. Lenglet C., Deriche R. , Faugeras O. (2003) Diffusion tensor magnetic resonance imaging: brain connectivity mapping. Technical Report No: 4983, INRIA, Sophia-Antipolis, France
38. Lenglet, C., M. Rousson, et al. (2006) DTI segmentation by statistical surface evolution. IEEE Transactions on Medical Imaging 25(6):685
39. Bozkaya U., Acar B. (2007) SMT: split & merge fiber tractography for DT-MRI. In: Lecture Notes in Computer Science, 4792:153-160, Springer (Proceedings of MICCAI 2007, Brisbane, Australia)
40. Bozkaya U., Acar B. (2006) SMT: mplit/merge fiber tractography for MR-DTI. In: Proceedings of ESMRMB 2006, Warsaw, Poland.
41. Yörük E., Acar B., Bammer R. (2005) A physical model for MR-DTI based connectivity map computation. In: Lecture Notes in Computer Science, 3749/1:213-220, Springer (Proceedings of MICCAI 2005, Palm Springs, CA, USA)
42. Bozkaya U (2006) SMT: split/merge fiber tractography for MR-DTI. M.S. Thesis, Boğiçi University, Biomedical Engineering Institute, Istanbul, Turkey
43. Lazar M, Alexander A.L. (2005) Bootstrap white matter tractography (BOOT-TRAC). Neuroimage 24:524-532.
44. Chib S., Greenberg E. (1995) Understanding the Metropolis-Hastings algorithm. The American Statistician 49:327-335
45. Sherbondy A., Akers D., Mackenzie R. et al (2005) Exploring connectivity of the brain's white matter with dynamic queries. IEEE Trans. Vis. Comput. Graph. 11:419-430.

# Strain Rate Tensor Estimation in Cine Cardiac MRI Based on Elastic Image Registration

Gonzalo Vegas Sánchez-Ferrero, Antonio Tristán Vega, Lucilio Cordero Grande, Pablo Casaseca de la Higuera, Santiago Aja Fernández, Marcos Martín Fernández and Carlos Alberola López

**Abstract** In this work we propose an alternative method to estimate and visualize the Strain Rate Tensor (SRT) in Magnetic Resonance Images (MRI) when Phase Contrast MRI (PCMRI) and Tagged MRI (TMRI) are not available. This alternative is based on image processing techniques. Concretely, image registration algorithms are used to estimate the movement of the myocardium at each point. Additionally, a consistency checking method is presented to validate the accuracy of the estimates when no golden standard is available. Results prove that the consistency checking method provides an upper bound of the mean squared error of the estimate. Our experiments with real data show that the registration algorithm provides a useful deformation field to estimate the SRT fields. A classification between regional normal and dysfunctional contraction patterns, as compared with experts diagnosis, points out that the parameters extracted from the estimated SRT can represent these patterns. Additionally, a scheme for visualizing and analyzing the local behavior of the SRT field is presented.

## 1 Introduction

Mechanical properties of the heart provide a way to determine early diagnosis and better patient follow-up. Actually, local motion abnormalities (measurable by means of mechanical anomalies) could precede electrocardiogram disorders [15]. Impaired myocardial perfusion appears as a consequence of reduced blood flow of the heart muscle. This is often analyzed with nuclear medicine imaging techniques or with MRI.

MRI is a very suitable imaging modality for blood flow and tissue motion measurement. It provides excellent contrast between soft tissues, and images can be acquired at positions and orientations freely defined by the practitioner. From a temporal sequence of MR images, boundaries and edges of tissues can be tracked by image processing techniques [24]. The local deformation of the myocardium can be

LPI, Universidad de Valladolid,
e-mail: {gonzalo,atriveg,lcorgra,casaseca}@lpi.tel.uva.es
e-mail: {sanaja,marcma,caralb}@tel.uva.es

described by the Strain Tensor (ST) which gives a measure of the deformation of an object relative to its original length. The instantaneous variation of the deformation is described by the Strain Rate Tensor (SRT) and also provides information about myocardial deformation.

There are different possibilities to estimate and visualize the ST. Among those modalities involving MRI, Tagged MRI (TMRI) and Phase Contrast MRI (PCMRI) are direct approaches to obtain information about the tensor. TMRI allows deriving a motion model of the underlying tissue by tracking a temporal sequence of images which have been previously marked by a pattern of dark lines (called tags). This pattern is achieved by modulation of the image intensity with a magnetic presaturation pulse [24]. The deformation field of the crossing line points can be calculated just following the temporal trajectories. On the other hand, PCMRI provides a measure of the velocity field by means of phase shifts induced in the transverse magnetization. The phase of the signal is directly related to the velocity of the material within each voxel. When TMRI or PCMRI are not available, the estimation of the SRT becomes a difficult task, even more if direct 3D information is not available. In section 3 both modalities are briefly explained.

This chapter is focused on an alternative to hardware techniques which need a TMRI or PCMRI scanner. This alternative is based on registration algorithms which are able to estimate a deformation field between two consecutive images deformed in a non rigid way. In case the deformation between adjoining frames is small enough, we can assume that the estimated deformation applies for the physics of the problem, and so we can identify the deformation field with the movement of the myocardial wall. With such a technique, the ST can be estimated from temporal sequences of conventional MRI.

Some non-rigid registration algorithms have been applied in MRI tagging and cine MRI. Concretely, Ledesma et al. [18] proposed a B-spline registration model that has demonstrated good results with subpixel accuracy. Chandrashekara et al. [9, 10], proposed a 4D B-spline registration model proving that the radial and longitudinal displacement provided by the deformation field shows high correlation coefficients compared to TMRI.

In this chapter, we demonstrate that registration algorithms are able to estimate the deformation field by means of a consistency checking methodology when no golden standard is available. This methodology provides an upper bound of the mean error of the estimated deformation field. Additionally it allows comparing different registration algorithms in terms of the consistency of the deformation field.

The posterior analysis of the SRT proves that the registration stage is accurate enough when consistency error is beneath the mean displacement of the deformation field because it provides useful information to detect abnormalities in the behavior of the myocardium.

Additionally, a scheme to visualize the tensor field is presented based on a geometric point of view of the tensor field. This scheme provides information about the local behavior of the tensor field.

The chapter is organized as follows: In section 2 we give a formal definition of the Strain Tensor. With this concept in mind, we can easily understand the different

MRI modalities which are usually employed to estimate the ST and the SRT, section 3 explains those modalities and their problems.

As an alternative to those modalities, when just cine MRI is provided, registration algorithms appears as a good alternative to estimate deformation fields. In section 4 different registration algorithms for general purpose are explained. These algorithms will be compared in further sections.

In order to get a good way to compare registration algorithms when no golden standard is provided, a phantom is a good alternative to see the real error in the deformation estimates. Section 5 describes the models usually employed and explains the phantom designed to compare registration algorithms.

Obviously, a methodology to compare registration performance in real data is desired, even more when no other MRI modalities are available. So, a consistency checking strategy seems to be very suitable. Section 6 explains the consistency strategy and shows the registration algorithms behavior in terms of consistency and real error when applied to the phantom. At the end of the chapter this strategy is applied and tested with real data.

Once the tensor field is estimated, a scheme to visualize the tensor field is needed in order to provide local information which can be useful to physicians. In section 7 a scheme based on a geometric point of view of the ST field is presented.

Section 8 shows the accuracy of registration algorithms by means of the consistency checking method exposed before and its usefulness in classifying normal and abnormal patients. We also see the coherence of the strain rate estimates by comparing normal and abnormal patient through the heart cycle. Additionally, a sequence of the heart cycle is presented to see the local behavior of the heart.

Finally, in section 9 the main conclusions extracted from this chapter are summarized.

## 2 Strain Tensor Definition

The strain tensor $E$ is a symmetric tensor which describes the strain of an object undergoing infinitesimal deformation.

For a simple description of the strain consider a one-dimensional object, with two arbitrary points $A$ and $B$ separated a distance $L$. After deformation these points move to position $A+u(A)$ and $B+u(B)$ where $u$ is the displacement. So, we can calculate the strain as the increase in the distance between $A$ and $B$ once deformed when $L$ tends to 0 (see Fig 1.(a)).

$$E = \frac{B+u(B)-A-u(A)-L}{L} = \frac{B-A+u(B)-u(A)}{L} - 1 \tag{1}$$

Since $B = A+L$ and the series expansion of u(A+L) for small deformations is $u(A+L) = A + \frac{\partial u}{\partial x}L$, the strain is:

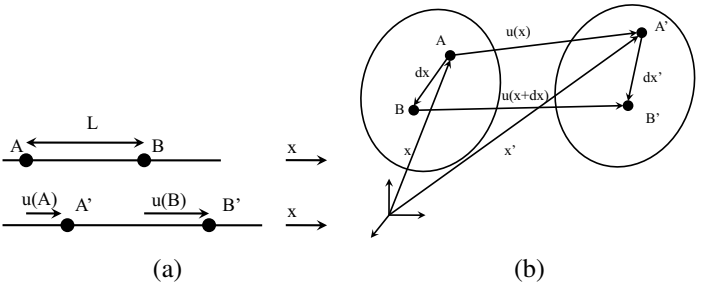



**Fig. 1** (a) 1D before and after deformation. (b) 3D body before and after deformation.

$$E = \frac{L + u(A) + \frac{\partial u}{\partial x}L - u(A)}{L} - 1 = \frac{\partial u}{\partial x} \tag{2}$$

A more general case for 3D strain, let us consider a point $A$ in a material with coordinates $\mathbf{x}$ (see Fig. 1b). If the point moves a small distance $\mathbf{u}(\mathbf{x})$, $\mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x})$.

Let consider a point $B$, $\mathbf{dx}$ in the neighbourhood of $\mathbf{x}$. After deformation it will be at the position given by:

$$\mathbf{x}' + \mathbf{dx}' = \mathbf{x} + \mathbf{dx} + \mathbf{u}(\mathbf{x} + \mathbf{dx}) \tag{3}$$

For small deformations we can approximate $\mathbf{u}$ by the first two terms in its Taylor Series:

$$\mathbf{x}' + \mathbf{dx}' \approx \mathbf{x} + \mathbf{dx} + \mathbf{u}(\mathbf{x}) + (\nabla \mathbf{u})\mathbf{dx} \tag{4}$$

Finally, the infinitesimal change in the position of the nearby particle is:

$$\mathbf{dx}' \approx \mathbf{dx} + (\nabla \mathbf{u})\mathbf{dx} \tag{5}$$

Note that if $\nabla \mathbf{u} = 0$, the transformation becomes a rigid translation. In order to distinguish rigid translations/rotations from deformation we can write the displacement matrix as a sum of symmetric and antisymmetric matrices:

$$\mathbf{dx}' = \mathbf{dx} + \frac{1}{2}\left((\nabla \mathbf{u}) - (\nabla \mathbf{u})^T\right)\mathbf{dx} + \frac{1}{2}\left((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T\right)\mathbf{dx} \tag{6}$$

The first two terms are the Taylor series of a rigid rotation and does not represent information about the deformation of the material. The last term represents the deformation of the material and is the strain tensor:

$$E = \frac{1}{2}\left((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T\right) \tag{7}$$

The SRT can be calculated with the same expresion by replacing the displacement field $\mathbf{u}$ with the velocity field $\dot{\mathbf{u}}$ associated to the myocardial motion.

# 3 MRI Modalities to Estimate Strain and Strain Rate Tensor

## *3.1 Tagged MRI*

The main contribution of TMRI or spatially modulated saturated magnetization processing techniques is their ability to obtain precise and reliable local measures of the myocardium motion [4]. This contribution is based on the intrinsic sensitivity to motion of MRI. In particular, it is possible to locally alter the tissue magnetization in a persistent manner (taking into consideration the relaxation times of MR) even in the presence of motion. TMRI image acquisition can be divided into two different stages:

*1. The generation of a spatially modulated saturated magnetization pattern.* A combination of radiofrequency (RF) and gradient pulses are used for this purpose. The techniques for generating this pattern have been improved from the ones that obtain a small set of saturated magnetization planes over the cardiac wall by means of selective RF excitation [39] to the modern formulation in terms of *Spatial Modulation of Magnetization* (SPAMM), that generates a set of saturated magnetization patterns along all the image by non selective excitation [5]. The main problem of this technique is the fading of the saturated patterns by the relaxation time and the repetition of RF excitations during image acquisition.

Complementary SPAMM (CSPAMM) [13] tries to confront this problem by using the subtraction of two images acquired with complementary SPAMM sequences to improve the contrast and allow for a complete tracking of the grid throughout the cardiac cycle. Finally, *Delays Alternating with Nutations for Tailored Excitation* (DANTE) pulse sequences are based on the generation of a set of RF pulses that are superimposed over the gradients, so acquisition times are reduced [21]. Multiple combinations and variations of these basic techniques have been developed during the last years.

*2. The generation of an image of the pattern deformation using pulse sequence synchronized with the electrocardiogram signal.* Almost any conventional acquisition technique for MRI can be used in this step. The choice of a given technique, as in the general case, must pay attention to the balance among the image resolution, its signal-to-noise ratio (SNR) and the acquisition time. Additionally, the contrast to noise ratio between tagged and untagged myocardium appears as an important factor for accurately detecting tags over the cardiac cycle. If the purpose is to measure the ST, spatial resolution of the grid has to supply at least two tagging lines along the myocardium and temporal resolution has to be sufficiently high to avoid tagging lines indefinition caused by motion.

A basic problem in TMRI motion reconstruction is to consider the component through the acquisition planes, that is, the longitudinal component. It is obvious that, due to movement, images acquired in different phases of the cardiac cycle could not represent the same myocardium material slice. Slice tracking methods [26] can help

in the correction of this effect, although a consistent approach for its treatment would typically lay on the combination of multiple long axis and short axis 2D views [20, 38] or on direct 3D TMRI acquisition [27]. Multi-slice 2D acquisition techniques are based on the selective excitation along the axis over which the slices are acquired. They are easily implemented, but the resolution along the axis of acquisition is limited. Volumetric techniques are based on the gradient modulation along three orthogonal directions. In general they present a larger SNR but they are more prone to motion related artifacts due to breathing. Two ways to partially overcome this problem are to avoid patient breathing by faster acquisition techniques or to couple the scanner to breathing, but there is no final solution, so the 3D reconstruction of the SRT by TMRI is susceptible of improvement mainly by a refinement of the acquisition techniques.

Grid tracking is the main task for the extraction of the myocardium motion measures in the analysis of TMRI. A great number of methods have been proposed in the recent years, as for instance the ones in [3, 8, 12, 23].

## *3.2 Phase Contrast MRI*

As noted in previous sections, the local *instantaneous* deformation of the myocardium is represented by the Strain-Rate (rate of deformation) Tensor (SRT). Its calculation is straightforward from the equation (7) above by just replacing the displacement field $\mathbf{u}$ with the velocity field $\dot{\mathbf{u}}$ associated to the myocardial motion [29]. Time-resolved velocity data enabling calculation of the SRT can be obtained from PCMRI, a non-invasive modality that can measure flow and displacement accurately.

The fundamental basis of PCMRI stands on the use of phase shifts for velocity encoding. Phase-contrast acquisition is performed by adding two opposing gradient pulses known as velocity-encoding to the imaging sequence of pulses. These bipolar gradients cause a phase shift in moving spins that is proportional to the velocity along the gradient's direction [14]. Since the direction of the velocity-encoding gradient is flexible, the encoding process can be repeated in order to obtain the complete velocity vector. Static spins on the other hand do not experience the aforementioned phase shift due to the symmetry of the bipolar gradient.

Conventional (anatomical) MRI is usually displayed as magnitude images. Velocity data in PCMRI is obtained from the phase of the complex signal (phase image). Since phase has a $2\pi$ range, values beyond this range do repeat, causing wraparound (aliasing). Stationary material is represented in the phase image as mid-grey (zero-value) pixels. On the other hand, increasing velocities are represented as either brighter (positive) or darker (negative) pixels.

The relationship between the phase image associated to the each direction and the corresponding velocity component is given by $\dot{u}_i = \text{VENC} \times \frac{\Delta\phi}{\pi}$ where $\dot{u}_i$ is the velocity component in the gradient direction, $\Delta\phi$ is the pixel phase shift and VENC is the velocity encoding value, which is an adjustable parameter that represents the

sensitivity of the acquisition to velocity measurements. A small VENC represents a highly velocity-sensitive image.

One of the main problems related to PCMRI is the effect of velocity aliasing or wraparound. This effect raises when the velocity phase shift exceeds $\pm\pi$. A phase shift outside this range cannot be distinguished from one within it, which is displayed instead. If the true ranges of velocities in the Region Of Interest (ROI) exceeds $2 \times$ VENC they cannot all be correctly displayed simultaneously due to aliasing. Since VENC is an acquisition parameter it cannot be changed after the image has been acquired. Thus, if aliasing appears excessive, acquisition should be repeated with a larger VENC. A simple way to reduce the effect of aliasing is to offset the measurable velocity range so that wraparound does not affect the ROI. Other approaches can be found in [6, 11, 36].

Another undesirable effect appearing in PCMRI acquisitions comes from the induction of Eddy currents in the scanner conductors [17]. The effect of these currents is the introduction of fiducial velocities in the images that result in static zones interpreted as moving tissues. A common methodology to reduce the Eddy currents effect is to apply linear models based on velocity measurements obtained from static regions of the image [25]. Correction is performed by comparison between the measured phases and those obtained from the model. Other artifacts such as those coming from the Maxwell effect associated to additional magnetic fields created during the acquisition can be easily removed [31].

## 4 Registration Methods

Registration is defined as the process of recovering an unknown geometrical mapping that transform one given image called target or fixed image into the source or moving image. In some cases, the warped image is the result of interest. In other cases, as in the problem at hand, we are interested in the mapping itself. Given a MRI section of the heart at a concrete moment, we consider it as the source image, and we try to register it with the target image, which is the corresponding section on a previous time slot. Under some assumptions, the recovered transformation may be considered as an estimate of the true displacement of physical points. This is only valid for small displacements, mainly due to the aperture problem. A large number of approaches to deformable image registration (i.e. not based on rigid movements or scaling) may be found in the literature [40], either based on landmark-matching or grey level-based techniques. We are interested in the latter approaches, where a cost function based on the similarity between the images to register is minimised according to some model for the movement. This way, we may choose between a transformation which models the underlying physical behavior of the organs being imaged, a smooth deformation parametrized by means of some basis function, or a strictly free-form deformation field defined by a vector field. Given the difficulty to accurately describe the complex physics of heart movement with enough degrees of

freedom, we try three state of the art algorithms based on parametric transformations and free form deformations[1].

### Local Correlation Coefficient Demons Registration

This registration algorithm is an adaptation of the well-known demons algorithm proposed by Thirion [30]. We use the implementation of Cachier et al. [7] instead, because of its more robust behavior.

The original demons proceeds by the alternative optimization of the cost function and the regularization of the deformation field by means of the convolution with a Gaussian kernel. The optimization is carried out following the optical flow equation, and works by aligning the iso-contours of the image. This algorithm has been proved to be a quasi-Newton optimization of the Mean Squared Error (MSE) between the images to register, relying on the assumption that the grey level of corresponding voxels does not vary from one image to the other, which makes it too sensitive to small changes in the illumination. Cachier et al. [7] propose a similarity measure based upon local (windowed) correlations between the images, i.e. the Local Correlation Coefficient (LCC), which is computed using convolutions with Gaussian kernels. This way, the algorithm is able to cope with severe changes in the illumination whenever the grey levels of corresponding voxels are linearly related, which is especially well suited for MRI.

### Normalized Mean Squared Difference Demons Algorithm

The problem with the LCC similarity measure is that it is not as local as the Mean Squared Difference (MSD), which results in a worse adaption to local variations of the deformation field. To partially overcome this difficulty, in [33] a new similarity measure based on LCC but with better local properties is proposed. Briefly speaking, it can be shown that the Cross Correlation (CC) may be seen as the expectation of the normalized difference between the two images being compared, $I$ and $J$: $\mathrm{CC} = E\left\{\left(\frac{I-E\{I\}}{\sigma_I} - \frac{J-E\{J\}}{\sigma_J}\right)^2\right\}$ where $\sigma_I$ (resp. $\sigma_J$) is the standard deviation of $I$ (resp. $J$). Since the expectations are replaced by spatial convolutions, the additional expectation of the squared difference introduces an extra blurring of the similarity measure, which motivates the introduction of the Normalised MSD (NMSD), defined as: $\mathrm{NMSD} = \left(\frac{I-E\{I\}}{\sigma_I} - \frac{J-E\{J\}}{\sigma_J}\right)^2$.

We cannot avoid the convolutions for the computation of the mean and standard deviation of the images, since they are necessary to normalize the differences and therefore to palliate the effect of the changes in illumination.

[1] Note that we are not trying to evaluate the goodness of a particular registration algorithm, but only to prove that image registration may be used to estimate cardiac motion.

Block-Matching Algorithm

Finally, we use a generic block matching algorithm to test parametric deformations. The images are sub-sampled in a grid and, for each node, the movement is estimated by comparing it with its neighbouring voxels inside a search window; the best matching voxel is chosen, and the displacement is computed as the vector from the original voxel to its best match. To estimate the goodness of fit between two given voxels, we take a neighborhood of a given fixed size around each of them, and then compute the linear cross correlation between the blocks. Due to noise as well as the aperture problem and some other artifacts, these are weak estimates of the true displacements, so further regularisation is needed. Like in [37], we normalize the similarities of each block so that they accomplish the requirements to be considered as probabilities, and use a Bayesian framework in which the local coherence is introduced as priors. The regularised displacements are extrapolated to non-grid voxels by means of a multi-level B-spline. Contrary to [37], where the smoothness of the interpolation is achieved by the variational optimisation of the spline parameters including a penalty term, we use the non-iterative strategy of [19] instead, which guarantees smooth and accurate representation of the interpolation nodes. Finally, we use an additional constraint to further improve registration accuracy: together with the transformation of interest, we estimate the inverse transformation (the one that drives the target image to the source image), then use the technique of [32] to pseudo-invert this transformation, and use the result to correct the desired deformation.

## 5 Phantom Design

Since the aim of this study is to quantify the motion of the LV and there is no ground truth, a model with known deformations should be constructed.

The model here prosed is designed to describe the motion of the LV in terms of the vector deformation field. This makes possible to calculate the exact movement of each pixel of the image from one moment to the following and, hence, the error of the deformation field estimate.

As the registration algorithms have to deal with MR images, the deformation of the phantom should be applied in images as closed as possible to the real MRI. With this idea in mind, the deformation field will be applied in a real image in a given instant (End Diastolic) transforming it into each one of the instants of the cardiac cycle. The scheme of the phantom is depicted in Fig. 2.

The phantom was developed from Late Diastolic images 2D+N. The longitudinal resolution was not good enough to consider it as a volume, so simulate longitudinal movements does not make sense. This does not suppose a problem because the phantom is designed to validate 2D+N movement in cine MRI and real data has no further information in longitudinal axis.

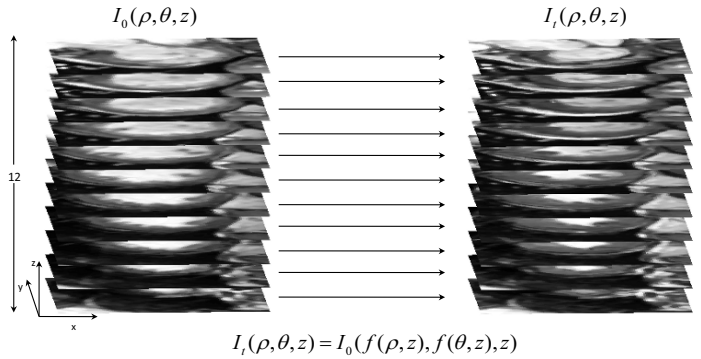



**Fig. 2** Phantom Scheme. The vector field which governs the deformation of the real image gives the complete cardiac cycle deforming the real image.

The movement was performed in a cylindrical coordinate system where the center was placed in the center of the segmentation provided by physicians. The segmentation defines the area of the maximum movement (endocardium contour) and the area from which the movement vanishes (beyond the epicardium contour and inside the endocardium). Two sorts of movements will deform the image. The first one consists of a radial contraction which depends on the position in the longitudinal edge and the endocardium/epicardium contours. The second is an angular movement which also depends on the longitudinal coordinate and the segmentation contours.

Let us call $(x,y,z)$ the coordinates of each voxel of the volume, where $z$ is the index of each slice and the pair $\mathbf{x} = (x,y) \in \Omega$ is each pixel in each slice. $\Omega_{end} \subset \Omega$ is the area inside the endocardium contour. In the cylindrical coordinate system $(\rho,\theta,z)$, where $\rho(x,y)$ and $\theta(x,y)$ are the radius and the angle related to the center of the endocardium segmentation. The radial and angular transformations are performed as it is described in Table 1.

Note that this contraction takes into consideration the endocardium contour in such a way that the maximum contraction is done in the endocardium neighbourhood. Then the contraction vanishes fast for points beyond the epicardium.

The transformation is, then, calculated as $I_t(\rho,\theta,z) = I_0(\rho^*,\theta^*,z)$. As an example, in Fig. 3 a cycle of the phantom is depicted with parameters: $A = 15$, $k = 0.01$, $B = 0.8$, $\alpha = 0.08$, $Z_{max} = 11$, $\rho_{max} = 7$. The superimposed grid shows a maximum deformation in a radial and angular way in the endocardium and then vanishes beyond the epicardium.

**Table 1** Radial and Angular transformation of the phantom†

Radial Transformation: $\rho^* = \rho + f_{r1}(t) \cdot f_{r2}(\mathbf{x}) \cdot f_{r3}(z)$

$$f_{r1}(t) = A \sin\left(\frac{\pi t}{3/4T}\right)^2 e^{-kt}$$

$$f_{r2}(\mathbf{x}) = \begin{cases} \cos\left(\frac{\pi}{2} \frac{F(\mathbf{x},\mathbf{C_{end}})}{\bar{\rho}_{end}}\right) & \text{if } \mathbf{x} \in \Omega_{end} \\ \cos\left(\frac{\pi}{4} \frac{F(\mathbf{x},\mathbf{C_{end}})}{\bar{\rho}_{epi} - \bar{\rho}_{end}}\right) & \text{if } \mathbf{x} \in \Omega \cap \bar{\Omega}_{end} \\ 0 & \text{if } F(\mathbf{x},\mathbf{C_{end}}) > 2(\bar{\rho}_{epi} - \bar{\rho}_{end}) \end{cases}$$

$$f_{r3}(z) = 1 - \frac{1-B}{Z_{max}} z$$

Angular Transformation: $\theta^* = \theta - f_{\theta 1}(t) \cdot f_{\theta 2}(\rho) \cdot f_{\theta 3}(z)$

$$f_{\theta 1} = \alpha \sin(\pi \tfrac{t}{T})^2$$

$$f_{\theta 2}(\rho) = \begin{cases} \alpha_{end} \sin(\frac{\pi}{2} \frac{\rho}{\bar{\rho}}) & \text{if } \mathbf{x} \in \Omega_{end} \\ \alpha_{end} - \alpha_{epi} \sin(\frac{\pi}{2} \frac{\rho - \rho_{end}}{\rho_{epi} - \rho_{end}}) & \text{if } \mathbf{x} \in \Omega_{epi} \cap \bar{\Omega}_{end} \\ \frac{\alpha_{end} - \alpha_{epi}}{2} + \frac{\alpha_{end} - \alpha_{epi}}{2} \cos(\pi \frac{\rho - \bar{\rho}_{epi}}{\rho_{max}}) & \text{if } \rho \leq \rho_{max} \\ 0 & \text{if } \rho > \rho_{max} \end{cases}$$

$$f_{\theta 3}(z) = \cos(\pi \tfrac{z}{Z_{max}})$$

† Where $T$ is the period of cardiac cycle, $A$ is the maximum contraction allowed in the endocardium contour, $k$ is a constant of temporal vanishment, $\bar{\rho}_{end}$ and $\bar{\rho}_{epi}$ is the mean radial component endocardium and epicardium respectively. $F(\mathbf{x},\mathbf{C_{end}})$ is the Euclidean distance transform between each point $\mathbf{x}$ and the endocardium contour. $B$, which takes values between 0 and 1, is the amplitude ratio allowed depending on the longitudinal direction, and $Z_{max}$ is the total number of slices. $\alpha$ is the maximum angular transformation, $\alpha_{end}$ is the ratio of the maximum angular value in the endocardium contour. $\alpha_{end}$ is the maximum decrease of $\alpha_{epi}$ in the epicardium contour and $\rho_{max}$ is the radial distance from $\rho_{epi}$ where the angular transformation becomes null.

## 6 Consistency Checking

Without a ground-truth of the deformation field to estimate (i.e. without TMRI or PCMRI), the validation of the results is difficult, so we must accomplish an indirect estimation of the accuracy. To assess the accuracy of the registration, we use similarity measures: **(1)** the Structural Similarity (SSIM) index [34], **(2)** the Quality Index based on Local Variance (QILV) [2] and **(3)** the Mean Square Error (MSE).

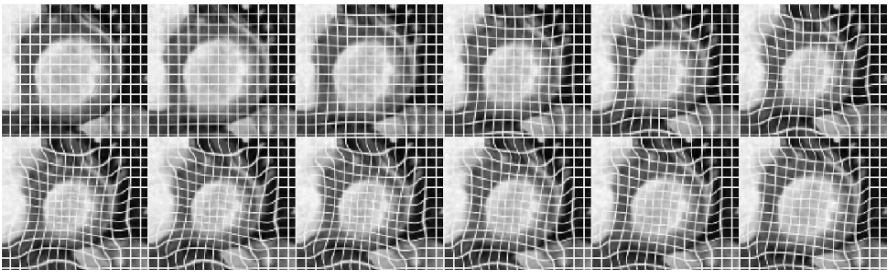


**Fig. 3** 12 of 20 Phantom Sequence instants for parameters: $A = 15$, $k = 0.01$, $B = 0.8$, $\alpha = 0.08$, $Z_{max} = 11$, $\rho_{max} = 7$. Time instants goes from left to right from top to bottom. The superimposed grid shows a maximum deformation in a radial and angular way in the endocardium and then vanishes beyond the epicardium.

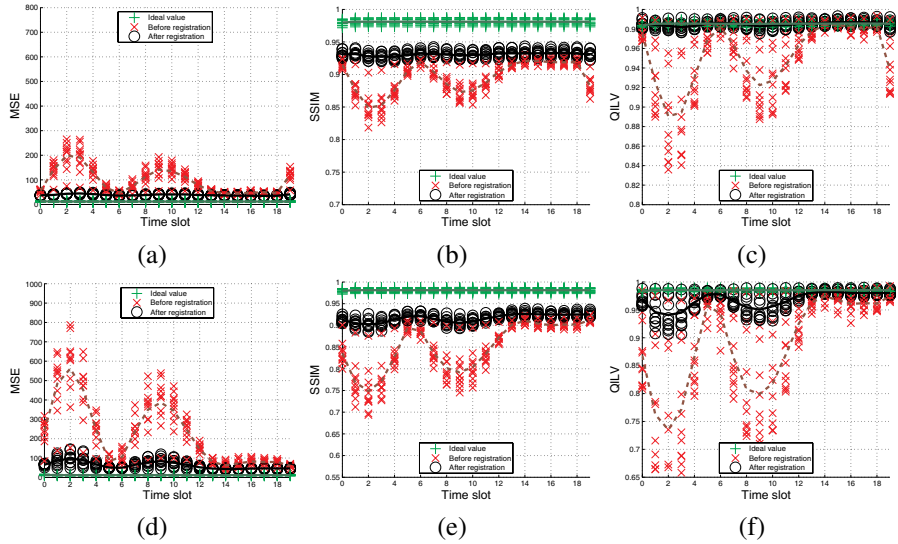



**Fig. 4** Similarity measures for each time slot and slice (MSE –a, d–, SSIM –b, e– and QILV –c, f–) for LCC Demons case. Top row: we register slot $t$ with slot $t+1$. Bottom row: slot $t$ with $t+2$. For each case, we superimpose the values for the 11 phantom slices in the same time slot together with the average value for all slices (solid and dashed lines). The results of the registrations are compared to the similarity before the registration and to the ideal value that could be achieved if the estimation of the deformation field were exact.

The MSE is a standard measure, so it does not need further explanation. Regarding SSIM and QILV, they have been widely applied in the literature as measures of the structural similarity between two given images; both indices are bounded: the closer to one, the better the image alignment.

The strategy consists in comparing the similarity measures before and after registration, as well as the *ideal value* we could get. This value is calculated from the same image by corrupting it with two independent Rician noises with variance esti-

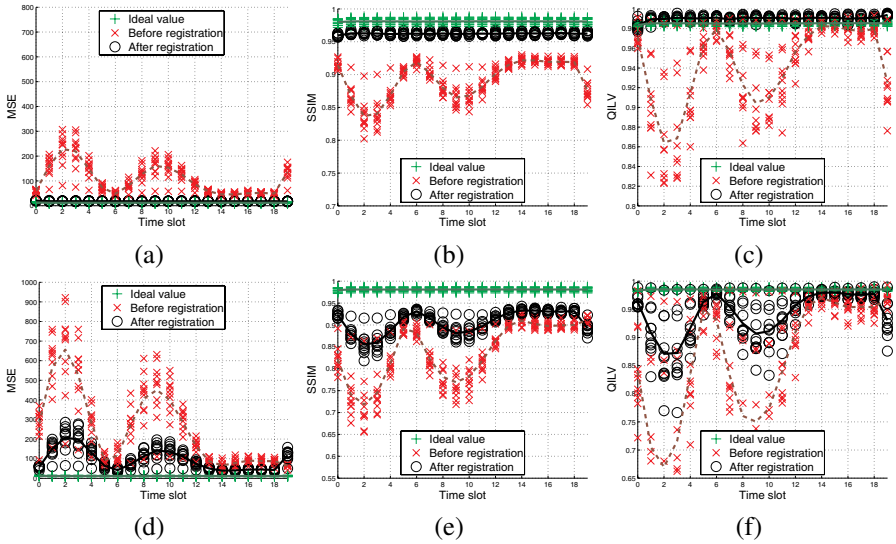



**Fig. 5** Similarity measures for each time slot and slice (MSE –a, d–, SSIM –b, e– and QILV –c, f–) for NMSDL Demons case. Top row: we register slot $t$ with slot $t+1$. Bottom row: slot $t$ with $t+2$. For each case, we superimpose the values for the 14 slices in the same time slot together with the average value for all slices (solid and dashed lines). The results of the registrations are compared to the similarity before the registration and to the ideal value that could be achieved if the estimation of the deformation field were exact.

mated from real data using the proposed method in [1] and then blurring the second one to simulate the interpolation effect of the registration algorithm. This way we obtain two different images $\mathscr{I}_1$ and $\mathscr{I}_2$ to compare and get an estimation of the best result we could obtain if the estimation of the deformation field were exact and the registered images differed only because of noise and interpolation artifacts.

The results obtained from the phantom are shown in Figs. 4-6 for each time slot from 0 to 19 (we register slot $t$ with $t+1$ and $t+2$), for each slice (12 slices in total), and the average of all the values for all slices. Results before and after registration together with the *ideal value* are shown. Figs. 4-6 demonstrate that the registration clearly improves the similarity between the images in all cases: mean values for all slices and individual values for each slice are closer to the ideal values. It is clear that the registration of non consecutive slots is more difficult than it is for consecutive slots, and therefore the registration results are poorer (note that the similarity measures, both before and after registration, are worse for the pairs $t+2,t$ than for the pairs $t,t+1$). Besides, not all time slots show the same similarity, but some of them are more difficult (and yield poorer results) than others. Those peaks shown in the simulated end-systole and end-diastole appear because the phantom was designed to move slower in those moments in order to simulate the real behavior of the heart.

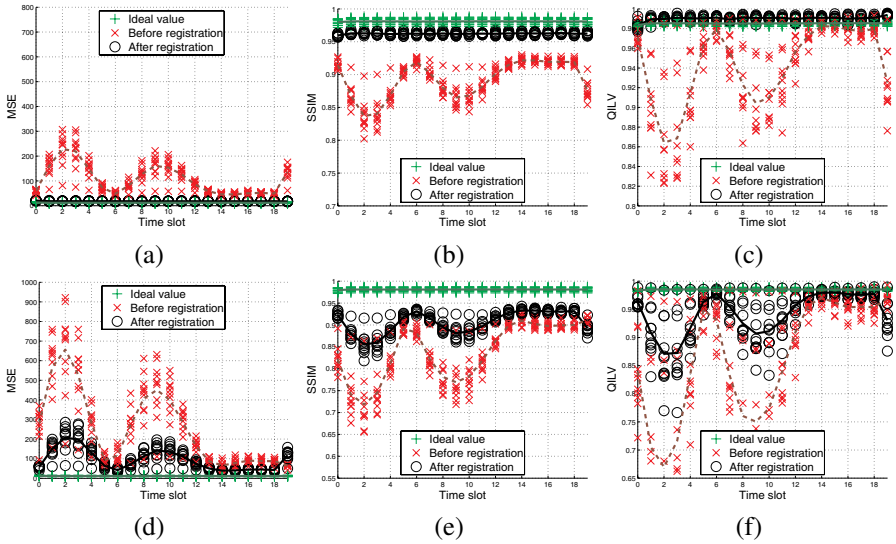



**Fig. 6** Similarity measures for each time slot and slice (MSE –a, d–, SSIM –b, e– and QILV –c, f–) for Block-Matching case. Top row: we register slot $t$ with slot $t+1$. Bottom row: slot $t$ with $t+2$. For each case, we superimpose the values for the 14 slices in the same time slot together with the average value for all slices (solid and dashed lines). The results of the registrations are compared to the similarity before the registration and to the ideal value that could be achieved if the estimation of the deformation field were exact.

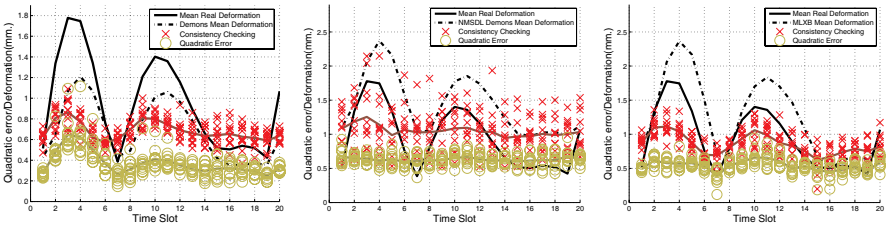
Once it has been shown that the algorithm is able to recover the deformations even for non consecutive time slots, we adapt the *consistency checking* methodology in [22] to estimate the accuracy of the results: for each time slot $t$, we estimate the deformation field $\mathscr{D}_{t,t+1}$ between this slot and slot $t+1$, and respectively $\mathscr{D}_{t+1,t+2}$ and $\mathscr{D}_{t+2,t}$. Then we compute for each image voxel the magnitude $\mathscr{M} = \|\mathscr{D}_{t,t+1} + \mathscr{D}_{t+1,t+2} + \mathscr{D}_{t+2,t}\|$. The composition of $\mathscr{D}_{t,t+1}$ and $\mathscr{D}_{t+1,t+2}$ should ideally equal $-\mathscr{D}_{t+2,t}$, and therefore $\mathscr{M}$ should be 0. If we consider that $\mathscr{D}$ are estimates of the true displacement field contaminated with some random noise, the residual $\mathscr{M}$ would be the modulus of the addition of three random variables, which in turn should be independent, since they correspond to three independent experiments. All in all, $\mathscr{M}$ may be considered as an estimate of the Euclidean error of the deformation field $\mathscr{D}$ whenever the estimate of $\mathscr{D}$ is unbiased.

The reason why we calculate $\mathscr{D}_{t+2,t}$ instead of $\mathscr{D}_{t,t+2}$ is that we want to get an unbiased measure of the deformation field so, this way, the fixed image and the moving image are never repeated.

The corresponding results are shown in Fig. 7, where we show the mean displacements $\|\mathscr{D}_t\|$ recovered by the registration algorithm, as well as the real mean displacements. Note that the maximum mean displacements correspond to the simulated early systole and late diastole, where the strain is maximum, which confirms once again that the estimation of the displacement fields is adequate. Comparing the

mean displacements and the mean consistency errors, the latter are well below, so we may conclude that the registration algorithm is able to recover the deformation field with an error in the order of the mean discrepancies (consistency errors). Note that to compute these discrepancies we accumulate the (independent) errors of three different fields, $\mathscr{D}_{t,t+1}$, $\mathscr{D}_{t+1,t+2}$ and $\mathscr{D}_{t+2,t}$; moreover, at the sight of Figs. 4-6, the error in the estimation of $\mathscr{D}_{t+2,t}$ is greater, so in fact the mean discrepancy overestimates the *true* error. This holds with the representation of the true squared error which is below the consistency error in all the cycle.

In order to calculate the true squared error, the real deformation field in each time slot is needed. Note that the phantom was designed modifying an initial image trough the whole cycle, so just the cumulative deformation field is known.




(a) LCC Demons (b) NMSDL Demons (c) Block-Matching

**Fig. 7** Consistency checking of the deformation fields. We present mean values of $\mathscr{M} = \|\mathscr{D}_{t,t+1} + \mathscr{D}_{t+1,t+2} + \mathscr{D}_{t+2,t}\|$ for all slices and time slots for the phantom, together with the mean values of the displacements, $\|\mathscr{D}_t\|$. The computation has been restricted to a ROI containing the myocardial wall.

## 7 Visualization

The interpretation of the ST by its components allows an intuitive way to represent them. The diagonal components ($E_{ii}$) may be interpreted as the unit elongation or compression of the material voxel in the $x_i$ direction. The other components ($E_{ij}$ with $i \neq j$) are the shear strains. So, in an infinitesimal square area strained without change of area, the shear strains components are approximately the decrease angle between the axes. In Fig. 8a an infinitesimal square is deformed without change of area, showing that the sum of shearing strains are approximately equal to the decrease in angle with the axis of the square element [35] and it is worthy to explain in detail:

Let us consider an infinitesimal square which is deformed without change of area (see Fig 8a). The tangent of $\gamma$ angle is:

$$\tan(\gamma) = \frac{u_y(A+L)}{L} \tag{8}$$

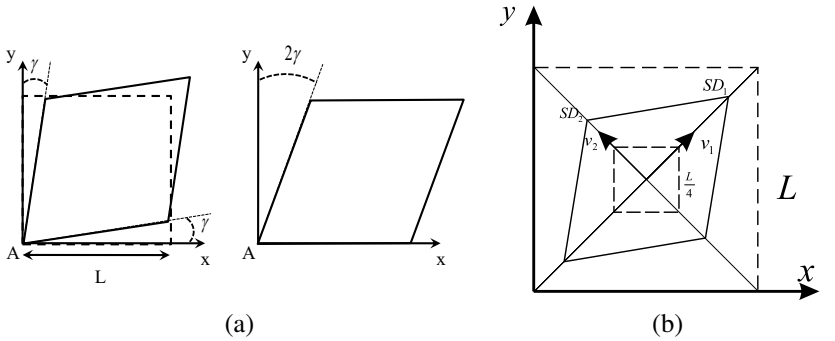



**Fig. 8** (a) Deformation of an infinitesimal square without change of area. The strain components are approximately the decrease in angle with the axis of the square. (b) Rhombus visualization of a ST. The dashed lines are the minimum and maximum allowed squares.

and, for small deformations ($L \to 0$), $\tan(\gamma) = \gamma$ and $u_y(A+L) \approx u_y(A) + \frac{\partial u_y}{\partial x} L$. Since, $u_y(A) = 0$ (and proceeding in the same way with $u_x$) we conclude:

$$\gamma \approx \frac{\partial u_y}{\partial x} \quad \gamma \approx \frac{\partial u_x}{\partial y} \tag{9}$$

and thus,

$$\gamma = \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) \tag{10}$$

This shows that the nondiagonal terms of the ST can be interpreted as the increase in angle so it makes sense to represent the strain by means of the principal directions given by the eigenvectors where is no shear strain.

If every voxel of an image is represented in the same way as an infinitesimal rectangle whose diagonals are oriented in the eigenvector directions and the position of each vertex is related to the eigenvalues, a more intuitive visualization of the tensor field may be achieved.

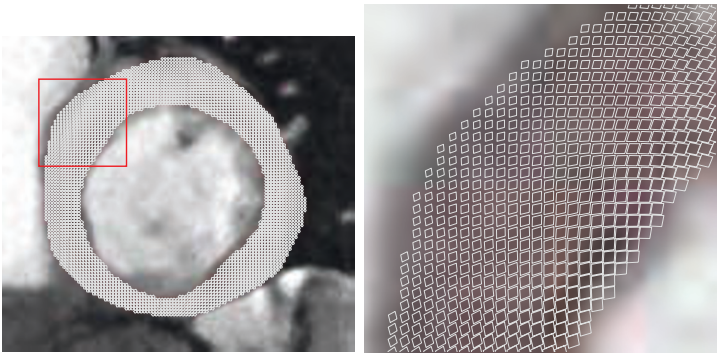
Since the eigenvalues can be either positive or negative, the positive eigenvalues are represented as elongations in the direction of its corresponding eigenvector, whereas negative eigenvalues are represented as contractions. Both eigenvalues are normalized by the largest absolute value of them. The length of each diagonal is:

$$SD_i = \sqrt{2}\left(\frac{L}{4} + \frac{L}{4}R_1^i + \frac{L}{2}R_2\right) \tag{11}$$

where $L$ is the side of the square, $i = 1,2$, $R_1$ is the normalization ratio computed as $R_1^i = \frac{1}{2}(\lambda_i + 1)$ and $R_2$ is the Euclidean norm of the deformation field in each instant. The minimum possible side of the square is L/4, and the maximum is L.

Obviously, when the eigenvalues are equal the eigenvectors do not add much information and the orientation of the square is random. It seems that a representation

by an ellipsoid could be better than a square, however when the tensor field is visualized all the tensors are seen as a whole and the orientation of groups of tensors are not affected by this effect and it is worth representing them as deformed squares in the direction of each diagonal in the same way as an infinitesimal square. Fig. 8b shows a possible representation of a ST. Fig. 9 shows the ST field in the phantom. Here the ST field has a contraction in the radial direction because the elongation of the epicardium wall increases less than the endocardium.



(a) (b)

**Fig. 9** (a) Tensor visualization. (b) Zoomed area. The ST field has a contraction in the radial direction because the elongation of the epicardium wall increases less than the endocardium

## 8 Real Data Results

### *8.1 Materials*

Cine MRI sequences of 17 patients were evaluated in this study. Every segment of the 16-segment model for wall motion recommended by the American Society of Echocardiography Committee on Standards was considered [28]. This model is divided into three different levels: Apical, Mid-heart and Base. The apical area comprises 4 segments (Anterior, Lateral, Inferior and Septal) whereas Mid and Base have 6 segments for each level (Anterior, Antero-lateral, Infero-lateral, Inferior, Infero-septum and Antero-septum). These segments were classified by expert cardiologists in the following 3 classes: 1. Normal, 2. Hypokinesia (diminished movement) and 3. Akinesia (negligible movement). Among them, 194 segments were classified as normal, 27 as Hypokinesia and 51 as Akinesia. Images size is $512 \times 512$ with pixel size $0.8594 \times 0.8594$ mm.

Demons registration algorithm was performed with 5 multiresolution levels with 50 iterations per level and a Gaussian regularization filter with $\sigma = 2.8$. Block-

Matching algorithm was perfomed with 5 multiresolution levels, a block size of 2, 5 iterations per level.

## *8.2 Registration Performance*

To assess the accuracy of the registration, we present a set of results based on the three image similarity measures explained before: SSIM index [30], QILV [1] and MSE.

In Figs. 10-12 we present the results for patient 10, for each time slot from 0 to 19 (we register each slot $t$ with $t+1$ and $t+2$), for each slice from 1 to 14, and the average values for all slices.

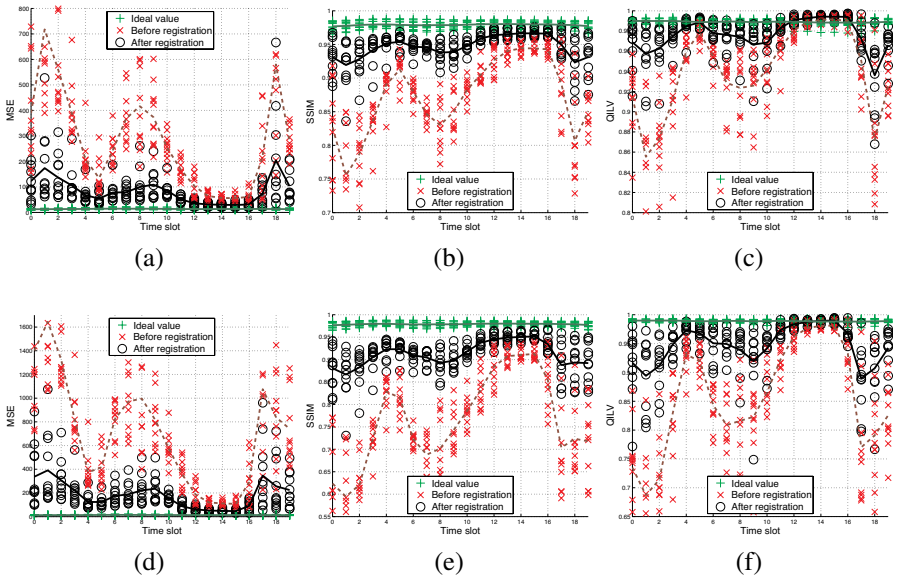



**Fig. 10** Similarity measures for each time slot and slice (MSE –a, d–, SSIM –b, e– and QILV –c, f–) for LCC Demons case. Top row: we register slot $t$ with slot $t+1$. Bottom row: slot $t$ with $t+2$. For each case, we superimpose the values for the 14 slices in the same time slot together with the average value for all slices (solid and dashed lines). The results of the registrations are compared to the similarity before the registration and to the ideal value that could be achieved if the estimation of the deformation field were exact.

We show the results before and after registration, together with an *ideal value*, computed for each slice and time slot by estimating the noise power $\sigma_n^2$ with the method explained in [1] and then denoising the corresponding image $\mathscr{I}$; then we corrupt $\mathscr{I}$ with two independent Rician noisy processes with power $\sigma_n^2$, and blur the second image to simulate the effect of the interpolation when we register, so

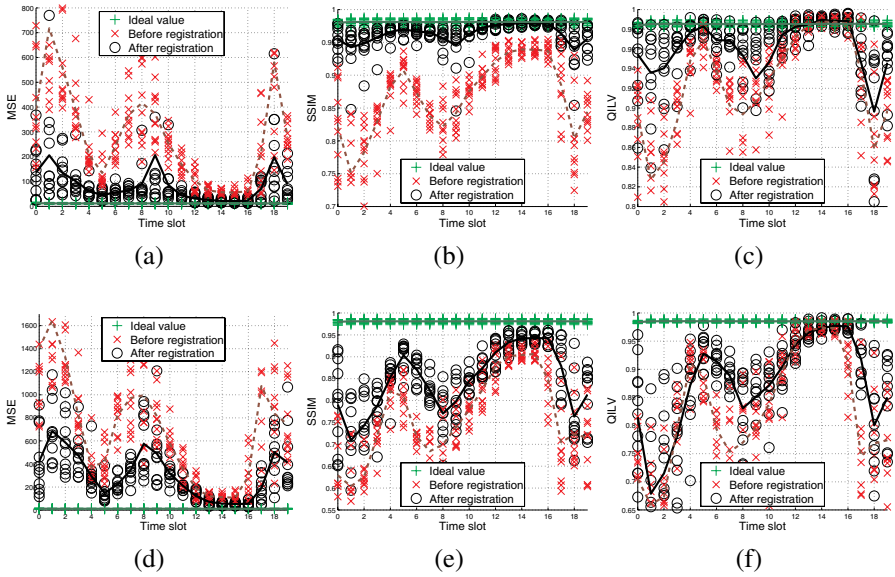



**Fig. 11** Similarity measures for each time slot and slice (MSE –a, d–, SSIM –b, e– and QILV –c, f–) for NMSDL Demons case. Top row: we register slot $t$ with slot $t+1$. Bottom row: slot $t$ with $t+2$. For each case, we superimpose the values for the 14 slices in the same time slot together with the average value for all slices (solid and dashed lines). The results of the registrations are compared to the similarity before the registration and to the ideal value that could be achieved if the estimation of the deformation field were exact.

that we have two synthetic images $\mathscr{I}_1$ and $\mathscr{I}_2$ that we compare. This way we obtain an estimation of the best result we could obtain if the estimation of the deformation field were exact and the registered images differed only because of noise and interpolation artifacts.

Figs. 10-12 demonstrate that the registration clearly improves the similarity between the images in all cases: mean values for all slices and individual values for each slice are much closer to the ideal values than they are to the values without registration. As in the case of the phantom, the registration of non consecutive slots is more difficult than it is for consecutive slots and therefore the registration results are poorer. Moreover, the shape of the averaged values resembles the shape of the cardiac cycle: the peaks of end systole and end diastole correspond to the slots where the myocardium moves the slowest.

Using the consistency checking methodology explained, the results shown in Figs. 10-12 suggest that in fact we have true estimates of the deformation fields, both for $\mathscr{D}_{t,t+1}$ and $\mathscr{D}_{t,t+2}$, so the proposed methodology makes sense. In Fig. 13 we show as well the mean displacements $\|\mathscr{D}_t\|$ recovered by the registration algorithms. Note that the maximum mean displacements correspond to early systole and late diastole, where the strain is maximum, which confirms once again that the estimation of the displacement fields is adequate in the real case. Comparing the mean

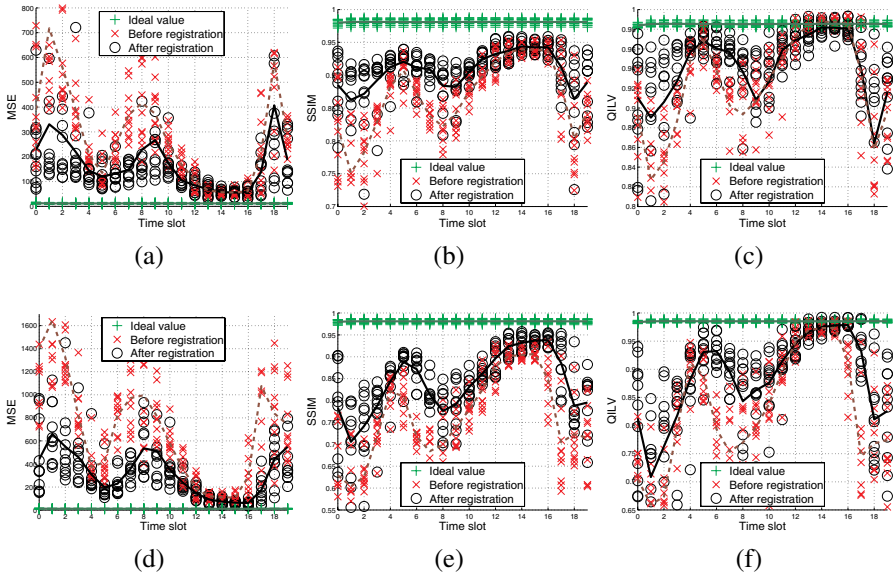



**Fig. 12** Similarity measures for each time slot and slice (MSE –a, d–, SSIM –b, e– and QILV –c, f–) for Block-Matching algorithm case. Top row: we register slot $t$ with slot $t+1$. Bottom row: slot $t$ with $t+2$. For each case, we superimpose the values for the 14 slices in the same time slot together with the average value for all slices (solid and dashed lines). The results of the registrations are compared to the similarity before the registration and to the ideal value that could be achieved if the estimation of the deformation field were exact.

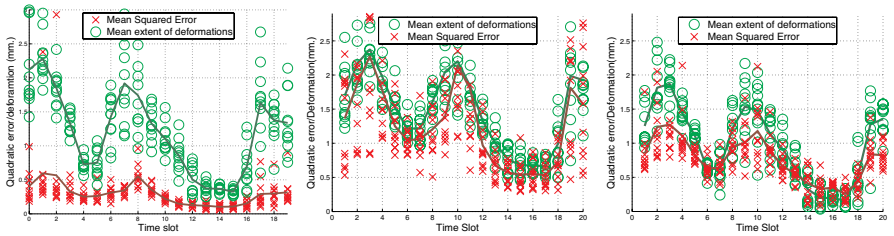
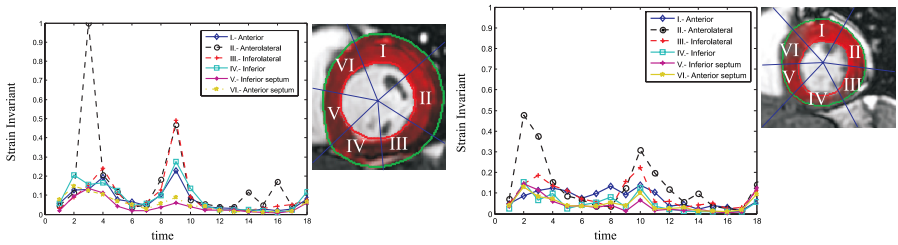



**Fig. 13** Demons Consistency checking of the deformation fields. We present mean values of $\mathscr{M} = \|\mathscr{D}_{t,t+1} + \mathscr{D}_{t+1,t+2} + \mathscr{D}_{t+2,t}\|$ for all slices and time slots for patient 10, together with the mean values of the displacements, $\|\mathscr{D}_t\|$. The computation has been restricted to a ROI containing the myocardial wall.

displacements and the mean consistency errors, the latter are lower, so we may conclude that the registration algorithms are able to recover in real data the deformation field with an error in the order of the mean discrepancies (consistency errors).




(a) Normal Patient (b) Patient with Akinesia

**Fig. 14** (a) Normalized strain invariant $I_t$ of a normal patient in a cardiac cycle. (b) Normalized strain invariant of a patient with a diagnosed Akinesia in the inferior area. The higher strain at the beginning and in the mid heart cycle shows the systole and diastole moments. In general, normal patient shows higher strain through all the heart cycle. In the Inferior and Inferolateral segments show an even lower strain in the whole cycle, especially in diastole, due to Akinesia.

## *8.3 Strain Rate Tensor Estimation Performance*

In order to see whether ST estimation provides a good description of the dynamic behavior of the heart, a classification was performed using the dynamic range of contraction and the maximum of the strain invariant $I_t = \lambda_1^2 + \lambda_2^2$ as input data. A Multilayer Perceptron with 3 layers was used with 2 neurons in the first layer, 2 in the second, and 1 in the output layer [16]. The network was trained with a Bayesian Regularization of the weights using the Levenberg-Marquardt algorithm. Testing was performed with a leave-one-out strategy.

A correct rate of 76.67% was reached by classifying into Normal and Abnormal (Hypokinesia or Akinesia) segments with the LCC Demons algorithm. We decided to use this algorithm because of its better performance in the consistency checking. This result makes clear that the SRT estimation can be considered as a feature for classifying.

In order to test the coherence of the strain estimation, a representation of the invariant $I_t = \lambda_1^2 + \lambda_2^2$ in each of the zones was performed for a patient who has an Akinesia diagnosed in the inferior area of the mid zone compared to a normal patient (see Fig. 14). In this representation, the strain rate shows its higher values at the beginning of the cycle (early systole) and in the mid heart cycle, when diastole begins. Curves obtained for the inferior and inferolateral, show a low strain through the cardiac cycle even on systole and diastole compared to the normal patient, so the zone with Akinesia presents low strain as it is expected.

Fig. 15 shows a sequence of the heart cycle represented with tensors drawn in the myocardial wall. The sequence is zoomed in to analyze the behavior of tensors in the Anterior and Anterior Septum area of the wall. The intensity of the deformation field is represented by a red level added to the gray level of the image. In the first frame, tensors are oriented to the endocardium contour showing elongations in that direction due to the contraction of the wall in the radial direction. An increased strain in the anterior-septum area is represented in the second frame with

overlapped tensors in the direction of the deformation. The third frame shows the myocardium is still contracting as an effect of the inertial movements. This behavior of the strain is coherent with the strain expected all over the cardiac cycle, where the strain peaks are in early diastole (when contraction begins) and in systole (when dilatation begins).

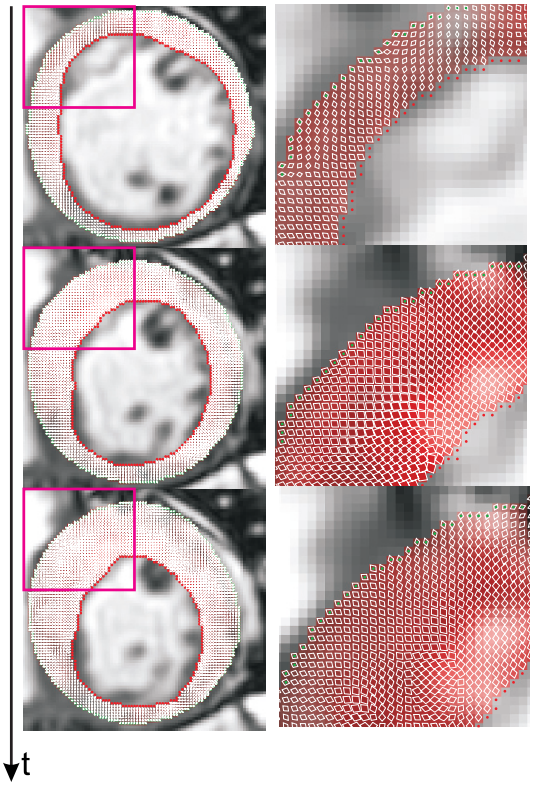



**Fig. 15** Sequence of the heart cycle with tensors represented in the myocardial wall. Right sequence corresponds to the zoomed Anterior and Anterior Septum area. The intensity of the deformation field is superimposed to the gray level of the image. The sequence shows how the strain increases in early systole until it reaches its maximum represented in the second frame with overlapped tensors. After reached its maximum, the strain diminishes.

## 9 Conclusion

In this chapter we have presented a methodology to estimate ST fields when no golden standard is available. With the proposed methodology, some parametric and non-parametric registration algorithms were used and tested.

To overcome the absence of a golden standard, a phantom was developed to test the performance of the registration algorithms. First, three similarity measures were used to demonstrate that registration algorithms provide a good registration by means of comparing the registered image and the static image. However this

method does not provide information about the error in the deformation field so a Consistency Checking method was additionally proposed when no golden standard is available.

This methodology has proved to provide an upper bound of the error as showed all the tests of registration algorithms. Moreover, when real data is used, results showed that the mean consistency error were below the mean displacements of the deformation fields provided by the registration algorithms so we conclude that the registration algorithms are able to recover the deformation field with an error in the order of the mean consistency error (as an upper bound).

As we have demonstrated the SRT field can be estimated from conventional MRI, it is worth to show whether this information is useful for diagnosis. With this idea in mind, we decided to use information from the SRT field and deformation field as inputs to classify between normal and abnormal patients. Results on real data yielded a correct classification rate of 76.76%. This result confirms that the ST fields estimated from the deformations provided by the registration algorithm can be used to detect abnormalities according to the professional diagnosis with good results. The study of strain in a patient with diagnosed Akinesia shows a coherent behavior of the strain curves that allows to detect systole, diastole and the areas with less strain. This classification study, though, should be extended taking into consideration information related to the ejection fraction and other classical measures which probably will increase the classification rate.

Additionally, a visualization scheme was proposed in order to visualize the behavior of the ST field in each time slot. Tensor visualization is consistent with the deformation of the LV wall through the cardiac cycle showing an elongation in the radial direction when endocardium is compressing, and a contraction when endocardium is expanding.

**Acknowledgements** The authors acknowlege the Spanish Comisión Interministerial de Ciencia y Tecnología for research grant TEC 2007-67073/TCM, the Fondo de Investigaciones Sanitarias for grant PI-041483 and the Junta de Castilla y León for grants VA026A07 and VA027A07, the ICICOR (Instituto de Ciencias del Corazón, Valladolid, Spain) and José Sierra (Centro PET Recoletas Valladolid) for the expertise exchange as well as the data provided to carry out this research and the Ministerio de Ciencia e Innovación for grant BES-2008-002109 of the research project MTM2007-63257.

## References

1. Aja-Fernández, S., Alberola-López, C., Westin, C.F.: Noise and signal estimation in magnitude MRI and Rician distributed images: A LMMSE approach. IEEE Tran. Image Proc. **17**(8), 1383–1398 (2008)
2. Aja-Fernández, S., San-José-Estépar, R., Alberola-López, C., Westin, C.F.: Image quality assessment based on local variance. In: Proc. IEEE EMBS, pp. 4815–4818. New York, NY, USA (2006)
3. Amini, A.A., Chen, Y., Elayyadi, M.: Tag surface reconstruction and tracking of myocardial beads from SPAMM-MRI with parametric B-spline surfaces. IEEE Tran. Med. Imaging **20**, 94–103 (2001)
4. Axel, L., A., M., Kim, D.: Tagged magnetic resonance imaging of the heart: a survey. Med. Image Anal. **9**, 376–393 (2005)

5. Axel, L., Dougherty, L.: MR imaging of motion with spatial modulation of magnetization. Radiol. **171**, 841–845 (1989)
6. Axel, L., Morton, D.: Correction of phase wrapping in magnetic resonance imaging. Med. Phys. **16**, 284–287 (1989)
7. Cachier, P., Pennec, X.: 3D non-rigid registration by gradient descent on a Gaussian-windowed similarity measure using convolutions. In: Proc. MMBIA, pp. 182–189. Hilton Head Island, SC, USA (2000)
8. Chandrashekara, R., Mohiaddin, R.H., Rueckert, D.: Analysis of 3-d myocardial motion in tagged mr images using nonrigid image registration. IEEE Tran. Med. Imaging **23**, 1245–1250 (2004)
9. Chandrashekara, R., Mohiaddin, R.H., Rueckert, D.: Cardiac motion tracking in tagged MR images using a 4D B-spline motion model and nonrigid image registration. In: Proc. IEEE Biomed. Imaging: Nano to Macro, pp. 468–471. Arlington, VA, USA (2004)
10. Chandrashekara, R., Mohiaddin, R.H., Rueckert, D.: Comparison of cardiac motion fields from tagged and untagged MR images using nonrigid registration. In: 3rd Int. Workshop FIMH, *LNCS*, vol. 3504, pp. 425–433. Barcelona, Spain (2005)
11. Chavez, S., Xiang, Q.S., An, L.: Understanding phase maps in MRI: A new cutline phase unwrapping method. IEEE Tran. Med. Imaging **21**, 966–977 (2002)
12. Chen, Y., Amini, A.A.: A map framework for tag line detection in spamm data using markov random fields on the b-spline solid. IEEE Tran. Med. Imaging **21**, 1110–1122 (2002)
13. Fischer, S.E., McKinnon, G.C., Maier, S.E., Boesiger, P.: Improved myocardial tagging contrast. Magn. Reson. Med. **30**, 191–200 (1993)
14. Gatehouse, P.D., Keegan, J., Crowe, L.A., Masood, S., Mohiaddin, R.H., Kreitner, K.F., Firmin, D.N.: Applications of phase-contrast flow and velocity imaging in cardiovascular MRI. Eur. Radiol. **15**, 2172–2184 (2005)
15. Guttman, M.A., Zerhouni, E.A., McVeigh, E.R.: Analysis of cardiac function from MR images. IEEE Comput. Graphics Appl. **17**(1), 30–38 (1997)
16. Haykin, S.: Neural Networks: A Comprehensive Foundation (2nd Edition). Prentice Hall (1998)
17. Jehenson, P., Westphal, M., Schuff, N.: Analytical method for compensation of eddy-current effects induced by pulsed magnetic field gradients. Med. Phys. **16**, 284–287 (1989)
18. Ledesma-Carbayo, M.J., Bajo, A., Santa-Marta, C., Perez-David, E., Garcia-Fernandez, M.A., M. Desco, A.S.: Fully automatic cardiac motion estimation from tagged MRI using non-rigid registration techniques. Com. Card. **33**, 305–308 (2006)
19. Lee, S., Wolberg, G., Shin, S.: Scattered data interpolation with multilevel B-splines. IEEE Tran. Vis. Comput. Graphics **3**(3), 228–244 (1997)
20. Moore, C.C., Lugo-Olivieri, C.H., McVeigh, E.R., Zerhouni, E.A.: Three dimensional systolic strain patterns in the normal human left ventricle: characterization with tagged mr imaging. Radiol. **214**, 453–466 (2000)
21. Mosher, T.J., Smith, M.B.: A DANTE tagging sequence for the evaluation of translational sample motion. Magn. Reson. Med. **15**, 334–339 (1990)
22. Netsch, T., van Muiswinkel, A.: Quantitative evaluation of image-based distortion correction in diffusion tensor imaging. IEEE Tran. Med. Imaging **23**(7), 789–798 (2004)
23. Osman, N.F., Kerwin, W.S., McVeigh, E.R., Prince, J.L.: Cardiac motion tracking using cine harmonic phase (HARP) magnetic resonance imaging. Magn. Reson. Med. **42**, 1048–1060 (1999)
24. Ozturk, C., Derbyshire, J., McVeigh, E.: Estimating motion from MRI data. Proc. IEEE **91**, 1627–1648 (2003)
25. Pelc, N.J., Sommer, F.G., Li, K.C., Brosnan, T.J., Herfkens, R.J., Enzmann, D.R.: Quantitative magnetic resonance flow imaging. Magn. Reson. Q. **10**, 125–147 (1994)
26. Rogers Jr., W.J., Shapiro, E.P., Weiss, J.L., Buchalter, M.B., Rademakers, F.E., Weisfeldt, M.L., Zerhouni, E.A.: Quantification of and correction for left ventricular systolic long-axis shortening by magnetic resonance tissue tagging and slice isolation. Circul. **84**, 721–731 (1991)

27. Ryf, S., Spiegel, M.A., Gerber, M., Boesiger, P.: Myocardial tagging with 3d-cspamm. J. Magn. Reson. Imaging **16**, 320–325 (2002)
28. Schiller, N.B., Shah, P.M., Crawford, M., DeMaria, A., Devereux, R., Feigenbaum, H., Gutgesell, H., Reichek, N., Sahn, D., Schnittger, I.: Recommendations for quantitation of the left ventricle by two-dimensional echocardiography. J. Am. Soc. Echocardiogr. **2**, 358–367 (1989)
29. Selskog, P., Heiberg, E., Ebbers, T., Wigström, L., Karlsson, M.: Kinematics of the heart: strain-rate imaging from time-resolved tree-dimensional phase contrast MRI. IEEE Tran. Med. Imaging **21**(9), 1105–1109 (2002)
30. Thirion, J.P.: Image matching as a diffusion process: an analogy with maxwell's demons. Med. Image Anal. **2**(3), 243–260 (1998)
31. Thunberg, P.: Accuracy and reproducibility in phase contrast magnetic resonance imaging. Ph.D. thesis, Linköping University, Linköping, Sweden (2004). Dissertation No. 899, ISBN 91-85295-41-8
32. Tristán, A., Arribas, J.I.: A fast B-spline pseudo-inversion algorithm for consistent image registration. In: Comput. Anal. Images Patterns, *Lecture Notes in Computer Science*, vol. 4673, pp. 768–775 (2007)
33. Tristán-Vega, A., Vegas-Sánchez-Ferrero, G., Aja-Fernández, S.: Local similarity measures for demons-like registration algorithms. In: IEEE Int. Symp. Biomed. Imaging. Nano to Macro., pp. 1087–1090. Paris, France (2008)
34. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: form error visibility to structural similarity. IEEE Tran. Image Proc. **13**(4), 600–612 (2004)
35. Wuensche, B., Lobb, R.: A toolkit for the visualization of stress and strain tensor fields in biological tissue. In: Proc. VIP, pp. 6–15. Sydney, Australia (1999)
36. Xiang, Q.S.: Temporal phase unwrapping for CINE velocity imaging. Magn. Reson. Imaging **5**, 529–534 (1995)
37. Xiao, G., Brady, J., Noble, J., Burcher, M., English, R.: Nonrigid registration of 3-D free-hand ultrasound images of the breast. IEEE Tran. Med. Imaging **21**, 405–412 (2002)
38. Young, A.A., Axel, L.: Three-dimensional motion and deformation of the heart wall: estimation with spatial modulation of magnetization –a model-based approach. Radiol. **185**, 241–247 (1992)
39. Zerhouni, E.A., Parish, D.M., Rogers, W.J., Yang, A., Shapiro, E.P.: Human heart: tagging with MR imaging–a method for noninvasive assessment of myocardial motion. Radiol. **169**, 59–63 (1988)
40. Zitová, B., Flusser, J.: Image registration methods: a survey. Im. Vis. Comp. **21**, 977–1000 (2003)

# Strain Tensor Elastography: 2D and 3D Visualizations

Darío Sosa-Cabrera, Karl Krissian, Javier González-Fernández, Luis Gómez-Déniz, Eduardo Rovaris, Carlos Castaño-Moraga and Juan Ruiz-Alzola

**Abstract** Elastography measures the elastic properties of soft tissues using principally ultrasound (US) or magnetic resonance (MR) signals. The elastic behavior of tissues can be analyzed with tensor signal processing. Different approaches have been developed to estimate and image the elastic properties in the tissue. In ultrasound elastography, the estimation of the displacement and strain fields is mostly based on measures computed from the Radio Frequency signals, such as time-domain cross-correlation. We propose to estimate the displacement field from two consecutive B-mode images using a multiscale optical flow method. The tensor strain field can then be plotted as ellipsoids, visualizing in a single image the standard scalar parameters that are usually represented separately. This technique can offer physicians the possibility of extracting new discriminant and useful parameters related to the elastic behavior of tissues. Although clinical validation is still needed, our experiments from finite element and ultrasound simulations display consistent and reliable results.

## 1 Introduction

Many pathologies such as breast cancer or cardiovascular disfunctions are highly correlated with changes in the elastic properties of the surrounding tissues. Being able to efficiently assess these changes either quantitatively or by providing efficient visual tools can lead to early detection of these diseases and better diagnosis and treatment. For example, the high prevalence of breast cancer make screening mammography being recommended every year for women older than 40. However, the use of X-ray radiography can increase the probability of tumors and can also give false alarms. Elasticity Imaging or Elastography measures the elastic properties of

Center for Technology in Medicine, Dept. Señales y Comunicaciones,
University of Las Palmas de Gran Canaria, SPAIN
e-mail: {dario,karl,jgonzalez,lgomez,erovaris,jruiz}@ctm.ulpgc.es

soft tissues in response to a controlled external stimulus by means of ultrasound or magnetic resonance imaging. Ultrasound (US) elastography is an imaging technique with high specificity that can potentially help eliminate a large number of biopsies for breast cancer detection without any secondary effects. In this work, we focus on ultrasound elastography, and we show how the elastic properties of tissues can be analyzed using tensorial signal processing tools.

The background section reviews the main concepts and protocols used in ultrasound elastography, basic tensor elasticity concepts, and the existing approaches for computing the displacement fields and visualizing tensorial images. Afterwards, the basis of strain tensor elastography is exposed. The methods used to carry out the experiments in this chapter are described in section 4. In sections 5 and 6, we present new visualizing techniques for 2D and 3D elastography respectively, and finally, we discuss new perspectives and conclude.

## 2 Background

### *2.1 Elastography*

Over the years, palpation has been a standard medical practice that relies on qualitative estimation of tissue elasticity. It has been used as a screening method for the detection of breast, prostate, thyroid, and liver abnormalities, based on the fact that the pathological state of soft tissues is normally correlated with changes in stiffness [24]. The concept of elasticity imaging was developed as a qualitative and quantitative methodology to map tissue elasticity, therefore adding new clinical information to the interpretation of ultrasound, computed tomography or other scans [16]. In 1991, Ophir and co-workers introduced an ultrasound technique (named ultrasound elastography), for imaging soft tissues stain profiles caused by a quasi-static compressive force which produces a relative deformation into the tissue [4, 22]. Since then, ultrasound elastography has become an emerging medical imaging tool to assist in the diagnoses of pathology involving tissue stiffness such as prostate tumours [25] and breast lesions [8, 9]. Elastography can be applied to any tissue accessible ultrasonically and which can be subjected to compression. The compression is often applied externally, whereas other techniques take advantage of natural sources of compression such as arterial pulsation [7] or respiration.

Several ultrasound-based techniques have been developed according to the type of excitation chosen: quasi-static compression [14], dynamic compression [20], or pulsed excitations [35]. In quasi-static compression elastography, data are recorded before and after a controlled mechanical compression has been applied to the tissue. Dynamic and pulsed techniques on the other hand make use of a pulsed low-frequency or a pulsed high-frequency excitation, respectively. Quasi-static compression elastography is specially indicated for the cases where there is a strong need to

determine axial displacements and for the cases related to reconstructing the inverse problem [5].

Quasi-static techniques appear as a natural choice for free-hand ultrasound elastography. The main characteristic of the free-hand implementation is twofold. First, it provides superior ease of use and versatility, and two, it has no need for additional hardware and in this manner can be include in a commercial system at a low cost. Furthermore, with the addition of a 3D position sensor, 3D ultrasound elastography can be carried out by scanning volumes by translating the probe in the elevational direction to sweep out the volume [17].

The complete description of an elastography technique deals with the estimation of the axial displacements and the strain field derived from sonograms. A big effort has been dedicated to this problem for over a decade [6, 27] and, generally, elastograms with high Signal-to-Noise Ratios (SRN) are obtained from cross-correlation methods. Traditionally, the correlation algorithms are applied to selected fractions of each Radio Frequency (RF) line by means of shifting windows in order to fit locally the pre- and post-compressed A-line pairs. Strain estimation is a non-stationary process, due to the fact that the pre-compression and post-compression RF echo signals (A-scan mode) are jointly non-stationary. In order to formulate the problem, it is convenient to approximate both RF signals to be jointly stationary, which can be assumed when small displacements are produced in the tissue (it is normal to consider displacements no longer than 1%). This assumption makes possible to de-correlate the RF signals in order to estimate the displacement. Additional stretching of the pre-compressed RF lines improves the efficiency of the correlation algorithms [21]. It has been shown that the strain estimation improves when using a local stretching approach [34] instead of a global strategy. Axial displacements are obtained from the strain field correlating each RF pre- and post-compressed A-scan lines. The estimation of lateral displacements require that pairs or even groups of A-scan lines be compared, which implies a high computational cost. A major disadvantage when using free-hand ultrasound elastography is that consecutive ultrasound frames are non-coplanar, so the level of signal de-correlation is increased. On the contrary, when using a mechanic actuator, the value of the deformation and pressure over the tissue are known precisely and, therefore, that information can be added to the correlation algorithms as a value to be used in the pre-stretching phase. Due to the nature of the free-hand ultrasound elastography technique, the knowledge about the tissue deformation comes from a mere estimation, resulting in a less accurate and more erroneous data source.

## *2.2 Tensor Elasticity*

The theory of elasticity provides a consistent set of equations that can be solved in order to obtain a unique point-wise description of the tensorial distribution experienced at each internal point (stresses) and displacements and deformations (strains) caused for a particular loading (applied forces) and geometry. If an elastic body is

compressed, its internal points are affected resulting in a tensional state determined by the stress tensor. This state produces a deformation, expressed by the strain tensor, which is related to the displacement.

When a stimulus is applied to an elastic solid, the stress tensor completely defines the state of tension at a point in the body. It is a rank-2 symmetric tensor $T$ that permits to calculate the tension vector $\mathbf{t}$ in any orientation $\mathbf{n}$ as the scalar product of a vector and a tensor

$$\mathbf{t} = T\,\mathbf{n}. \tag{1}$$

For the sake of simplicity, we will consider rank-2 tensors as matrices and use their corresponding matricial notations. Deformation caused by a stimulus is related to the displacement vector field $\mathbf{u}$ between the initial and the final positions of the points:

$$\mathbf{u}(x,y,z) = u_x(x,y,z)\mathbf{x} + u_y(x,y,z)\mathbf{y} + u_z(x,y,z)\mathbf{z}. \tag{2}$$

The displacement gradient $\nabla\mathbf{u}$ is a tensor (Jacobian matrix) expressed as:

$$\nabla\mathbf{u} = \begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} & \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} & \frac{\partial u_z}{\partial y} & \frac{\partial u_z}{\partial z} \end{pmatrix}, \tag{3}$$

that can be decomposed as the sum of the symmetric strain tensor $\Gamma$ with the normal and shearing strains and the anti-symmetric vorticity tensor $\Omega$ [18]:

$$\nabla\mathbf{u} = \Gamma + \Omega, \tag{4}$$

where $\Gamma = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^t)$ and $\Omega = \frac{1}{2}(\nabla\mathbf{u} - \nabla\mathbf{u}^t)$. The strain tensor $\Gamma$ measures the changes of shape, while the vorticity tensor $\Omega$ informs about local rotations. Assuming small deformations and no rotations, which is a reasonable assumption in our case of quasi-static compression, we can focus on the strain tensor component. The (unitary) strains are expressed as:

$$\Gamma = \varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right), \tag{5}$$

where the index notation $\{x_1, x_2, x_3\}$ stands for $\{x, y, z\}$ and $(u_1, u_2, u_3)$ for $(u_x, u_y, u_z)$. The unitary angular deformation or shear-strain which characterizes the shearing is denoted $\gamma$ and for $i \neq j$, it is defined as $\gamma_{ij} = 2\varepsilon_{ij}$. A more extended description of the theory can be found in [31].

Different parameters describing different elastic or viscoelastic properties of the materials have been investigated and visualized:

- Longitudinal strains: $\varepsilon_x$, $\varepsilon_y$ and $\varepsilon_z$.
  Depending on the axis disposition, they are named axial strain, lateral strain and elevational strain. In the standard elastographic set-up, axial and lateral strains correspond to the plane of the 2D probe, the former in the compression axis. The elevational direction is used to define the out-of-plane parameters.

- Shear strains: $\gamma_{xy}$, $\gamma_{xz}$, $\gamma_{yx}$.
  Normally we refer with shear strain to that occurring in the plane defined by the 2D probe.
- Poisson's ratio. $\nu = \frac{-\varepsilon_l}{\varepsilon_a}$.
  Some researchers refers to the lateral-to-axial strain ratio $= \frac{\varepsilon_l}{\varepsilon_a}$.
  In both cases, $l$ and $a$ subindex meaning lateral and axial longitudinal strains.
- Axial-shear strain $= \frac{\partial u_x}{\partial y}$, considering $x$ the axial axis and $y$ the lateral. Mathematically, this parameter is known as the *cross-derivative xy*.

Some examples are the axial strain, the shear strain, Young's modulus, shear modulus, Poisson's ratio, viscosity, shear viscosity, lateral-to-axial strain ratio, etc. In [31], other parameters with clinical potential are presented. One of them is the vorticity that can be extracted from the rotation tensor and is related to the amount of circulation or rotation. Another parameter used to visualize elastic properties is the *strain index*, related to the trace of the tensor. The latter is not described physically in the theory of elasticity, and is more related to the tensorial, mathematical or algebraic properties of the tensor. Other different properties of the strain and/or stress tensors may be investigated with this tensorial approach in order to obtain more tools for the clinicians to help the diagnosis of different pathologies related with changes in the elastic properties of tissues. As there are many pathologies, with their specific behaviors and physical reactions, some parameters can suit better to one pathology than another.

Scalar visualizations can be represented in gray scale (the darker areas representing stiffer regions, i.e. less strain), or in pseudo-color images. The gray scale has been the standard for ultrasound imaging in general and also for elasticity imaging. However, the color images for the elastograms are becoming widely used. This is due to the fact that elastograms are usually represented beside the B-mode image or onto it. Therefore, it is easier to appreciate the colored image in contrast to the gray scale one. In this work, both of them are used and, in any case, the clinician should decide which one brings him the more relevant information.

## *2.3 Displacement Estimation*

Ultrasonic strain assessment methods rely on the estimation of displacements between ultrasound signals acquired before and after mechanical compression. Once an array of one, two or even three dimensional displacements estimates has been calculated, strain is computed from a spatial derivative of the displacement to produce a strain image. These methods require highly accurate estimates of the small deformations that occur between successive frames in an ultrasound scan. Several methods have been described in the literature. Conventionally, each displacement estimate is produced by identifying the closest matching windows of the pre-deformed frame in the post-deformed frame. Either radio frequency (RF) or complex baseband signals are used for accuracy.

A measure of similarity is required in these window-matching approaches. Normalized correlation coefficient has been traditionally utilized, although many alternative estimators have been reported with nearly identical performance figures [37], such as the sum of squared differences (SSD), sum of absolute differences (SAD), and phase root seeking of the complex cross-correlation function [27]:

$$SSD = \sum_{t=i}^{t=i+W} (x_1(t) - x_2(t))^2 \tag{6}$$

$$SAD = \sum_{t=i}^{t=i+W} |x_1(t) - x_2(t)|, \tag{7}$$

where $x_1(t)$ and $x_2(t)$ denote a pair of signals which represent the reference and the delayed signals received by the transducer, which may have been de-correlated by physical processes and additive noise introduced from electronic sources. Under these conditions, a pattern matching algorithm can be used to estimate the shift between $x_1(t)$ and $x_2(t)$. The normalized-cross correlation (NCC) method determines the displacement that maximizes normalized correlation between pairs of windowed ultrasonic data over a predefined search region (Fig. 1). NCC coefficient can be defined as [40]:

$$NCC = \frac{\sum_{t=i}^{t=i+W} x_1(t)\, x_2(t)}{\sqrt{\sum_{t=1}^{t=i+W} x_1^2(t)} \sqrt{\sum_{t=1}^{t=i+W} x_2^2(t)}}, \tag{8}$$

where $t$ is the iteration index, $i$ is the in-window index and $W$ is the window size.

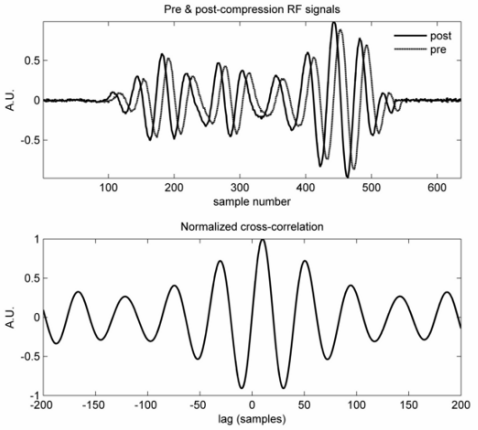



**Fig. 1** Precompressed and postcompressed RF data and the corresponding normalized cross-correlation.

The corresponding maximum of the NCC coefficient is an indicator of the accuracy of the displacement estimate. To estimate displacements with subsample ac-

curacy, parabolic, cosine or sync interpolation is required [3]. Due to its simplicity, parabolic interpolation is frequently used [3].

Alternatively, to find the absolute maximum of the cross-correlation function, time shift can be estimated from the phases of either the analytic (complex) RF signals or base-banded signals. Pesavento et al. [26] proposed a Phase Root Seeking algorithm to find the phase root of the correlation function of the analytic signals. This root is found using an iterative formula, derived from a Newton iteration, in which the derivative of the phase was approximated by the transducer center frequency.

In addition, other methods such as adaptive stretching [2] and spectral cross-correlation [36], estimate the strain directly without involving the use of gradient operators but at huge computational cost and loss of precision, although they have been shown to produce quality elastograms for large applied strains. Spectral methods estimate the strain field in a direct manner [1, 11] based on the Fourier scaling properties, which state that compression of a temporal signal (RF in this case) produces a proportional expansion in the corresponding power spectrum. In this way, spectral methods formulate the problem in the frequency domain, while the conventional methods use the temporal domain.

## 3 Strain Tensor Elastography (STE)

An important contribution of this work is the visualization of the full strain tensor for elasticity imaging. Traditionally this medical imaging modality has visualized scalar parameters components of the strain and the elastic moduli tensors or related parameters. We propose a new technique, Strain Tensor Elastography, hereinafter STE, which visualizes in one image the standard scalar parameters that are usually represented separately in elasticity imaging. By using this technique physicians would have additional information. Besides, it offers them the possibility of extracting new discriminant and useful parameters related to the elastic behavior of tissues. Synthetic experiments from finite element and ultrasound simulations, and commercial phantoms are presented in this section. This tensorial approach as well as its visualization, can be extended to other elasticity imaging techniques.

The strain tensor will be used for the new visualizations presented in this chapter. It gives a relative measure of the elasticity of different tissues that can be used for the diagnosis of several pathologies. As opposed to the estimation of the elastic modulus tensor, the estimation of the strain tensor does not require any assumption about isotropy or problem configurations to shorten the independent constitutive equations. Also, it does not add computational cost to the procedure. The visualization schemes applied here to the strain tensor can equally be applied to the stress tensor.

Tensors and tensor fields are basic tools in differential geometry and physics that describe geometric and physical quantities which remain invariant under coordinate transformations, as the ones we face in the theory of elasticity. In a 3D Euclidean

space, such as ordinary physical space, the number of components of a tensor is $3^n$, denoting $n$ the dimensionality of the space. The strain and stress tensors are both second order tensors, therefore defined by nine components in 3D and four components in 2D. As this two tensors are by definition symmetric, we have six independent components for the 3D case and three for the 2D case.

## *3.1 Filtering the Strain Tensor Field*

The estimated strain tensor field is noisy and susceptible to be filtered. The authors in [29] proposed a non-iterative anisotropic method to regularize vector and higher order tensor fields having in mind DT-MRI applications. The description that follows below has been inspired from that article. We extended its use to the strain tensor field, obtaining a more aligned and anisotropically averaged tensor field, while the edges are preserved. Results are shown in Fig. 2.

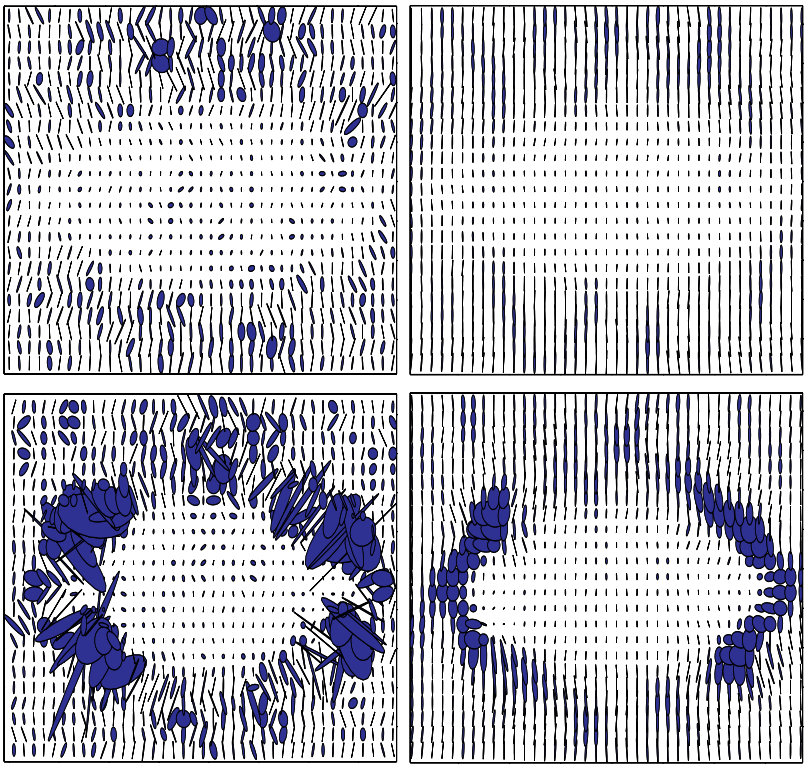


**Fig. 2** Tensorial images filtering pipeline. *Top*: bounded case (malign); *bottom*: unbounded case (benign). *Left*: noisy data (US simulation); *right*: anisotropically filtered data after regularization.

The method is based on first estimating the local structure of the field and, then, combining the outputs of a set of steerable filter basis responses to the tensor field, which are adaptively weighted depending on the local structure [28].

The filtering scheme for tensor fields is based on the independent application of the same scalar filter to every component, though the local structure tensors are obtained from the whole input tensor field and not from each independent component. We first give details of the scalar filter to be applied to each component. The scalar filter consists of a bank of steerable filters forming a basis: one of the filters is low-pass (ideally a Wiener filter), intended to get the coarse information, and the remaining ones form a set of high-pass filters oriented in different directions, that give the level of detail at each direction.

As the results obtained for our case tend to round the strain tensor ellipses, loosing the orientation although preserving the borders, we finally decided to filter each component of the tensor separately, what gave us better results. However the anisotropic filtering might yield better results optimizing it for our images through weighting, masks and different borders strategies.

### *3.2 Strain Tensor Visualization*

As mentioned before, most representations used in elasticity imaging are based on scalar parameters, which can be grouped in a matrix format to form the strain tensor (see equation 5). Each component of this tensor (4 for the 2D case and 9 for the 3D case), is obtained from the first order derivatives of the displacement vector field, through pre and post-compression operations. The derivatives are usually computed by means of a Least Square Estimator filter. Tensors formulations are not widely used in image processing and computer vision, but they are gaining a increasing interest. For example, in the field of biomedical research, they are used in application of diffusion tensor magnetic resonance imaging (DT-MRI) [10, 38], and in cardiac strain rate imaging (SRI) [30], where major efforts have been focused on representing the tensor field.

Although the DT-MRI visualization techniques are quite well developed, their application to strain tensor fields is not straightforward, since the strain tensor does not always satisfy the positive semidefinite condition. In our case, the relevant clinical information can be recovered from the absolute value of the tensor eigenvalues. Further explanation for the sign of the eigenvalues is detailed in section 5.

In this work, we applied two different displacement algorithms: the first one based on the standard elastography cross-correlation for the Radio Frequency (RF) A-lines, and the second one through the optical flow method applied to the B-mode images. Note that the B-mode consists of calculating the envelope for each RF A-line and this information is obtained directly from the sonogram image, in contrast to the RF-lines, which are not always available for the researcher (it depends on the ecograph hardware/software architecture). Representation of results by means of the strain tensor is independent to the applied method to estimate the strain. In Figs. 3

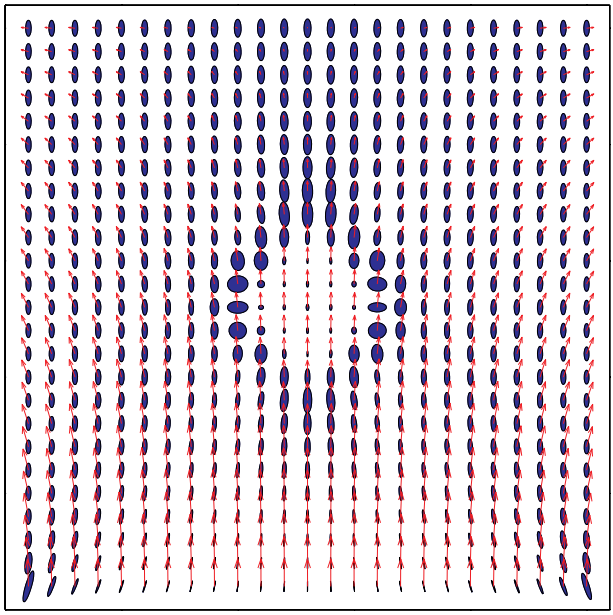


**Fig. 3** Displacement vector field overlaid on the strain tensor field for a virtual phantom, ideal case. Unbounded case (benign) with symmetric boundary conditions.

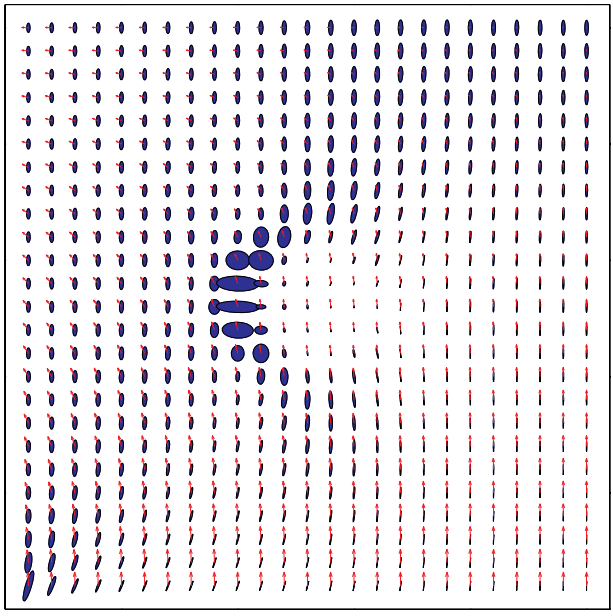


**Fig. 4** Displacement vector field overlaid on the strain tensor field for a virtual phantom, ideal case. Unbounded case (benign) with non symmetric boundary conditions.

and 4, we present results for a virtual phantom. As it can be seen, the vector field at each point of the region-of-interest (ROI) is superimposed on the strain tensor with different boundary conditions, i.e. the ideal case output of the Finite Element Method. These images show the potential of the tensorial representation and the information it provides.

# 4 Experiments

## *4.1 Field II Simulations*

Field II is a program for simulating ultrasound imaging using linear acoustics and summed spatial impulse responses through a series of MATLAB (Mathworks, USA) scripts [12].

Field II proposes a point scatterer approach for simulation. It approximates the medium as filled with elementary objects distributed randomly from which ultrasound deflects and disperses, i.e. *scatters*. The contributions of the spatial impulse responses at each scatterer are thus summed to form the simulated image. These US reflective points emulate the real tissue scatters in the body insonified by the US transducer.

Parameters Used in the Simulation

The probe parameters model an L5-12 MHz ultrasound probe (Prosonic Co., Korea) commonly used in breast imaging. The probe has a central frequency of 7.5 MHz, 80% fractional bandwidth, 128 physical elements and captures a linear image sector from the phantom image built with 200,000 scatterers distributed according to a Gaussian distribution. The simulated B-mode dynamic range was 42 dB. Sample rate for the simulations was 100 MHz. The mechanical deformation produced with the Finite Element (FE) simulation, was reproduced by displacing each scatterer. Synthetic pre- and post-compression B-mode images were generated using FIELD II ultrasound simulator.

## *4.2 Commercial Phantom*

Figure 5 shows a convention scan (a) and elastogram (b) of a breast elastography phantom (Model 059, Computerized Imaging Reference Systems, Inc., USA) which contains several solid masses that appear isoechoic to the simulated breast tissue under normal ultrasound, but are three times stiffer than the background. Depicted lesion is about 10 mm in size. This phantom was imaged with a linear array of

128 elements with a 7.5-MHz centroid frequency, 80% fractional bandwidth transducer of an Ultrasonix 500-RP ultrasound scanner (Ultrasonix Medical Corporation, Canada).

## 5 2D Strain Tensor Elastography

Visualization of the tensor field improves the understanding and interpretation of tensor data. In the last years, tensor field visualization has brought great interest within researchers, partly due to the rapid improvement of graphics hardware, and to recent advances in fiber visualization given by DT-MRI.

As in [30], we propose to visualize tensors as ellipsoids colored according to the sign of the largest eigenvalue representing stretch or compression in the principle direction. We chose a range of colors from blue to red for representing shortening to stretching respectively, when displaying the tensorial image alone or overlaid on a gray scale image. In the case where the tensorial image is displayed over a color image, the black color is used for shortening, allowing a distinction from the background colors.

The ellipses are aligned with the the principal directions of the tensor, given by its eigenvectors, and representing maximum and minimum values for the strain. This representation, called the Lamé ellipsoids, has been used to visualize the stress and the strain tensors in [23].

In our case, as in Figs. 5 and 2, all the ellipses appear in a unique color: blue or black. Later in this chapter, other visualization schemes are presented where the ellipses (ellipsoids in 3D) have different colors. Because we are applying a compression in the axial direction to an incompressible target, that deformation expands in the target to both lateral directions, namely the Poisson's effect. The axes of the ellipses aligned with the axial direction, where the compression is applied, have higher values. The strain, i.e. the derivative of the displacement, gives an idea of the change in the displacement. In the quasi-static elastographic set-up, this change is expected to be higher in the axial direction.

This representation visualizes the complete tensor in one plot giving information about magnitude, direction and the ratio between the strain principle values. This representation can detect possible abnormalities and regions with different local behavior than the surrounding.

Two sets of experiments are shown next. First, we present the results for a virtual phantom with the malign and benign tumors simulated. Afterwards, we report the experiments carried out with our optical flow estimation algorithm on physical phantoms.

## 5.1 Tumor Discrimination Experiments

Two different cases of boundary conditions between the inclusion (the simulated tumor) and its surrounding (the background) were studied: one of them with the inclusion loosely bounded to the background and the other fully connected, simulating respectively the benign and malign tumors. These experiments were carried out with virtual phantoms and displacements were estimated with cross-correlation of the RF A-lines of pre- and post-compression data.

The experiments used to investigate the vorticity, showed that an inclusion may rotate when a compression is applied, depending on the boundary conditions. The strain tensor is expected to behave differently for this two cases, i.e. when the inclusion is firmly bounded and when it is loosely bounded to the surrounding, especially at the boundaries of the tumor. We used the same experiments to test the capabilities of the standard Lamé visualizations of the strain tensor. The output of the finite element software is considered the ground truth for comparison of the ultrasound simulated phantoms (Figs. 5–7).

## 5.2 Optical Flow Experiments

For this set of experiments, we use a multiscale variational method to estimate the displacement field. Further information on this technique can be found in [32, 33].

Three experimental phantoms were devised to progressively take into account both the elastic response of the tissue and the dynamic range of conventional ultrasound images:

- Phantom A: Simulated data generated using Finite Element Analysis.
- Phantom B: Computer-simulated ultrasound images.
- Phantom C: A commercial breast elastography phantom, model 059, CIRS Inc., USA.

Phantom C contains several lesions three times stiffer than the background. In Fig. 8, the displacement field (red vectors) is overlaid on the B-mode image for phantom C. A green square delimits the ROI, whose elastograms are shown afterwards in Figs. 11 and 12. The scalar and tensorial elastograms for phantoms A and B, are plotted separetely in Fig. 9, while the plot of the ellipses overlaid on the axial strain for both of them, are presented in Fig. 10.

Besides the color-coded information about the stretching or shortening of the tissue locally, the ellipses present the principal directions of deformation, their magnitude, and the ratio between these magnitudes. While the scalar representations show the deformation in axial or lateral directions, the ellipses account for both the magnitude and the direction of the tissue deformation.

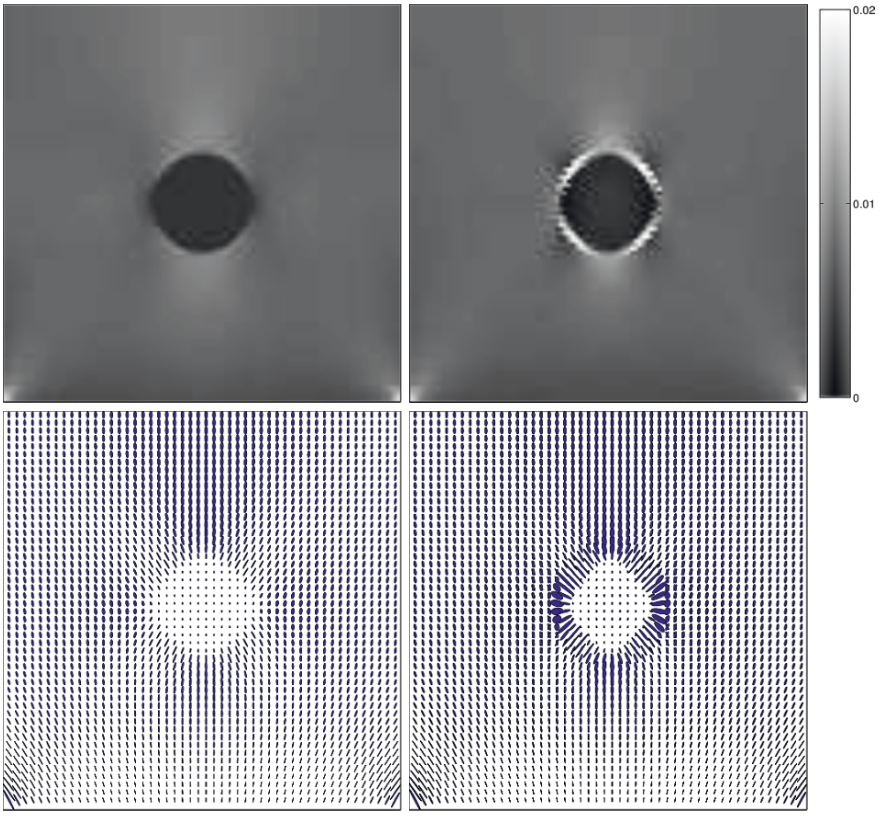



**Fig. 5** Scalar and tensorial images comparison for a virtual phantom, ideal case. *Top*: axial elastograms; (0.02 = 2%). *Bottom*: tensor elastograms. *Left*: bounded case (malign); *right*: unbounded case (benign).

## 6 3D Strain Tensor Elastography

For the 3D experiments, finite element simulations have been used. Because we compute three eigenvectors and eigenvalues for each voxel instead of two, the sign of the eigenvalues must be considered in a different way. The aim of this section is to give a set of tools and considerations for a useful 3D visualization of the strain tensor.

In order to have usability for the clinicians, some parameters such as the plot of the whole volume or different slices through the axial, lateral and elevational planes, the size of the ellipsoids, the orientation of the views, etc., should be determined by the physician.

We used the an interface generated by the free software AMILab [15], to allow tuning the parameters for a specific application. A screenshot is presented in Fig. 13.

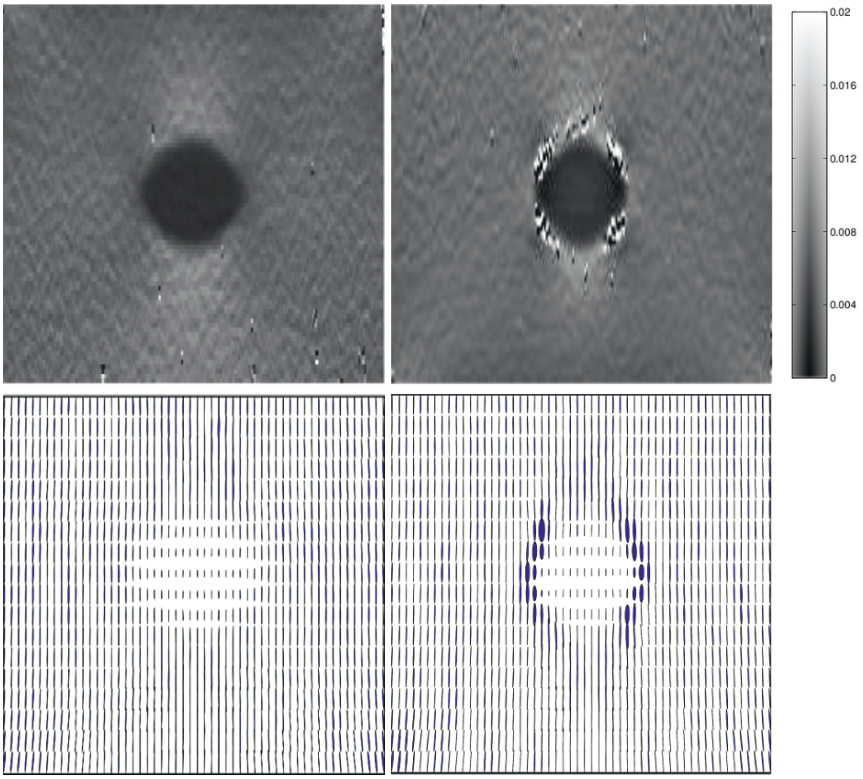



**Fig. 6** Scalar and tensorial images comparison for an US simulation of the virtual phantom. *Top*: axial elastograms; (0.02 = 2%). *Bottom*: tensor elastograms. *Left*: bounded case (malign); *right*: unbounded case (benign).

In this section, we present visualizations corresponding to phantom A, an ideal case as we do not estimate the displacements but obtain them directly from the FEA. The visualizations presented hereafter are independent from the estimation algorithm used to obtain the displacement field.

## *6.1 Visualizations*

The ellipsoids represented at each voxel are Lamé visualizations of the strain tensor as described in section 5. In Figs. 14–16, several 3D visualizations of phantom A are presented.

Different coloring stragegies have been experimented. First, we colored the ellipses according to the following three parameters:

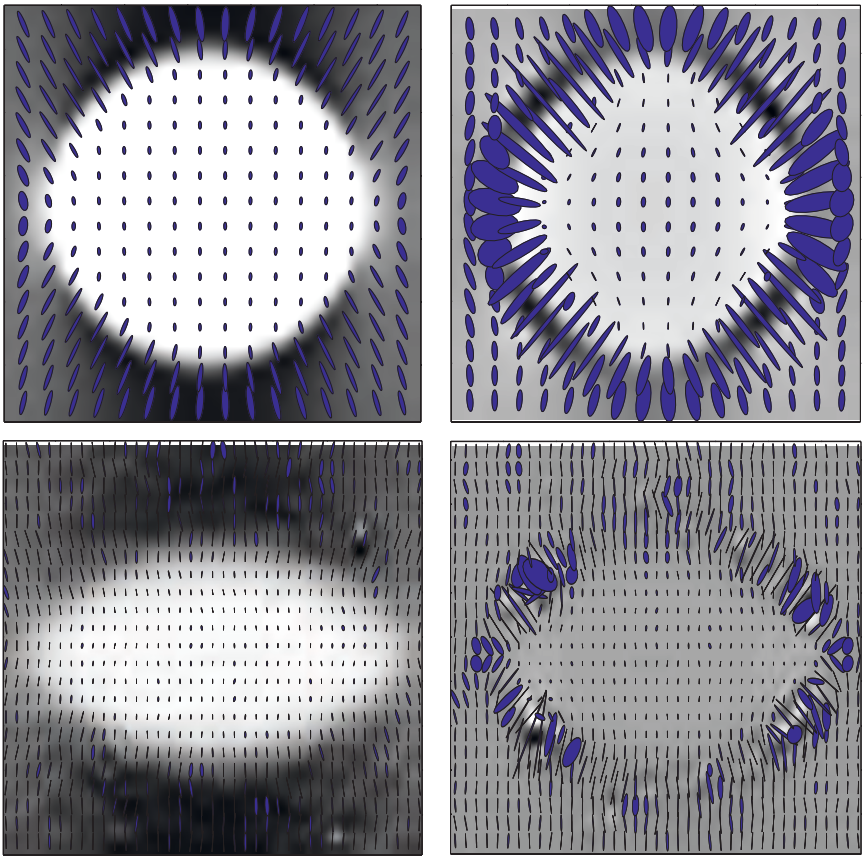


**Fig. 7** Tensorial images overlaid on the axial strain. Comparison of the virtual phantom between the Ideal case and the US simulation. *Top*: Ideal case; *bottom*: US simulation *Left*: bounded case (malign); *right*: unbounded case (benign). Note that the tensor field dimensions are different between the ideal case and the US simulated phantoms because several ultrasonic beams cover the areas corresponding to the nodes of the FE, resulting on the elliptical shape of the inclusions.

$$c_l = \frac{\lambda_1 - \lambda_2}{\sum_i \lambda_i},\ c_p = \frac{\lambda_2 - \lambda_3}{\sum_i \lambda_i},\ c_s = \frac{\lambda_3}{\sum_i \lambda_i} \tag{9}$$

where $c_l$, $c_p$ and $c_s$ stand for linear and planar anisotropy and isotropy coefficients, as described in [39]. The red, green and blue components of the color are set as proportional to $c_l$, $c_p$ and $c_s$ respectively.

Although the ellipsoids within a stiff object that is compressed inside a softer one (as it is the case of phantom A), are more spherical, having a bigger $c_s$ and a smaller $c_l$, this color coding permits a first discrimination of the inclusion from the background, and renders different colors within each tissue class.

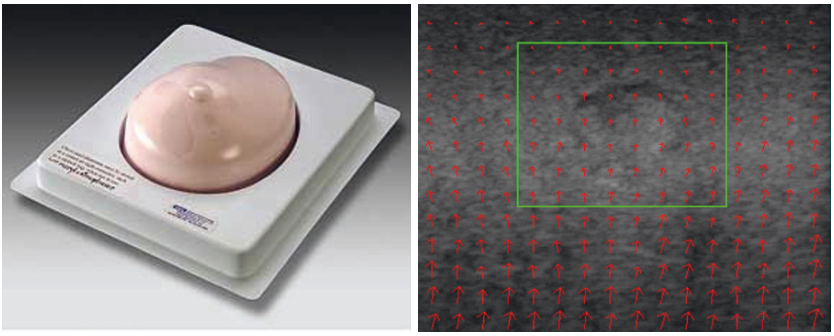


**Fig. 8** Left, picture of the commercial breast elastography phantom. Right, estimated displacement field overlaid on the pre-compression US B-mode image of the phantom C.

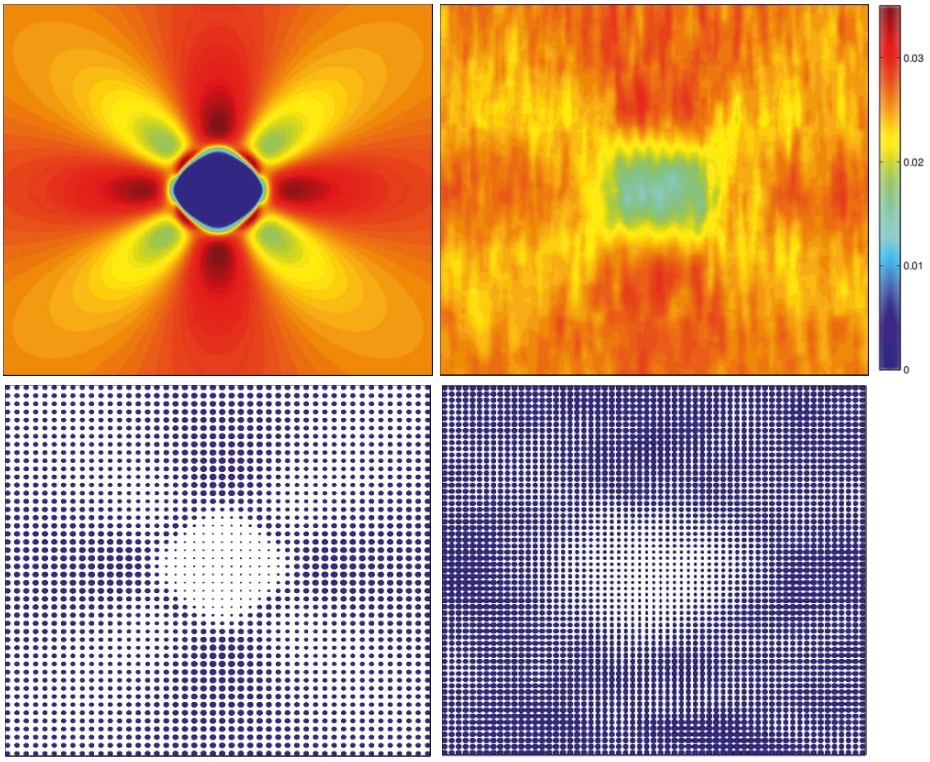



**Fig. 9** Scalar versus tensorial and ideal versus US-simulated elastograms. *Top*: axial elastograms; *bottom*: tensorial elastograms. *Left*: phantom A (ground truth); *right*: phantom B (US simulation). Note that the tensor field dimensions are different between the ideal case and US simulation because several ultrasonic beams cover the areas corresponding to the nodes of the FE.

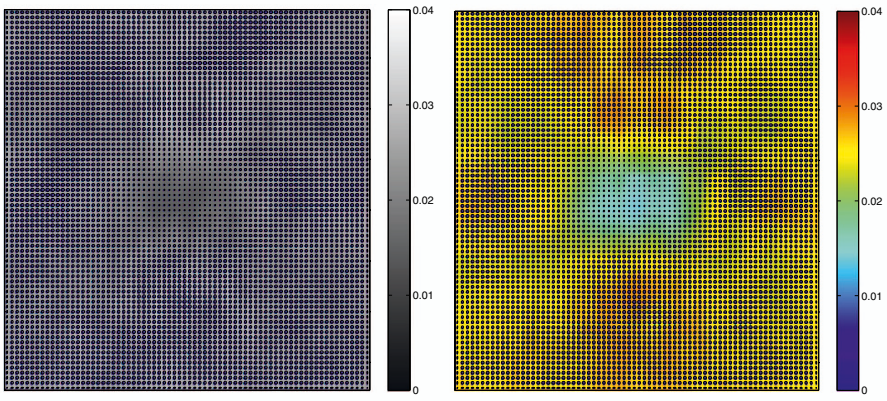



**Fig. 10** Tensorial images overlaid on the axial strain for phantom B (virtual). *Left*: mapped in gray; *right*: mapped with colors. (0.02 = 2%).

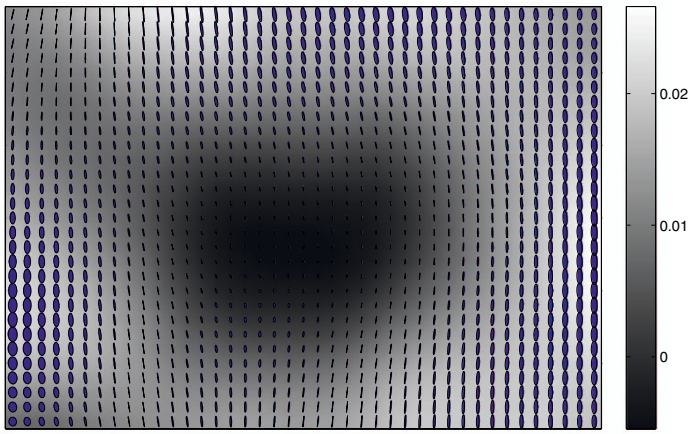



**Fig. 11** Tensorial image overlaid on the axial strain mapped to gray scale for phantom C (tissue-mimicking commercial breast); (0.02 = 2%).

We also tried a volume factor with linear operations from the three eigenvalues $\lambda_1$, $\lambda_2$ and $\lambda_3$, and finally decided to design the color coding using the $c_l$ for the red component and $\lambda_1$ for the green and blue components. As in the elastographic setup, the compression is applied vertically, $\lambda_1$, the biggest eigenvalue, will be most probably oriented in the Y axis, and it will determine the biggest shape difference between the background and the inclusion.

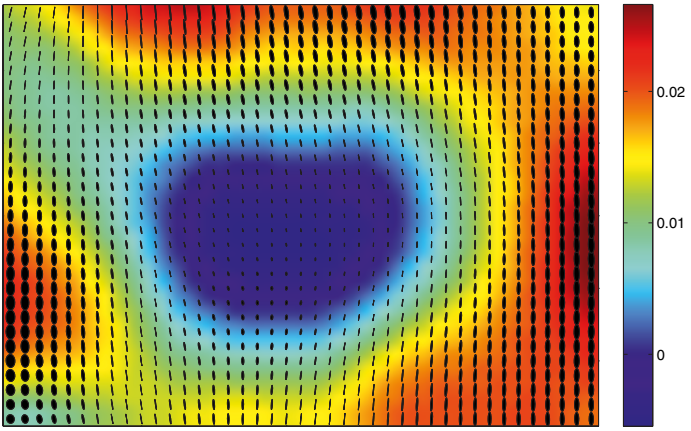



**Fig. 12** Tensorial image overlaid on the axial strain mapped to colors for phantom C (tissue-mimicking commercial breast); (0.02 = 2%).

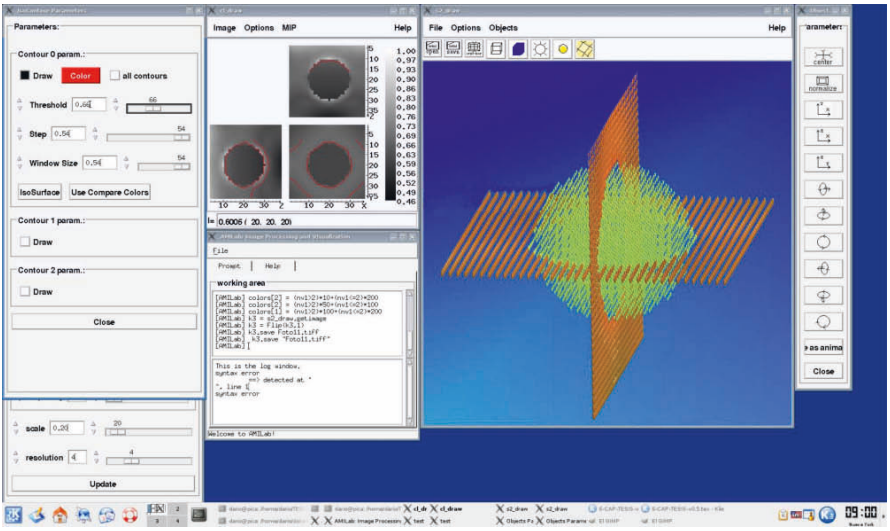



**Fig. 13** Screenshot of AMILab freeware used to visualize 3D tensor fields.

## 7 Conclusions

Other works visualize scalar values such as axial strain, shear strain and Poisson's ratio, and have shown the usefulness of the information contained in them to assess the mobility of the tumor and therefore its malignancy [8, 13]. As the strain tensor integrates these parameters, its representation may provide new criteria and addi-

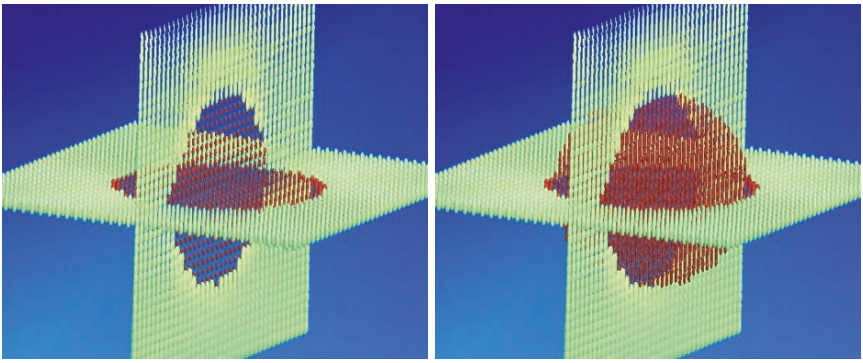


**Fig. 14** Visualization of the 3D strain tensor from phantom A using two slices orthogonal slices. The image at the right displays all the voxels of the inclusion based on its segmentation.

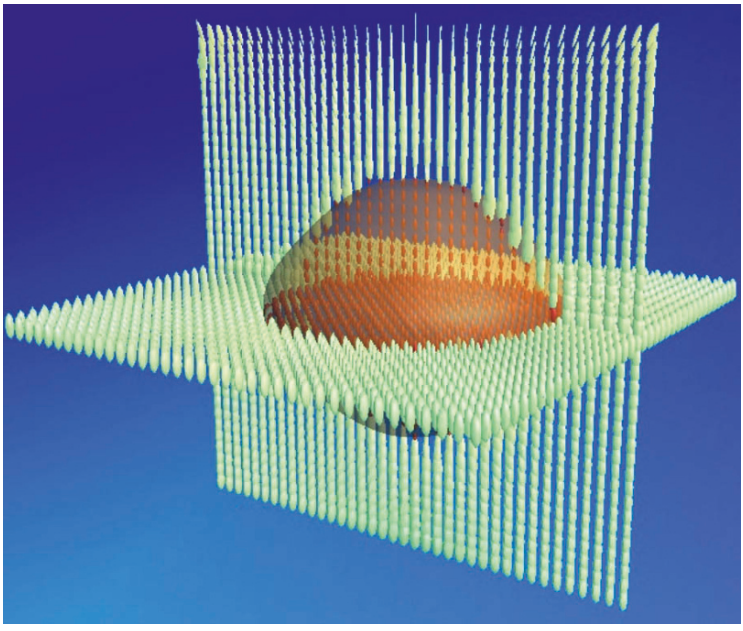


**Fig. 15** 3D strain tensor image of phantom A. The surface of the inclusion is rendered with 50% of opacity to allow the visualization of the ellipsoids.

tional information which complement the scalar images. The color-coded ellipses representation of the strain tensor has a big potential in studying the elastic behavior of tissues.

The visualization approach presented in this chapter, may be applied to many applications in different elastographic modalities, not only quasi-static elastography. As an example, tensorial representations may be useful detecting the calcification in the arteries [19]. Further investigation must be done, to find other ways to visualize

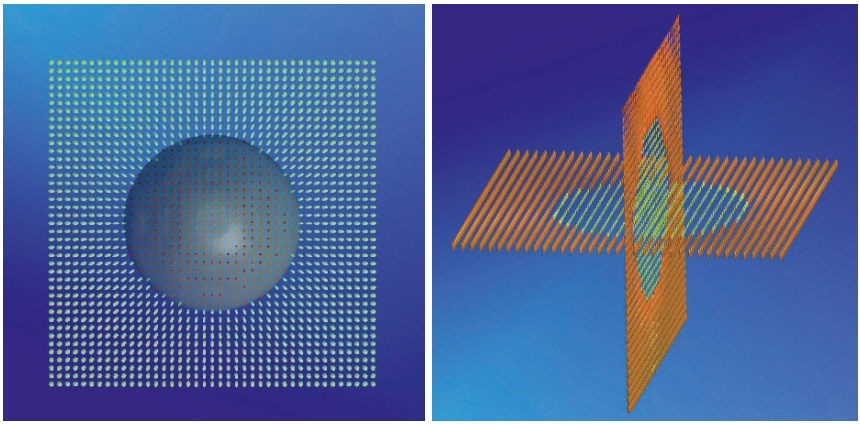


**Fig. 16** Two more example showing different points of view for the 3D strain image, the one at the *right* with the surface object rendered. The image at the *left*, has the inclusion's voxels visible too.

the strain tensor in elastography suitable to handle both the information and the utility for the physicians, extracting information clinically useful for the diagnosis and prognosis of diseases such as breast and prostate cancer. Following the approach presented in this work, novel parameters for the visualization of the elastic properties of the tissues might be found.

Commercial and gelatin phantoms, and clinical validation are under study, as well as other tensorial data representations that can be extracted from the mathematical approach presented here.

**Acknowledgements** Supported by: USIMAG grant (TEC2004-06647-C03-02) from the Spanish Government, SIMILAR Network of Excellence (FP6-507609) from the EU and the FLUID Specific Targeted Research project (No 513633) from the EU.

## References

1. S.K. Alam, F.L. Lizzi, T. Varghese, E.J. Feleppa, and S. Ramachandran. Adaptive spectral strain estimators for elastography. *Ultrasound Imaging*, 26(3):131–149, July 2004.
2. S.K. Alam, J. Ophir, and E.E. Konofagou. An adaptive strain estimator for elastography. *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, 45:461–472, 1998.
3. I Céspedes, Y Huang, J Ophir, and S Spratt. Methods for estimation of subsample time delays of digitized echo signals. *Ultrason Imaging*, 17(2):142–71, April 1995.
4. I. Céspedes and J. Ophir. Reduction of image noise in elastography. *Ultrasound Imaging*, 15:89–102, 1993.
5. E.J. Chen, J. Novakofski, W.K. Jenkins, and W.D. Jr. O'Brien. Young's modulus measurements of soft tissues with application to elasticity imaging. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 43(1):191–194, Jan 1996.

6. R.J. Dickinson and C.R. Hill. Measurement of soft tissue motion using correlation between a-scans. *Ultrasound in Medicine and Biology*, 8:263–271, 1982.
7. M. Dighe, Unmin Bae, Michael L Richardson, Theodore J Dubinsky, Satoshi Minoshima, and Yongmin Kim. Differential diagnosis of thyroid nodules with us elastography using carotid artery pulsation. *Radiology*, 248(2):662–9, August 2008.
8. B.S. Garra, I. Céspedes, J. Ophir, S. Spratt, R. A. Zuurbier, C. M. Magnant, and M. F. Pennanen. Elastography of breast lesions: initial clinical results. *Radiology*, 202:79–86, 1997.
9. T.J. Hall, Y. Zhu, and C.S. Spalding. In vivo real-time freehand palpation imaging. *Ultrasound in Medicine and Biology*, 29(3):427–35, March 2003.
10. M. Horsfield and D. Jones. Applications of diffusion-weighted and diffusion tensor mri to white matter diseases–a review. *NMR Biomed.*, 15(7–8):570–577, 2002.
11. K. Hoyt, F. Forsberg, and J Ophir. Analysis of hybrid spectral strain estimation technique in elastography. *Physics in Medicine and Biology, Institute of Physics Publishing*, 5:197–209, 2006.
12. Jorgen Arendt Jensen. Field: A program for simulating ultrasound systems. In *10th Nordic-Baltic Conference on Biomedical Imaging*, pages 351–353, 1996.
13. E.E. Konofagou, T. Harrigan, and J. Ophir. Shear strain estimation and lesion mobility assessment in elastography. *Ultrasonics*, 38:400–404, 2000.
14. E.E. Konofagou and J. Ophir. A new elastographic method for estimation and imaging of lateral displacements, lateral strains, corrected axial strains and poisson's ratios in tissues. *Ultrasound in Medicine and Biology*, 24(8):1183–1199, 1998.
15. Karl Krissian. Amilab interpreted language for image processing. `http://serdis.dis.ulpgc.es/$\sim$krissian/HomePage/Software/AMILab/`.
16. G. Lan, S.K. Alam, and K.J. Parker. A new vibration theory for sonoelasticity imaging. In *Ultrasonics Symposium*, 2:879–882, 1993.
17. A. Lorenz, A. Pesavento, M. Pesavento, and H. Ermert. Three-dimensional strain imaging and related strain artifacts using an ultrasonic 3d abdominal probe. *Ultrasonics Symposium, 1999. Proceedings. 1999 IEEE*, 2:1657–1660, 1999.
18. L.E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Ed. Prentice Hall, 1969.
19. R.L. Maurice, M. Daronat, J. Ohayon, E. Stoyanova1, F.S. Foster, and G. Cloutier. Non-invasive high-frequency vascular ultrasound elastography. *Phys. Med. Biol.*, 50:1611–1628, 2005.
20. K.R. Nightingale and M.L. Palmeri. Elasticity imaging: Dynamic approaches, 2007. tutorial in IEEE Ultrasonics Symposium, New York, NY, USA.
21. J. Ophir, S.K. Alam, B. Garra, F. Kallel, E.E. Konofagou, T. Krouskop, C. Merritt, R. Righetti, and R. Souchon. Elastography: Imaging the elastic properties of soft tissues with ultrasound. *Journal of Medical Ultrasonics*, 29:155–171, 2002.
22. J. Ophir, S.K. Alam, B.S. Garra, F. Kallel, E. E. Konofagou, T. A. Krouskop, and T. Varghese. Elastography: Ultrasonic estimation and imaging of the elastic properties of tissues. *Journal of Engineering in Medicine*, 213:203–233, 1999.
23. Federico París. *Teoría de la Elasticidad*. Escuela Técnica Superior de Ingenieros Industriales de Sevilla, 1996.
24. K.J. Parker, S.R. Huang, R.A. Musulin, and R.M. Lerner. Tissue response to mechanical vibrations for sonoelasticity imaging. *Ultrasound in Medicine and Biology*, 16:241–246, 1990.
25. A. Pesavento and A. Lorenz. Real time strain imaging–a new ultrasonic method for cancer detection: First study results. In *IEEE Ultrasonics Symposium*, 2:1647–1652, 2001.
26. A. Pesavento, A. Lorenz, and H. Ermert. Phase root seeking and the Cramer-Rao-lower bound for strain estimation. *Ultrasonics Symposium, 1999. Proceedings. 1999 IEEE*, 2:1669–1672, 1999.
27. A. Pesavento, C. Perrey, M. Krueger, and H. Ermert. A time-efficient and accurate strain estimation concept for ultrasonic elastography using iterative phase zero estimation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 46(5):1057–1067, September 1999.

28. M. A. Rodríguez-Florido. *Procesado Anisótropo de Campos Tensoriales Multidimensionales y sus Aplicaciones al Filtrado y Segmentación de Imágenes Médicas*. Phd dissertation, University of Las Palmas de GC, 2004.
29. M.A. Rodríguez-Florido, C.-F. Westin, and J. Ruiz-Alzola. DT-MRI regularization using anisotroipic tensor field filtering. *IEEE International Symposium on Biomedical Imaging*, pages 15–18, April 2004.
30. P. Selskog, E. Heiberg, T. Ebbers, L. Wigstrom, and M. Karlsson. Kinematics of the heart: strain-rate imaging from time-resolved three-dimensional phase contrast mri. *IEEE Trans Med Imaging*, 21(9):1105–1109, 2002.
31. D. Sosa-Cabrera. *Novel Processing Schemes and Visualization Methods for Elasticity Imaging*. Phd dissertation, University of Las Palmas de GC, 2008.
32. D. Sosa-Cabrera, J. González-Fernández, C. Castaño-Moraga, L. Gómez-Déniz, L. Álvarez-León, and J. Ruiz-Alzola. A multiscale variational optical flow method to estimate discontinuous motion fields for ultrasound elastography. In *Proceedings of Fifth the International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, 5:31, 2006.
33. D. Sosa-Cabrera, J. González-Fernández, L. Gómez-Deniz, and J. Ruiz-Alzola. Characterization of a multiscale variational optical flow method for elastography. In *Proceedings of the IEEE Ultrasonics International Symposium*, 2007.
34. S. Srinivasan, F. Kallel, R. Souchon, and J. Ophir. Analysis of an adaptive strain estimation technique in elastography. *Ultrasonic Imaging*, 24:109–118, 2002.
35. Mickaël Tanter, Jeremy Bercoff, Laurent Sandrin, and Mathias Fink. Ultrafast compound imaging for 2-d motion vector estimation: application to transient elastography. *IEEE Trans Ultrason Ferroelectr Freq Control*, 49(10):1363–74, October 2002.
36. T. Varghese, E. E. Konofagou, J. Ophir, S.K. Alam, and M. Bilgen. Direct strain estimation in elastography using spectral cross-correlation. *Ultrasound in Medicine and Biology*, 26(9):1525–1537, 2000.
37. F. Viola and W.F. Walker. A comparison of the performance of time-delay estimators in medical ultrasound. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 50(4):392–401, April 2003.
38. C.F. Westin, S.E. Maier, H. Mamata, A. Nabavi, F.A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor mri. *Medical Image Analysis*, 6:93–108, 2002.
39. C.F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. Jolesz. Geometrical diffusion measures for mri from tensor basis analysis. 1997.
40. Reza Zahiri-Azar and Septimiu E Salcudean. Motion estimation in ultrasound images using time domain cross correlation with prior estimates. *IEEE Trans Biomed Eng*, 53(10):1990–2000, October 2006.

# Part V
# Storage, Visualization and Interfaces

# Similar Tensor Arrays – A Framework for Storage of Tensor Array Data

Anders Brun[1,2,3], Marcos Martin-Fernandez[4], Burak Acar[5], Emma Munoz-Moreno[4], Leila Cammoun[6], Andreas Sigfridsson[2,3,7], Dario Sosa-Cabrera[8], Björn Svensson[2,3], Magnus Herberthson[9], and Hans Knutsson[2,3]

**Abstract** This chapter describes a framework for storage of tensor array data, useful to describe regularly sampled tensor fields. The main component of the framework, called Similar Tensor Array Core (STAC), is the result of a collaboration between research groups within the SIMILAR network of excellence. It aims to capture the essence of regularly sampled tensor fields using a minimal set of attributes and can therefore be used as a "greatest common divisor" and interface between tensor array processing algorithms. This is potentially useful in applied fields like medical image analysis, in particular in Diffusion Tensor MRI, where misinterpretation of tensor array data is a common source of errors. By promoting a strictly geometric perspective on tensor arrays, with a close resemblance to the terminology used in differential geometry, STAC removes ambiguities and guides the user to define all necessary information. In contrast to existing tensor array file formats, it is minimalistic and based on an intrinsic and geometric interpretation of the array itself, without references to other coordinate systems.

[1] Centre for Image Analysis, Box 337, SE-751 05, SLU, Uppsala, Sweden. anders@cb.uu.se (corresponding author) · [2] Division of Medical Informatics, Department of Biomedical Engineering, Linköping University, Linköping, Sweden. {bjosv,knutte}@imt.liu.se · [3] Center for Medical Image science and Visualization (CMIV), Linköping University, Linköping, Sweden. · [4] Laboratorio de Procesado de Imagen (LPI), Dept. Teoría de la Señal y Comunicaciones e Ingeniería Telemática, Universidad de Valladolid, Spain. marcma@tel.uva.es, emunmor@lpi.tel.uva.es · [5] Electrical & Electronics Engineering Department, Boğaziçi University, Istanbul, Turkey. acarbu@boun.edu.tr · [6] Signal Processing Institute (ITS), Ecole Polytechnique Fédérale Lausanne (EPFL), Lausanne, Switzerland. leila.cammoun@epfl.ch · Division of Clinical Physiology, Department of Medicine and Care, Linköping University, Linköping, Sweden. sigge@imv.liu.se · [7] Center for Technology in Medicine, Dept. Señales y Comunicaciones, University of Las Palmas de Gran Canaria, Spain. dario@ctm.ulpgc.es · [8] Department of Mathematics, Linköping University, Linköping, Sweden. maher@mai.liu.se

# 1 Introduction

Trying to describe facts of the world, facts of worlds that may only exists in our imagination, or regularly sampled tensor fields, it is appropriate to start with a quote:

> *"Wovon man nicht sprechen kann, darüber muss man schweigen."*[1]
> – Ludwig Wittgenstein, from Tractatus Logico-Philosophicus [19]

In this chapter, we define a minimalistic tensor array file format called Similar Tensor Array Core (STAC). It is designed to capture the essence of tensor arrays and it makes no effort to describe anything else. If something is not a regularly sampled tensor field, it cannot be described by STAC. It is also with a reference to the late thinking of Wittgenstein that we now write this chapter. We define a canonical standard for storage of tensor arrays, to be used as such. However, we also define this standard to curiously see what response such an action will trigger in on the tensor image processing community.

Tensors and tensor fields are basic tools in differential geometry and physics, to describe geometric and physical quantities that remain invariant under coordinate transformations. Examples include mathematics and physics in general [10], continuum mechanics [8], general relativity [16], diffusion in the human body [17], and local image features in 2-D and higher dimensional images [1]. In computer programs, tensors and tensor fields are often implemented using arrays, with indices corresponding to both spatial dimensions (e.g. $x$, $y$ and $z$) and tensor indices (e.g. $i$ and $j$). Due to the lack of support for tensors in most programming languages, programmers have themselves come up with different conventions for storing tensor data. One of the things that has caused confusion in Diffusion Tensor MRI applications is the relation between the basis vectors of the tensors and the basis vectors of the space where the tensors have been sampled. Perhaps due to a limited view of diffusion tensors, that they are simply $3 \times 3$ matrices, the geometric nature of tensors has sometimes been overlooked and lost. For this reason, we propose a standard for storage of tensor arrays, to promote a geometric view of tensors in the image processing community and make it easier to exchange data between computer programs and between researchers.

We propose a compact file format that is able to store regularly sampled tensor fields, in arbitrary dimensions. The approach is minimalistic, canonical and geometric. It can store tensor arrays, and nothing but tensor arrays. There is a unique encoding of a particular tensor array and the tensor components are strictly coupled to the geometric arrangement of the array of samples. It aims to capture the very essence of tensor fields, using arrays, in a way that is compatible with mathematics, physics and computer programming. We have divided the work on Similar Tensor Arrays into two parts:

- Similar Tensor Array Core (STAC). This is the basic data type for storage of tensor arrays. It promotes simplicity and safety in processing, communication and storage of regularly sampled tensor field data.

[1] "Whereof one cannot speak, thereof one must be silent."

- Similar Tensor Array Extensions. Application specific attributes and conventions, e.g. for storing diffusion tensor MRI data and structure tensor fields. Additional pieces of information that might be necessary are for instance SI units, experimental parameters and the relation to application specific coordinate systems.

It is fair to say that STAC alone is sufficient to mathematically describe any regularly sampled tensor field and one may think of STAC as a tensor array datatype. In many applications however, it is also convenient to embed other information than the tensor array in a dataset and therefore there is a need for extensions. This chapter only describes STAC, but it includes examples on how STAC alone can be used in real applications.

We first review the few existing standards for storage of tensor arrays, then we present our geometrical interpretation of arrays and explain how images and later tensor arrays can be included in the same geometric framework. The necessary attributes are described, named and defined with mathematical formulas. Finally, STAC is described, which includes a Python-inspired syntax and details on how to store data as files on disk. A special section is dedicated to examples of STAC syntax and common operations on tensor arrays. An overview of basic linear algebra and tensor mathematics is presented in an appendix, which may be useful in particular if the reader is unfamiliar with the tensor index notation or other tensor-related concepts that are used throughout this chapter.

## 2 Related Work

To the best of our knowledge, there are few well-documented file formats that are capable of storing sampled tensor field data, even though there are evidently many computer programs that use tensors. Within the medical imaging and visualization community, two file formats for storage of sampled tensor fields should be mentioned; one is the Visualization Toolkit (VTK) [14] from Kitware and the other one is the Nearly Raw Raster Data (NRRD) [11] by Kindlmann.

VTK is able to store 0th, 1st and 2nd order tensors in 3-D using its data format. It supports the formats: structured points, structured grid, rectilinear grid, polygonal data and unstructured grid. It does not have support for separating covariant and contravariant tensor indices, concepts that we present in more detail in the appendix. A distinction between covariant and contravariant indices is important, since these two classes of indices behave very differently. VTK also lacks support for tensors of higher-order than two. VTK is on the other hand versatile in describing the geometry of the dataset, going far beyond regularly spaced rectangular sampling by using so-called unstructured grids where data, such as tensors, may be attached to arbitrary points and cells in space. VTK is a popular library for visualization and it has been used in many applications such as the 3-D Slicer [6]. Writers and readers for this file format are available in the extensive VTK software library, that is coded in C++ and easily accessible from scripted languages like Python [15] and TCL [13].

The other format, the NRRD, is a data format for N-dimensional data arrays. It is a fairly complex data format and it has several features that make it suitable for medical applications. It is not only a tensor array file format, but also a file format that is able to handle non-spatial dimensions and store for instance RGB color images. NRRD has been used in NA-MIC [20] (National Alliance for Medical Image Computing) and is supported by a set of freely available command line tools and software libraries from the same author. The NRRD format is flexible and a NRRD file header can be used as a wrapper to many other array data file formats, including STAC[2]. The downside of such flexibility is of course that the data file reader must be able to handle many special cases, and thus it is very different from the design principles of STAC. Since NRRD is also a software library for processing of N-dimensional array data, written in C, the most convenient use of NRRD is probably through this library.

In summary, both VTK and NRRD have capabilities going beyond storing regularly sampled tensor fields. In contrast, the goal of this chapter is to present a minimal data object, STAC, which can only store tensor arrays. To this core data format, additional layers of abstraction (extensions) may be added in the future to provide extended functionality for particular applications. In for instance Diffusion Tensor MRI, additional information about the data acquisition protocol and SI units may be useful to store. It should however be pointed out that for *processing* of tensor arrays, the STAC part of tensor data is sufficient in many cases, including e.g. filtering, segmentation, registration, resampling, fiber tracking and visualization[3]. Just like PNG image files [2] can be used in different applications such as microscopy and astronomy, STAC can be used to store and process tensor arrays in different applications: STAC is a "tensor image file format".

## 3 Geometric Arrays

In general, an array contains a set of objects, indexed by a fixed number of positive or non-negative integers. Throughout this chapter, we will follow the convention that indices start at 1. Arrays have no explicit connection to geometry, but a natural extension is to regard its $d$ indices as coordinates in a vector space $V = \mathbb{R}^d$ spanned by an orthonormal (ON) basis. In this interpretation, each array element corresponds to a point in space and elements are uniformly spaced in all $d$ dimensions, like in Fig. 1.

If elements are regarded as space-filling, like "pixels" or "voxels", a natural interpretation is that each point is regarded as a cell that extends 0.5 coordinate-units from its center position in both directions for each dimension, forming a square in 2-D and a cube in 3-D.

[2] From personal communication with Gordon Kindlmann at the Tensor Toolbox Workshop in Las Palmas de Gran Canaria in 2006, where we concluded that NRRD did not have support for a metric tensor at the time, but indirectly it already supported the notion of contravariant and covariant indices.

[3] With the reservation that STAC contains no information about handedness.

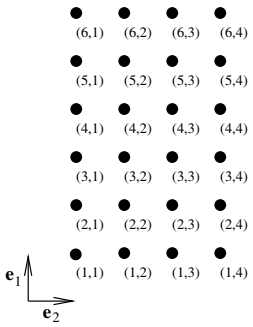



**Fig. 1** The geometry of a $4 \times 6$ array placed in its natural coordinate system in $\mathbb{R}^2$ along with the usual Euclidean ON basis.

In medical imaging, it is sometimes necessary to consider the actual shape of each sample to be different from a square or cube. The sample can be an average over a diamond shape, the value in a point (an average over a Dirac distribution), an average over a sinc-function or something else. It is also possible to make a distinction between regarding a datapoint as 1) a sample or 2) a reconstruction coefficient. A sample is a value obtained from some functional, e.g. a linear functional such as a weighted average using a Gaussian kernel. A reconstruction on the other hand is when the value is interpreted a coefficient, which together with a basis function, represents a reconstructed signal. Most algorithms in image processing and image analysis do not take the actual shape of the pixels/voxels into account, i.e. the samples are regarded as either box averages or point values. In STAC, the exact shape of a sample is undefined. The anisotropy of the sample pattern is stored however, in the so called metric tensor, which is described in the following sections.

## 4 Scalar Arrays

With a geometric interpretation of arrays, as regularly spaced points in a vector space, we now proceed to define a data format for geometric scalar arrays. Scalars are different from higher-order tensors, because they are not geometric objects and their representation is invariant to any change of coordinate system. A scalar value is just a number. Table 1 describes a minimalistic data format for geometric scalar arrays. This simple data format specifies the scalar array data, the number of spatial dimensions, the spatial size of the array and an optional metric tensor to encode the equivalent of pixel- or voxel size. From their mathematical definitions, some of the values of these attributes are bounded and coupled to eachother. This short file format also includes optional naming of the array and its indices, to make it easy to identify the scalar array with the notation of a corresponding mathematical scalar field. While not particularly important for the processing of array data, it may be

| Variable name | Mathematical notation |
|---|---|
| `[scalar_name]` | $f$ |
| `[array_index_names]` | $c^i = [c^1, c^2, \ldots, c^d]^\mathsf{T}, 1 \leq c^i \leq m(i)$, $c^i \in \mathbb{N}$, $\mathbf{c} \in V$ |
| `[array_metric_tensor]` | $g_{ij} \in V^* \otimes V^*$, stored in row-order and p.d. |
| `array_size` | $m(1)\ m(2)\ \ldots\ m(d), m(i) \in \mathbb{N}$ |
| `array_dimensionality` | $d = \dim(V) \in \mathbb{N}$ |
| `data` | $f(c^1, \ldots, c^d)$, stored in row-order |

**Table 1** A table of a minimalistic scalar array format.

convenient to keep the original names of variables and indices, to be faithful to the terminology of a particular data modality or scientific paper.

The metric tensor is a generalization of what is commonly known as pixel- or voxel spacing in digital imaging. Given the metric tensor, it is possible to measure the length of a vector in the geometric array,

$$||\mathbf{c}|| = \sqrt{\mathbf{c}^\mathsf{T}\mathbf{G}\mathbf{c}} = \sqrt{\sum_{i,j=1}^{d} c^j c^i g_{ij}},$$

and more generally to calculate the scalar product between two vectors $\mathbf{a}$ and $\mathbf{c} \in V$,

$$\langle \mathbf{a}, \mathbf{c} \rangle = \mathbf{a}^\mathsf{T}\mathbf{G}\mathbf{c} = \sum_{i,j=1}^{d} a^j c^i g_{ij}.$$

Using the metric it possible to for instance define a unit circle in $V$, i.e. the set of all vectors $\mathbf{c}$ such that

$$\mathbf{c}^\mathsf{T}\mathbf{G}\mathbf{c} = \sum_{i,j=1}^{d} c^j c^i g_{ij} = 1.$$

Choosing the metric appropriately enables the encoding of oblique sampling patterns, which is not possible with pixel- and voxel-spacing alone. However, the metric does not reveal any information about the orientation of the scalar array in relation to any real world coordinate system, it only contains information on how to measure distances and angles within $V$. Depending on the application, these distances may be measured in some physical unit such as meter (m), millimeter (mm) or feet (ft), or they may be dimensionless.

While there is no notion of handedness or any other information on how to transform the geometric scalar array into the real world in which we live, our simple scalar array format does not say anything about how the scalar array should be displayed on a screen. It could be displayed rotated, large or small, upside-down and even mirrored. Only the relative distances of the data points are known. However, most algorithms in image processing do not need more information than a metric to process scalar image data. An image that is rotated 90° and mirrored does not

need any special treatment because most algorithms are invariant to such geometric transformations.

## 5 Tensor Arrays

Some of the most annoying problems related to storage of tensor arrays, at least in the medical imaging community, can be explained by geometric transformations. A typical scenario is that somewhere in the process of image acquisition and reconstruction of tensor arrays, there has been a change of coordinate system for the image volume (translation, rotation, scaling and/or mirroring) without an appropriate transformation of the tensor components. Tensors that were originally aligned with the image volume are suddenly oriented completely wrong. The effect of this can be seen in Fig. 2. The mathematically correct way to transform tensors is to let the tensors transform and change coordinates in a similar way that the overall geometry change, like in Figs. 3 and 4.

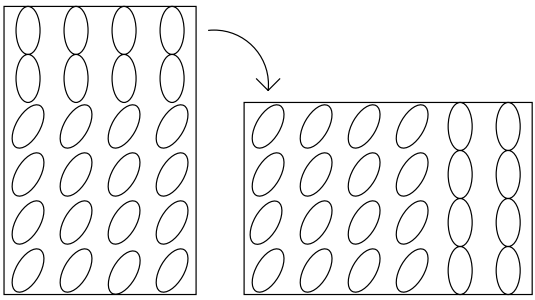


**Fig. 2** An example of transforming only the geometry without changing the tensors. Note that the relation between the tensors and the geometry changes during the transformation. Typically this is not a desirable outcome.

The components of the tensors are related to a basis. In STAC, similar to the commonly used conventions in differential geometry, the basis vectors of the tensors are derived from the coordinate system of the array, which is also known as the coordinate basis. The $i$:th basis vector in $V$ is the tangent vector obtained as the partial derivative of the position in $V$ when all coordinates but $i$ are fixed. In STAC, this happens to be the standard ON basis for $V$. The interesting thing is that whenever we change coordinate system, for instance when the data is resampled or transformed into some physical coordinate system like RAS [6], either these basis vectors must be transformed or the tensor components must change to respect the new coordinate basis. This is described in more detail later in examples and in the appendix. One analogy of coordinate transformations and tensors is to imagine that the tensor

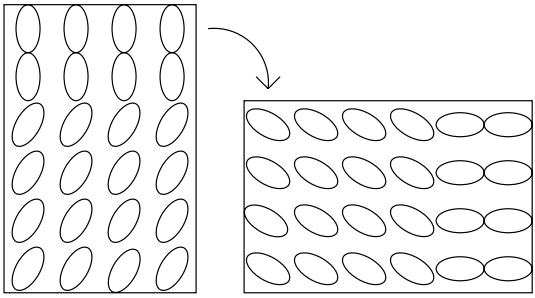


**Fig. 3** An example of transforming both the geometry and the tensors. Note that the tensors and the geometry are aligned before and after the transformation. This is usually the correct way to transform tensor arrays, since tensors in this chapter are regarded as geometric objects that are attached to the image or volume.

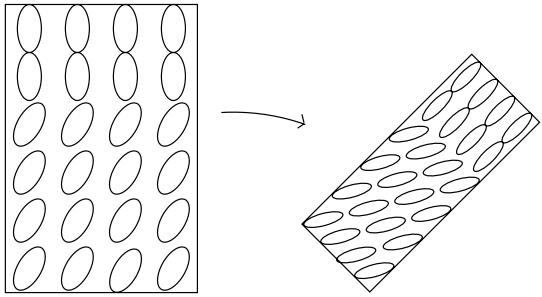


**Fig. 4** Another example of transforming both the geometry and the tensors, which includes both rotation and anisotropic scaling.

glyphs[4] are painted on a data volume made of rubber. When the volume is rotated and stretched, the tensor glyphs are rotated and stretched in a similar manner, i.e. they are tightly related to the geometry of the dataset.

There are exceptions from this rule. During stretching and rotation of DT-MRI data, e.g. when performing medical image registration of two human brains, the transformation of the tensors should only reorient them and now allow them to change in shape [1].Thus, it can be concluded that image registration of DT-MRI tensor arrays is a more complex process than simply a mapping of one coordinate system onto another.

## *5.1 The Tensor Array Core Attributes*

The core of the tensor array standard, STAC, is an almost minimal set of parameters describing the tensor array as a computational and geometrical object. In particu-

[4] Tensor glyphs are graphical representations of tensors, e.g. arrows or ellipsoids.

lar it lacks references to the physical world, including units (e.g. V, m/s and T). Despite its minimalistic qualities, it encodes a self-contained block of information that captures the essence of a regularly sampled tensor field. For many tasks in basic tensor array processing, including filtering, generation of tensor streamlines and calculation of the trace, this core of data attributes can serve as a "greatest common divisor" in the pipeline between algorithms.

The lack of references to a physical world has implications on the visualization of tensor arrays, as described earlier for scalar arrays. Given only the core data, there is in fact no way to determine whether a tensor array should be displayed to a user in a regular or mirrored fashion, i.e. the handedness of the dataset is missing. Even though handedness is important in some applications, for instance in visualization, this information has not been included in the core. In contrast to the metric tensor, handedness not an intrinsic geometrical property of the data. Most tensor array processing algorithms, e.g. filtering and interpolation techniques, are invariant under both mirroring and rotation. If an application needs this information to be stored, then it is an issue that has to be addressed in an extension to STAC, because STAC itself is only concerned with the *intrinsic* geomery of tensor arrays. The STAC attributes are described in Table 2. This table contains both optional and required data fields.

The most important data field is perhaps the dimensionality of the vector space $V$ in which the tensor array lives, $d = \dim(V)$. If the dimensionality is 3, the array extends in three spatial dimensions and each of the tensor indices are numbered from 1 to 3. If for instance the array is a $256 \times 256$ 2-D slice of (3-D) diffusion tensors, this can only be encoded by a $256 \times 256 \times 1$ volume.

The spatial indices of the array may also be regarded as the spatial coordinates describing where each tensor is located in $V$. The tensor order describe the number of tensor indices each tensor has, while the index types encode whether each tensor index is contravariant or covariant. Some redundancy has been allowed for clarity and error checking. Optional parameters for storing a metric tensor and give natural names to the tensor array object and indices have also been added. All optional parameters are denoted within square brackets.

## 5.2 Storing Tensor Arrays to Disk

While the core attributes give a schematic view of a tensor array format, further specifications are needed to store a tensor array to disk. The tensor array data is divided into two parts:

- The STAC header file, contains the STAC attributes describing the data. This file has the extension `.stach`.
- The STAC data file, contains the actual tensor components. This file has the extension `.stacd` or `.stacd.gz`.

Readers and a writers for these STAC files have been implemented in MATLAB, from the attributes in Table 2 and following some principles as detailed below.

### 5.2.1 The Header File

The attributes of the STAC header are named and defined in Table 2, all attributes except `data` are stored in the `.stach` file. The header file consists of a series of lines that define the attributes, see for instance Example 1 and 2. The syntax is a subset of the Python programming language [15], including only strings, integers, floating point numbers, lists, variable assignments and comments. The metric tensor is stored as a list, serialized in row-order. Because the syntax is a subset of Python, `.stach` files can also easily be tested and parsed in any Python interpreter.

### 5.2.2 The Data File

To store the actual array data, the format and byte order of the floating point numbers in the array, and the ordering of the array itself, must be specified. STAC requires floating point data to be stored in the IEEE-754 double precision format using big-endian byte order [9]. This requirement has been added for simplicity and it is sufficient for a wide range of applications. Single precision floats will not be more efficient in terms of data size if appropriate data compression is used. Integer or fixed point data could also save some space, but integers are unnatural to use to describe tensors.

Row-major order, or "lexicographic order", has been chosen for the serialization of the array. This means that first index varies slowly and last index varies fast, when storing a multi-dimensional array in a sequential memory, which is also the storage convention for arrays in C-languages. In STAC it has been chosen to ensure that the tensor components for a single tensor are stored at nearby memory locations. It is worth to note that MATLAB and Fortran uses column-major order instead and a conversion is needed when data is read or written in these programming languages. The reason for not allowing both row- and column-major order is simplicity. Neither of these two schemes are optimal in all situations and ongoing research in scientific computing is investigating other orderings for storing multi-dimensional arrays, such as z-order (Morton order) [18].

Storing the data with a file extension `.stacd.gz` is also allowed, meaning that data has been compressed with gzip in accordance to RFC 1951 [4] and RFC 1952 [5].

| Variable name | Mathematical notation |
|---|---|
| [tensor_name] | $T$ |
| [tensor_index_names] | $s(1)\ s(2)\ \ldots\ s(n), 1 \leq s(p) \leq d, s(p) \in \mathbb{N}$ |
| tensor_index_types | $\in \{'contravariant', 'covariant'\}$ |
| tensor_order | $n \in \mathbb{N}$ |
| [array_index_names] | $c^i = [c^1, c^2, \ldots, c^d]^\mathsf{T}, 1 \leq c^i \leq m(i)$, $c^i \in \mathbb{N}$, $\mathbf{c} \in V$ |
| [array_metric_tensor] | $g_{ij} \in V^* \otimes V^*$ stored in row-order and p.d. |
| array_size | $m(1)\ m(2)\ \ldots\ m(d), m(i) \in \mathbb{N}$ |
| array_dimensionality | $d = \dim(V) \in \mathbb{N}$ |
| data | $T(j) = T(j_1, \ldots, j_{d+n}) =$ $T(c^1, \ldots, c^d, s(1) \ldots s(n)) =$ $T(c^1, \ldots, c^d) \cdots_{s(p)} \cdots^{s(q)} \cdots$, where $j$ maps to a row-order of $(j_1, \ldots, j_{d+n})$ and $1 \leq j \leq d^n \prod_{i=1}^{d} m(i)$. |

**Table 2** A table of the STAC attributes.

## 6 Examples of Use

Below we present some examples, to show what the file headers look like and to explain some of the most important consequences of using STAC for tensor array processing.

*Example 1. (*Storing diffusion tensor MRI data*).* Here the STAC file format is used to store a diffusion tensor MRI dataset. The sampling is often anisotropic in medical image volumes, making the metric tensor deviate from unity. In this particular example, the voxel size is 1mm × 1mm × 3mm.

```
# Similar Tensor Array Core header (\textsc{stac})
# File Format release 0.9
array_dimensionality = 3
array_size = [128, 128, 32]
array_index_names = ["r", "a", "s"]
array_metric_tensor = [1, 0, 0,
                       0, 1, 0,
                       0, 0, 9]
tensor_order = 2
tensor_index_types = ["contravariant","contravariant"]
tensor_index_names = ["alpha", "beta"]
tensor_name = "T"
description = """A diffusion tensor volume.
All tensors are positive semi definite (PSD),
The metric unit corresponds to 1 millimeter.
The unit of the tensor T^{ab} is second^-1."""
```

To access the tensor component $T(62,67,15)^{23}$, a lookup should be performed in the data from the corresponding `.stacd` file. Since the array is serialized in row-major order, this particular component would be found as a double precision floating point number in big-endian byte order, from byte index 18142745[5] to 18142752[6] (indices start at 1).

*Example 2. (*Storing local structure tensors*).* STAC is used to store the local structure tensor field [1] of a 2-D image. This is a second order covariant tensor field, considering that a simple structure tensor estimate can be obtained from averaging outer products of gradient vectors. Images usually have a unit aspect ratio, i.e. the metric tensor is the unit matrix.

```
# Similar Tensor Array Core header (\textsc{stac})
# File Format release 0.9
array_dimensionality = 2
array_size = [64, 64]
array_index_names = ["x", "y"]
array_metric_tensor = [1, 0,
                       0, 1]
tensor_order = 2
tensor_index_types = ["covariant", "covariant"]
tensor_index_names = ["i", "j"]
tensor_name = "S"
description = """A local structure tensor field
for a 64 x 64 pixel image."""
```

*Example 3. (Inheriting the metric from a physical coordinate system).* In medial applications, a tensor array will frequently be accompanied by a transformation from the array to a physical coordinate system, which is often aligned with the anatomy of a human. A common coordinate system is called RAS [6], after the directions of its basis vectors: Right, from the patients perspective. Anterior, i.e. from the back to the belly. Superior, from feet to head. These directions constitute a right-handed coordinate system.

One way to specify the relations between tensor array indices and real-world coordinates is to define an affine transformation,

$$\mathbf{x} = \mathbf{A}\mathbf{c} + \mathbf{t},$$

where $\mathbf{A}$ is a linear transformation and $\mathbf{t}$ is a translation. It is up to the user to choose the basis vectors for the RAS system, which can be measured in e.g. meters (m) or millimeters (mm). In either case, there is a natural metric in the RAS system, which can be transferred to the IJK or tensor array index space, to be stored in the `array_metric_tensor` attribute,

[5] $(62-1)\cdot 128\cdot 32\cdot 3\cdot 3\cdot 8+(67-1)\cdot 32\cdot 3\cdot 3\cdot 8+(15-1)\cdot 3\cdot 3\cdot 8+(2-1)\cdot 3\cdot 8+(3-1)\cdot 8+1$

[6] $(62-1)\cdot 128\cdot 32\cdot 3\cdot 3\cdot 8+(67-1)\cdot 32\cdot 3\cdot 3\cdot 8+(15-1)\cdot 3\cdot 3\cdot 8+(2-1)\cdot 3\cdot 8+(3-1)\cdot 8+8$

$$g_{ij} = \mathbf{G} = \mathbf{A}^{\mathsf{T}}\mathbf{A},$$

which will have e.g. the unit $m^2$ or $mm^2$ depending on the choice of basis vectors. The expression for the components of the metric tensor, in the tensor array index space, can also be derived as a coordinate change of the metric, which is a covariant tensor, from the world space to index space.

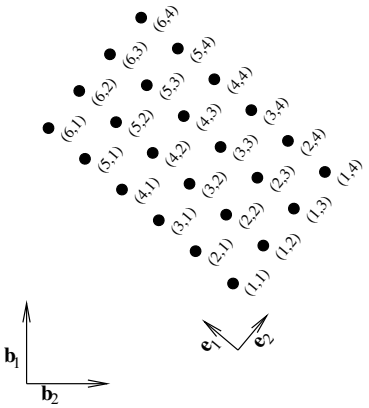



**Fig. 5** Illustrating a change of coordinates, from a coordinate system expressed in the $\mathbf{b}_i$ basis to the tensor array expressed in the $\mathbf{e}_i$ basis. This scenario appears in both resampling of a tensor array and when there exist a transformation from the tensor array index space to a physical world coordinate system such as the RAS system.

*Example 4. (Resampling of tensor arrays).* A common task in image processing is resampling, either to increase or decrease the resolution of a image or volume. Resampling can also be performed in an anisotropic manner, e.g. to restore an isotropic sampling pattern in a volume that has been acquired from anisotropic and/or oblique sampling. In the tensor array framework, resampling can be seen as a change of coordinates, which induce a coordinate change of all tensors in the volume, including the metric tensor if it has been defined. An affine change of coordinates from $\mathbf{c}$ to $\tilde{\mathbf{c}}$ can be described by

$$\tilde{\mathbf{c}} = \mathbf{A}\mathbf{c} + \mathbf{t}, \tag{1}$$

e.g. with

$$\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

to describe an upsampling by a factor 2. Since tensors are expressed in components relative to the coordinate basis, all contravariant tensor indices transform like contravariant vector fields $\mathbf{v} = v^i$,

$$\tilde{\mathbf{v}} = \mathbf{A}\mathbf{v}$$

This is analogous to Eq. 1, i.e. index position vectors $\mathbf{c}$ are contravariant vectors. A covariant vector field $\mathbf{w}^{\mathsf{T}}$, and covariant indices of tensors in general, transform in a

dual manner,

$$\tilde{\mathbf{w}}^{\mathsf{T}} = \mathbf{w}^{\mathsf{T}}\mathbf{A}^{-1},$$

or equivalently in index notation

$$\tilde{w}_i = \sum_{j=1}^{n} w_j (\mathbf{A}^{-1})_{ji}.$$

The metric tensor $g_{ij}$, which is a second order covariant tensor, is thus transformed according to

$$\tilde{\mathbf{G}} = \tilde{g}_{ij} = \sum_{a=1}^{n}\sum_{b=1}^{n} g_{ab}(\mathbf{A}^{-1})_{ai}(\mathbf{A}^{-1})_{bj} = (\mathbf{A}^{-1})^{\mathsf{T}}\mathbf{G}(\mathbf{A}^{-1}).$$

In summary, when resampling a tensor array, both the tensor components and the metric tensor need to be transformed according to the rules for coordinate changes if the resampled tensor array should faithfully represent the original tensor field.

*Example 5. (Calculating eigenvalues and invariants).* In many applications, tensors are synonymous with $2 \times 2$ or $3 \times 3$ positive definite symmetric matrices. The eigenvalues of such matrices represent properties of the tensor that are invariant under rotations and unitary transformations, see e.g. [17]. However, this requires the tensors to be expressed in an orthogonal basis and in STAC, tensors are stored relative to the coordinate basis, which may not be orthogonal. To further complicate things, the definition of eigenvalues and eigenvectors suggests that the matrix should be interpreted as a linear transformation, $\mathbf{Ax} = \lambda\mathbf{x}$, where $\mathbf{x}$ is mapped to $\lambda\mathbf{x}$, i.e. a transformation from a contravariant vector to a contravariant vector. This implies that the matrix $\mathbf{A}$ is a mixed tensor, having one covariant index and one contravariant index. In many applications where eigenvalues of tensors are calculated however, tensors are either second order contravariant (like in Diffusion Tensor MRI) or second order covariant (like in structure tensors). To derive eigenvalues that are invariant to coordinate changes, we need to use the metric tensor to convert the tensors from either type (2,0) to (1,1) or from (0,2) to (1,1). This is done by raising or lowering one index of the tensor, a procedure that is well known in tensor algebra and is further described in the appendix.

For a contravariant tensor array in e.g. DT-MRI, we must calculate the eigenvalues and eigenvectors of

$$T^{ib}g_{bc}x^b = \lambda x^i,$$

or in matrix notation

$$\mathbf{TGx} = \lambda\mathbf{x}.$$

For a covariant tensor array, e.g. a structure tensor field, eigenvalues and eigenvectors are derived from

$$T_{ab}g^{bi}x^a = \lambda x^i,$$

equivalent to

$$\mathbf{TG}^{-1}\mathbf{x} = \lambda\mathbf{x}$$

in matrix notation.
It is worth mentioning that eigenvalues are not invariant to the unit of the metric tensor. The choice of metric unit, e.g. whether the metric tensor measures length in meters (m) or millimeters (mm) in the physical world, will have an impact on the eigenvalues, even though the eigenvectors will be the same. To further complicate things, tensors sometimes have non-geometric units. In Diffusion Tensor MRI, diffusion may be measured by ($mm^2s^{-1}$) and in this case the ($s^{-1}$) part is a non-geometrical aspect of the data that has to be accounted for outside the STAC header or in an extension specific for this application.

## 7 Discussion

The STAC approach described here is a minimalistic framework for the storage of tensor arrays. One of the main purposes of this framework is actually to point out how little information is needed to store regularly sampled tensor fields that can be interpreted, visualized and processed by anyone. Some of its properties may be new to a novice user, especially to those who think of tensors as $3 \times 3$ matrices. The positive thing is that STAC will guide these users to actually learn more about tensors and how they transform under coordinate changes. And if the user knows this, then the user knows virtually everything about tensors–because tensors and tensor arrays are in fact very simple geometric objects.

**Acknowledgements** We gratefully acknowledge the support of the SIMILAR Network of Excellence[7], the European research task force creating human-machine interfaces similar to human-human communication of the European Sixth Framework Programme (FP6-2002-IST1-507609), and the Manifold-Valued Signal Processing project by the Swedish Research Council (Vetenskapsrådet, grant 2004-4721). We also acknowledge the support of the tensor grand challenge from Benoit Macq, Jean-Philippe Thiran, Richard Kitney, Carlos Alberola-López and other people within the medical work package of the SIMILAR NoE.

Parts of this chapter have previously been published in the PhD thesis *Manifolds in Image Science and Visualization* [3]. The STAC file format is the result of a collaboration within the SIMILAR network of excellence and was first presented at the *Tensor Toolbox Workshop* held in Las Palmas de Gran Canaria in November 2006. Readers and writers for the STAC file format were developed by Anders Brun, Marcos Martin-Fernandez and Emma Munoz-Moreno.

## Appendix – Tensor Mathematics

This appendix contains some basic definitions and explanations of the mathematics and notation of tensor algebra. For a more complete introduction to tensors and index notation, see for instance [16, 10, 8].

[7] http://www.similar.cc

## *7.1 Linear Algebra*

To be able to introduce tensors, it is convenient to first define vectors, vector spaces and related concepts.

### 7.1.1 Vector Spaces

Let $V$ be a vector space with $\dim(V) = d$. A basis for $V$ is a set of elements $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\} \subset V$, which are linearly independent and spans $V$, i.e. for any vector $v \in V$ there is a set of coordinates $x^i$ such that

$$\mathbf{v} = \sum_{i=1}^{d} x^i \mathbf{b}_i$$

and

$$\sum_{i=1}^{d} x^i \mathbf{b}_i = \mathbf{0}$$

has the unique solution $x^i = 0$.

### 7.1.2 Linear Maps

A linear map $f$ is a map between two vector spaces $V$ and $W$, $f : V \to W$, that is additive and homogeneous:

$$\begin{aligned} f(\mathbf{u}+\mathbf{v}) &= f(\mathbf{u}) + f(\mathbf{v}) \\ f(\lambda \mathbf{u}) &= \lambda f(\mathbf{u}). \end{aligned}$$

### 7.1.3 The Dual Vector Space

The dual vector space $V^*$ is the space of all linear maps $w : V \to \mathbb{R}$. Thus, $w(u) \in \mathbb{R}$ and

$$\begin{aligned} w(\mathbf{u}+\mathbf{v}) &= w(\mathbf{u}) + w(\mathbf{v}) \\ w(\lambda \mathbf{u}) &= \lambda w(\mathbf{v}) \end{aligned}$$

A simple example is the function $w(\mathbf{v}) = \mathbf{a} \cdot \mathbf{v}$, where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, or more general, $w(\mathbf{v}) = \langle \mathbf{a}, \mathbf{v} \rangle$ where $\mathbf{u}, \mathbf{v} \in V$, for some $V$ equipped with an inner product. $V^*$ is a vector space with $\dim(V^*) = d$. An element $w \in V^*$ operating on a vector $\mathbf{v} = \sum_{i=1}^{d} x^i \mathbf{b}_i$ may be decomposed,

$$w(\mathbf{v}) = w(\sum_{i=1}^{d} x^i \mathbf{b}_i) = \sum_{i=1}^{d} x^i w(\mathbf{b}_i) = \sum_{i=1}^{d} x^i w_i.$$

Apparently, the action on the elements of a basis $B$ uniquely determines the action on any vector expressed in that basis.

A dual basis to $B$, $W = \{\mathbf{b}^i\}$, is defined by

$$\mathbf{b}^i(\mathbf{b}_j) = \delta^i{}_j \tag{2}$$

where $\delta^i{}_j$ is the Dirac delta function,

$$\delta^i{}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}.$$

$V$ and $V^*$ are different vector spaces and there is not necessarily a way to identify a vector $\mathbf{v} \in V$ with an element $w \in V^*$, unless there is an inner product defined. Then, $\mathbf{v}$ may be identified with the element $w$, defined by $w(\mathbf{u}) = \langle \mathbf{v}, \mathbf{u} \rangle$. One interpretation of a dual vector is that it measures some aspect of an ordinary vector. If ordinary vectors are geometrically depicted as arrows of different length, a dual vector can be thought of as the slope of a scalar function defined in $V$ or a level curve to a linear scalar function in $V$, see Fig. 6.

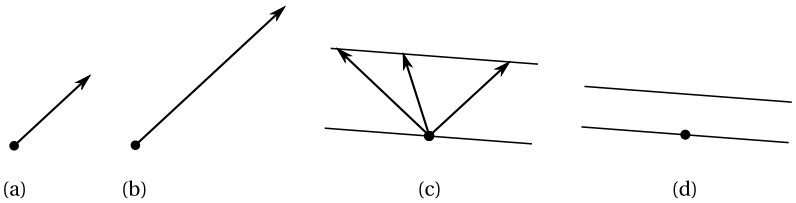



**Fig. 6** **(a)**: A contravariant vector $x^i$. **(b)**: A contravariant vector $2x^i$. **(c)**: a covariant vector $w_i$ and various contravariant vectors $z^i$, for which $z^i w_i = 1$. **(d)**: A covariant vector $2w_i$. Note that the graphical representation or glyph, of a contravariant vector, which can be thought of as a level curve of a scalar function, gets narrower when the coefficients are doubled. This behavior is different from the arrow representing a contravariant vector, which gets longer when the coefficients are doubled.

From Eq. 2 we note the convention that ordinary or contravariant basis vectors are written in boldface with a lower index, $\mathbf{b}_i$, while covariant basis vectors are written using boldface with an upper index, $\mathbf{b}^i$. Consequently, the coordinates of a covariant vector are denoted $x^i$, and the coordinates of a covariant vector are with a lower index, $w_i$. From now on, a vector is often denoted by its coordinates, $x^i$ or $w_i$, which is practical since it then becomes possible to distinguish between contravariant and covariant vectors. Sometimes we also use the notation $\mathbf{v}$, usually to denote a contravariant vector, when there is no way to mix up covariant and contravariant vectors. This notation is the most well-known notation for most readers after all.

### 7.1.4 The Einstein Summation Convention

Since many expressions involving vectors, matrices and soon also tensors, include summations, it is now time to introduce the so-called Einstein summation convention. It means that indices that occur in several places in an expression are summed over from 1 to $d$, where $d = \dim(V)$, e.g.

$$\mathbf{v} = \sum_{i=1}^{d} x^i \mathbf{b}_i = x^i \mathbf{b}_i$$

or

$$w(\mathbf{v}) = w(\sum_{i=1}^{d} x^i \mathbf{b}_i) = \sum_{i=1}^{d} x^i w_i \mathbf{b}^i(\mathbf{b}_i) = x^i w_i.$$

For vectors, this results in a slightly shorter notation. For higher-order tensors however, this notation is more even practical.

### 7.1.5 Coordinate Changes

Coordinate changes in the vector space $V$ induce a dual coordinate change in $V^*$ if the dual basis is assumed. Let $x^i$ denote $\sum_{i=1}^{d} x^i \mathbf{b}_i$ and $w_i$ denote $\sum_{i=1}^{d} w_i \mathbf{b}^i$. Introduce a coordinate change in the contravariant coordinates, $\tilde{x}^i = t^i{}_j x^j$. Then, regardless of coordinate system, we have

$$\begin{aligned} x^i w_i &= \tilde{x}^i \tilde{w}_i \\ &= x^j t^i{}_j T_i^k w_k \Rightarrow \\ t^i{}_j T_i^k &= \delta_j{}^k \end{aligned}$$

for some coordinate change $T_i^k$ in the dual space. Thus, coordinates of dual vectors in $V^*$ must transform inversely to coordinate changes in $V$. The following example gives some intuition to coordinate changes from a simple example in physics.

*Example 0.1.* Consider a capacitance consisting of two charged metal plates separated by a gap $d = 0.5$m with a potential difference $U = 100$V, depicted in Fig. 7. Then, the field strength $E = 200\text{V}/\text{m}$, since it satisfies the equation $U = d \cdot E$. By changing the spatial coordinate system from meter to feet we obtain $d = 1.64$ft, $U = 100$V and $E = 60.98\text{V}/\text{ft}$. Length is a contravariant vector and the coordinate of $d$ increases during the coordinate change. Field strength is a gradient, a covariant vector, and is coordinate decreases from this coordinate change. Thus, there are two types of vectors; covariant and contravariant, which are dual. The type of a vector is often hinted by the associated physical unit, i.e. whether the spatial unit (m, ft, ...) is in the numerator or denominator.

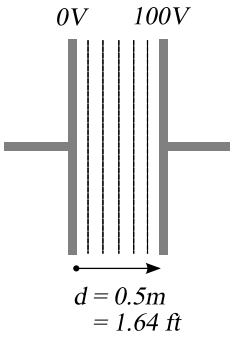



**Fig. 7** When the unit is changed from m to ft, the components of ordinary contravariant vectors and covariant gradient vectors transform differently.

### 7.1.6 Inner Products and Metrics

An inner product $\langle \mathbf{u}, \mathbf{v} \rangle$, or equivalently for our purposes a metric $g(\mathbf{u}, \mathbf{v})$, is a bi-linear map (linear in each argument) $g : V \times V \to \mathbb{R}$ with two additional properties,

$$\begin{aligned} \langle \mathbf{u}+\mathbf{v}, \mathbf{w} \rangle &= \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle \\ \langle \lambda \mathbf{u}, \mathbf{w} \rangle &= \lambda \langle \mathbf{u}, \mathbf{w} \rangle \\ \langle \mathbf{u}, \mathbf{v} \rangle &= \langle \mathbf{v}, \mathbf{u} \rangle \\ \langle \mathbf{v}, \mathbf{v} \rangle &\geq 0. \end{aligned}$$

From linearity we have,

$$g(x^i \mathbf{b}_i, y^j \mathbf{b}_j) = \sum_{i,j=1}^{d} x^i y^j g(\mathbf{b}_i, \mathbf{b}_j), = \sum_{i,j=1}^{d} x^i y^j g_{ij} = x^i y^j g_{ij},$$

i.e. in a given basis $d^2$ components $g_{ij} = g(\mathbf{b}_i, \mathbf{b}_j)$ completely define the action of this map on any pair of vectors. Again, a coordinate change in $V$ induces a change in the components $g_{ij}$. Let $\tilde{x}^i = x^j {t_j}^i$, then

$$\begin{aligned} g_{ij} x^i y^i &= \tilde{g}_{ij} \tilde{x}^i \tilde{y}^i \\ &= \tilde{g}_{ij} x^k {t_k}^i y^m {t_m}^j \Rightarrow \tilde{g}_{ij} = (t^{-1})^k{}_i g_{km} (t^{-1})^m{}_j, \end{aligned}$$

i.e. the components of $g_{ij}$ transform dually relative to the contravariant vectors $x^i$ and $y^j$, because the metric is an example of a second order covariant tensor.

## *7.2 Tensors*

Tensors generalize scalars, vectors and matrices to higher dimensions. Sometimes the word "tensor" is used for any multi-dimensional array with more indices than a matrix, more than two, but we use the term in a more precise meaning that is in agreement with the notation in physics and differential geometry. In these research fields tensors are geometric objects that are invariant under coordinate changes, just like vectors. In physics, the word "tensor" usually refers to what in mathematics would be called a "tensor field" but in both domains it is meaningful to think of tensors as objects defined pointwise in a vector space.

Many spatial quantities in physics are tensors, for instance: velocity (m/s), diffusion ($m^2/s$) and electric field strength (V/m). In mathematics, contravariant vectors are those that behave like we are used to, while the covariant vectors are gradients. Examples of higher-order tensors in mathematics are quadratic forms.

A tensor $F$ is defined as multi-linear map,

$$F : \underbrace{V^* \times \ldots \times V^*}_{r} \times \underbrace{V \times \ldots \times V}_{s} \to \mathbb{R},$$

i.e. a map that is linear in each of its arguments. Its order is $r+s$ and it has type $(r,s)$, meaning that it operates on $r$ covariant vectors and $s$ contravariant vectors. In some contexts, order is called rank and type is called valence, which can be confusing since rank is also used to describe the rank of matrices. Similar to vectors and the metric previously defined, the action of tensors can be defined by components that are derived from the action on all combinations of basis vectors $\{\mathbf{w}^i\}$ in $V^*$ and $\{\mathbf{b}_j\}$ in $V$,

$$F^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s} = T(\mathbf{w}^{i_1},\ldots,\mathbf{w}^{i_r},\mathbf{b}^{i_1},\ldots,\mathbf{b}^{i_r}).$$

The number of components is $d^{r+s}$. If the coordinates are changed, $\tilde{x}^i = t_k{}^i x^k$, then each contravariant index is transformed as a vector and each covariant index is transformed as a dual vector,

$$\tilde{F}^{abc}_{xyz} = F^{abc}_{xyz} t_a{}^i t_b{}^j t_c{}^k \ldots (t^{-1})^x{}_m (t^{-1})^y{}_d (t^{-1})^z{}_o \ldots$$

In physics, this is sometimes how tensors are defined, i.e. as objects that transform according to certain transformation laws.

### 7.2.1 Outer Products

The outer product of two vectors, $F$ and $G$, having type $(r,s)$ and $(p,q)$, is defined by

$$(F \otimes G)((x_{i_1})_a,\ldots(x_{i_{r+p}})_a,(y_{j_1})^a,\ldots,(y_{j_{s+q}})^a) =$$
$$F((x_1)_a,\ldots(x_r)_a,(y_1)^a,\ldots,(y_s)^a)G((x_1)_a,\ldots(x_p)_a,(y_1)^a,\ldots,(y_q)^a)$$

where $(x_i)_a$ refers to the i:th covariant vector.

### 7.2.2 Cartesian Tensors

It is common in e.g. continuum mechanics to work solely using Cartesian vectors and tensors. This means that an ON basis is used and the basis and dual basis coincide and there is no need to differentiate between upper and lower indices.

### 7.2.3 Index Gymnastics

Many operations in tensor analysis can be performed by manipulation of the indices, which is sometimes known as index gymnastics. A contravariant vector $x^i$ may for instance be transformed to a covariant vector by multiplication with the metric $g_{ij}$, $x_i = g_{ij}x^j$. It is called to "lower" an index. In a similar fashion, an index may be "raised", $w^i = g^{ij}w_j = (g^{-1})^{ij}w_j$.

## References


1. D. C. Alexander, C. Pierpaoli, P. J. Basser, and J. C. Gee. Spatial transformations of diffusion tensor magnetic resonance images. *IEEE Transactions on Medical Imaging*, 20(11):1131–1139, 2001.
2. T. Boutell. PNG (Portable Network Graphics) Specification Version 1.0. RFC 2083 (Informational), March 1997.
3. A. Brun. *Manifolds in Image Science and Visualization*. PhD thesis, Linköping University, Sweden, SE-581 85 Linköping, Sweden, 2007. Dissertation No 1157, ISBN 978-91-85715-02-2.
4. L. P. Deutsch. RFC 1951: DEFLATE compressed data format specification version 1.3, May 1996.
5. L. P. Deutsch. RFC 1952: GZIP file format specification version 4.3, May 1996.
6. D. Gering. A System for Surgical Planning and Guidance using Image Fusion and Interventional MR. Master's thesis, Department of Electrical Engineering and Computer Science, MIT, Boston MA, USA, 1999.
7. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995.
8. J. H. Heinbockel. *Introduction to Tensor Calculus and Continuum Mechanics*. 2008. http://www.math.odu.edu/ jhh/bookpdf.zip.
9. IEEE Computer Society Standards Committee. Working group of the Microprocessor Standards Subcommittee and American National Standards Institute. *IEEE Standard for Binary Floating-Point Arithmetic*. ANSI/IEEE Std 754-1985. IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1985.
10. C. J. Isham. *Modern Differential Geometry for Physicists (World Scientific Lecture Notes in Physics)*. World Scientific Publishing Company, 1989.
11. G. Kindlmann. Nearly Raw Raster Data homepage. http://teem.sourceforge.net/nrrd, November 2005.
12. National Alliance for Medical Image Computing. http://www.na-mic.org, November 2006.
13. J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley Professional, March 1994.
14. W. J. Schroeder, K. M. Martin, L. S. Avila, and C. Charles Law. *Vtk User's Guide*. Kitware Inc©, May 2000.
15. G. van Rossum. *The Python Language Reference Manual*. Network Theory Ltd., September 2003.

16. R. M. Wald. *General Relativity*. University of Chicago Press, Chicago, June 1984.
17. C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization of diffusion tensor MRI. *Medical Image Analysis*, 6(2):93–108, 2002.
18. D. S. Wise, J. D. Frens, Y. Gu, and G. A. Alexander. Language support for morton-order matrices. In *PPoPP '01: Proceedings of the eighth ACM SIGPLAN symposium on Principles and practices of parallel programming*, pages 24–33, New York, NY, USA, 2001. ACM Press.
19. L. Wittgenstein. *Tractatus Logico-Philosophicus*. Routledge & Kegan Paul, London, second edition, 1922.

# User Interfaces to Interact with Tensor Fields

Susana Merino-Caviedes and Marcos Martín-Fernández

**Abstract** Nowadays there is a growing interest in tensor medical imaging modalities. In Diffusion Tensor Magnetic Resonance Imaging (DT-MRI), each pixel is valued with a symmetric second-order tensor describing the spatial properties of diffusion at that point. Therefore, it provides significantly more information than scalar modalities, but this causes the complexity of the interfaces dealing with them to grow. In this chapter, the current situation of user interfaces for tensor fields is reviewed. Tensor user interfaces are difficult to design, given the difficulty of mentally integrating data with so many parameters. This is why a considerable effort must be invested in order to achieve intuitive and easy-to-use interfaces. The display of tensor information plays an important role in this, and we review several existing visualization methods for tensor fields. We must point out that, although most of the applications are graphical interfaces, there are also examples of command-line tools and multimodal interfaces employing virtual environments. We study some of the current medical user interfaces for diffusion tensor fields.

## 1 Introduction

Medical imaging has proven to be an invaluable tool for physicians. Among the existing modalities, Magnetic Resonance Imaging (MRI) obtains good quality images without exposure to ionizing radiation, measuring instead the response of hydrogen atoms to strong magnetic fields. Diffusion Tensor MRI (DT-MRI) is an imaging method that measures the water diffusivity in different space directions and models it as a second-order tensor. Therefore, this modality detects information about the internal structure of a tissue that other techniques cannot uncover [3]. For example, it can detect the fibrous structure of white matter, which is seen by other imaging modalities as an homogeneous tissue. Neurology and neurosurgery can greatly ben-

Laboratory of Image Processing, University of Valladolid, Spain
e-mail: smercav@lpi.tel.uva.es, e-mail: marcma@tel.uva.es

efit from DT-MRI, as it allows the *in vivo* computation of tractographies, the pathways of the nervous fibers. Some medical applications where diffusion tensor imaging plays an important role are [42]: research on multiple sclerosis [27], leucoaraiosis, cerebral ischemia, diffuse axonal injury, image-guided neurosurgery [43], neurooncology, white matter abnormalities in schizophrenia [36], epilepsy, etc.

However, due to the complexity of this imaging modality, images must be processed before meaningful information can be extracted from them. It is then fundamental to develop user interfaces for DT-MRI, in order to profit from the advantages offered by this modality. Additionaly, they should use visualization methods that represent the information given by the tensors at each point of the image in an intuitive way for the user. This is not an easy task, and some research has been conducted in the last years to accomplish this goal.

There are not too many software applications for DT-MRI, but they are growing in number, and they can be classified into command-line tools, graphical user interface (GUI) applications, and multimodal interfaces. In addition, as most of these interfaces are fairly recent, they are evolving, improving their usability or adding new functionalities to the already existing ones.

The rest of the chapter is organized as follows. Firstly, some concepts about diffusion tensors and DT-MRI are explained in Section 2. Then, an overview of the existing visualization techniques for tensor fields is given in Section 3, as they are an important part for most user interfaces. Next, the command-line tools, GUIs and multimodal interfaces are respectively reviewed in Sections 4, 5 and 6. They are followed by a discussion in Section 7, and some conclusions in Section 8.

## 2 Diffusion Tensor MRI

The design of user interfaces for diffusion tensor imaging requires at least some knowledge of tensor theory, especially in order to understand the visualization techniques of this kind of data.

Diffusion is a property of a physical medium that measures the Brownian motion of the molecules present in it. When the medium is isotropic, only a scalar is needed in order to characterize diffusion completely. When it is anisotropic, however, the diffusion depends on the direction along which we are measuring it. This is why a symmetric tensor is needed to characterize diffusion in this case [3, 32]. We will refer to such a tensor as $D = D_{ij}$, where $ij = 1,2,3$. It must be noted that lower indices will be used in spite of them being contravariant. This is done for clarity of notation. In matrix form, the diffusion tensor is usually represented as:

$$D = \begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{pmatrix} \tag{1}$$

DT-MRI is a method for measuring the relative diffusion coefficients of water molecules in different directions of an image [32], using magnetic resonance imag-

ing (MRI). These measures are coded as a diffusion tensor at each point of the image.

Diffusion tensors cannot be directly observed from the real world. Instead, they are estimated from a set of diffusion weighted images (DWI) and a baseline image. The DWI are taken using different noncollinear magnetic field gradients, and the baseline image is taken with a null gradient. We name the former $S_i$, the latter $S_0$, and the unitary gradients $\hat{\mathbf{g}}_i$, where $i = 1, 2, \ldots, N$. These images follow the Stejskal-Tanner equation system [46], given below in its logarithmic form:

$$\log(S_i) = \log(S_0) - b\hat{\mathbf{g}}_i^T D\hat{\mathbf{g}}_i \tag{2}$$

where $b$ is the diffusion sensitization [38], which depends on gradient pulse characteristics and gyromagnetic properties of protons, and $D$ is the diffusion tensor.

Since there are six degrees of freedom in the diffusion tensor, at least six DWI images $S_i$ besides the baseline are needed to completely determine the diffusion tensor. We must take into account that the diffusion tensor is obtained through an estimation, and that the presence of noise in the images affects this computation, adding a degree of uncertainty to the result. Frequently, more than six DWI images are used to determine the diffusion tensor in order to achieve better estimations.

Diffusion tensors are symmetric and positive semi-definite. This means that all their eigenvalues $\lambda_i$ are nonnegative, and their eigenvectors $\mathbf{e}_i$ form an orthogonal basis. This is of great importance in DT-MRI visualization. Eigenvalues are usually listed in a decreasing order, so that $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are called the major, medium and minor eigenvalues respectively.

Given that diffusion tensors are symmetric, only six components of the $3 \times 3$ matrix are independent quantities, which means that we have six degrees of freedom in order to characterize a tensor at each point of a DT-MRI image.

Data from a DT-MRI image are representations of the diffusion tensor corresponding with infinitesimal volumes (voxels) of a human body. The molecule mobility in different directions is characterized by the degree of anisotropy, which allows to determine not only the type of tissue but also fiber orientation. There are several parameters used to facilitate the interpretation of the data. These parameters can be classified as diffusivity measurements, anisotropy measurements and orientation related parameters:

1. **Mean diffusivity**. The goal of this parameter is to obtain a measure of the diffusion in a voxel, free of anisotropy effects, and to provide a value that is invariant with tissue direction or orientation. Using this parameter, tissues with the same mean diffusivity characteristics will ideally have the same value, independently of its direction or orientation.

$$\langle D \rangle = \frac{trace(D)}{3} = \frac{D_{11} + D_{22} + D_{33}}{3} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \tag{3}$$

2. **Anisotropy** is the property of presenting different characteristics depending on the orientation. It is the opposite property to isotropy, which is characterized by presenting equal properties in all directions.

- The fractional anisotropy (FA) characterizes the ellipsoid eccentricity and has the expression [4, 33]:

$$FA = \frac{\sqrt{3\left((\lambda_1 - \langle D\rangle)^2 + (\lambda_2 - \langle D\rangle)^2 + (\lambda_3 - \langle D\rangle)^2\right)}}{\sqrt{2(\lambda_1^2 + \lambda_2^2 + \lambda_3^2)}} \tag{4}$$

  where $\langle D\rangle = (\lambda_1 + \lambda_2 + \lambda_3)/3$ is the mean diffusivity. FA ranges from 0 (isotropic) to 1 (completely anisotropic).
- The relative anisotropy (RA) measures the variations in the eigenvalues and is as follows [46]:

$$RA = \frac{\sqrt{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2}}{\sqrt{2}(\lambda_1 + \lambda_2 + \lambda_3)} \tag{5}$$

- The volume ratio (VR) coefficient is the ratio between the volume of the ellipsoid defined by the diffusion tensor and a sphere with the mean diffusivity as radius [5]:

$$VR = \frac{\lambda_1\lambda_2\lambda_3}{\langle D\rangle^3} \tag{6}$$

  This coefficient ranges from 0 (completely anisotropic) to 1 (isotropic), so the measure $(1 - VR)$ is also used in order to make the range consistent with that of FA.
- The anisotropy shape coefficients $c_l, c_p, c_s$ were introduced in [46], and they measure which kind of anisotropy is present in a tensor, whether linear, planar or spherical, respectively. They have the property that $c_l + c_p + c_s = 1$. There are different expressions for them, and we give the ones in [31], because of their relevance with respect to barycentric mapping:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \tag{7}$$

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \tag{8}$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{9}$$

3. **Fiber orientation**: DT-MRI facilitates information regarding the direction followed by the fibers present in the white matter of the nervous system. These fibers are composed by axons with a myelin layer, and the water molecules can move more easily along the fibers than across them. Therefore, the fiber orientation is taken to be the direction of maximum diffusion, which is given by the major eigenvector of the diffusion tensor.

These coefficients are widely used for tensor processing and visualization. The latter is described in the following section.

## 3 Visualization of Diffusion Tensor Fields

The comprehensive visualization of multidimensional data like a tensor field is a challenging task. Several methods have appeared in recent years, which are summarized in Table 1. Due to their relevance to DT-MRI user interfaces, glyphs and tract visualization are explained in the following subsections.

**Table 1** Overview of tensor visualization techniques.

| | |
|---|---|
| Glyphs | Ellipsoids, cuboids, cylinders, superquadrics [30], composite shapes [46]. |
| Tract visualization | Hyperstreamlines[11], streamtubes (coupled with streamsurfaces) [49]. |
| Volume visualization | Volume deformation [50], topological lines [52], hyperLIC [51], reaction-diffusion textures [31]. |
| Rendering techniques | Barycentric mapping [31], hueballs [31], lit-tensors [31]. |

### *3.1 Glyphs*

A popular way of visualizing tensor fields is to represent the tensor at each point of the field with a glyph [46], which is a geometric figure whose form depends on the characteristics of the tensor it represents. A glyph is composed of a single or multiple geometric primitives, whose shape may codify eigenvectors and eigenvalues, anisotropy coefficients, or other measures like curvature, shear, rotation, etc., of the tensor field.

The most popular shape for a glyph is an *ellipsoid*, by which tensor eigenvectors and their corresponding eigenvalues are represented. The ellipsoid's major axis is orientated along the major eigenvector $\mathbf{e}_1$, and the medium and minor axes in the direction of the medium and minor eigenvectors $\mathbf{e}_2$ and $\mathbf{e}_3$, respectively. We must point out that this representation is valid only for positive–definite tensors, in order to make sure that the eigenvalues are real and nonnegative and the eigenvectors are orthonormal. The size of the ellipsoid axes will be proportional to the corresponding eigenvalues.

Depending on the relative values of the eigenvalues, the ellipsoid will take one form or another:

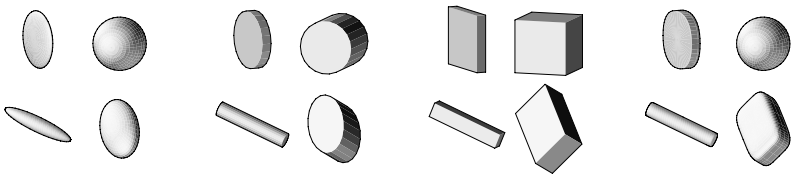
- If $\lambda_1$ is visibly greater than the other eigenvalues, the ellipsoid shape will be elongated in the major eigenvector direction. This happens where diffusion is very linear as for example in some brain regions where neural fiber tracks, also called tracts, form coherent bundles along the same direction.
- If $\lambda_1 \simeq \lambda_2$, the ellipsoid shape resembles a disk, and the smaller $\lambda_3$, the flatter the disk will be. This type of tensors are often seen in regions where a fiber bundle

bifurcates or where two bundles cross each other, and bring uncertainty in the computation of the direction of tracts, as seen in Section 3.2.

- If $\lambda_1 \simeq \lambda_2 \simeq \lambda_3$, the resulting ellipsoid looks like a sphere. When this happens, there is no preferred diffusion direction.

We must take into account that the ellipsoid size may suffer great variations from a region to another, which is not good for visualization. Laidlaw *et al.* [37] propose a normalization for ellipsoids, by which the major axis is given an unitary value, and the size of the other axes is computed accordingly, in order to preserve the proportion between them. Color can be employed to provide the information lost in the normalization.

On the other hand, ellipsoids are not the only shapes one can use for representing tensor eigenvectors and eigenvalues. In [30], cuboids and cylinders were used to represent eigenvectors and eigenvalues instead of ellipsoids. In Fig. 1, glyphs using these three geometrical figures can be observed: ellipsoids in Fig. 1(a), cylinders in Fig. 1(b) and cuboids in Fig. 1(c).



(a) Ellipsoids (b) Cylinders (c) Cuboids (d) Superquadrics

**Fig. 1** Comparison between glyphs using different geometric figures.

*Composite shapes* [46] were developed to provide a better appreciation of the exact value of the anisotropy shape coefficients. Instead of ellipses or other individual shapes, a composite of a headless arrow, a circle and a sphere, each of them proportional to the coefficients $c_l$, $c_p$ and $c_s$ respectively is used. Unfortunately this glyph also presents ambiguity, so they must be given a coloring depending on the tensor anisotropy to solve this problem. The hue is interpolated between the blue linear case, the yellow planar case and the red spherical case.

In order to overcome the problems presented by the glyphs seen above, Kindlmann proposed a new kind of glyphs called *superquadric glyphs* [30]. The basic idea is to use superquadrics to represent a tensor, depending on the anisotropy coefficients $c_l$ and $c_p$, and using the eigenvectors to orientate the glyph.

The shape of a superquadric is controlled by two parameters. Spheres, cubes and cylinders can be expressed as particular cases of a superquadric. The expression of the geometric primitive of a superquadric glyph varies depending on the anisotropy:

- Case $c_l \geq c_p$: $q(x,y,z) \equiv \left(y^{2/\alpha_l} + z^{2/\alpha_l}\right)^{\alpha_l/\beta_l} + x^{2/\beta_l} - 1 = 0$
- Case $c_l < c_p$: $q(x,y,z) \equiv \left(x^{2/\alpha_p} + y^{2/\alpha_p}\right)^{\alpha_p/\beta_p} + z^{2/\beta_p} - 1 = 0$

where $\alpha_l = (1 - c_p)^\gamma$, $\beta_l = (1 - c_l)^\gamma$, $\alpha_p = (1 - c_l)^\gamma$ and $\beta_p = (1 - c_p)^\gamma$, and $\gamma$ controls the edge sharpness. The resulting superquadric tensor glyphs for $\gamma = 3$ can be seen in Fig. 1(d). We must point out that their axes are scaled by the tensor eigenvalues.

## *3.2 Tract Visualization*

One of the main applications of DT-MRI is the computation of tractographies of the brain white matter, where the structure of the tracts[1] appearing in the image is obtained from the diffusivity measure at each voxel. Usually a large number of seed points are selected as starting points for tract computation, and afterwards, a culling algorithm is applied to select the most representative ones [49]. Finding the tracts passing by a voxel is not a trivial problem, mainly due to the following reasons:

- DT-MRI images have a poor resolution compared to the size of nerve fibers, whose section can be measured in microns, whereas the volume of a DT-MRI voxel is of the order of $mm^3$. This is called the partial volume problem. Fortunately, it can be partially solved, because nerve fibers often form large bundles going in the same direction.
- Tractographies are often computed following the principal direction of diffusion (PDD), given by $\mathbf{e}_1$. When the tensor anisotropy is clearly linear, this method yields adequate results. Nevertheless, where planar or spherical anisotropy prevails, there is ambiguity about the direction of the tract at that point [46].

Tractography visualization is an important task which overlaps with some tensor field visualization techniques, such as hyperstreamlines [11] or streamtubes [49]. Streamline visualization methods can also be employed, but only the information about the tensor major eigenvector is displayed in that case.

### 3.2.1 Hyperstreamlines

Following the philosophy of streamlines, Delmarcelle and Hesselink [11] propose *hyperstreamlines* as a model for the visualization of positive-definite tensor fields. This concept was inspired by streamlines as a representation technique for vector fields, where each vector in the field is tangent to a streamline.

The shape of hyperstreamlines is similar to a tube with an elliptical cross-section. The tube direction at each point is computed from the vector field formed by the major eigenvector of each tensor in the field, similarly to streamlines. In addition, the axes of the cross-section ellipse are proportional to the medium and minor eigenvalues, and oriented in the directions of their corresponding eigenvectors.

[1] Tract: a bundle of myelinated nerve fibers following a path through the brain.

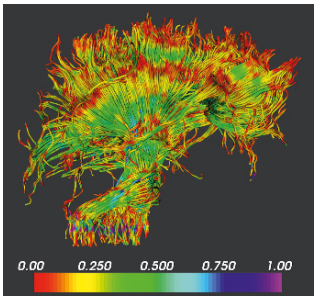



**Fig. 2** Visualization of DT-MRI using hyperstreamlines, whose color is proportional to the tensor linear anisotropy at each voxel.

Therefore, this method can be used to represent all tensor information at each point of a hyperstreamline, and a sense of continuity is achieved as well. It is not possible, however, to represent this information in the whole volume, because hyperstreamlines are still discrete elements and may suffer from cluttering, as it happens with glyphs.

An example of hyperstreamlines can be seen in Fig. 2. In this picture, hyperstreamlines are given a color at each point depending on their linear anisotropy. Tract color usually depends on anisotropy measures, but other options can be selected. Brun *et al.* [6] use Laplacian eigenmaps to assign color to tracts computed from a DT-MRI volume, so that similar fibers are similarly colored. Ehricke *et al.* [12] employ the probability that a point is crossed by a tract and the conditional probability that a sample belongs to the tract.

### 3.2.2 Streamtubes and streamsurfaces

*Streamtubes*, proposed by Zhang *et al.* [49], are a tract visualization technique which is very similar to hyperstreamlines, but tries to overcome some of its limitations. Firstly, a hyperstreamline cross-section can get quite large, which makes smaller the number of them that can be visualized in a volume. This can become problematic in the analysis of biological tissue. Secondly, in regions where the anisotropy is mainly planar, hyperstreamlines are not very effective [49].

For all these reasons, Zhang *et al.* propose to split the volume in regions of linear anisotropy and planar anisotropy, so that the former are visualized by *streamtubes*, and the latter by *streamsurfaces*. Herein lies the problem of setting adequate thresholds for the regions. In [49], these thresholds are manually defined.

Streamtubes are in essence very similar to hyperstreamlines. Their main difference lies in how the cross-section is computed. Although its shape is an ellipse in both techniques, the major axis, corresponding to the medium eigenvector and oriented along this direction, is forced to be constant, and the length of the minor axis, oriented along the minor eigenvector, will be computed so that the proportion between axes is maintained. This way, more elements may be represented than with

hyperstreamlines, although the information in the whole volume still cannot be visualized in one representation. In order to avoid ambiguity between different tensors with the same quotient $\frac{\lambda_2}{\lambda_3}$, the streamtube is given a red hue proportional to the linear anisotropy coefficient $c_l$. That means that the more linear the diffusion is, the redder the streamtube will be.

*Streamsurfaces* are represented in regions where $c_p$ has a high value, which means that the anisotropy is highly planar. At each point, they try to be an approximation to the surface defined by the major and medium eigenvectors, which will lie on the tangent plane to the streamsurface at that point. As it happened with streamtubes, a streamsurface is given a hue proportional to the planar anisotropy coefficient $c_p$, but in this case the chosen color is green.

## 4 Command Line Interfaces

Tensor visualization techniques improve the understanding of the information contained in DT-MRI volumes. However, people working with these images need a user interface to access the aforementioned methods, as well as other image processing tasks. Two command-line interfaces for DT-MRI are described below: the Teem software [23] and the interface Camino [9]. This type of interfaces is based on typing commands that will be executed by the system.

### *4.1 Teem*

Teem is a collection of libraries created by Gordon Kindlmann, whose purpose is representing, processing and inspecting scientific raster data [23]. On top of these libraries two user interfaces have been built. The first one is a command line interface, and the second one is a simple graphical user interface called Deft. However, both of them can work together.

The NRRD (Nearly Raster Raw Data) file format [21], which allows n-dimensional data to be stored, is a fundamental component of these interfaces. The command-line tool designed to manage these files is *unu*. Some of the capabilities of this command are: creation, conversion, saving in different file formats, type conversion, mapping, certain pixelwise mathematical operations, creation of histograms, and others like cropping, resampling, merging, etc.

On the other hand, *tend* is employed for diffusion tensor processing and analysis. Some of the tasks that can be performed with it are: estimating tensors from DWI images, generating synthetic tensor fields, computing eigenvectors and eigenvalues, generating postscript renderings of glyphs, computing anisotropy indices, or modifying some of the tensor characteristics.

Teem commands follow the syntax: `command action [options]`. For example, typing `unu head myfile.nrrd`, the header of the file *myfile.nrrd* is

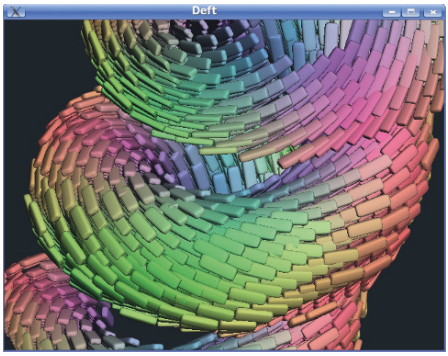



**Fig. 3** Synthetic tensor data created with *tend* and visualized using *Deft*.

printed on the standard output. If no options are given, an explanation about the command is shown.

The goal of the interface Deft is to provide interactive tensor visualization. It consists of a render window and other windows for controlling the visualization, which are called *Deft::Viewer*, *Deft::Hyperstreamline*, *Deft::TensorGlyph* and *Deft::TriPlane*. In Fig. 3, a synthetic tensor volume is shown, using superquadric glyphs. The commands employed for the creation of this figure, using a linux environment, are:

```
tend helix -s 29 30 31 -ip 0.1 0.3 0.6 -mp -0.8 0.1 0.4 \
-r 50 | Deft_dti -i - -a fa -gsc 1200 -atr 0.65 \
-fr 940.805 307.575 1025.14 -at 0 0 0 \
-up -0.521774 -0.557152 0.646014 -rh -ar \
-dn -164.709 -df 164.709 -fv 5.41056 -is 640 480
```

Teem uses the GNU Lesser General Public License [18], so that it can be freely distributed and modified, and can also be used with non-free libraries. In fact, these libraries have been used in the 3D Slicer framework, in ITK for reading data in NRRD files, etc.

## *4.2 Camino*

Camino [9] is another free open-source command-line tool for DT-MRI. It is written in Java, so theoretically it could run on any platform, but it is designed for a Unix-style interface, with shell wrappers for each program and documentation as man pages. In order to process data, a pipeline must be built by connecting several commands, so that the output of one of them is the input of the following one. A Camino program reads raw data from a scanner or from a data synthesizer. A scheme file must be provided with the data acquisition parameters. As of version 1.5, the output of the pipeline is in the form of raw data, which means that other

applications must be employed if the user wants to visualize the result. The only exception is the `vcthreshselect`, which loads the F-test data from the output of the command `voxelclassify` and previews the voxel classification.

## 5 Graphical User Interfaces for Tensor Processing

Nowadays the most frequently seen paradigm for user interfaces is the graphical user interface (GUI). In them, the user is offered symbols and visual metaphors to represent the elements of the interface. A pointer is employed to interact with these elements, often by direct manipulation (clicking, dragging, etc). The user controls the pointer using a mouse, a touchpad, etc. A 2D desktop screen and a keyboard are also present in these interfaces.

Most of the applications that work with DT-MRI images have a graphical user interface (GUI). We will review 3D Slicer, MedINRIA, BioTensor and BioImage.

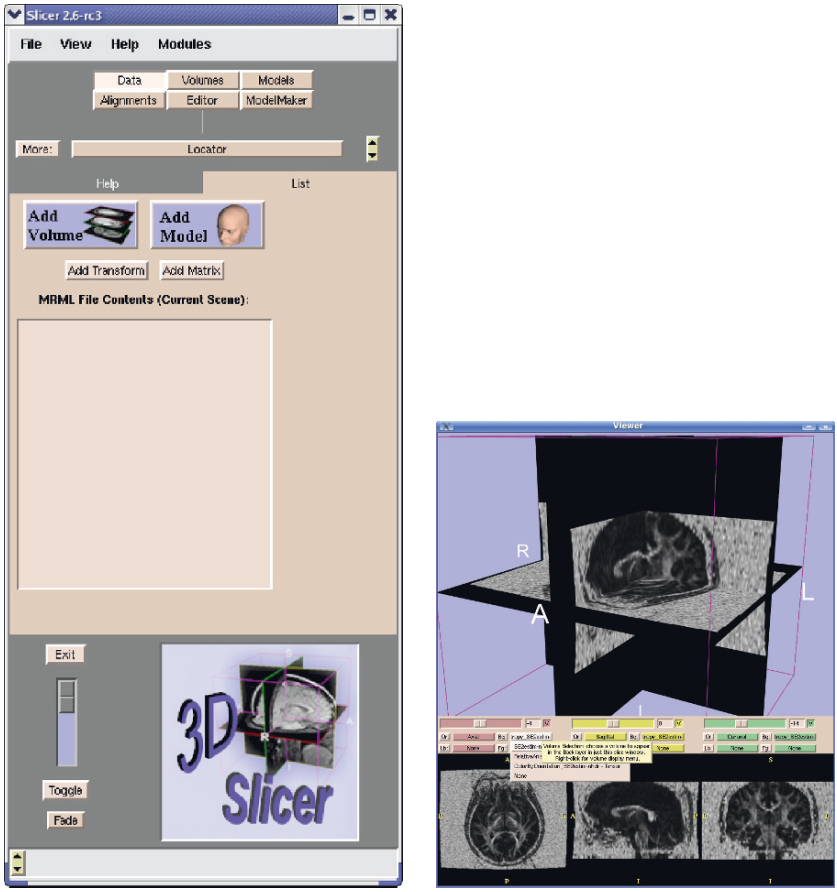
In order to test the interfaces, either synthetic data or a real DT-MRI volume were used. The volume was acquired using a General Electrics scanner of 1.5 T, with $b = 1000\,s \cdot mm^{-2}$, 15 gradient directions, 8 acquisitions per slice, and voxel size of $1.015 \times 1.015 \times 3\,mm^3$.

### *5.1 3D Slicer*

*3D Slicer* [13, 25, 26] was developed by the MIT AI Laboratory [17] and Brigham and Women's Hospital at Harvard [16], and its current development is mainly supported by the National Alliance for Medical Image Computing (NA-MIC) [20]. 3D Slicer is a software environment oriented to image-guided medicine, in particular for pre-operative planning, surgical guidance, and diagnostic visualization [25]. It allows not only volume data visualization of different kinds of medical image modalities, but also image processing techniques like filtering, segmentation and registration. The main goals of this software are to provide a common development platform for researchers, a familiar user interface for image processing and visualization tasks, and the transfer of algorithms and other techniques from developers to users [40]. 3D Slicer does not provide any guarantee of clinical accuracy or reliability for research.

This package is open source, and is portable to several operating systems. It is composed of modules, and its architecture is designed so that new modules can be added without having to rebuild the whole application. 3D Slicer version 2 uses Tcl/Tk [45] for the user interface, VTK [41] for visualization and ITK [28] for image processing. From now on, we refer to 3D Slicer version 2.6.

One of the available modules for 3D Slicer is a module for diffusion tensor image visualization called DTMRI. It allows the user to load a DT-MRI volume and pro-




(a) (b)

**Fig. 4** (a) Main window and (b) viewer window of 3D Slicer.

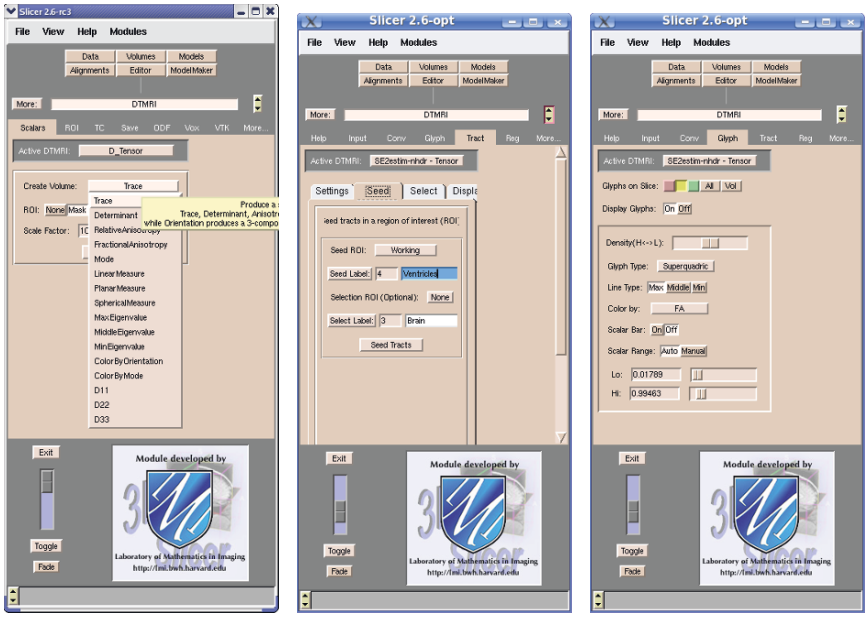
cess it to get some image characteristics, like eigenvalues or anisotropy coefficients, and also to perform a number of tensor processing and visualization methods.

The application interface is divided in three windows: a main window, a viewer window, and a command console. The first two can be observed in Figs. 4(a) and 4(b), respectively. The preferred input device is a conventional mouse, although sometimes the keyboard is needed. It is nonetheless possible to use it in conjunction with other devices.

Looking at Fig. 4(a), we can see how the main window is divided. It has a menu bar at the top of the window, some buttons, each for a Slicer 3D module, a frontal panel divided into tabs, and at the bottom there is a small window with controls for the viewer window, although it is not always active. A study of website design [7] for placement and presentation of navigation menus showed that top and

left placements for navigation elements, and the tabbed navigation style are preferred by users. Therefore, the layout of 3D Slicer follows these directives.

The viewer window showed in Fig. 4(b) is divided into a window for 3D volume visualization and three small windows for axial, sagittal or coronal slices. Nonetheless this layout can be changed in the main window. It can be seen that there are some user controls in the viewer window that help the navigation throughout the volume, changing the point where the slices intersect, allowing to show or hide slices in the visualization, or choosing what volume to represent. A helpful feature for users is that a tooltip box is showed when the pointer is over an element a certain amount of time, as can be seen in Fig. 4(b). The user can zoom the volume in and out using the right button of the mouse, rotate it with the left button, and also pan it using the middle one.




(a) (b) (c)

**Fig. 5** Main window panels of the DTMRI module for: (a) Choosing which tensor characteristic to compute. (b) Computing a tractography with a seed ROI. (c) Choosing the glyph visualization properties.

3D Slicer provides three visualization methods for DT-MRI volumes. Firstly, volumes of anisotropy indices or tensor invariants can be represented by orthogonal 2D slices. In Fig. 4(b), a volume FA is shown. Secondly, a tract visualization method based on tubes is available. Thirdly, glyphs can be employed either for slices or in the whole volume. In Fig. 6(a), a region of a coronal slice is visualized using

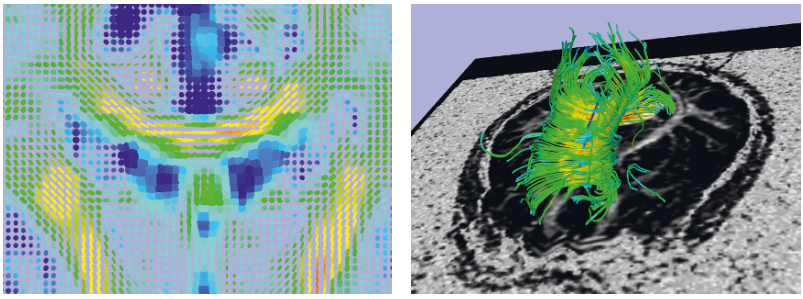


**Fig. 6** (a) Visualization of a coronal slice using superquadric glyphs, colored by the FA. (b) Resulting tracts colored by the FA value at each point.

superquadric glyphs. However, it can be noted that their renderization is not too good. In Fig. 5(a), the diferent options of the DTMRI module for computation of some of the tensor characteristics are shown. In Fig. 5(b) the tractography panel can be seen. In this case, an ROI has been chosen as the seeding strategy for the tractography. Finally, the panel for using glyphs in the visualization is shown in Fig. 5(c).

If the user wants to select a region of interest (ROI), the Editor module should be employed. It has a direct access from the main window, and offers methods to accomplish this, like freehand drawing or thresholding. Slicer offers ways to make calculations on ROIs using another module called VolumeMath.

Finally we try the tractography visualization method offered by the DTMRI module, for which we will use the Tract panel. We make a tractography employing user-selected seed points, which are taken from the position of the mouse cursor while the key "S" is pressed. Once a seed point is selected, the corresponding tract is computed. In Fig. 6(b), the visualization of the results obtained this way can be observed.

At the present time, the code of 3D Slicer version 2 is being ported to a newer version that is called Slicer3. It has almost completely changed the user interface, employing a user-centered design and changing the GUI programming language from Tcl/Tk to C++ by using the KWWidgets library [34]. The DTMRI module for Slicer3 is currently being developed.

## *5.2 MedINRIA*

MedINRIA [19] is a software environment originally developed for DT-MRI processing and visualization, but now including modules for other purposes. It was initiated by Pierre Fillard and Nicolas Toussaint, members of the INRIA research team Asclepios at Sophia Antipolis. Like 3D Slicer, it uses VTK [41] for visualization and ITK [28] for image processing, but the wxwidgets library [24] is employed

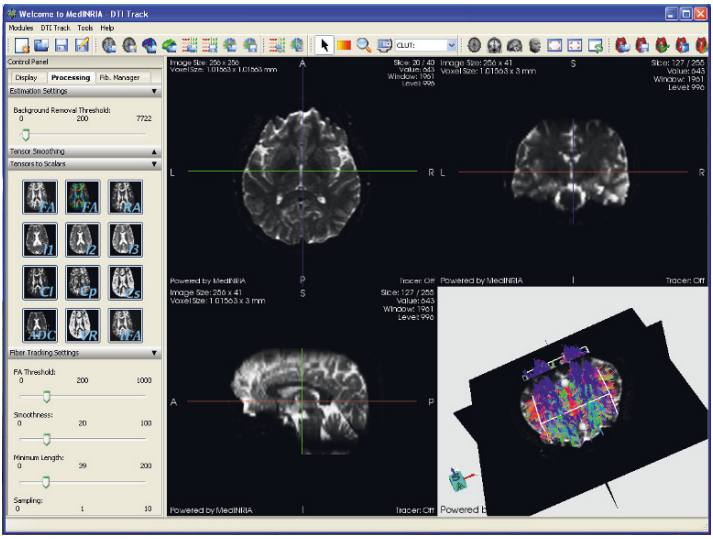



**Fig. 7** MedINRIA DTI-Track module interface.

for the user interface. It can be freely used, but not redistributed, and its source code is not available. It can be downloaded from [19]. As of version 1.7.0, MedINRIA consists of several modules, two of which are specific for diffusion tensor images: the *DTI-Track* and the *Tensor Viewer*.

The DTI-Track module is mainly oriented to the processing of DT-MRI volumes, as well as fiber tracking. It employs Log-Euclidean metrics [2] in its computations, which is protected by a french patent. This ensures that the computed tensors remain in the space of positive semidefinite tensors. The aim of this module is to provide clinicians with the necessary tools for DTI analysis and fiber tracking [19].

In Fig. 7, the DTI-Track module is shown. The interface has a menubar and several toolbars with icons in order to provide an easier access to the menu functions. It also has a tabbed control panel, where each panel is divided into groups of settings, which can be hidden from view. This is interesting, because thanks to it the user can concentrate on the elements of the interface which are relevant to a certain task, and omit the others. The volume and 2D slices are visualized in a dedicated area, and there is a log window, that can also be hidden from view.

In order to employ this module, a DTI Study must be created from a set of DWI images and their respective gradients. Several file formats can be used, although DICOM series must be preprocessed using the ImageViewer module, which also belongs to MedINRIA. Once a study is created, image processing techniques can be applied. As with 3D Slicer, several scalar anisotropy measures can be computed. However, tracts are computed in the whole volume, and bundles can be selected afterwards, using a cropping box or one or more regions of interest (ROI). MedINRIA can compute statistical information about fiber bundles, namely histograms of anisotropy indices and other tensor invariants, their maximum and minimum val-

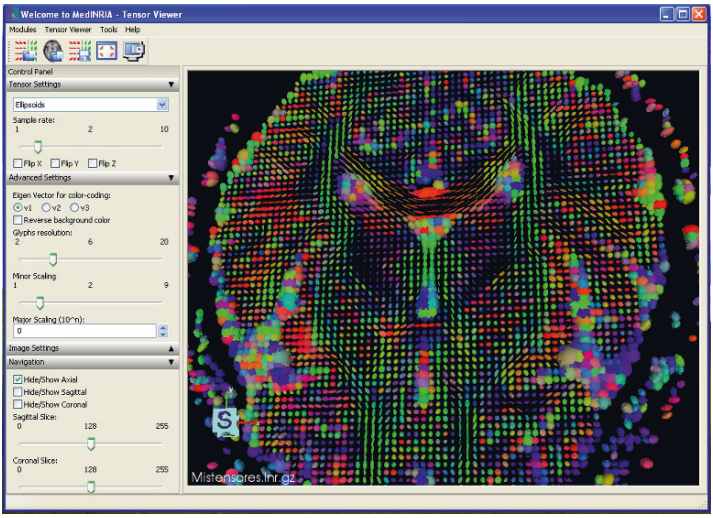



**Fig. 8** MedINRIA Tensor Viewer module interface.

ues, mean and standard deviation, and others like the bundle volume, the number of fibers and some information about their length. Tensors can also be computed and saved so that they can be used with the TensorViewer module.

As in 3D Slicer, the volumes are visualized using 2D orthogonal slices, but volume rendering can also be employed. Tracts can be visualized as streamlines, 3D ribbons or 3D streamtubes.

The TensorViewer module represents the tensor volume using glyphs. They can be lines, arrows, disks, cylinders, cubes, ellipsoids and superquadrics. They are placed in orthogonal slices which can be shown or hidden from view, and the user can set a sampling rate on the voxels that appear in the representation. In Fig. 8, the Tensor Viewer interface is shown when a tensor volume is visualized using cuboid glyphs.

## *5.3 BioTensor*

BioTensor is a medical application that processes and visualizes tensor fields. It can estimate tensors from an adequate set of DWI images, which are previously registered to prevent global distortions caused by eddy currents [15]. The visualization capabilities are the representation of individual tensor samples, the determination of anisotropy regions using isosurfaces of different anisotropy metrics, and the visualization of white matter tracts, obtained by following the principal eigenvector or using the tensorline algorithm [44].

In Fig. 9, this user interface is shown. It is divided in three parts (from left to right): the tensor processing panel, the rendering window, and a panel for visualiza-

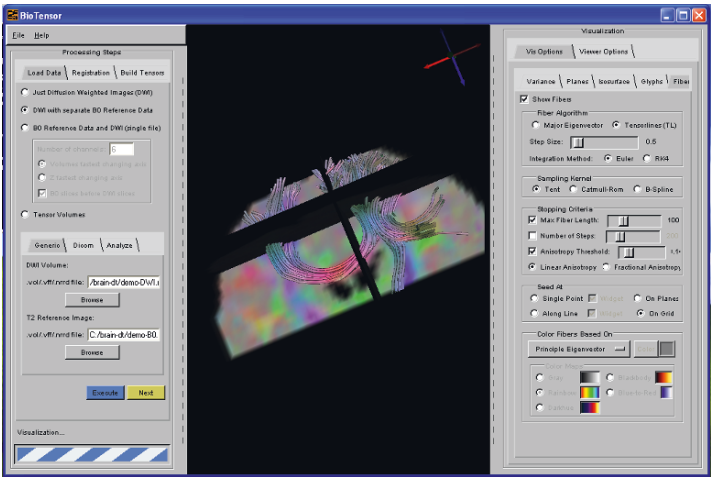



**Fig. 9** BioTensor interface.

tion options. These panels can be detached from the others. We can observe in Fig. 9 that tabbed windows are widely used in this interface.

## *5.4 DtiStudio*

Another interface for DT-MRI is DtiStudio [29]. It is implemented using Visual C++ and OpenGL, and it must be run on a Windows operating system. It is able to read data proceeding from different scanners (Philips, General Electric, Siemens and Toshiba), and perform tasks like estimating the tensor volume from the diffusion weighted images, computing anisotropy indices and fiber tracking. In Fig. 10, this interface can be observed.

## *5.5 Other Graphical Tools*

Other interfaces for DT-MRI are:

- Applications provided by scanner manufacturers.
- The Diffusion Tool of the BioImage Suite [39].
- A DT-MRI add-on for Analyze 7.0 [14].
- A diffusion plugin written for the Neurolens application [22].

In addition, Wünsche [47] developed a toolkit for visualizing and exploring tensor fields.

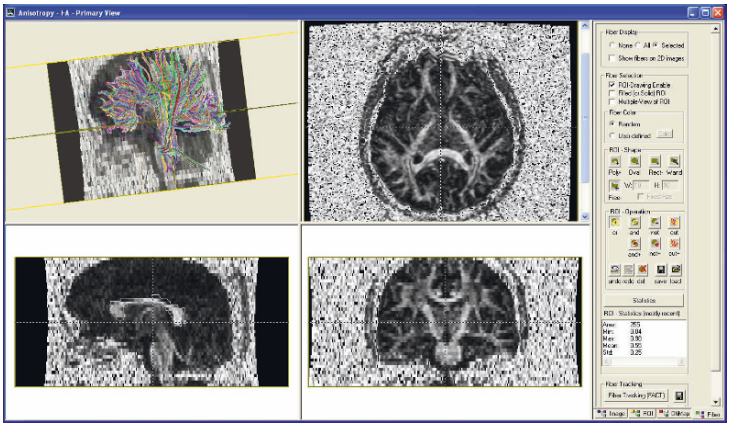


**Fig. 10** DtiStudio interface.

## 6 Multimodal Interfaces for Tensor Processing

In later years, human-computer interaction (HCI) professionals have tried to make interfaces that emulate human-human interaction. This has motivated some research on multimodal interaction, where other input and output devices are used beyond a keyboard, a mouse and a conventional screen. Nowadays some interfaces that include modalities beyond those of the GUI paradigm are starting to be accesible to the average user. Some examples are GPS navigators that include speech recognition and the iPhone, which allows interaction by touching the screen.

Not every user interface for DT-MRI images belongs to the graphic user interface paradigm. There are reports of using a virtual environment for this purpose in [48, 49]. We explain below the interaction devices used in this interface.

### *6.1 Hardware Devices*

Hardware devices are meant to facilitate and improve the possibilities for system interaction and/or visualization. These devices should improve the perception or communication with the system. Other desirable qualities would be their intuitive use, functionality, ergonomy, etc.

Input devices are all the features or systems whose goal is to capture or to transmit the user's stimuli to the system. The devices that capture user motion can be classified as discrete, continuous or hybrid. Keyboards and mice belong to the discrete and hybrid categories, respectively. There are devices that capture other types of information, like auditory (sounds, speech) or visual input.

Output devices transfer information from the system to the user. Most of them use the visual or the auditory channel, but there are also devices that employ haptic

and tactile feedback to communicate with the user. Some devices of interest for the rest of the section are described below.

- Trackers
  One of the most important aspects of 3D interaction in virtual worlds is providing a correspondence between the physical and virtual environments. As a result, having accurate tracking is extremely important to make a virtual environment usable. Currently there are a number of different tracking technologies in the marketplace. The different types of trackers are: magnetic, mechanical, acoustic, inertial, vision/camera and hybrids. These devices often describe an object location using three parameters for its spatial coordinates and another three for its orientation.
- Wand
  A wand is an input device for gesture recognition [8]. It includes a tracking systems and a number of buttons. Its appearance is similar to that of a remote controller.
- The CAVE
  The CAVE is a recursive acronym for Cave Audio-Visual Experience Automatic Virtual Environment. It is composed of three to six faces of a cube surrounding the viewer.These faces are translucent display screens, where stereo images are projected onto [10]. A head-tracking device must be included, in order to adapt the projections to the user's perspective.
  Depending on how many faces there are, it can be classified as either fully immersive (six faces) or partially immersive (three to six faces). This device has also the advantage that more than one person can use it at the same time. However, the perspective is only modified according to the user that is wearing the tracker. It also takes advantage of peripheral vision. However, its dimensions are very large, and it is very expensive.
- Partially Immersive Displays
  While fully immersive displays support the feeling of being a part of the virtual environment, in partially immersive displays the user has the sensation of looking at it from the outside [35]. Some devices that belong to this group are panoramic displays, stereo monitors, etc.
- Haptic Devices
  Haptic devices provide a force feedback to the user movements that simulate the sense of touch.

## *6.2 Immersive Interface for DT-MRI*

There are reports of using virtual immersive environments with tensor field representations. In [48, 49] the use of the CAVE for tract visualization is reported. Thanks to the CAVE, more than one person can use the virtual environment concurrently.

A wand was employed as input device to interact with the virtual world. A ray was simulated as if it came out from the wand, with the same orientation and direc-

tion as the latter. This ray stopped when it hit an object [48]. The user could also interact with the system by leaning toward the represented DT-MRI volume, which made the visualization grow larger [48]. This method was chosen to emulate the natural human behavior when a closer look from an object is wanted.

There is a compromise between visual fidelity and interactivity. When the level of detail is high, the rate of frames per second that can be displayed decreases, diminishing the interactive capacity of the system. This is solved by using different degrees of resolution for different actors in the visualization, prioritizing the objects of more interest to the user [48]. The neural structures (represented in this case study by streamtubes and streamsurfaces) are given high priority, whereas the resolution of anatomical landmarks varies depending on the goal pursued by the visualization. For example, if the visualization goal is brain tumor surgery, the blood vessels surrounding the tumor are given higher priority [48].

### *6.3 Partially Immersive Interface for DT-MRI*

Another virtual reality system is used in [12] for the real-time visualization of fiber tracts in a collaborative environment. For this purpose, a passive stereo display with backprojection and two high resolution beamers were used as output devices, whereas the interactivity was given by an optical tracking system that included a flystick as input device.

### *6.4 DTInteract*

Another multimodal interface is reported in [1], where the addition of a haptic device for interacting with a DT-MRI user interface was studied. The available visualization techniques are 2D orthogonal slices of anisotropy indices or the baseline volume, hyperstreamlines for tractographies, and the major eigenvector at each point of the volume could be visualized using arrow-shaped glyphs. A haptic device was used for controlling the position of the slices, and determining the placement and shape of a spherical ROI. However, the haptic device is not connected to the main interface.

## 7 Discussion

Tensor theory is well established, and tensors are widely used in physics and engineering, so there is little work nowadays done in that respect. However, data visualization and intuitive user interfaces for second-order tensor fields are a relatively new research subject, which is still developing, mainly due to the large amount of information contained in tensors.

Tensor visualization has developed greatly in recent years, and many techniques have arisen, some of which are reviewed in this document. Most of them take as input a second-rank positive definite tensor, like the diffusion tensor, which has been employed in this document. Other tensors do not have real non-negative eigenvalues and can represent a rotation, which complicates both its representation and its interface.

Glyphs allow the visualization of all the information associated to a tensor. As most of them may suffer from ambiguity, color is often used to solve this problem, giving them a hue computed as the interpolation of the tensor anisotropy coefficients $c_l$, $c_p$ and $c_s$. Glyphs, however, have also drawbacks. In the first place, a glyph corresponds to only one tensor, so the continuity of the tensor field is lost in the visualization. Another essential point is the number and position of the represented glyphs, in order to avoid a sensation of cluttering, which might happen if too many glyphs are represented. Furthermore, overlapping of glyphs must be avoided.

Hyperstreamlines and streamtubes are able to provide continuity to the tensor field, at least in the direction of the principal eigenvector, and they are especially well suited for the representation of tractographies. They are not well suited when the region anisotropy is not linear, but streamtubes are a solution for regions of linear anisotropy, leaving the representation of planar anisotropy regions to stream-surfaces. These techniques can also suffer from cluttering, and therefore, selection methods should be employed in order to make the visualization clearer to the user.

These visualization techniques are an important part of current user interfaces for tensor fields. We have described user interfaces for DT-MRI images. Most of them are graphical user interfaces (3DSlicer, MedINRIA, BioTensor, etc.), but, as we have seen, an interface for DT-MRI visualization has been implemented using an immersive virtual environment. Also, the Teem libraries can be employed as a command line interface, or as a graphical interface using Deft. One of the characteristics of an interface is its usability. We have found documented user feedback for 3D Slicer and the immersive virtual environment seen in Section 6.2. Nevertheless, it could be interesting to conduct usability tests on the reviewed interfaces, in order to have quantitative data for a comparison.

A survey about 3D Slicer was conducted from March 2002 to November 2004, whose results can be accessed at [13]. Looking at the answers given by the users can be helpful to determine how effective the user interface of this application is. Nevertheless, we must point out that there are factors in interface design that users only notice when they are absent [7], e.g., if a function in the interface is not working properly, it is immediately brought into attention, but otherwise it passes unnoticed. Therefore, this kind of feature is unlikely to appear on the survey, unless they are unsatisfactory to the participants. One of the most frequent comments is that 3D Slicer, at least for versions lower than 2.0, needed a better graphic user interface (GUI), with a cleaner appearance. Also, the addition of pictures and tooltips to the buttons was desired, which as we have seen is solved in version 2.6. A user found the menus too long. Also, pop-up windows were disliked for entering data and should only be used for errors and warnings. In the current version these windows are also employed for showing measurement results. It was noted, however that sometimes

the GUI was unresponsive. Furthermore, a wide variety of bugs were reported. The user interface to the data visualization (viewer window) was one of the best liked features, as it was displaying the pixel map and the model on a single screen. Interaction between the three 2D dataset views would be appreciated, though. It was also judged useful the ability to overlay labelmaps onto gray scales and to rotate 3D volumes. A user wished for a memo function that allowed to put notes on 2D slices. Interactive volume rendering was a desired new feature. One of the most often mentioned themes was the drawing features. In general, many users wanted more and better tools for this purpose, like for example the ability to draw on the surface anatomy. However, the fact that editing needed multiple layers was disliked. On the other hand, people complained about the measurement tools of Slicer. In particular, measuring distances was difficult and time consuming. A certain comment attracted our attention. Someone admitted not using most of the available features because he was relatively new to Slicer 3D. However, he/she had been using 3D Slicer for almost a year. In fact, most of the participants said that they only worked with a few modules. This shows that 3D Slicer has a large range of functionalities, but many of them are not used. Users also wished for a full tutorial and a user-friendly manual. The newer version of 3D Slicer, Slicer3, has taken into account many of these comments, adopting a user-centered design for its new interface.

As for user feedback on the immersive virtual environment for DT-MRI visualization, it was reported that the physicians and medical students that worked with the system found it easy to use, that the learning time was just a few minutes, and that the system provided a better understanding of the 3D structures than a desktop 2D representation [49]. These systems are, however, very expensive and can be quite seldom accessed.

This is why it could be interesting to develop new interfaces with more affordable interaction devices, briefly reviewed in Section 6.1. These devices could open up possibilities in the addition of new modalities to these interfaces, like gesture recognition (used in the CAVE environment [49]) and others. Using devices with more degrees of freedom could also be a motivating factor for the research into new interaction techniques with tensor fields, taking into account the multidimensionality of tensors.

## 8 Conclusions

Thanks to the computational power provided by current computers, tensor field manipulation has become a reality. The equations and the theory were developed a long time ago, but for the time being, widespread use has not been possible. A tensor provides a great deal of information, but for the same reason, making good user interfaces to interact with these fields is a difficult task, which is being strongly researched and developed.

One of the most active fields related to this purpose is tensor visualization. We have seen a large number of methods for representing tensor fields. All of them have

advantages and drawbacks, so each one is more adequate to certain tasks than the others. For example, the natural way for visualizing tractographies are hyperstreamlines and streamtubes. One of the issues in glyph visualization and tractographies is to choose the number and position of the elements to be represented. If this is not done, we might have cluttered visualizations, which would be confusing to the user. However, if there is a small number of them, they might not be enough to convey a sufficiently detailed representation of the tensor field, which could diminish the comprehension of the tensor field.

Hardware devices are an important part of any user interface, which conditions the interaction techniques that can be employed on it. Although most of the reviewed user interfaces employ standard devices like a conventional display, a mouse and a keyboard, the use of a virtual environment has been reported. The user feedback points out that this system was easy to learn, and that 3D representations of volume tensor data helps the understanding of these images. This supports the idea that multimodal interfaces using emerging hardware, like stereo displays or input devices with a larger number of degrees of freedom, can help in the design of more intuitive and usable interfaces. However, their cost is frequently high and many of them are still under development.

There are not too many implemented user interfaces for tensor visualization, and although the majority are graphical interfaces, some command-line and multimodal interfaces have also been reported. One of the GUIs is 3D Slicer version 2, which has a module for DT-MRI visualization. Although users are satisfied overall with it, they have reported that a better graphic user interface should be provided, and that there is a large part of the available functions that they do not use. They would also like to interact more with the data. This has been taken into account in the development of Slicer3. Other medical image applications have modules, plug-ins or add-ons that deal with tensor images, especially DT-MRI volumes, like MedINRIA, BioTensor, Teem/Deft, etc. Tensor user interfaces are being strongly researched and developed. Some multimodal interfaces for DT-MRI have been developed, which received positive feedback from their users.

**Acknowledgements** Some of the figures were taken using the interfaces 3D Slicer 2.6, MedINRIA 1.7, BioTensor, DtiStudio and Teem/Deft. The authors acknowledge the Spanish Comisión Interministerial de Ciencia y Tecnología for research grant TEC 2007–67073/TCM, the Junta de Castilla y León for grants VA026A07 and VA027A07, and the Consejería de Sanidad de Castilla y León for grant GRS 292/A/08.

## References

1. Aksoy, M., Avcu, N., Merino-Caviedes, S., Diktaş, E.D., Martín-Fernández, M.A., Girgin, S., Marras, I., Muñoz Moreno, E., Tekeli, E., Acar, B., Bammer, R., Martin-Fernández, M., Sahiner, A.V., Üskudarli, S.: DTI Application with Haptic Interfaces. In: Proc. eNTERFACE'07 Workshop Multimodal Interfaces, pp. 1–9. Istanbul, Turkey (2007)

2. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors. Magn. Reson. Med. **56**(2), 411–421 (2006)
3. Basser, P.J., Jones, D.K.: Diffusion-tensor MRI: theory, experimental design and data analysis–a technical review. NMR Biomed. **15**(7-8), 456–467 (2002)
4. Basser, P.J., Pierpaoli, C.: Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. J. Magn. Reson. Ser. B **111**, 209–219 (1996)
5. Bihan, D.L., Mangin, J., Poupon, C.l., Clark, C.A., Pappata, S., Molko, N., Chabriat, H.: Diffusion tensor imaging: concepts and applications. J. Magn. Reson. Imaging **13**(4), 534–546 (2001)
6. Brun, A., Park, H., Knutsson, H., Westin, C.: Coloring of DT-MRI fiber traces using Laplacian eigenmaps. In: Proc. 9th Int. Conf. Comput. Aided Syst. Theory (EUROCAST), pp. 564–572. Las Palmas de Gran Canaria, Spain (2003)
7. Burrell, A., Sodan, A.: Web Interface Navigation Design: Which Style of Navigation-Link Menus Do Users Prefer? In: Proc. 22nd Int. Conf. Data Eng. Workshops, pp. 1–10. Atlanta, GA, USA (2006)
8. Cho, S., Oh, J.K., Bang, W., Chang, W., Choi, E., Jing, Y., Cho, J., Kim, D.Y.: Magic wand: a hand-drawn gesture input device in 3-D space with inertial sensors. In: Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on, pp. 106–111. Kokubunji, Tokyo, Japan (2004)
9. Cook, P.A., Bai, Y., Nedjati-Gilani, S., Seunarine, K.K., Hall, M.G., Parker, G.J., Alexander, D.C.: Camino: Open-Source Diffusion-MRI Reconstruction and Processing. In: 14th Sci. Meet. Int. Soc. Magn. Reson. Med., p. 2759. Seattle, WA, USA (2006)
10. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., Hart, J.C.: The CAVE: audio visual experience automatic virtual environment. Commun. ACM **35**(6), 64–72 (1992)
11. Delmarcelle, T., Hesselink, L.: Visualizing Second-Order Tensor Fields with Hyperstreamlines. IEEE Comput. Graphics Appl. **13**(4), 25–33 (1993)
12. Ehricke, H.H., Klose, U., Grodd, W.: Visualizing MR diffusion tensor fields by dynamic fiber tracking and uncertainty mapping. Comput. & Graphics **30**, 255–264 (2006)
13. Electron. Resour.: 3D Slicer. Medical Visualization and Processing Environment for Research. `www.slicer.org`. Online: June 2006
14. Electron. Resour.: Analyze 7.0. New Features/Enhancements. `http://www.analyzedirect.com/Analyze/enhancements.asp`. Online: November 2006.
15. Electron. Resour.: BioTensor Tutorial. `http://software.sci.utah.edu/doc/User/Tutorials/BioTensor/BioTensor.html`. Online: July 2006
16. Electron. Resour.: Brigham and Women's Hospital website. `http://www.brighamandwomens.org/`. Online: June 2006
17. Electron. Resour.: Computer Science and Artificial Intelligence Laboratory. `http://www.csail.mit.edu/index.php`. Online: June 2006
18. Electron. Resour.: GNU Operating System. Licenses. `http://www.gnu.org/licenses/licenses.html`. Online: August 2008
19. Electron. Resour.: MedINRIA. Asclepios Research Project — INRIA Sophia Antipolis. `http://www-sop.inria.fr/asclepios/software/MedINRIA/`. Online: November 2006
20. Electron. Resour.: National Alliance for Medical Image Computing. `http://www.na-mic.org`. Online: June 2006
21. Electron. Resour.: Nearly Raw Raster Data. `http://teem.sourceforge.net/nrrd/index.html`. Online: December 2006
22. Electron. Resour.: Software. `http://arnaud.guidon.free.fr/blog/?page_id=4`. Online: November 2006
23. Electron. Resour.: Teem Summary. `http://www.na-mic.org/Wiki/index.php/TeemSummary`. Online: November 2006.
24. Electron. Resour.: wxWidgets. Cross-platform GUI Library. `http://www.wxwidgets.org`. Online: September 2008.

25. Gering, D., Nabavi, A., Kikinis, R., Eric, W., Grimson, L., Hata, N., Everett, P., Jolesz, F., Wells III, W.: An Integrated Visualization System for Surgical Planning and Guidance using Image Fusion and Interventional Imaging. In: Med. Image Comput. Comput.-Assist. Interv. (MICCAI) (1999)
26. Gering, D.T.: A System for Surgical Planning and Guidance using Image Fusion and Interventional MR. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (1999)
27. Goldberg-Zimring, D., Mewes, A.U.J., Maddah, M., Warfield, S.K.: Diffusion Tensor Magnetic Resonance Imaging in Multiple Sclerosis. J. Neuroimaging **15**(s4), 68S–81S (2005)
28. Ibanez, L., Schroeder, W., Ng, L., Cates, J.: The ITK Software Guide. `http://www.itk.org/ItkSoftwareGuide.pdf`, second ed. (2005)
29. Jiang, H., van Zijl, P.C.M., Kim, J., Pearlson, G.D., Mori, S.: Dtistudio: Resource program for diffusion tensor computation and fiber bundle tracking. Comput. Methods Programs Biomed. **81**(2), 106–116 (2006)
30. Kindlmann, G.: Superquadric Tensor Glyphs. In: Jt. EUROGRAPHICS–IEEE TCVG Symp. Vis., pp. 147–154, (2004)
31. Kindlmann, G., Weinstein, D., Hart, D.: Strategies for Direct Volume Rendering of Diffusion Tensor Fields. IEEE Trans. Vis. Comput. Graphics **6**(2), 124–138 (2000)
32. Kingsley, P.B.: Introduction to diffusion tensor imaging mathematics: Part I. Tensors, rotations, and eigenvectors. Concepts Magn. Reson. Part A **28A**(2), 101–122 (2006)
33. Kingsley, P.B.: Introduction to diffusion tensor imaging mathematics: Part II. Anisotropy, diffusion-weighting factors, and gradient encoding schemes. Concepts Magn. Reson. Part A **28A**(2), 123–154 (2006)
34. Kitware, Inc.: KWWidgets. `http://www.kwwidgets.org/Wiki/KWWidgets`. Online: september 2008
35. Kjeldskov, J.: Interaction: Full and partial Immersive Virtual Reality Displays. In: Proc. IRIS24, pp. 587–600. Bergen, Norway (2001)
36. Kubicki, M., Westin, C.F., McCarley, R.W., Shenton, M.E.: The Application of DTI to Investigate White Matter Abnormalities in Schizophrenia. Ann. N. Y. Acad. Sci. **1064**, 134–148 (2005)
37. Laidlaw, D.H., Ahrens, E.T., Kremers, D., Avalos, M.J., Jacobs, R.E., Readhead, C.: Visualizing diffusion tensor images of the mouse spinal cord. In: Proc. Vis. '98., pp. 127–134, 527. Research Triangle Park, NC, USA (1998)
38. Melhem, E.R., Mori, S., Mukundan, G., Kraut, M.A., Pomper, M.G., van Zijl, P.C.M.: Diffusion Tensor MR Imaging of the Brain and White Matter Tractography. Am. J. Roentgenol. **178**(1), 3–16 (2002)
39. Papademetris, X., Jackowski, M., Rajeevan, N., Constable, R.T., Staib, L.H.: BioImage Suite: An integrated medical image analysis suite. `http://www.bioimagesuite.org/`. Online: september 2008.
40. Pieper, S.D., Halle, M., Kikinis, R.: 3D Slicer. In: Proc. 2004 IEEE Int. Symp. Biomed. Imaging: From Nano to Macro, pp. 632–635. Arlington, VA, USA (2004)
41. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit. An Object Oriented Approach to 3D Graphics. Kitware, Inc., New Jersey (2002)
42. Sundgren, P.C., Dong, Q., Gómez-Hassan, D., Mukherji, S.K., Maly, P., Welsh, R.: Diffusion tensor imaging of the brain: review of clinical applications. Neuroradiol. **46**, 339–350 (2004)
43. Talos, I.F., O'Donell, L., Westin, C.F., Warfield, S.K., Wells III, W., Yoo: Diffusion Tensor and Functional MRI Fusion with Anatomical MRI for Image Guided Neurosurgery. In: Proc. 6th Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI 2003). Montréal, Canada (2003)
44. Weinstein, D., Kindlmann, G., Lundberg, E.: Tensorlines: Advection-Diffusion based Propagation through Diffusion Tensor Fields. In: IEEE Vis. '99, pp. 249–253. San Francisco, CA, USA (1999)
45. Welch, B.B.: Practical Programming in Tcl and Tk. Prentice Hall PTR, New Jersey (1997)
46. Westin, C.F., Maier, S., Mamata, H., Nabavi, A., Jolesz, F.A., Kikinis, R.: Processing and visualization for diffusion tensor MRI. Med. Image Anal. **6**, 93–108 (2002)

47. Wünsche, B.C.: A Toolkit for the Visualization of Tensor Fields in Biomedical Finite Element Models. Ph.D. thesis, Department of Computer Science. University of Auckland (2004)
48. Zhang, S., Demiralp, c., Keefe, D.F., DaSilva, M., Laidlaw, D.H., Greenberg, B.D., Basser, P.J., Pierpaoli, C., Chiocca, E.A., Deisboeck, T.S.: An immersive virtual environment for DT-MRI volume visualization applications: a case study. In: Proc.Conf. Vis. '01, VIS '01, pp. 437–440. IEEE Computer Society, Washington DC, WA, USA (2001)
49. Zhang, S., Demiralp, c., Laidlaw, D.H.: Visualizing Diffusion Tensor MR Images Using Streamtubes and Streamsurfaces. IEEE Trans. Vis. Comput. Graphics **9**(4), 454–462 (2003)
50. Zheng, X., Pang, A.: Volume Deformation For Tensor Visualization. In: IEEE Vis. 2002, pp. 379–386. Boston, MA, USA (2002)
51. Zheng, X., Pang, A.: HyperLIC. In: IEEE Vis. 2003, pp. 249–256. Seattle, WA, USA (2003)
52. Zheng, X., Parlett, B.N., Pang, A.: Topological Lines in 3D Tensor Fields and Discriminant Hessian Factorization. IEEE Transactions on Visualization and Computer Graphics **11**(4), 395–407 (2005)

# T-flash: Tensor Visualization in Medical Studio

J. Wiklund, V. Nicolas, P. Rondao, M. Andersson and H. Knutsson

**Abstract** Tensor valued data are frequently used in medical imaging. For a 3-dimensional second order tensor such data imply at least six degrees of freedom for each voxel. The operators ability to perceive this information is of outmost importance and in many cases a limiting factor for the interpretation of the data. In this paper we propose a decomposition of such tensor fields using the Tflash tensor glyphs that intuitively conveys important tensor features to a human observer. A matlab implementation for visualization of single tensors are described in detail and a VTK/ITK implementation for visualization of tensor fields have been developed as a Medical Studio component.

## 1 Introduction

An efficient system development process is crucially dependent on the designers ability to interpret system behavior. In the present case the system output is a tensor field and the channel chosen to convey the information is the designers visual system. It is equally important that the information is presented in a way that is easily understood by the end user. In a medical setting it is paramount that the visualiza-

Johan Wiklund
Linköping University, Sweden. e-mail: jowi@imt.liu.se

Vincent Nicolas
Université catholique de Louvain, Belgium. e-mail: vincent.nicolas@uclouvain.be

Patrice Rondao Alface
Université catholique de Louvain, Belgium. e-mail: patrice.rondao@uclouvain.be

Mats Andersson
Linköping University, Sweden. e-mail: matsa@imt.liu.se

Hans Knutsson
Linköping University, Sweden. e-mail: knutte@imt.liu.se

tion techniques used are able to reduce the complicated tensor field information to relevant parameters that are useful in the clinical environment.

Tensor valued image data are frequently used to represent targeted structures in modern medical imaging analysis, e.g. *diffusion tensors*, *strain tensors* and *local structure tensors*. For second order tensors, the by far most common type today, the information stored in each voxel is a $3 \times 3$ tensor for 3-dimensional data. In the simple case of symmetric tensors this implies 6 degrees of freedom (inner dimensions). Visualizing such tensor fields is not a trivial task, showing volume slices of individual tensor components will, for example, not make much sense to a human. The tensor visualization problem is well recognized and a number of approaches have been presented, e.g. [3].

This chapter presents new tensor visualization software, including a tensor glyph generator, that we believe to be preferable in many situations.

## 2 The Tflash Tensor Glyph

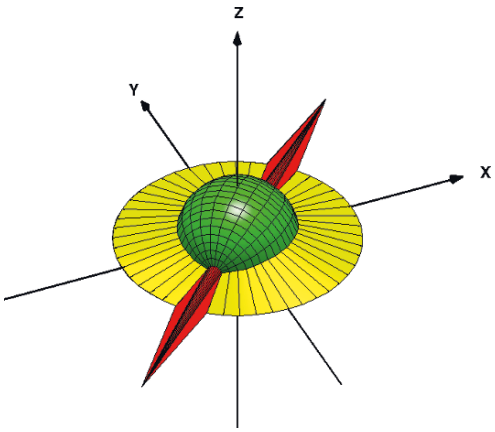



**Fig. 1** Tflash tensor glyph for $\lambda_1 = 1.00$ $\lambda_2 = 0.50$ $\lambda_3 = 0.25$, see text for details.

The tensor visualization tool presented in this chapter is designed for visualization of 2:nd order symmetric tensor fields. Normally the tensor fields are dense. The standard way of decomposing a symmetric second order tensor is

$$\boldsymbol{T} = \lambda_1 \boldsymbol{e}_1 \boldsymbol{e}_1^T + \lambda_2 \boldsymbol{e}_2 \boldsymbol{e}_2^T + \lambda_3 \boldsymbol{e}_3 \boldsymbol{e}_3^T \qquad \lambda_1 \geq \lambda_2 \geq \lambda_3$$

where $\lambda_i$ are the eigenvalues and $\boldsymbol{e}_i$ the corresponding eigenvectors.
The glyph is inspired by the traditional decomposition of the (symmetric second order) local structure tensor into three parts, the linear part, $\boldsymbol{T}_1$, the planar part, $\boldsymbol{T}_2$, and the isotropic part, $\boldsymbol{T}_3$ where

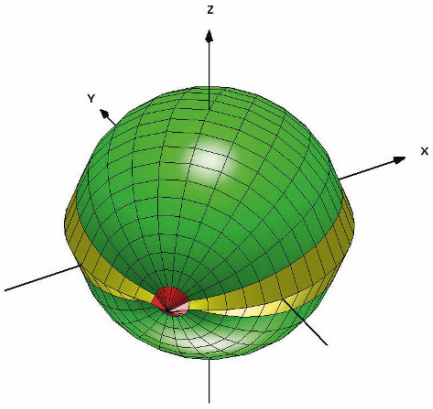



**Fig. 2** An almost isotropic tensor, $\lambda_1 = 0.67 \;\; \lambda_2 = 0.63 \;\; \lambda_3 = 0.59$

$$\boldsymbol{T} = \boldsymbol{T}_1 + \boldsymbol{T}_2 + \boldsymbol{T}_3$$

where

$$\begin{aligned} \boldsymbol{T}_1 &= (\lambda_1 - \lambda_2)\,\boldsymbol{e}_1\boldsymbol{e}_1^T \\ \boldsymbol{T}_2 &= (\lambda_2 - \lambda_3)\,(\boldsymbol{e}_1\boldsymbol{e}_1^T + \boldsymbol{e}_2\boldsymbol{e}_2^T) \\ \boldsymbol{T}_3 &= \lambda_3\,(\boldsymbol{e}_1\boldsymbol{e}_1^T + \boldsymbol{e}_2\boldsymbol{e}_2^T + \boldsymbol{e}_3\boldsymbol{e}_3^T) \end{aligned}$$

The tensor glyph is based on a parametric sphere surface rendering. The magnitude of the isotropic part corresponds to the radius of the sphere. The sphere is rendered in green, see fig. 1. The poles of the sphere are oriented in the direction of $\boldsymbol{e}_1$ and the planar part is visualized as a red spear through the poles where the length of the spear is proportional to the magnitude of $\boldsymbol{T}_1$. The red part of the glyph is generated by expanding the segments of the sphere located around the poles. Finally $\boldsymbol{T}_2$ is visualized by an expansion of the two longitudes of the sphere that corresponds to a plane orthogonal $\boldsymbol{e}_3$. This area of the glyph is rendered in yellow, see fig. 1.

Since the Tflash glyph is based on ad distorted sphere the visualization becomes very efficient and the distinct shape together with the coloring improves the observers ability to perceive the geometry of the glyph from all views. The difficulties in perceiving the correct geometry using more fundamental glyphs as e.g. an ellipsoid is one of the main reasons for the introduction of the Tflash glyph.

In fig. 2 another example of the Tflash tensor glyph is shown. The eigenvectors are identical to the tensor in fig.1 but the distribution of the eigenvalues correspond to a much more isotropic tensor with $\lambda_1 = 0.67 \;\; \lambda_2 = 0.63 \;\; \lambda_3 = 0.59$. For the

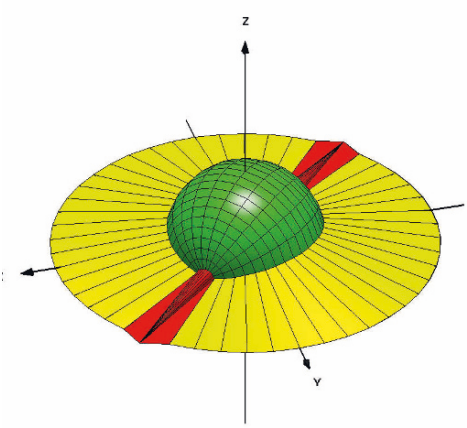



**Fig. 3** Planar part, $\boldsymbol{T}_2$, is dominant, $\lambda_1 = 0.90 \;\; \lambda_2 = 0.85 \;\; \lambda_3 = 0.35$.

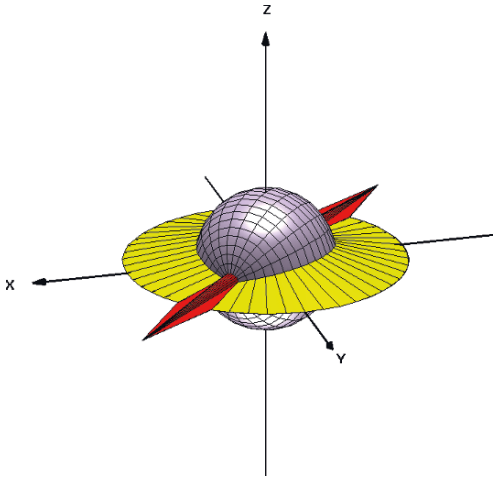



**Fig. 4** A negative $\lambda_3$ is visualized by a change in color of the isotropic part, $\lambda_1 = 1 \;\; \lambda_2 = 0.6 \;\; \lambda_3 = -0.30$.

tensor in fig. 3 the planar part, $\boldsymbol{T}_2$, is dominant ($\lambda_1 = 0.95 \;\; \lambda_2 = 0.90 \;\; \lambda_3 = 0.85$). The yellow part is oriented in a direction orthogonal to $\boldsymbol{e}_3$.

As the local structure tensor describes the directed energy distribution the Fourier domain there exists no physical interpretation of an indefinite local tensor. Negative eigenvalues ($\lambda_3 < 0$) may, however, occur due to e.g. phase interference within the filters in complex or low SNR signal neighborhoods. Although this is a relatively rare event it indicates that the second order tensor model is not sufficient to represent

such a neighborhood and it is important to convey this information to the observer of the tensor glyph.

The shape of the Tflash tensor glyph is based on the absolute value of the eigenvalues but a negative sign of the eigenvalues cause a distinct change in the color rendering. In fig. 4 $\lambda_1 = 1$ $\lambda_2 = 0.6$ $\lambda_3 = -0.30$ which introduce a blue haze in the isotropic part. It would be straight forward to let the sign of the eigenvalues affect the shape of the glyph as well. This would on the other hand make the glyph more difficult to interpret and could cause ambiguities in certain views.

## 3 Implementation as a MedicalStudio Component

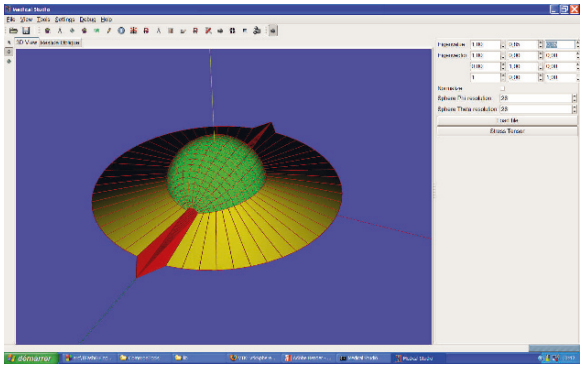


**Fig. 5** A single glyph represented in MedicalStudio.

MedicalStudio is a crossplatform framework for visualization, interaction and processing of medical images. It is based on popular open source libraries such as VTK for visualization, ITK for image processing, GTKmm for graphical user interface and DCMTK for Dicom compatibility. Intended to be flexible, the implementation allows an easy integration of new components. Mono and multimodal registration, atlas-based segmentation, 3D reconstruction, augmented visualization are some examples of the already available components.

Tensor field visualization by Tflash glyphs has been implemented on the basis of the vtkTensorGlyph class of VTK. Starting from a parameterized sphere primitive, the spear and circle are created by displacing the poles and a meridian in the normal direction according to the eigenvalues (see Fig. 5). The color code is implemented by adding face color attributes. The final orientation of the glyph is then obtained through a transform matrix defined by the tensor eigenvectors.

The integration into MedicalStudio is done by creating only one graphical component. This component drives the new TflashGlyphFilter and exposes its parameters to the user, which allows to easily test several configurations according to the

loaded data. Visualization of the generated glyphs, navigation through the volume, data management, interaction management, etc., a lot of components are already implemented into MedicalStudio, letting the programmer to concentrate on its main task.

## 4 Result

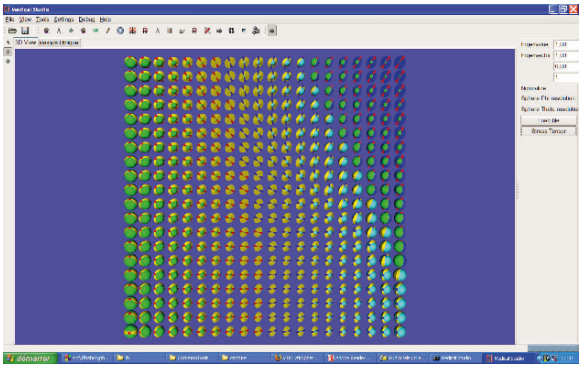


**Fig. 6** Phantom regular grid illustrating some of the many possible configurations of a tensor glyph.

Fig.6 represents the Tflash glyph on a phantom regular grid with various eigenvalues sets and slowly varying orientations. This figure shows the capability of the Tflash to naturally highlight global principal directions as well as to illustrate the various local properties of the tensor data. An illustration of the Tflash glyph on DTI tensor data is represented on Fig. 7.

When compared to other glyphs, the Tflash necessitates a very small amount of polygons to be rendered (i.e. as many as the well-known minimalist ellipsoids) while describing as much information as the superquadrics glyph [2]. This point facilitates a fast and fluid rendering of the tensor data. However, there is still no objective way to compare tensor visualization tools. Actually, benchmarking tools are strongly needed to test the performances of the state of the art tensor visualization techniques. Their performances mainly relate to the psycho-visual efficiency as well as to the rendering efficiency. This is an open issue that deserves a common effort of the tensor processing community.

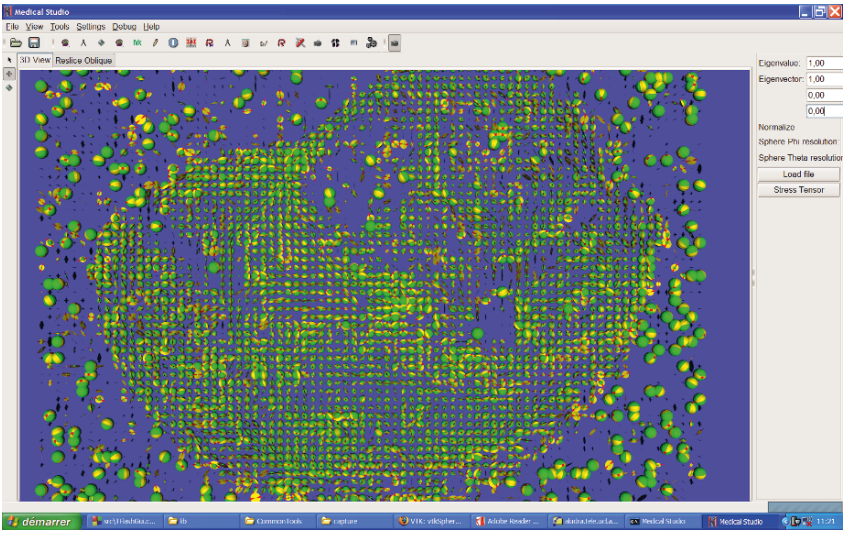



**Fig. 7** Tflash and DTI tensor visualization in MedicalStudio.

## 5 Conclusion

This paper has presented the Tflash glyph as an efficient tool to visualize tensor data. This glyph enables to univocally represent tensor eigenvector directions as well as the relative amplitudes of tensor eigenvalues in a very simple 3D model. The generation of the glyph is very simple and fast since it only necessitates the morphing of a very small number of points of a generic sphere. The color code is also simple and clear. This chapter has also presented the MedicalStudio platform as well as the easy integration of the Tflash as a component.

**Acknowledgements** We gratefully acknowledge the support from EU-FP6 (through the SIMILAR Network of Excellence), VINNOVA (The Swedish Governmental Agency for Innovation Systems), CMIV (the Center for Medical Image Science and Visualization at Linköping University), VR (the Swedish Research Council), and SSF (the Swedish Foundation for Strategic Research through the MOVIII Strategic Research Center).

## Appendix: Matlab code for Tflash

```
function [ht, hxy, hxz, hyz] = Tflash(Lambda, EigVec, norm)
% [ht, hxy, hxz, hyz] = Tflash(Lambda, EigVec, norm)
%
% Tflash visualizes a symmetric 2nd order 3 dimensional tensor.
% The object consists of three parts corresponding to the
% eigenvalues. The absolute value of the eigenvalues in descending
```

```
% order maps to a spear, a disc and a sphere. The orientation of the
% different parts are given by the corresponding eigenvectors. For
% positive eigenvalues the spear is red, the disc yellow and the sphere
% green. A negative eigenvalue will change the color to hazy purple.
%
%   Lambda: Eigenvalues (3 component row vector [l1, l2, l3])
%   EigVec: Corresponding eigenvectors (3x3 matrix [e1; e2; e3])
%   norm:   normalize with largest eigenvalue
%   ht:     Handle to tensor shape object
%   hxy:    Handle to xy-plane object
%   hxz:    Handle to xy-plane object
%   hyz:    Handle to xy-plane object
%
% Author: Johan Wiklund, jowi@isy.liu.se
% Department of Medical Engineering
% Link\"{o}ping University


if nargin ~= 3
  help Tflash
  return
end

% Check sign of Lambda
Ls = Lambda >= 0;
Lambda = abs(Lambda);

% Normalize EigVec
EigVec = diag(sum(EigVec'.^2+eps).^-0.5)*EigVec;

% Sort in descending order
[Lambda,I]=sort(Lambda,'descend');
EigVec=EigVec(:,I);
Ls = Ls(I);

N=24;
if norm
  L0=1.0;
  L1=Lambda(2)/Lambda(1);
  L2=Lambda(3)/Lambda(1);
else
  L0=Lambda(1);
  L1=Lambda(2);
  L2=Lambda(3);
end

if L2 == 0
  L2 = eps;
end

gr = [0.95 0.85 1.0];
% Define colormap
if Ls(3)
  csphere=[0 1 0];
```

```
else
  csphere=gr;
end
if Ls(2)
  cdiscus=[1 1 0];
else
  cdiscus=gr;
end
if Ls(1)
  cspear=[1 0 0];
else
  cspear=gr;
end
cgrid=[0 0 0];
colormap([csphere; cdiscus; cspear; cgrid]);

% Sphere controlled by L2
[X,Y,Z]= sphere(N);
X=L2*X;
Y=L2*Y;
Z=L2*Z;

% Discus controlled by L1
edge=L1/L2;
long0=1;
long1=N/2+1;
long2=N+1;
X(:,long0)= edge*X(:,long0);
X(:,long1)= edge*X(:,long1);
X(:,long2)= edge*X(:,long2);
Z(:,long0)= edge*Z(:,long0);
Z(:,long1)= edge*Z(:,long1);
Z(:,long2)= edge*Z(:,long2);

% Spear controlled by L0
Z(1,:)= -L0;
Z(N+1,:)= L0;

% Rotate shape
v1=EigVec(:,1)';
v2=EigVec(:,2)';
v3=EigVec(:,3)';

T=[v2;v3;v1];
NC=[X(:) Y(:) Z(:)]*T;
X=reshape(NC(:,1), N+1, N+1);
Y=reshape(NC(:,2), N+1, N+1);
Z=reshape(NC(:,3), N+1, N+1);

% Set colors
C=zeros([N,N]);
C(:,1)=1;
C(:,N/2)=1;
C(:,N/2+1)=1;
```

```
C(:,N)=1;
C(1,:)=2;
C(N,:)=2;

% Double longitudes at discontinuities
C1=N/2;
C2=N/2+1;
C3=N/2+2;
X=[X(:,1:2) X(:,2:C1) X(:,C1:C2) X(:,C2:C3) X(:,C3:N) X(:,N:N+1)];
Y=[Y(:,1:2) Y(:,2:C1) Y(:,C1:C2) Y(:,C2:C3) Y(:,C3:N) Y(:,N:N+1)];
Z=[Z(:,1:2) Z(:,2:C1) Z(:,C1:C2) Z(:,C2:C3) Z(:,C3:N) Z(:,N:N+1)];
C=[C(:,1:2) C(:,2:C1) C(:,C1:C2) C(:,C2:C3) C(:,C3:N) C(:,N)];

% Double latitudes at discontinuities
X=[X(1:2,:) ; X(2:N,:) ; X(N:N+1,:)];
Y=[Y(1:2,:) ; Y(2:N,:) ; Y(N:N+1,:)];
Z=[Z(1:2,:) ; Z(2:N,:) ; Z(N:N+1,:)];
C=[C(1,:) ; C(1:N,:) ; C(N,:)];

% Render surface
ht=surf(X,Y,Z,C);
caxis([0 3]);
set(ht, 'FaceLighting', 'phong')
%set(ht, 'EdgeColor', 'none');
set(ht, 'EdgeColor', cgrid);
%set(ht, 'AmbientStrength', 0.5);
%set(ht, 'DiffuseStrength', 0.8);
%set(ht, 'SpecularStrength', 1.0);
%set(ht, 'SpecularCol\orReflectance', 0.7);
%set(ht, 'BackFaceLighting', 'lit');
%set(ht, 'FaceLighting', 'phong',...
%       'EdgeColor', [0.0 0.0 0.0],...
%       'AmbientStrength', 0.9,...
%       'DiffuseStrength', 0.8,...
%       'SpecularStrength', 1.0,...
%       'SpecularColorReflectance', 0.7,...
%       'BackFaceLighting', 'lit');
axis([-1 1 -1 1 -1 1]);
axis vis3d;
axis off;

% Render eigenvectors
%v1=1.5*v1;
%v2=1.5*v2;\
%patch([0 v1(1)], [0 v1(2)], [0 v1(3)], [1.0 1.0 1.0]);
%patch([0 v2(1)], [0 v2(2)], [0 v2(3)], [1.0 1.0 1.0]);

% Render coordinate axis
as=0.95;
aw=0.01;
vert1=[-1.0 1.0 as  as  as  as]';
vert2=[ 0.0 0.0 aw  aw -aw -aw]';
vert3=[ 0.0 0.0 aw -aw -aw  aw]';
faces=[1 2 2; 2 3 4; 2 4 5; 2 5 6; 2 6 3];
```

```
% X-axis
hx=patch('Vertices', [vert1 vert2 vert3], 'Faces', faces, 'FaceColor',
[0 0 0]); set(hx, 'LineWidth', 1.0);
% Y-axis
hy=patch('Vertices', [vert2 vert1 vert3], 'Faces', faces, 'FaceColor',
[0 0 0]); set(hy, 'LineWidth', 1.0);
% Z-axis
hz=patch('Vertices', [vert2 vert3 vert1], 'Faces', faces, 'FaceColor',
[0 0 0]); set(hz, 'LineWidth', 1.0);

% Labels
text(1.1, 0, 0, '\bf X', 'HorizontalAlignment', 'Center');
text(0, 1.1, 0, '\bf Y', 'HorizontalAlignment', 'Center');
text(0, 0, 1.1, '\bf Z', 'HorizontalAlignment', 'Center');

% Axis planes
hold on
coord=[-1:0.1:1]'*ones(1,21);
zc=zeros([21,21]);
C=3*ones([21,21]Tflash(L,E));
% XY plane
X=coord;
Y=coord';
Z=zc;

% Check matlab version
verstr = version;
ver = str2num(verstr(1:3));

if ver < 6
    hxy=mesh(X,Y,Z,C);
    set(hxy, 'Visible', 'off');
    % XZ plane
    hxz=mesh(X,Z,Y,C);
    set(hxz, 'Visible', 'off');
    % YZ plane
    hyz=mesh(Z,X,Y,C);
    set(hyz, 'Visible', 'off');
else
    hxy=mesh(X,Y,Z,C);
    set(hxy, 'Visible', 'off', 'FaceAlpha', 0.5, 'EdgeAlpha', 0.5);
    % XZ plane
    hxz=mesh(X,Z,Y,C);
    set(hxz, 'Visible', 'off', 'FaceAlpha', 0.5, 'EdgeAlpha', 0.5);
    % YZ plane
    hyz=mesh(Z,X,Y,C);
    set(hyz, 'Visible', 'off', 'FaceAlpha', 0.5, 'EdgeAlpha', 0.5);
end
hold off

% Camera properties
camproj('perspective');
camva(7);
```

```
% Light!
l1=light('Position',  v3);
l2=light('Position', -v3);

rotate3d on;
```

## References

1. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995.
2. Gordon Kindlmann. Superquadric tensor glyphs. In *Proceeding of The Joint Eurographics - IEEE TCVG Symposium on Visualization*, pages 147–154, May 2004.
3. Gordon Kindlmann. *Visualization and Analysis of Diffusion Tensor Fields*. PhD thesis, University of Utah, 2004.
4. H. Knutsson. Representing local structure using tensors. In *The 6th Scandinavian Conference on Image Analysis*, pages 244–251, Oulu, Finland, June 1989. Report LiTH–ISY–I–1019, Computer Vision Laboratory, Linköping University, Sweden, 1989.
5. V. Nicolas and B. Macq. Medical Studio. http://www.medicalstudio.org.

DOI 10.1007/978-1-84882-299-3_22

**ERRATUM**

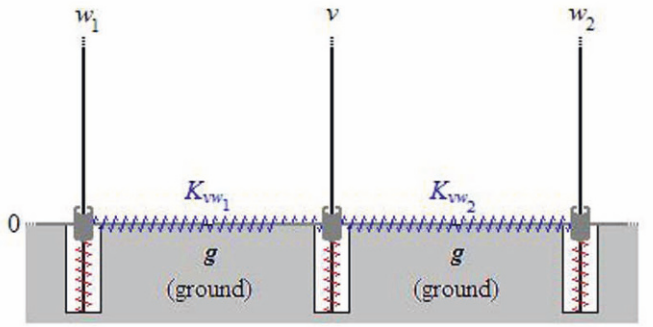
# Tensors in Image Processing and Computer Vision

**Edited by Santiago Aja-Fernández, Rodrigo de Luis García, Dacheng Tao, Xuelong Li**

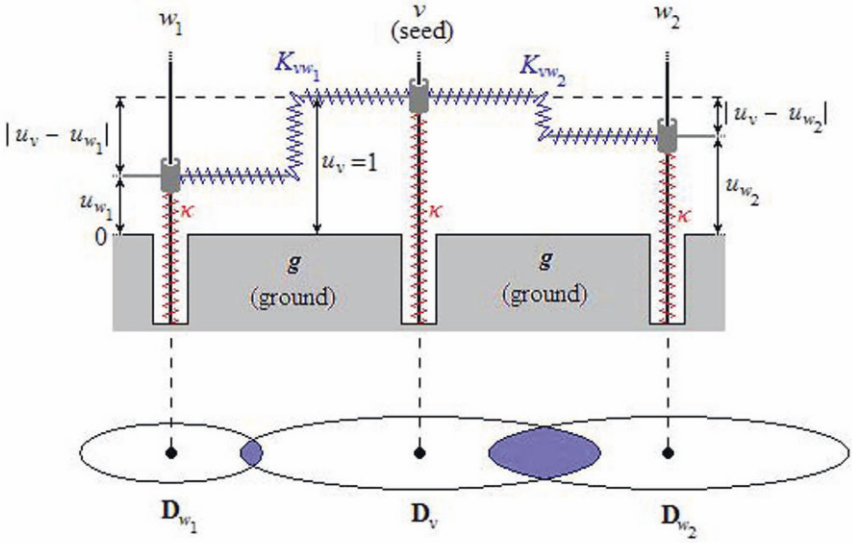
ISBN: 978-1-84882-298-6



The Chapter entitled *DT-MRI Connectivity and/or Tractography?: Two New Algorithms* by Burak Acar and Erdem Yörük, DOI 10.1007/978-1-84882-299-3_16, pages 335-353 contains a figure that is incorrect, i.e., Fig. 1 on page 341. The publisher regrets that by mistake the present figure was not replaced. The right figure is printed on the next page:


$w_1$
$v$
$w_2$
$K_{vw_1}$
$K_{vw_2}$
0
$g$
(ground)
$g$
(ground)

(a)


$w_1$
$v$
(seed)
$w_2$
$K_{vw_1}$
$K_{vw_2}$
$|u_v - u_{w_1}|$
$u_v = 1$
$|u_v - u_{w_2}|$
$u_{w_1}$
$u_{w_2}$
0
$K$
$K$
$K$
$g$
(ground)
$g$
(ground)
$\mathbf{D}_{w_1}$
$\mathbf{D}_v$
$\mathbf{D}_{w_2}$

(b)

Springer

# Index