Nadia Magnenat-Thalmann (Ed.)

LNCS 5903

Modelling the Physiological Human

3D Physiological Human Workshop, 3DPH 2009
Zermatt, Switzerland, November/December 2009
Proceedings

Springer

Lecture Notes in Computer Science 5903

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Nadia Magnenat-Thalmann (Ed.)

Modelling the Physiological Human

3D Physiological Human Workshop, 3DPH 2009
Zermatt, Switzerland, November 29 - December 2, 2009
Proceedings

Springer

Volume Editor

Nadia Magnenat-Thalmann
MIRALab/C.U.I, University of Geneva
Battelle, 7 route de Drize, 1227 Carouge/Geneva, Switzerland
E-mail: thalmann@miralab.unige.ch

Library of Congress Control Number: Applied for

CR Subject Classification (1998): J.3, I.3, I.3.5, I.4, I.5, H.5.2

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-642-10468-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-10468-8 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12804917 06/3180 5 4 3 2 1 0

Preface

This book presents recent advances in the domain of the 3D physiological human that were presented last December at the Workshop on 3D Physiological Human 2009 that was held in Zermatt, Switzerland. This workshop was funded by the "Third Cycle in Computer Science of Western Switzerland" named CUSO, the European project Focus K3D (ICT-2007-214993), the European Marie Curie project 3D Anatomical Human (MRTN-CT-2006-035763) and the European Network of Excellence InterMedia (NoE-IST-2006-038419).

3D physiological human research is a very active field supported by several scientific projects. Many of them are funded by the European Union, such as the 3D Anatomical Human project and those present in the seventh framework programme "Virtual Physiological Human"(FP7-ICT-2007-2). One of the main objectives of the research on 3D physiological human is to create patient-specific computer models for personalized healthcare. These models are used to simulate and hence better understand the human physiology and pathology. There is also a synergy in this research in the way medical information is distributed: to have any model available anytime, anywhere on any mobile equipment.

A collection of scientific articles was proposed to highlight the necessity to exchange and disseminate novel ideas and techniques from a wide range of disciplines (computer graphics, biomechanics, knowledge representation, human–machine interface, mobile computing, etc.) associated with medical imaging, medical simulation, computer-assisted surgery and 3D semantics. The emphasis was on technical novelty along with current and future applications for modeling and simulating the anatomical structures and functions of the human body.

The book is divided into six main parts: Segmentation Methods, Anatomical and Physiological Modeling, Simulation Models, Motion Analysis, Medical Visualization and Medical Ontology. It includes 19 papers that were carefully selected by an International Program Committee.

December 2009 Nadia Magnenat-Thalmann

Organization

3DPH 2009 was organized by MIRALab, University of Geneva.

Workshop Chairs

Nadia Magnenat-Thalmann	MIRALab/University of Geneva, Switzerland
Seunghyun Han	MIRALab/University of Geneva, Switzerland
Jinman Kim	MIRALab/University of Geneva, Switzerland
Daniel Thalmann	EPFL, Switzerland

Program Co-chairs

Nadia Magnenat-Thalmann	MIRALab/University of Geneva, Switzerland
Nuria Oliver	Telefónica R&D, Barcelona, Spain
Eric Stindel	University Hospital of Brest, France

Local Committee Chairs

Caecilia Charbonnier	MIRALab/University of Geneva, Switzerland
Jérôme Schmid	MIRALab/University of Geneva, Switzerland

International Program Committee

Massimiliano Baleani	IOR, Italy
Raffaele Bolla	DIST/University of Genoa, Italy
Werner Ceusters	State Center of Excellence in Bioinformatics & Life Sciences, USA
Jian Chang	Bournemouth University, UK
Gary E. Christensen	University of Iowa, USA
Albert C.S. Chung	Hong Kong University of Science and Technology, Hong Kong
Hervé Delingette	INRIA, France
Scott Delp	Stanford University, USA
Mark de Zee	SMI, Denmark
Tor Dokken	SINTEF, Norway
Guy A. Dumont	University of British Columbia, Canada
Daniel Espino	IOR, Italy
David D. Feng	Sydney University, Australia
Stephen Ferguson	University of Bern, Switzerland
John Fisher	Institute of Medical & Biological Engineering, UK

Andrea Giachetti	CRS4 and University of Verona, Italy
Enrico Gobbetti	CRS4, Italy
James K. Hahn	George Washington University, USA
Hans-Christian Hege	Zuse Institute Berlin, Germany
Tobias Heimann	INRIA, France
Pheng-Ann Heng	The Chinese University of Hong Kong, Hong Kong
Pierre Hoffmeyer	HUG/University of Geneva, Switzerland
Wolfgang Hürst	Utrecht University, The Netherlands
Nigel John	University of Wales, UK
Chris Joslin	Carleton University, Canada
Leo Joskowicz	The Hebrew University of Jerusalem, Israel
Jinman Kim	MIRALab/University of Geneva, Switzerland
WonSook Lee	University of Ottawa, Canada
Vincent Luboz	Imperial College London, UK
Nadia Magnenat-Thalmann	MIRALab/University of Geneva, Switzerland
Dimitris N. Metaxas	State University of New Jersey, USA
Nassir Navab	Technische Universität München, Germany
Bernhard Preim	University of Magdeburg, Germany
Nicolas Pronost	EPFL, Switzerland
Hong Qin	State University of New York at Stony Brook, USA
Ewald Quak	SINTEF, Norway
Michela Spagnuolo	CNR, Italy
Eric Stindel	University Hospital of Brest, France
Daniel Thalmann	EPFL, Switzerland
Andrew Todd-Pokropek	UCL, UK
Remco Veltkamp	Utrecht University, The Netherlands
Franz-Erich Wolter	University of Hannover, Germany

Sponsoring Projects and Institutions

3D Anatomical Human (MRTN-CT-2006-035763)
http://3dah.miralab.unige.ch

InterMedia (NoE-IST-2006-038419)
http://intermedia.miralab.unige.ch

Focus K3D (ICT-2007-214993)
http://www.focusk3d.eu

Conférence Universitaire de Suisse Occidentale (CUSO)
http://www.cuso.ch

Table of Contents

I Segmentation

II Anatomical and Physiological Modeling

III Simulation Models

IV Motion Analysis

V Medical Visualization and Interaction

VI Medical Ontology

Vessels-Cut: A Graph Based Approach to Patient-Specific Carotid Arteries Modeling

Moti Freiman[1], Noah Broide[1], Miriam Natanzon[1], Einav Nammer[2],
Ofek Shilon[2], Lior Weizman[1], Leo Joskowicz[1], and Jacob Sosna[3]

[1] School of Eng. and Computer Science, The Hebrew Univ. of Jerusalem, Israel
[2] Simbionix Ltd, Israel
[3] Dept. Of Radiology, Hadassah Hebrew University Medical Center, Israel
`freiman@cs.huji.ac.il`

Abstract. We present a nearly automatic graph-based segmentation method for patient specific modeling of the aortic arch and carotid arteries from CTA scans for interventional radiology simulation. The method starts with morphological-based segmentation of the aorta and the construction of a prior intensity probability distribution function for arteries. The carotid arteries are then segmented with a graph min-cut method based on a new edge weights function that adaptively couples the voxel intensity, the intensity prior, and geometric vesselness shape prior. Finally, the same graph-cut optimization framework is used for nearly automatic removal of a few vessel segments and to fill minor vessel discontinuities due to highly significant imaging artifacts. Our method accurately segments the aortic arch, the left and right subclavian arteries, and the common, internal, and external carotids and their secondary vessels. It does not require any user initialization, parameters adjustments, and is relatively fast (150–470 secs). Comparative experimental results on 30 carotid arteries from 15 CTAs from two medical centres manually segmented by expert radiologist yield a mean symmetric surface distance of 0.79mm (std=0.25mm). The nearly automatic refinement requires about 10 seed points and took less than 2mins of treating physician interaction with no technical support for each case.

1 Introduction

Minimally invasive endovascular surgeries such as carotid, coronary, and cerebral angiographic procedures are frequent in interventional radiology. They require experienced physicians and involve time-consuming trial and error with repeated contrast agent injection and X-ray imaging. This leads to outcome variability and non-negligible complication rates. Training simulators such the ANGIO Mentor^{TM} (Simbionix Ltd, Israel) have the potential to significantly reduce the physicians' learning curve, reduce the outcome variability, and improve their performance. A key limitation is the simulators' reliance on hand-tailored anatomical models generated by a technician from CTA scans, which are impractical to produce patient-specific simulations in a clinical environment.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 1–12, 2009.

Carotid arteries segmentation from CTAs is a challenging task because of significant intra and inter-patient carotid intensity and geometry variance, intensity values overlap of carotid arteries and neck vertebrae, and streaking artifacts, among others. Numerous automatic and semi-automatic vascular structure segmentation methods have been developed recently [1]. The main approaches rely on intensity values [2], geometric shape [3], edge-based active contours [4,5], statistical active shape models [6], and contour tracking [7,8,9]. Since these segmentation algorithms have been developed for specific regions, such as the carotid bifurcation, their performance on the segmentation of the entire vascular anatomy often yield inaccurate vessel diameters, miss vessels segments and entire small vessels, include non-vessel anatomical structures, and incorrectly model vessel bifurcations and pathologies. These failures are often due to severe stenosis around the carotid bifurcation [8] and the local vessels appearance and characteristics variability in various regions of the body. In addition, most methods require extensive user interaction and expert technical support for the adjustment of non-intuitive parameters, which difficult their routine clinical use.

A promising approach is the graph min-cut segmentation method [10,11]. It classifies the voxel nodes that separate the objects of interest and the background based on weighted voxel adjacencies. The advantages of graph min-cut segmentation are that it is generic and that it is nearly parameter-free. However, it relies on significant user interaction and requires a technical expert to fine-tune the vessels intensity priors to capture the small vessels variability. It does not incorporate vessels geometric information, is computationally intensive, and has extensive memory requirements.

Recent improvements to the graph min-cut interactive segmentation method address some of these drawbacks. Slabaugh and Unal [12] add an elliptical shape prior term to the edges cost function. Sinop and Grady [13] use a Laplacian pyramid to accelerate the segmentation and to reduce the memory requirements. Ning et al. [14] improve the segmentation with graph-cut active contours based on prior object surface estimation. Rother et al [15] require user-defined enclosing rectangular regions around the objects of interest. While this method is useful for extracting simple objects in 2D images of natural scenes, it is laborious for 3D scans of complex vascular structures. A common key drawback of all these methods is that their generic segmentation framework is often ill-suited for volumetric vessels segmentation.

In this work we present a nearly automatic graph min-cut segmentation method for the carotid arteries and the aorta. It consists of automatic segmentation step followed by nearly automatic removal of few vein segments and filling minor vessel discontinuities due to large intensity variations. The main contributions of this work are: 1) a new graph-cut with tubular shape prior formulation for vessels segmentation, 2) a novel edge weights function that adaptively couples intensity with geometric vesselness shape prior, and; 3) a new nearly automatic method for segmentation of the entire carotids system.

The automatic step does not require any user initialization or parameters adjustment. It can segment the entire head and neck CTA scans ($512 \times 512 \times 750$

voxels) in 150–470secs, depending on the scan resolution. We then use the same graph-cut optimization framework to interactively remove a few vein segments and to fill minor vessel discontinuities due to large intensity variations. Experimental results on 30 carotid arteries from 15 scans from two medical centers show that our method is accurate, reliable, and fast enough to generate patient specific models for interventional radiology simulation with only 2mins of treating physician intervention and no technical support.

2 Method

Our segmentation method consists of: 1) automatic aortic arch segmentation, 2) automatic carotid, vertebral, and subclavian arteries segmentation, and; 3) interactive graph-based refinement. We describe each stage in detail next.

2.1 Aortic Arch Segmentation

The aorta is the dominant arch-like vessel above the pulmonary artery in the lower region of a head and neck CTA scan. Aortic arch segmentation relies on prior anatomical knowledge of the aortic arch structure: the aorta location, its estimated radius, and its relative brightness.

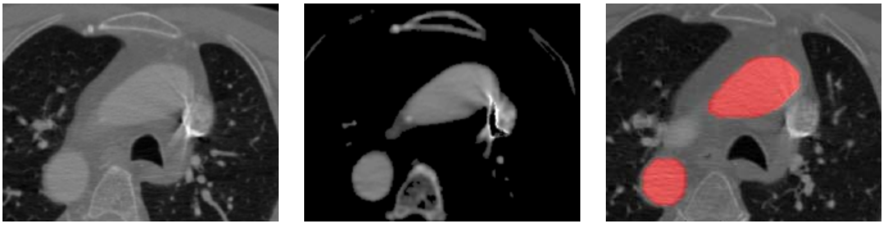
Our aortic arch segmentation algorithm starts with automatic Region of Interest (ROI) selection and enhancement, followed by morphological segmentation. The lower fourth of the CTA scan is first selected, downsampled (1:9), and smoothed with a Gaussian kernel (Fig. 1a). Next, background voxels whose values are outside the vessels intensity range (typically 0-600 HU) are eliminated. Finally, a log-based intensity transform is applied to enhance the dominant blood vessels and conceal small vessels (Fig. 1b). A morphological ellipsoid operator is then applied to the enhanced ROI. Each voxel is classified based on the high homogeneity and low number of background voxels of the ellipsoid voxel neighborhood whose dimensions are derived from the typical aorta radius.

The resulting classification includes three connected components: the aortic arch, the pulmonary artery, and the spinal cord. We grade these components by low variance and arch-like structure and choose the one with the highest score (Fig. 1c). Finally, mean and standard deviation aorta intensity values are computed for their use as prior Intensity Probability Distribution Function (IPDF).

2.2 Carotid Arteries Segmentation

We use a min-cut graph segmentation approach [10] to segment the arteries. We combine the estimated aorta IPDF, the geometric tube-like shape prior based on multi-scale Hessian eigen-analysis [3], and the local image gradients into a graph-based image volume representation.

Let $G = (V, E)$ be the image graph, where $V = \{v_1, ...v_n, v_s, v_t\}$ are the graph nodes such that node v_i corresponds to voxel i and terminal nodes v_s and v_t correspond to object and background, respectively. Graph edges



(a) Original slice (b) After enhancement (c) Segmentation result

Fig. 1. Automatic aorta segmentation: (a) CTA scan axial slice after downsampling and smoothing; (b) enhanced image; (c) aortic arch segmentation

$E = \{(v_i, v_s), (v_i, v_t), (v_i, v_j)\}$ consist of: 1) edges (v_i, v_s) from voxels to the object terminal node; 2) edges (v_i, v_t) from voxels to the background terminal node, and; 3) edges (v_i, v_j) between adjacent voxels (4 or 8 neighbors for 2D images, 6 or 26 neighbors for 3D images). The cost of a cut $|C|$ that divides the graph into two parts, one associate with the source vertex (e.g. object class), and one associate with target vertex (e.g. background class) is defined as the sum of the weights of the cut edges $e \in C$. The segmentation is obtained by minimization of the cut cost:

$$\underset{e \in C}{\operatorname{argmin}} \sum_{e \in C} w_e \tag{1}$$

Edges weights (w_e) are assigned as follows. Edge weights $w(v_i, v_s)$ represent the posterior probability that voxel v_i is related to the vessels (object) or to the background (target):

$$w(v_i, v_s) = Pr(v_i \in I(O)) \cdot Pr(v_i \in S(O)) \tag{2}$$

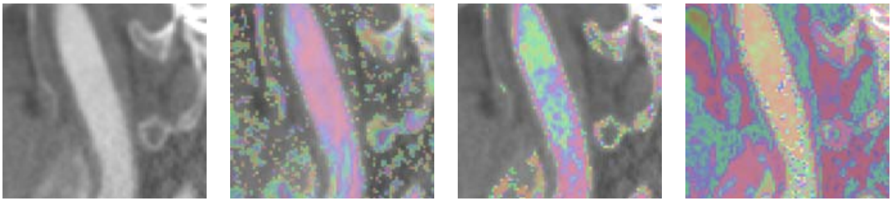
where $Pr(v_i \in I(O))$ is the probability that the voxel belongs to the object class based on voxel's intensity v_i and object's IPDF $I(O)$. The term $Pr(v_i \in S(O))$ is the geometrical shape information prior of the current voxel computed using Frangi's multi-scale Hessian based vesselness filter [3]:

$$M(\sigma) = \begin{cases} 0 & \lambda_2 > 0 \text{ or} \\ & \lambda_3 > 0, \\ \left(1 - \exp\left(-\frac{R_A^2}{(2a)^2}\right)\right)\left(\exp\left(-\frac{R_B^2}{(2b)^2}\right)\right)\left(1 - \exp\left(-\frac{S^2}{(2c)^2}\right)\right) & \text{otherwise} \end{cases} \tag{3}$$

where

$$R_A = \frac{|\lambda_2|}{|\lambda_3|} \qquad R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}} \qquad S = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$$

and σ is the scale at which the measure is computed. R_A differentiates between plate and line-like structures, R_B measures the deviation from blob-like structures, and S differentiates between foreground (vessel) and background (noise).



(a) Original image (b) Vesselness term (c) Object intensity (d) Background intensity

Fig. 2. Illustration of the new graph edge weights function: (a) original CTA sagittal slice; (b) shape-based probability map of each voxel to belong to the object class; (c-d) intensity based probability of each voxel to belong to the object (background) class. Red pixels denote high membership values.

Constants a, b and c are predefined weights determining the influence of R_A, R_B and S. The vesselness measure value is close to 1 for voxels with tube-like structures, and close to 0 otherwise. The vesselness measure is computed for each voxel and for each scale. Finally, for each voxel, the maximal value among the different scales is set to the vesselness measure for the voxel.

Edge weights $w(v_i, v_t)$ represent the probability of each voxel to belong to background class:

$$w(v_i, v_t) = Pr(v_i \in I(B)) \cdot Pr(v_i \notin S(O)) \tag{4}$$

where $Pr(v_i \in I(O))$ is the probability that the voxel belongs to the background class based on voxel's intensity v_i and background IPDF, implicitly computed as described by Freiman et al [16], and $Pr(v_i \notin S(O)) = 1 - Pr(v_i \in S(O))$.

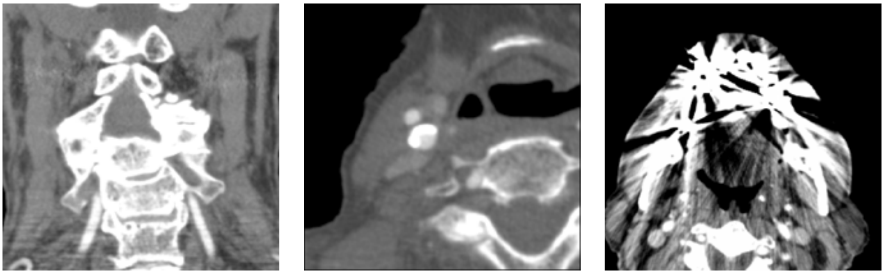
Edges weight $w(v_i, v_j)$ represent the magnitude of the local gradient between the adjacent voxels:

$$w(v_i, v_j) = \exp\left(-\frac{(I(v_i) - I(v_j))^2}{\sigma(Pr(v_i \in S(O)))}\right) \tag{5}$$

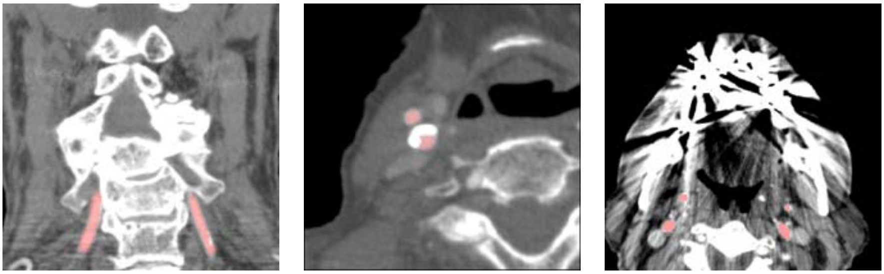
where σ is linearly depend on the geometrical shape information $Pr(v_i \in S(O))$. The adaptive weighting is designed to cope with small intensity differences inside the vessels while accounting for these differences along the vessel surface. Fig. 2 illustrates the different weights of each voxel on a sagittal slice.

The coupling of both intensity and geometrical shape information yield a robust and accurate segmentation method that correctly segments both healthy and pathological cases with mild or severe stenosis and calcifications without any parameter tuning. Fig. 3 illustrates the performance of our method on several representative challenging cases. Note that our method successfully separates between the vessels and surrounding structures, including bones, calcifications, and dental implants artifacts.

To cope with the memory requirements of the graph representation, the algorithm divides the scan volume into several block regions with small overlap based on the aorta ROI information. The graph min-cut is then computed for each block independently, and the results are merged together.



(a) Vertebral arteries (b) Severe stenosis (c) Implant artifacts



(d) (e) (f)

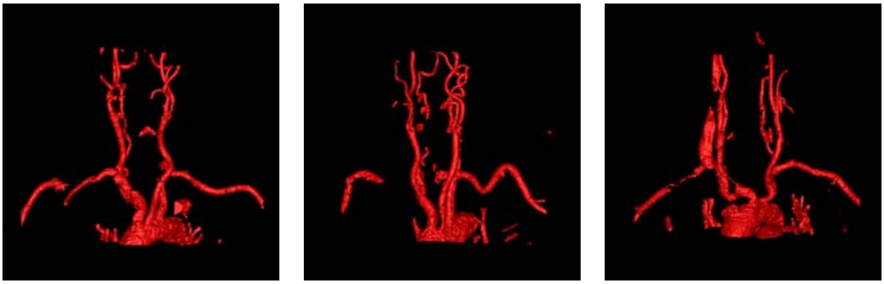
Fig. 3. Segmentation results on several representative challenging datasets: (a)-(c) original CTA images; (d)-(f) 2D view of the segmentation results superimposed on the original CTA slices

2.3 Nearly Automatic Graph-Based Vessel Adding/Removal

Inevitably, the global arteries segmentation step may produce disconnected vessel segments or may miss entire small vessels or include the internal jugular veins (Fig. 4).

To allow the treating physician to fix these flaws interactively, we have developed a spatially constrained graph min-cut interactive editing tool based on the graph min-cut segmentation framework. The nearly automatic tool requires the user to identify two points for each defective or missing vessel. It computes the segmentation of the vessel in two steps: 1) vessel trajectory estimation using a shortest-path algorithm; and, 2) optimal vessel surface computation using spatially constrained graph min-cut.

In the first step, it computes a path inside the vessel as the weighted shortest path between the graph nodes that contain the vessel endpoints. The edge weights compound local image and seed intensity information and vessel path geometric characteristics. In the second step, it constructs a Vessel Region of Interest (VROI) from the vessel path and the estimated vessel radius and computes the min-cut of the subgraph inside it. The min-cut identifies the graph nodes (voxels) at the boundary of the object (vessel) and background, thus producing the desired vessel boundary segmentation. Fig. 5 illustrates the method. We describe each step in detail next.



(a) CTA scan 1 (b) CTA scan 2 (c) CTA scan 3

Fig. 4. Spatial visualization of the surface vessel meshes constructed from three CTA scans after automatic segmentation

1. Weighted shortest path computation

The vessel path is computed by finding the weighted shortest path between the vessel seed points v_s and v_f that is inside the vessel. The shortest path is the sequence of edges connecting v_s to v_f for which the sum of its edge weights is minimum. We use Dijkstra's shortest-path algorithm whose worst-case complexity is $O(n^2)$, where n is the number of image voxels (much less when the vessel seed points are near).

To robustly and accurately find the vessel path, we use a hybrid edge weighting function with intensity and geometric information. The edge weight is the sum of: 1) local intensity difference; 2) seed deviation intensity difference; 3) path smoothness; and 4) path length penalty.

The local intensity difference term is the squared difference of the edge voxel intensity values:

$$(I(v_i) - I(v_j))^2 \tag{6}$$

Since its value is large at boundary crossings, it prevents the path from leaving the vessel region.

The seed deviation intensity difference term is the sum of the relative squared differences of the seeds and edge end voxel intensity values:

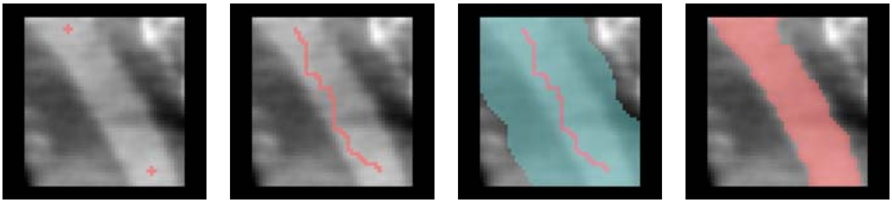
$$(I(v_j) - I(v_s))^2 + (I(v_j) - I(v_t))^2 \tag{7}$$

This term prevents the edges in the path from diverging too much from the intensity values of the user-selected seed points. Its effect is to prevent the path to move along locally smooth tissues with low edge weights rather than moving inside the noisy vessel.

The path smoothness term is the angle between the edge voxels gradient directions:

$$|\cos^{-1}(\nabla v_i \cdot \nabla v_j)| \tag{8}$$

where ∇ is the normalized voxel gradient. This term prevents edges with large gradient differences to be added to the path.



(a) original image (b) weighted shortest path (c) vessel region of interest (d) vessel boundary segmentation

Fig. 5. Illustration of the segmentation process on a clinical coronal CTA slice of the carotid artery: (a) detail of the original image showing the start and end vessel seed points (cross); (b) weighted shortest path between the two seeds; (c) vessel region of interest, and; (d) vessel boundary segmentation

The path length term is the Manhattan distance contribution of the edge. This term penalizes long paths.

2. Optimal vessel boundary segmentation

The vessel boundary segmentation step starts by defining a Vessel Region of Interest (VROI) from the vessel path and the estimated vessel radius. It then updates the graph edge weights according to the VROI and computes the graph min-cut to identify vessel boundary voxels [14].

The VROI is computed as follows. First, two VROI boundaries are computed from each seed point by taking the perpendicular of the path at the seed point and symmetrically extending it by twice the estimated vessel radius. Within the band defined by the two line segments, all nodes that are at a distance of twice the estimated vessel radius from the vessel path are included in the VROI.

Next, the vessel boundary segmentation is formulated as a min-cut problem over the corresponding graph. We use the same graph representation as in the previous step, with two additional terminal nodes v_s, v_t corresponding to object and background classes, respectively. In addition to the edges between voxels (v_i, v_j), we add two edges for each voxel node v_i: $(v_i, v_s), (v_i, v_t)$. The edges (v_i, v_s) are from voxels to the object terminal node and the edges (v_i, v_t) are from voxels to the background terminal node.

Edges weights are assigned as follows. Edge weights $w(v_i, v_s)$ represent the probability that voxel v_i is related to the vessels (object) or to the background:

$$w(v_i, v_s) = \exp\left(-\frac{I(v_i) - \mu_p}{\sigma_p}\right)^2 \cdot k \tag{9}$$

where μ_p is the intensity mean value along the computed path, σ_p is the standard deviation and k is a scalar multiplier that depends on the distance between the current voxel v_i and the computed path. It represents the probability that the voxel belongs to the object class based on voxel's intensity v_i and object's mean intensity value combined with spatial information that prefer voxels that are closer to the path computed before.

(a) CTA scan 1 (b) CTA scan 2 (c) CTA scan 3

Fig. 6. Spatial visualization of the surface vessel meshes constructed from three CTA scans after the nearly automatic refinement. 3D movies are available at: `http://www.cs.huji.ac.il/~freiman/vessels-cut`

Edge weights $w(v_i, v_t)$ represent the probability of each voxel to belong to background:

$$w(v_i, v_t) = 1 - w(v_i, v_s) \tag{10}$$

We use the inverse of the object weight $w(v_i, v_s)$, instead of explicit modeling of the background intensity.

Edges weight $w(v_i, v_j)$ represent the magnitude of the local gradient between the adjacent voxels:

$$w(v_i, v_j) = \exp\left(-\frac{I(v_i) - I(v_j)}{\sigma_p}\right)^2 \tag{11}$$

The optimal surface that separates the image into a vessel object and background is the min-cut at the resulting graph.

3 Experimental Results

We evaluate our segmentation method on 30 carotid arteries obtained from 15 clinical CTA scans acquired at two sites, each with its CT machine (Siemens and GE). For the first ten scans, patients were administered 100cc of non-iodinated contrast agent with a rapid injection aid at 3-4cc per sec. These CTAs have $512 \times 512 \times 750$ voxels of size $0.5 \times 0.5 \times 0.55mm^3$. For the remaining five scans, patients were administered 125cc of non-iodinated contrast agent with a rapid injection aid at 5cc per sec. These CTAs have $512 \times 512 \times 120$ voxels of size $0.35 \times 0.35 \times 1.25mm^3$.

The scans include varying levels of stenosis and dental implants streaking artifacts. Ground-truth segmentations of the arteries and vessels were obtained manually by a 3D segmentation expert and validated by an expert radiologist.

Each volume was divided into five regions: 1) aortic arch; 2) subclavian arteries; 3) common carotid arteries; 4) common carotid bifurcation into internal

Table 1. Comparison metrics and scores for 15 CTA scans. The first column is the segment type. The second column is the Absolute Volume Difference (AVD) from the ground-truth in %. The third column is the Average Symmetric Surface Distance (ASSD) in mm. The fourth is the Root Mean Square Symmetric Surface Distance (RMS SSD) in mm. The fifth column is the Maximal Symmetric Surface Distance (MSSD) in mm. The sixth column is the Volumetric Overlap Error (VOE) in %.

	AVD (%)		**ASSD (mm)**		**RMS (mm)**		**MSSD (mm)**		**VOE**	
	mean	std	mean	std	mean	std	mean	std	mean	std
Aortic Arch	14.95	9.55	1.03	0.56	1.58	0.79	16.48	24.54	19.06	8.67
Subclavian arteries	23.99	13.66	0.74	0.45	1.30	1.03	9.06	5.28	33.09	10.28
Common carotid	12.74	8.64	0.29	0.14	0.55	0.20	5.63	4.10	18.07	6.92
Carotid bifurcation	16.69	13.25	0.43	0.39	0.71	0.74	4.37	4.24	26.59	18.75
Internal & external carotids	24.65	19.40	0.70	0.58	1.47	1.29	12.04	8.82	40.74	24.65
Entire arteries	16.43	9.87	0.79	0.25	1.68	0.94	20.51	7.81	24.11	6.95

and external carotid arteries; and, 5) internal and external carotid arteries. The bifurcation region was defined as 25mm above and below the bifurcation. Segmentation results of each region were evaluated separately following the methodology in [17]. Both volumetric, and surface based measures were computed. Fig 6 shows three examples of the vessels segmentation using our method. Note the high quality of the proposed method segmentation results.

Table 1 summarizes the results of the 15 cases for individual segments and for all segments. Our method successfully segmented all datasets accurately without any failure from datasets with both mild and severe stenosis and with calcifications, without any parameter tuning, unlike the results reported on [8] for CTA images. The accuracy of our method around the carotid bifurcation is almost twice better than the results reported in [8]. Our overall accuracy is similar to the results reported in [8], while segmenting much more challenging regions, including the subclavian, aortic-arch, and internal and external carotid secondary bifurcations. The segmentation performance difference around the carotid bifurcation and of other parts of the carotid system stems from the varing noise and imaging artifacts caused by the intensity degradation near the shoulders bones and dental implants on the upper level of the internal and external carotid arteries.

The automatic segmentation mean computation time on a standard PC dual-core 2.4GHz with 3GB of memory machine was 470secs (std=212 secs) for the ten high resolution scans, and 154 secs (std=12 secs) for the five lower resolution scans. The nearly automatic refinement stage required on average 10 seeds and took the radiologist less than 2mins for each case with no technical support. This sets apart our method from existing semi-automatic and interactive methods which are more laborious and time consuming. These results indicate that our method is accurate, robust and easy to use for patient specific modeling of the carotid arteries system.

4 Conclusion

We have developed a nearly automatic graph-based method for patient-specific modeling of the aorta and the carotid, vertebral, and subclavian arteries for patient-specific simulations from CTA scans. The method automatically generates a vessel segmentation which is then refined by the treating physician with an easy-to-use tool to produce a mesh for simulation. Our results show that the proposed method is accurate, robust, easy to use, and can be integrated into existing simulators for patient-specific simulations. We are currently integrating the algorithm into the simulation platform and are extending it to other vascular structures and procedures, such as liver vessels and abdominal aortic aneurism.

Acknowledgment

This research is supported in part by MAGNETON grant 38652 from the Israeli Ministry of Trade and Industry.

References

1. Kirbas, C., Quek, F.: A review of vessel extraction techniques and algorithms. ACM Comput. Surv. 36(2), 81–121 (2004)
2. Kim, D., Park, J.: Connectivity-based local adaptive thresholding for carotid artery segmentation using MRA images. Image and Vis. Comp. 23(14), 1277–1287 (2005)
3. Frangi, A., Niessen, W., Vincken, K., Viergever, M.: Multiscale vessel enhancement filtering. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 130–137. Springer, Heidelberg (1998)
4. Lorigo, L., et al.: Curves: Curve evolution for vessel segmentation. Med. Image Anal. 5, 195–206 (2001)
5. Nain, D., Yezzi, A., Turk, G.: Vessel segmentation using a shape driven flow. In: Barillot, C., Haynor, D.R., Hellier, P. (eds.) MICCAI 2004. LNCS, vol. 3216, pp. 51–59. Springer, Heidelberg (2004)
6. Lekadir, K., Merrifield, R., Guang-Zhong, Y.: Outlier detection and handling for robust 3-D active shape models search. IEEE Trans. Med. Imaging 26(2), 212–222 (2007)
7. Schaap, M., et al.: Bayesian tracking of tubular structures and its application to carotid arteries in CTA. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 562–570. Springer, Heidelberg (2007)
8. Manniesing, R., Viergever, M., Niessen, W.: Vessel axis tracking using topology constrained surface evolution. IEEE Trans. Med. Imaging 26(3), 309–316 (2007)

9. Friman, O., Hindennach, M., Peitgen, H.O.: Template-based multiple hypotheses tracking of small vessels. In: Proc. of the 5th IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro. ISBI 2008, pp. 1047–1050 (2008)
10. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. Int. J. of Comp. Vision 70(2), 109–131 (2006)
11. Kang, L., Xiaodong, W., Chen, D., Sonka, M.: Optimal surface segmentation in volumetric images - a graph-theoretic approach. IEEE Trans. Patt. Anal. and Mach. Intell. 28(1), 119–134 (2006)
12. Slabaugh, G., Unal, G.: Graph cuts segmentation using an elliptical shape prior. In: Proc. of the 2005 IEEE Int. Conf. on Image Processing, ICIP 2005, vol. 2, pp. 1222–1225 (2005)
13. Sinop, A., Grady, L.: Accurate banded graph cut segmentation of thin structures using laplacian pyramids. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4191, pp. 896–903. Springer, Heidelberg (2006)
14. Ning, X., Narendra, A., Ravi, B.: Object segmentation using graph cuts based active contours. Comp. Vision and Image Understanding 107(3), 210–224 (2007)
15. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)
16. Freiman, M., Eliassaf, O., Taieb, Y., Joskowicz, L., Sosna, J.: A bayesian approach for liver analysis: Algorithm and validation study. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) MICCAI 2008, Part I. LNCS, vol. 5241, pp. 85–92. Springer, Heidelberg (2008)
17. Ginneken, B., Heimann, T., Styner, M.: 3D segmentation in the clinic: A grand challenge (2007), `http://www.sliver07.org`

Interactive Segmentation of Volumetric Medical Images for Collaborative Telemedicine

Jérôme Schmid, Niels Nijdam, Seunghyun Han, Jinman Kim,
and Nadia Magnenat-Thalmann

MIRALab, University of Geneva, Switzerland
{`schmid,nijdam,han,jinman.kim,thalmann`}`@miralab.unige.ch`

Abstract. Teleradiology, which enables distribution and sharing of digital medical images for collaborative diagnosis, has enjoyed rapid success due to the advances in telecommunication and multimedia technologies. However, the interactive collaboration mechanisms that control the editing among multiple users on the same object is often limited to simple 'locking' concept, where only one user edits an object, while all other users become only viewers. In this study, we introduce a new collaborative mechanism for networked teleradiology systems, demonstrated with the collaborative segmentation of 3D medical images. Our system enables concurrent annotation / region of interest (ROI) control by multiple users on the same image data, which can be useful for collaborative diagnosis as well as for teaching and training purposes. A preliminary prototype system is developed and the result suggests a promising collaborative mechanism for teleradiology applications.

1 Introduction

Telemedicine is a rapidly developing application system in the medical domain that harnesses the advances in telecommunication and multimedia technologies [1] and has lead to many healthcare benefits, e.g., telesurgery applications that use image-guided robotics and real-time consultation (teleconferencing) for distance-based surgery [2]; and telemonitoring systems for elderly prevention and care by using wireless sensory devices to communicate the status/condition of patients with physicians [3]. In particular, teleradiology - a telemedicine system that involves the electronic transmission of digital radiographic images (e.g., Magnetic resonance imaging (MRI), and X-Ray) from one geographical location to another, has revolutionized the healthcare practices by enabling efficient distribution and sharing of medical images. Teleradiology has evolved from video/audio/text based conferencing and shared 2D images running on imaging workstations with local area network (LAN) in year 2000 [4], to current state-of-the-art systems that are capable of streaming volumetric medical images in real-time via a wireless network [5]. Moreover, modern systems now support thin client/cloud computing [6] that processes (e.g., volume rendering) and stores all the medical image data in a client-server relationship, thus not requiring imaging workstations to access and view the images.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 13–24, 2009.

Despite the remarkable growth in teleradiology, little progress has been made in the mechanism that enables interactive collaboration, i.e., computer supported cooperative work (CSCW) technologies [7,8,9], among multiple users. In teleradiology applications, collaboration tools include the usual video/audio/text for communication among the users (teleconferencing), image navigation, and image editing, which we refer to hereon as 'collaborative editing'. Basic image editing involves appending information to the image without changing the image's state, i.e., textual annotations; drawing regions of interest (ROIs); and measurements using a ruler. For editing that results in a changed state of the image, such as brightness/contrast (B/C), and lookup table (LUT), a simple 'lock and release' has become the default control mechanism. In such a strict locking mechanism with collaborative applications (e.g., [5,10]), one user requests the object, applies changes on it, while all other users cannot make changes to it until the object is 'released'. Unfortunately, these studies do not allow concurrent editing, where every user's action can be executed timely and collaboratively, because only a single user who has ownership of the object can edit it.

In this study, we present our ongoing research in the real-time and interactive collaborative editing for telemedicine applications. The ability to collaboratively edit medical images in teleradiology can, e.g., introduce new approaches for diagnosis and image interpretation in situations where multiple experts can cooperate in the diagnosis [1]. Of course, such collaboration tools can also find much useful application for teaching and training. Unlike with conventional strict locking mechanism, in our approach, we relax the time constraint in collaborative editing in order to increase real-time and interactive performance of users. To achieve this, we introduce a two-tiered sharing mechanism with time constraints that is based on the iteration of the image editing process, i.e., multiple users may request edits within a predefined time range, where these requests are then managed (filtered) based on image processing constraints. The filtered request is executed and all the users are acknowledged of the processed requests. To demonstrate and measure our control mechanism in a teleradiology application, we have developed a collaborative system which enables semi-automated segmentation to be performed among multiple users in an interactive and collaborative manner. Image segmentation is the most common of image manipulation tools that are used to annotate and define regions of interest (ROI) for communication among the users. We illustrate the proposed concept with two case scenarios: (i) An experienced physician is over-looking and guiding the iterative segmentation of an image by one or more trainees ("teacher-students" scenario); and (ii) Two or more experts are simultaneously segmenting a same volumetric image data ("expert-expert" scenario). In the next section, the different components of the proposed framework will be presented along with the description of the two scenarios. Then, implementation details and experimental results will be detailed followed by discussions and conclusions.

2 Methods

Our collaborative editing framework consists of three main components: (i) Collaborative editing mechanism; (ii) Iterative 3D segmentation; and (iii) Communication substrate for collaboration; as shown in Fig. 1. Once a user initiates the collaboration session, i.e., loads the image and sends invitation to other users to join, multiple users can connect to the session and start the collaboration. When one of the users selects a segmentation command, an iteration of the segmentation is executed and the results are relayed to all the users in the session. The results are composed of the image data (slice) and segmentation data (segmentation overlay). At any point, any user can collaboratively segment the image by adjusting the segmentation parameters via interactive image annotation. These parameters are then inserted as new constraints in the segmentation and used in the next segmentation iteration.

For the collaborative segmentation, we have selected to process MRI data of the lower limbs, providing an interesting multi-user case scenario, where the segmentation is challenging due to the poor image quality and inhomogeneous intensities within its structure (imposed by hardware and protocol restrictions) [11] and thus can benefit from multiple users to concurrently aid in the segmentation process. Moreover, this data offer many different types of anatomical structures that needs to be segmented, e.g., bones and various types of muscles, thus providing multiple structures to be segmented concurrently. We will utilize our previously developed deformable model segmentation algorithm [11,12] for its

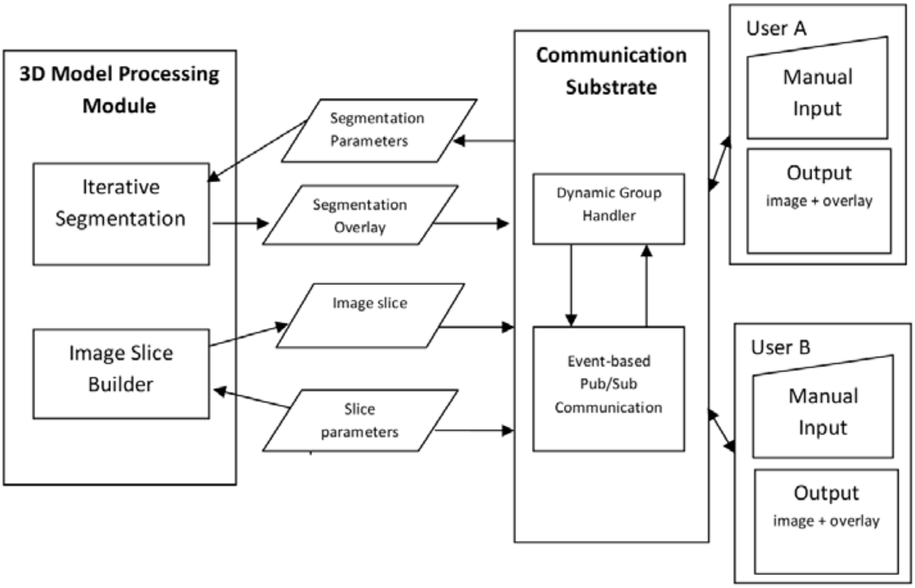


Fig. 1. The proposed collaborative telemedicine framework

inherent characteristics to evolve from rough to optimal segmentation results via an iterative and intuitive visual feedback. The advantage of visual feedback is the ability for the users to understand the segmentation process as it iterates to the final (optimized) result.

3 Collaborative Editing Mechanism

To enable real-time and interactive performance of users over conventional strict 'locking' based concurrency control mechanism that is commonly found in teleradiology applications, we relax the time constraints of the control mechanism by using a two-tiered sharing approach: semantics-tier (3D Image + 3D deformable models) and representation-tier (2D slice image + segmentation overlay) sharing, as shown in Fig. 2.

In this approach, semantics can instantiate more than one representation, which is shared by all users in a collaborative session. A polymorphic representation of the shared semantics is dynamically generated when a group of users request edits (e.g., new slice position on the stack, B/C change, segmentation constraints) within a predefined time range. The generated representation is replicated to each user of the group, not only to support direct manipulation on the shared representation, but also to continuously support users to collaborate even in the event of a transient network failure. Users of a group can temporally

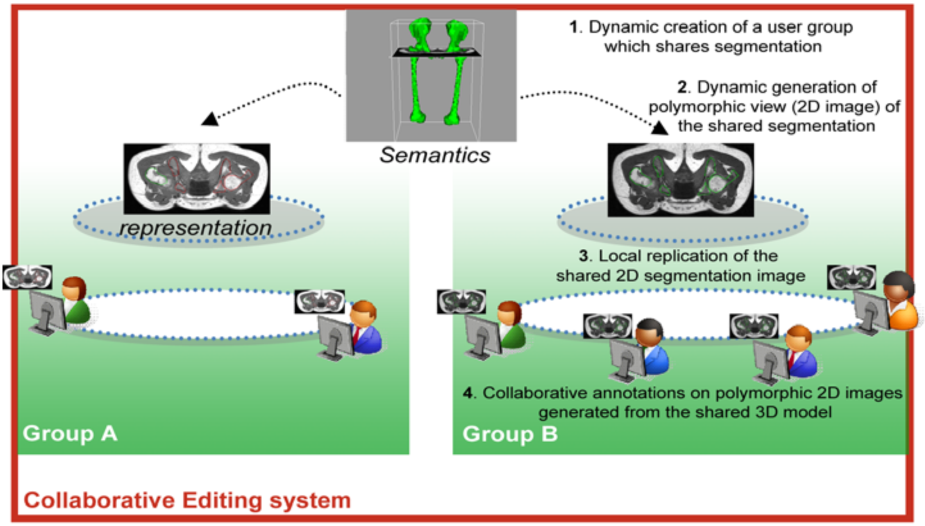


Fig. 2. Two-tiered semantics and representation sharing architecture for collaborative editing. In this example, two groups (i.e. two collaborative sessions) share the same semantics, while each member of a group shares a same polymorphic representation of the semantics.

manipulate, update and annotate on the representation within a pre-defined time frame. Updates within the time range on the shared image in a group are not directly affected to the semantics without agreement among all users in a group. Two types of event dissemination, filter and aggregator, are provided for perceptual consistency among users. A filter restricts the event dissemination scope and is used for distributing local constraint changes in a group. An aggregator broadens the scope and is used for distributing changes of global properties to groups which share the global properties.

For our collaborative editing, the image processing algorithm module must follow these constraints:

- Iterative: the algorithm must be iterative thereby enabling different users to amend, stop and resume the segmentation process. Most importantly, each iteration has to provide meaningful intermediate results that can be correctly interpreted by users. This provides efficient monitoring of the algorithm evolution.
- Multiple parameters: The algorithm must be able to accept multiple parameters (changes) from multiple clients. Parameters can be global (e.g., a weight coefficient) or local (e.g., local constraints, see examples in next section). Appropriate decision rules have to be designed to fuse the various changes or constraints.
- Locality of changes: Changes made by users should have an immediate local influence. These changes may ultimately affect the process in a global but progressive manner.

4 Multi-user Iterative Image Segmentation

We employ a segmentation algorithm based on [11,12,13,14]. This is an iterative deformable model segmentation algorithm where user-initialized meshes evolve under the influence of 'forces'. Each mesh vertex behaves like a lumped-mass particle that follows the Newtonian law of motion. A discrete first-order system of differential equations is then numerically and iteratively solved according to

$$\begin{aligned} P_{t+dt} - P_t &= V dt \ . \\ V_{t+dt} - V_t &= M^{-1} F(P, V) dt \ . \end{aligned} \tag{1}$$

where, t and dt denote time and time-step respectively, M is the particle mass, P is its position, V is its velocity and $F(P, V)$ is the total force vector. Depending on the choices made for the particle state, various numerical schemes are possible (e.g., forward Euler method takes forces and velocities at t, i.e. $V = Vt$ and $F(P, V) = F(Pt, Vt)$), resulting in a tradeoff between stability, accuracy and speed of convergence. Our method uses a fast implicit integration scheme [15].

External forces rely on image information (e.g., gradients, intensity distribution) to drive the model towards the desired anatomical boundaries. Conversely, internal forces only exploit the current geometrical configuration of the mesh to regulate its evolution. We exploit the use of prior knowledge to considerably improve the robustness and the quality of the segmentation. Principal Component Analysis (PCA) is often used to describe the modes of variations among shapes to segment and is used in many segmentation studies, i.e., bone [16,17,18,11] and other structures [19,20,12,14]. Taking advantage of this ability, our priors are defined using PCA of global shape variations, in addition to Markov Random Field (MRF) for local deformations [11].

Such segmentation framework allows the simultaneous segmentation of various structures of interest. To cope with models interpenetration, efficient collision detection and response are implemented. Coupled with a multi-resolution approach (from coarse to fine), a fast and interactive segmentation algorithm is derived.

Furthermore, users can easily manually correct the segmentation process by using what we refer to as 'constraint points' [14]. Three types of constraint points (See Fig. 3) are available:

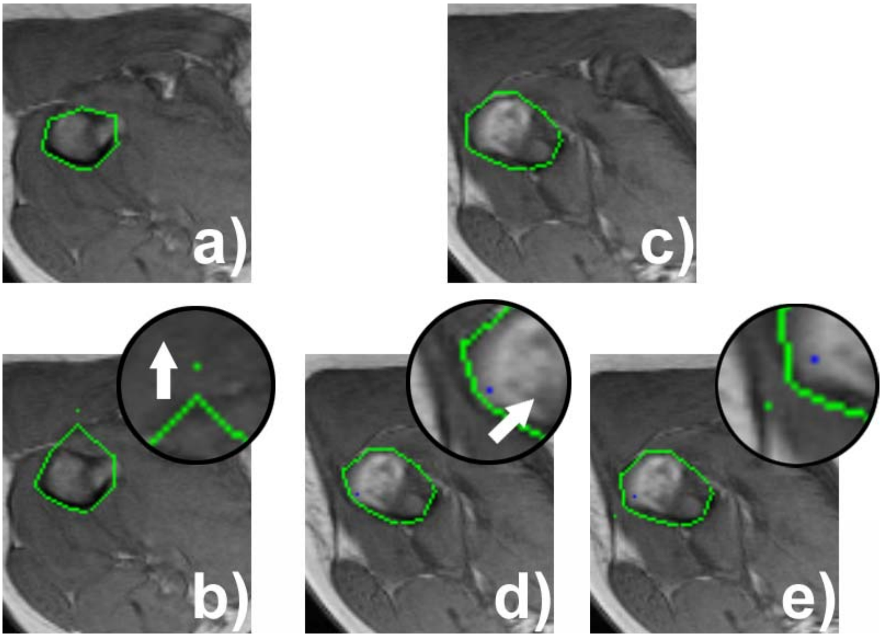


Fig. 3. Interactive segmentation by constraint points. a-b) Model changes after the insertion of an internal point (green dot); c-d) Model changes after the insertion of an external point (blue dot); White arrows indicate directions of change. e) Model subject to the influence of one internal and one external point. Here the weights of the associated forces are equal, thus the model stabilizes itself between both points (as in "expert-expert" scenario, see next section).

1. Internal points: the model evolves to include the points in its interior.
2. External points: the model evolves to exclude the points from its interior.
3. Frontier points: the model evolves to have the points lying exactly on its surface.

Constraint points are seamlessly translated as external forces in our framework, each force being weighted by a coefficient. Next section will explain how such weights can be tuned to account for the various collaborative segmentation scenarios. These external forces have a local influence while the modification of the force weight can globally affect the segmentation. This segmentation algorithm is thus a good candidate for our collaborative application as it fulfills the constraints defined in Sect. 3 and allows the concurrent segmentation of multiple structures. In this case, the models contour and the constraint points are overlaid in the slice and represent what was previously denoted in Sect. 3 as "segmentation overlay" (See also Fig. 1).

5 Collaborative Telemedicine Implementation

In order to illustrate the concepts of the proposed medical collaborative system, two scenarios were designed and evaluated:

1. "Teacher-students" scenario: a teacher shows to students the reading of radiological images and the concept of semi-automatic segmentation for ROI annotation on the images. In this scenario, both the teacher and the students each has access to their own computers which have installed our proposed collaborative editing medical viewer. A group session is started and all the students observe the segmentation evolution as the teacher initiates the segmentation and changes the viewing slice across the dataset. Then the teacher shows how to manually and locally correct the segmentation evolution by inserting various constraint points. Once this simple principle is explained, students are invited to interact with the segmentation process. The teacher are assigned greater 'weight' to the constraint points than the students, thus, enabling the teacher to override students' constraints.
2. "Expert-expert" scenario: two (or more) experts are segmenting the same dataset. Each of them can monitor and modify the segmentation in different parts of the dataset hence expediting the segmentation process. As with the "teacher-students" scenario, when experts work in the same slice, weights may be allocated based on the experience of the users, thus giving more priority on the segmentation constraints to the more experienced user.

The proposed system is implemented based on client-server architecture. The server owns the semantics and constitutes the 3D processing module (Fig. 1), while clients share and edit the polymorphic representations. It benefits not only to easily keep consistency and integrity of the shared semantics on a server, but also to allow users to exploit resource-limited devices to participate in a collaboration session. The processing server side has a 3D rendering view and

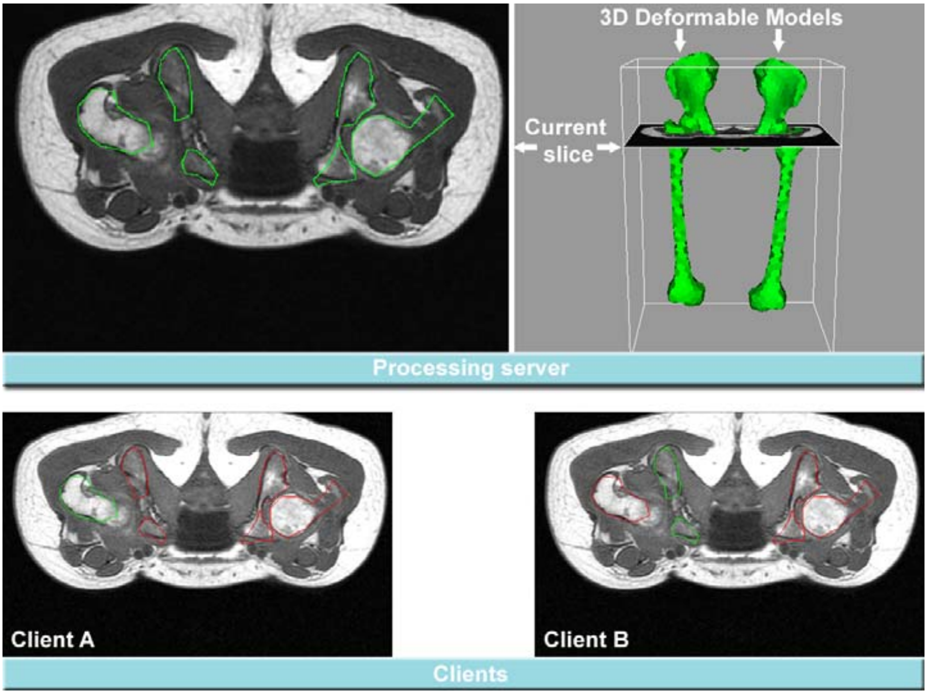


Fig. 4. Processing server-side (top) has both the 2D and 3D views whereas the client-side (bottom) has only the 2D view. As shown in this example, the server and client views are synchronized, and different clients (varying color annotations) are concurrently making changes to the segmentation. Each client can apply modification to a selected model (in green overlay on the client view). The segmentation constantly evolves on both the 3D and 2D views.

a 2D view. In this current prototype, the 3D rendering is only for illustrative purposes and not used in the collaborative editing, and only the 2D view was distributed to the users. The views can be synchronized (Fig. 4) which means that all the users observe the same slice. This particular mode is appropriate in the "teacher-students" scenario, in which the teacher first segments while the students only observe. A second mode consists in letting the users to view a slice independently to the other users in the same session. This option is essential for the second case scenario, in which experts can segment different parts of the same dataset. The 2D viewer consists of the standard image manipulations controls - LUT change, B/C, scaling, and annotation, in addition to be able to change image slices (moving up and down in the image stack).

We have developed a simple control mechanism for the segmentation manipulation. Each client is assigned to a unique annotation color and can either add or remove constraint points on the image (via mouse selection) as a segmentation parameter. Per iteration cycle, the server compiles all the parameters and executes the control based on time-stamp (first-come, first-serve) event management.

Multiple manipulations are possible, i.e., LUT change and segmentation change, in a single iteration cycle.

6 Experimental Results

Our system was primarily evaluated for interactive performance among multiple users during collaborative segmentation with two examples. The first example was based on the bone segmentation consisting of 4 different bone models (left-right femur and hip bone) on an image of $483 \times 358 \times 270$ dimensions. The second example was operated on an image of smaller size ($260 \times 511 \times 242$) but with a considerably higher number of models, representing the 21 muscles that were simultaneously segmented. For each example, the MRI image was loaded on the server and any number of users were able to join the collaborative group session. Raw images (no compression) were used in the transmission for testing purposes, and for performance measurements, only orthogonal image slices of the volume were used. We measured the frame per second (fps) at different levels as described below:

- Segmentation: the fps of an iterative step was measured
- Client update: the fps of the client update was measured. Client update was triggered every time a new slice with corresponding overlay (meshes contours, constraint points) was received and prepared for rendering. The time to receive an answer to a request from the server was hence also considered here. Please note that the rendering was performed in another thread, thus fps on the rendering was not representative of the client responsiveness and was not measured at this level.

The expert-expert scenario was considered in the first example, i.e., two users with their own computer concurrently segmented the same image. In the second case, 20 users joined in a collaborative session and the teacher-students scenario was simulated, i.e. most of the time one user explored and interacted with the segmentation process while the other only observed. An average update fps of a client was measured over 100 segmentation iterations.

Simulation results, shown in Table 1, reveal our system does not expose any significant degradation in system responsiveness and preserves interactive performance of users in varying conditions. Measured fps at the client level was satisfactory for collaborative editing ($\tilde{4}$ fps) and was mostly bound by the segmentation process. The complexity of the segmentation, in terms of memory consumption and execution speed was mostly related to the size of the segmented image and the number of models that were simultaneously segmented. These factors were expected to also have an impact on the system as the image data and the models contours (overlaid on the slice at the client side) need to be sent over the network. Additionally, the number of clients introduced additional load at the server side. Optimizations of the segmentation code and the use of more computational resources are expected to greatly improve the interactivity. The client was very light in terms of memory usage (< 1 MB) and CPU consumption ($< 1\%$).

Table 1. The fps with varying number of users, models and image size, averaged over 100 iterations (simulation hardware configuration (both clients and a server): Windows XP, Pentium 4 CPU at 3.40GHz, 2Gb of RAM connected to LAN)

	Example 1	Example 2
3D image size ($W \times H \times D$ pixels)	$483 \times 358 \times 270$	$260 \times 511 \times 242$
Number of models	4	21
Segmentation iteration fps	4.6	3.8
Client update	4.2	3.5
Number of clients	2	20

7 Discussions and Conclusions

In this paper, we presented a collaborative segmentation of musculoskeletal data among multiple users for two typical case scenarios: teacher-students and expert-expert collaboration. Our preliminary results demonstrate that collaborative editing can be useful in teleradiology context and has many potential clinical applications. Our system performance results indicate that, prior to code optimization, it can support large number of users for online collaborative editing of 3D volumetric medical images. In this study, only a simple medical image viewer was developed as a prototype. As such, only a small number of most common image manipulation and navigation tools were incorporated. Because the primary aim of this paper was in the feasibility of collaborative segmentation, we suggest that our current viewer is sufficient in demonstrating the capabilities of our collaborative editing mechanism. As our system is designed to be modular, the system can be integrated to other, more complete, medical image viewers, such as the popular open source image viewers built using ITK [21]. Our framework is also not restricted to the use of the deformable model segmentation algorithm [11] as presented in this study. This algorithm was chosen for its inherent characteristics to evolve from rough to optimal segmentation results via an iterative and intuitive visual feedback. The advantage of visual feedback is in the ability for the users to understand the segmentation process as it iterates to the final result. Our collaborative editing supports any image processing algorithms which adhere to the requirements given in Section 2.1. In future studies, we will experiment with other types of segmentation algorithms commonly found in medical applications. Similarly, the 3D MRI data was selected only as a case study and can be changed to any other medical imaging modality. Depending on the usage, the number of concurrent users can vary, for example, 2 to 3 users are expected for collaborative diagnosis, to 20+ users for collaborative teaching. Our networking component is designed to support large number of users and therefore the increase in users will not have a large effect in the overall system performance.

In our further research, we will derive new algorithms for concurrency control between the users, especially ‘commit agreement’ procedures; design and develop a collaborative editing medical image viewer; and conduct vigorous evaluation involving clinical usability and system performance. The mobility aspect will also be studied. We conducted some preliminary tests on a mobile device (UMPC) over a wireless network, and similar fps was observed. For such mobile devices, future work will mostly focus on optimizations from the server side (e.g., compression techniques, scalability studies) as the client application is already very light. Further user studies on our collaborative editing mechanism and its impact in clinical teleradiology will be also performed.

Acknowledgments. We would like to thank our partners: University Hospital of Geneva and University College London for providing us with the MRI datasets. This work is supported by the 3D Anatomical Human (MRTN-CT-2006-035763) and Intermedia (NoE-IST-2006-038419) projects funded by the European Union.

References

1. Wootton, R., Craig, J., Patterson, V.: Introduction to telemedicine, 2nd edn. The Royal Society of Medicine Press Ltd., London (2006)
2. Marescaux, J., Leroy, J., Gagner, M., Rubino, F., Mutter, D., Vix, M., Butner, S., Smith, M.: Transatlantic robot-assisted telesurgery. Nature 413, 379–380 (2001)
3. Rialle, V., Lamy, J., Noury, N., Bajolle, L.: Telemonitoring of patients at home: a software agent approach. Comput. Meth. Programs. Biomed. 72(3), 257–268 (2003)
4. Zhang, J., Stahl, J., Huang, H., Zhou, X., Lou, S., Song, K.: Real-time teleconsultation with high-resolution and large-volume medical images for collaborative healthcare. IEEE Trans. Inform. Tech. Biomed. 4(2), 178–185 (2000)
5. Park, S., Kim, W., Ihm, I.: Mobile collaborative medical display system. Comput. Meth. Programs Biomed. 89(3), 248–260 (2008)
6. Constantinescu, L., Kim, J., Chan, C., Feng, D.: Automatic mobile device synchronization and remote control system for high-performance medical applications. In: IEEE Proc. Engineering in Medicine and Biology Society (EMBS), pp. 2799–2802 (2007)
7. Delaney, D., Ward, T., McLoone, S.: On consistency and network latency in distributed interactive applications - a survey. Presence, Part I 15(4), 465–482 (2006)
8. Morillo, P., Orduna, J., Fernandez, M., Duato, J.: Improving the performance of distributed virtual environment systems. IEEE Trans. Parallel Distr. Syst. 16(7), 337–649 (2005)
9. Lee, D., Lim, M., Han, S., Lee, K.: Atlas: A scalable network framework for distributed virtual environments. Presence 16(2), 125–156 (2007)
10. Simmross-Wattenberg, F., Carranza-Herrezuelo, N., Palacios-Camarero, C., Casaseca-de-la Higuera, P., Martn-Fernández, M., Aja-Fernández, S., Ruiz-Alzola, J., Westin, C., Alberola-López, C.: Group-slicer: A collaborative extension of 3d-slicer. J. Biomed. Informat. 38(6), 431–442 (2005)
11. Schmid, J., Magnenat-Thalmann, N.: Mri bone segmentation using deformable models and shape priors. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) MICCAI 2008, Part I. LNCS, vol. 5241, pp. 119–126. Springer, Heidelberg (2008)

12. Schmid, J., Sandholm, A., Chung, F., Thalmann, D., Delingette, H., Magnenat-Thalmann, N.: Musculoskeletal simulation model generation from MRI datasets and motion capture data, pp. 3–19. Springer, Heidelberg (2009)
13. Gilles, B., Moccozet, L., Magnenat-Thalmann, N.: Anatomical modelling of the musculoskeletal system from mri. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4190, pp. 289–296. Springer, Heidelberg (2006)
14. Gilles, B.: Anatomical and Kinematical Modelling of the Musculoskeletal System from MRI. PhD thesis, Université de Genève (2007)
15. Volino, P., Magnenat-Thalmann, N.: Implementing fast cloth simulation with collision response. In: Computer Graphics International 2000, pp. 257–266. IEEE Publisher, Los Alamitos (2000)
16. Leventon, M.E., Grimson, W.E.L., Faugeras, O.: Statistical shape influence in geodesic active contours. In: IEEE Conf. Computer Vision Pattern Recognition (CVPR), vol. 1, pp. 316–323 (2000)
17. Fripp, J., Crozier, S., Warfield, S., Ourselin, S.: Automatic segmentation of the bone and extraction of the bone-cartilage interface form magnetic resonance images of the knee. Phys. Med. Biol. 52, 1617–1631 (2007)
18. Seim, H., Kainmueller, D., Heller, M., Lamecker, H., Zachow, S., Hege, H.C.: Automatic segmentation of the pelvic bones from ct data based on a statistical shape model. In: Botha, C., Kindlmann, G., Niessen, W., Preim, B. (eds.) Eurographics Workshop on Visual Computing for Biomedicine, pp. 93–100. Delft, Eurographics Association, The Netherlands (2008)
19. Heimann, T., Munzing, S., Meinzer, H., Wolf, I.: A shape-guided deformable model with evolutionary algorithm initialization for 3d soft tissue segmentation. In: Karssemeijer, N., Lelieveldt, B. (eds.) IPMI 2007. LNCS, vol. 4584, pp. 1–12. Springer, Heidelberg (2007)
20. Costa, M.J., Delingette, H., Novellas, S., Ayache, N.: Automatic segmentation of bladder and prostate using coupled 3d deformable models. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 252–260. Springer, Heidelberg (2007)
21. Ibanez, L., Schroeder, W., Ng, L., Cates, J., et al.: The ITK software guide. Kitware (2003)

Simultaneous Segmentation and Correspondence Establishment for Statistical Shape Models

Marius Erdt[1], Matthias Kirschner[2], and Stefan Wesarg[2]

[1] Fraunhofer Institute for Computer Graphics, Darmstadt, Germany,
Department of Cognitive Computing & Medical Imaging
`marius.erdt@igd.fhg.de`
[2] Technische Universität Darmstadt, Graphisch-Interaktive Systeme, Germany
`matthias.kirschner@gris.tu-darmstadt.de`,
`stefan.wesarg@gris.tu-darmstadt.de`

Abstract. Statistical Shape Models have been proven to be valuable tools for segmenting anatomical structures of arbitrary topology. Being based on the statistical description of representative shapes, an initial segmentation is required – preferably done by an expert. For this purpose, mostly manual segmentation methods followed by a mesh generation step are employed. A prerequisite for generating the training data based on these segmentations is the establishment of correspondences between all training meshes. While existing approaches decouple the expert segmentation from the correspondence establishment step, we propose in this work a segmentation approach that simultaneously establishes the landmark correspondences needed for the subsequent generation of shape models.

Our approach uses a reference segmentation given as a regular mesh. After an initial placement of this reference mesh, it is manually deformed in order to best match the boundaries of the considered anatomical structure. This deformation is coupled with a real time optimization that preserves point correspondences and thus ensures that a pair of landmark points in two different data sets represents the same anatomical feature.

We applied our new method to different anatomical structures: vertebra of the spinal chord, kidney, and cardiac left ventricle. In order to perform a visual evaluation of the degree of correspondence between different data sets, we have developed well adapted visualization methods. From our tests we conclude that the expected correspondences are established during the manual mesh deformation. Furthermore, our approach considerably speeds up the shape model generation, since there is no need for an independent correspondence establishment step. Finally, it allows the creation of shape models of arbitrary topology and removes potential error sources of landmark and correspondence optimization algorithms needed so far.

1 Introduction

State-of-the-art clinical diagnosis, therapy planning, and intra-operative navigation is based on three-dimensional image data. For this, anatomical structures

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 25–35, 2009.

have to be extracted from the images. This segmentation can be done employing a wide range of techniques: region-based approaches, contour-oriented algorithms, atlas-based approaches, and techniques incorporating prior knowledge about the typical shape of these structures and their most likely variation. The latter methods typically make use of statistical shape models (SSM) [1] that have been proven to be a valuable tool for segmenting organs like the liver [6], the heart [15], and the pelvic bones [12].

For deriving statistical information from a set of meshes created from a number of data sets, the so-called *correspondence problem* has to be solved. This means, that all meshes must contain the same number of nodes (called landmarks), and for two different meshes each node of data set A is required to have a corresponding node in data set B, both representing the same anatomical feature.

The first statistical shape models were constructed from landmarks which were manually placed on training images [1]. This manual annotation process is very time-consuming and regarded as intractable in 3D, due to the size and complexity of the shapes. Therefore, many researchers focused on the development of algorithms that establish correspondence automatically or semi-automatically. Recent overviews of automatic correspondence algorithms can be found in [3,5]. In typical semi-automatic methods, a sparse set of manual landmarks is defined which corresponds to predominant and unambiguously identifiable features. Additional landmarks are then automatically placed in between, either equally spaced according to the contour length in 2D [10] or by subdivision surfaces in 3D [13].

Beyond the tractability problem, the process of manual landmarking is often criticized for suffering from inter- and intra-observer variability. However, this does not imply that manually placed landmarks have lower quality than their automatically determined counterparts, as it remains unclear whether the objective functions applied in automatic methods really measure ‘true’ correspondence. Evaluation studies give an inconsistent picture: In the study of Styner et al. [13], a semi-automatic method based on manually defined landmarks and subdivision surfaces produced worse landmarks than optimization algorithms based on DetCov [8] and MDL [2]. It is unclear whether the manually defined landmarks or rather the automatic subdivision scheme accounts for the poor performance of the semi-automatic method in this evaluation. On the other hand, in a study from Ericsson and Karlsson [4], models learned from manually defined landmarks performed better than those constructed from automatically established landmarks. Though the latter study was restricted to 2D shapes and therefore excluded all complications that occur in 3D, the results indicate that manual landmarks may be better than their reputation.

(Semi-)automatic algorithms require that the training shapes are provided as surfaces. In practice, these surfaces are reconstructed from segmentations, which requires that either an expert delineates contours on training images manually or that automatic segmentation algorithms are used. Manual delineation is time-consuming and tedious, though by far easier than consistently placing landmarks.

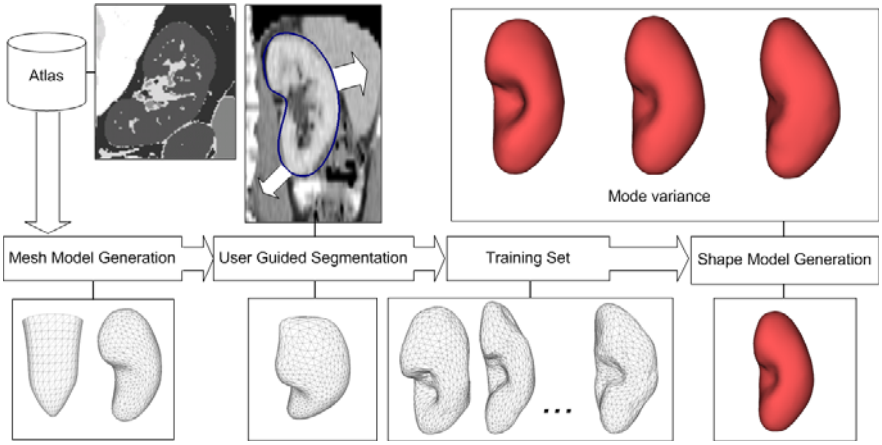


Fig. 1. Workflow of shape model generation. One triangulated reference mesh is created for every organ. The reference mesh is deformed by the user and simultaneously optimized to enforce correspondence of the deformed models with the reference shape. The resulting surfaces are then used as training data input for the shape model generation.

If the segmentations are generated automatically, the resulting training shapes are restricted by the accuracy of the applied segmentation algorithm.

While segmentation of training shapes and establishing correspondence are treated independently in case of (semi-)automatic correspondence algorithms, manual placement of landmarks integrates both aspects. Hence it is promising to develop tools which support the user during placement of landmarks, thereby making the process tractable.

In this work we present a method for simultaneous segmentation and point correspondence establishment for SSMs. In our approach a reference mesh is manually deformed and at the same time optimized in real time to preserve point correspondence. The resulting meshes can be directly used for building a shape model. In addition, SSMs of arbitrary topology can be easily constructed using our method, whereas many automatic, parameterization-based methods are either restricted to shapes of specific topology, like genus-0 surfaces [7], or require an artificial decomposition of the training shapes into several patches [12]. In the latter approach, correspondence is established independently on the patches and the results are merged afterwards, which introduces discontinuities at the cuts.

2 Methods

An overview of the shape model generation process is given in figure 1. The first step is the construction of a polygonal reference organ model that can be taken as a basis for all training data sets of the shape model. The next step is a user guided segmentation. Here, the mesh is three dimensionally deformed by the user

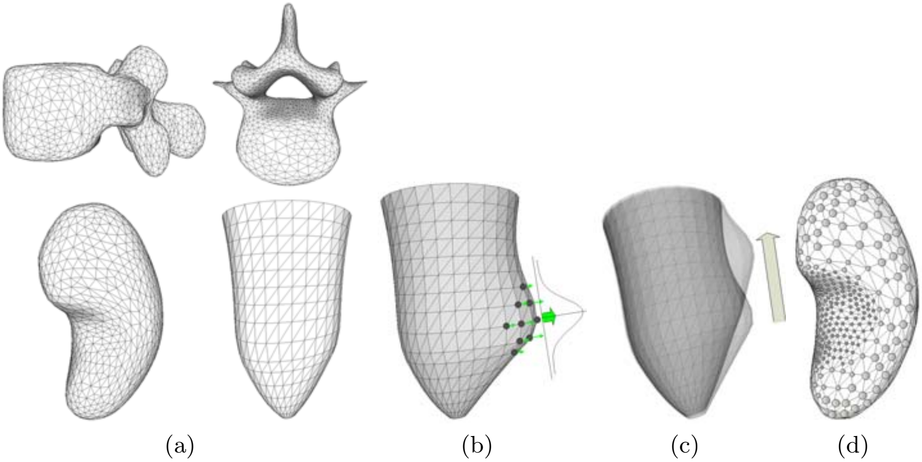


Fig. 2. (a) Generated mesh models for user guided adaptation: vertebra (top), kidney (lower left) and cardiac left ventricle (lower right). (b) 3D mesh deformation using a Gaussian weighting force. (c) Global shape preservation during lateral movement. (d) Local weights on the kidney shape: a soft area which maps the vessel and ureter connection region is embedded into a stiff capsule.

to match the organ boundaries in the data set. In order to ensure that points in the reference mesh and the deformed mesh denote the same feature points and therefore correspond to the same region, the mesh is globally optimized in each deformation step. The results are deformed training meshes with regularly distributed points that can be directly taken for shape model generation in the last step.

2.1 Reference Mesh Construction

In order to create a segmentation by using manual mesh deformation, a reference model is needed for every organ that can be adapted to the data set by the user. This model should be represented by a regularly distributed point cloud with an adequate number of points in order to ensure that all local organ features in the current data set can be mapped to the shape model.

The organ models are constructed based on a clinically validated reference segmentation — in our case taken from an organ atlas [14]. For simple organ shapes like the left ventricle we uniformly sample points along the volume's main axis in order to create a regular point grid. In the case of complex organ shapes, the binary segmentation is first resampled in order to remove the typical staircase artifacts resulting from image reconstruction in CT or MRI. Secondly, morphological closing and opening is applied to close holes inside the organ (e.g. vessels that are not classified as organ tissue). Afterwards, the *Marching Cubes Algorithm* generates a polygonal tessellation from the binary mask. Since the points of the resulting mesh are in most cases not regularly distributed, a

Laplacian smoothing is applied iteratively until all polygons are of comparable size. Fig. 2(a) shows the results of the model generation of vertebra, kidney and cardiac left ventricle.

In order to preserve an anatomic correct shape during deformation and to prevent the user from mapping smooth regions to areas of high curvature, we added local weight constraints to the kidney model (compare fig. 2(d)). Here, the kidney capsule is modeled with a 5 times higher rigidity than the area containing ureter and vessel connections. This leads to a more robust adaptation while being able to map the well known regions of high frequency boundaries at the same time.

2.2 Model Deformation

User Guided Adaptation. The first step of the manual segmentation process is the selection of an organ and the placement of the according model in the data set by the user. After placement, the model can be scaled and rotated in order to ease the adaptation and to roughly align important feature points of the model to the underlying data (e.g. inferior and superior renal capsule, which denote the lower and upper boundaries of the kidney capsule, respectively).

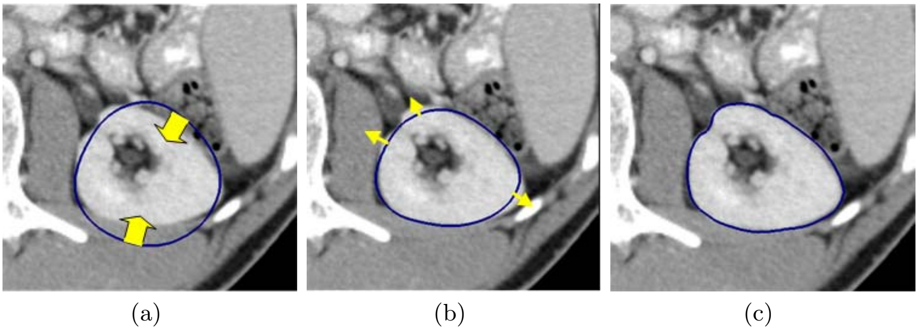
The subsequent step is a fine grained segmentation by directly deforming the mesh. This is done by pulling the boundaries of the mesh towards the real image boundaries in the three 2D standard views of medical imaging (axial, sagittal and coronal image planes). The user driven force at a given point is propagated to adjacent points using a 3D Gaussian $G_\sigma(x, y, z)$ (compare fig. 2(b)). The user can switch between three different scales of the standard deviation σ of G_σ. Initially, it is suitable to select the highest value of σ, which results in a non-local or stiff deformation of the mesh around the user movement vector (fig. 3(a)). In order to adapt to areas of high curvature, σ may be lowered which results in a softer deformation until only points in a vicinity are affected (fig. 3(b)). This procedure is repeated in a couple of different slices using the standard views and changing the value of σ until the mesh is properly fitted to the data (fig. 3(c)).

In order to keep areas of corresponding points matched during the deformation process, lateral movement does not change the global shape of the organ, i.e. a local deformation can be *pulled* back and forth on the surface (compare fig. 2(c)). This also prevents self-intersection or folding of the surface.

Optimization. Relying only on the described mesh deformation would lead to highly irregularly distributed point clouds after adaptation. Moreover, the quality of the resulting meshes would directly depend on the number of refinement steps, since no position correction is applied.

In order to use the user guided segmentation meshes as training input for the shape model generation, we optimize all point coordinates in real time such that the global shape of the reference mesh as well as the point distances are preserved.

Similar to [9], we define two energies E_{shape} and E_{force} to regularize the mesh deformation. E_{shape} denotes a shape preservation energy defined as



(a) (b) (c)

Fig. 3. Mesh adaptation in axial view. (a) Mesh after placement, scaling and rotation. The user forces the mesh into the real boundaries using stiff deformations (arrows). (b) Fine adaptation using soft mesh deformation. (c) End result of manual segmentation.

$$E_{\text{shape}} = \sum_{i \in P} w_i \sum_{j \in N(i)} \left((p_i - p_j) - (r_i - r_j)\right)^2 , \tag{1}$$

with p and r being the points of the deformed model and reference model respectively. The set of point indices P is the same for both r and p. $N(i)$ denotes the set of all direct neighbors of point p_i. w_i is a weight, that adds locally variant stiffness to the model as described in section 2.1. The described term ensures that the point distances in the deformed mesh remain similar to the distances in the reference mesh.

The energy E_{force} contains the user movement force towards the boundaries of the organ and is defined as

$$E_{\text{force}} = \sum_{i \in P} (p_i - s_i)^2 , \tag{2}$$

where s is the new point position resulting from the user force and weighted by G_σ *without* optimization.

The final point coordinates are now obtained minimizing

$$E = E_{\text{force}} + E_{\text{shape}} . \tag{3}$$

Equation (3) can be transformed into a linear system by setting the partial derivatives $\frac{\delta E}{\delta p_i}$ and $\frac{\delta E}{\delta p_j}$ to zero and bringing the resulting system to the form $A\boldsymbol{p} = \boldsymbol{b}$, with $\boldsymbol{p}$ containing the new point coordinates of the mesh. This overdetermined system can be solved in a least squares sense. Its solution is obtained by solving the normal equations $\left(A^{\mathrm{T}}A\right)\hat{\boldsymbol{p}} = A^{\mathrm{T}}\boldsymbol{b}$ with $\hat{\boldsymbol{p}}$ denoting the optimal solution vector for the new point coordinates. For performance reasons, x-,y- and z-coordinates are computed separately.

Since every point in the model mesh has a very limited number of direct neighbors (usually less than 7), most entries of A (and also $A^{\mathrm{T}}A$) will be zero. Therefore, sparse linear system solver [11] can compute the result very efficiently.

Also note that matrix $A^{\mathrm{T}}A$ and A^{T} can be precomputed and loaded together with the model, because vector $\boldsymbol{b}$ contains all non-constant expressions. The whole optimization for a mesh of 2500 points is 70 ms on a 2.4 GHz Quad Core PC. The deformation can therefore be performed in real time which allows a very smooth editing of the meshes. This is also very important for the use and acceptance of the application by the user since accuracy and speed should not be lower than a comparable conventional manual segmentation system.

2.3 Shape Model Generation

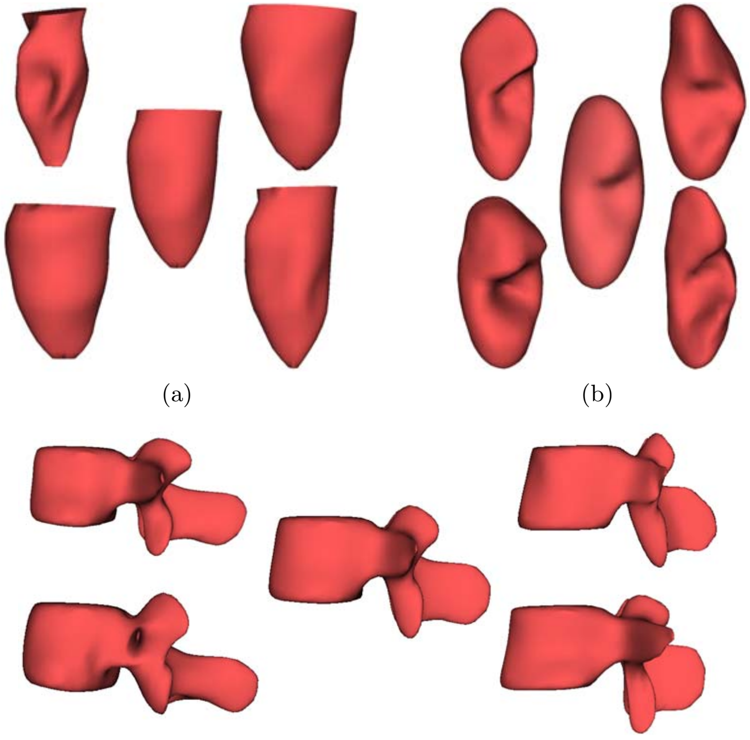
With the described procedure, a set of training meshes is created. These shapes are identified by their N corresponding landmarks. A single training shape can be represented as training vector $x_i \in \mathbb{R}^{3N}$, where x_i is simply the concatenation of the 3D coordinates of all landmarks. Given a set of M training vectors $\{x_1, \ldots, x_M\}$ representing the training shapes, a SSM is generated as follows [1]: Using the Procrustes method, the shapes are aligned into a common coordinate system, i.e. the representation becomes independent of translation, scaling and rotation. Principal component analysis (PCA) is used in order to capture the statistics of the aligned training shapes: Given the mean $\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$ and the covariance matrix $C = \frac{1}{M-1}\sum_{i=1}^{M}(x_i - \bar{x})(x_i - \bar{x})^T$, the eigenvectors $p_1, \ldots, p_{3N}$ of C and their corresponding eigenvalues $\lambda_1 \geq \ldots \geq \lambda_{3N}$ are obtained through diagonalization of C. Note that the eigenvectors are ordered by decreasing eigenvalues. The eigenvalue λ_i describe the variance along the principal axis p_i. In order to reduce the dimensionality of the model, axes with small variance are excluded from the model. We choose the smallest dimension t such that $\sum_{i=1}^{t} \lambda_i$ captures 95% of the variance of the training data set. The set of shapes modeled by the SSM are all shapes $\hat{x}$ in the form

$$\hat{x} = \bar{x} + Pb$$

where $P = (p_1|\ldots|p_t)$ is the matrix of retained eigenvectors and the shape parameters b_i are restricted to be in the interval $[-3\sqrt{\lambda_i}, 3\sqrt{\lambda_i}]$. As $\bar{x}$ and P are fixed, each shape $\hat{x}$ can be uniquely defined by the t-dimensional parameter vector $\boldsymbol{b}$.

3 Results

We tested our approach by building SSMs for three different anatomical structures of varying topology. For the vertebra and kidney, static volume data from CT was used (10 and 16 data sets, respectively) while the cardiac left ventricle is represented by 20 dynamic images provided as 3D cine MRI data reconstructed along the ventricle's main axis. An experienced user without medical background needs around 5 to 10 minutes for segmenting a kidney and a vertebra sufficiently precise and around 7 minutes for a ventricle. These times are comparable to other manual segmentation systems. Therefore, the proposed segmentation step is suitable to be used in practice.



(a) (b)

(c)

Fig. 4. Principal modes of variation for the SSM of the LV (a), kidney (b) and vertebra (c). The variation of the two largest eigenmodes between $-3\sqrt{\lambda_i}$ (left) and $3\sqrt{\lambda_i}$ (right) are shown together with the mean mesh (middle).

The deformed meshes were directly used for the shape model generation as described in section 2.3. Fig. 4 shows the principle modes of variation for all built SSMs. For each model, the mean (middle) is shown as well as the variation of the two largest eigenmodes between $-3\sqrt{\lambda_i}$ (left) and $3\sqrt{\lambda_i}$ (right). As it can be seen in fig. 4(a), the first mode maps the deformation of the ventricle as it is visible in the cine MRI data. A comparison between fig. 4(c) and fig. 5 (showing the set of training shapes of the vertebra) shows that the variation of the training shapes is well mapped to the first two modes of the generated shape model.

3.1 Correspondence Visualization

In order to support visual evaluation of correspondence, we developed a new visualization technique.

Coarse correspondence of regions is obtained by a color transfer function which maps landmark indices to hue in HSV space. Value is fixed to 1.0, and saturation to 0.4, which avoids loud colors. Our technique requires that adjacent landmarks have similar indices. If this is not the case, a renumbering of the landmarks indices is necessary. In our implementation, we assign the index 0 to an arbitrarily selected landmark, and then start a breadth first search from this landmark on the graph induced by the connectivity structure of the model. New landmark indices are determined by the order in which they are discovered during the breadth first search.

Visualization of correspondence on a fine-grained level of detail is achieved by means of glyphs. Since visualizing all landmarks would obstruct perception, we restrict to a subset S which is determined automatically. Our approach aims at finding landmarks corresponding to predominant features, which at the same time cover approximately uniformly the whole surface of the training meshes. We sort the landmarks by decreasing Gaussian curvature on a single, arbitrarily selected training mesh M. We then iterate through the sorted landmarks, and add a landmark to S if its euclidean distance on M to landmark that was previously added to S is larger than a specified threshold. All landmarks in S are visualized by spherical glyphs. In order to assign color to the glyphs, we use a similar transfer function as described above, but set the saturation of 1.0 to obtain a better accentuation.

Fig. 5 shows the correspondence visualization for the training sets of the vertebra. On the vertebra, prominent feature points are well visible and therefore mismatching points can be identified easily using our visualization approach. As it can be seen, correspondence in terms of mapping areas from one shape to the other is established well. In addition, the feature points mapping local high curvature are also placed on corresponding positions over the set of training shapes. In case of the kidney, the correspondence cannot fully reach the quality of the vertebra models. This is due to the fact that the kidney does not have many

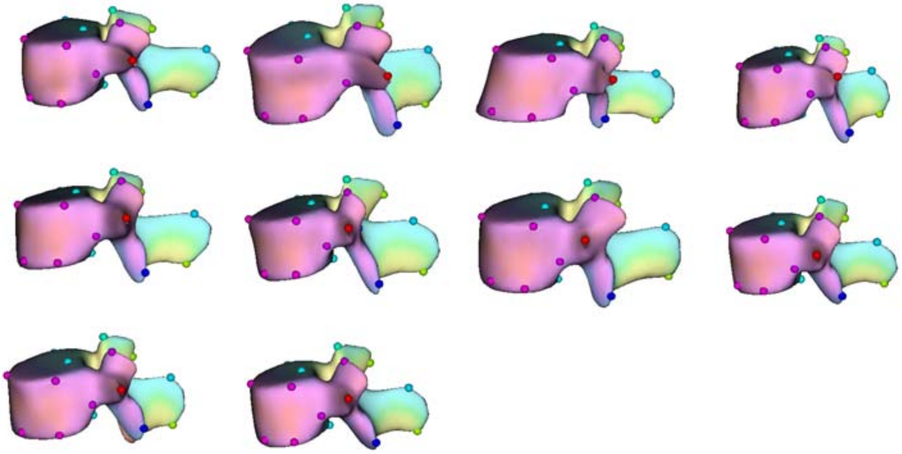


Fig. 5. Color coded visualization of point-to-point correspondence of the vertebra training set. Points evenly placed on features of locally high curvature are shown as glyphs.

uniquely identifiable points compared to the vertebra. The correspondence of the cardiac left ventricle models is comparable to the vertebra. Over all cine images, the position of the ventricle does not change. This eases the adaptation process and removes the error that may be introduced to a wrong initial orientation of the model.

4 Discussion

We propose a method for simultaneous segmentation and point correspondence establishment for statistical shape models using real time mesh adaptation and optimization. We applied our approach to different anatomical structures and built SSMs for the vertebra of the spinal chord, kidney, and cardiac left ventricle. In order to perform a visual evaluation of the degree of correspondence between different data sets, we have developed a color coded visualization based on local curvature. From our tests we conclude that the resulting correspondences established during the manual mesh deformation can be directly used to build SSMs even for structures of varying topology. Therefore, our approach considerably speeds up the shape model generation, since there is no need for an independent correspondence establishment step. In future work, we plan to further enhance the method to visually emphasize correspondence during adaptation in order to ensure that important feature points are mapped to the same landmark. In this context, our method can also be used as a refinement tool for the results of automatic shape model generation approaches. Furthermore, we plan to quantitatively evaluate the quality of the resulting SSMs using standard measures [7]. Other important aspects to be further evaluated will be the bias introduced by the choice of the template mesh and the dependency of landmark correspondence quality on the user guided model initialization.

References

1. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models - their training and application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
2. Davies, R.H., Twining, C.J., Cootes, T.F., Waterton, J.C., Taylor, C.J.: A minimum description length approach to statistical shape modeling. IEEE Transactions on Medical Imaging 21(5), 525–537 (2002)
3. Davies, R.H., Twining, C.J., Taylor, C.J.: Statistical Models of Shape - Optimization and Evaluation. Springer, Heidelberg (2008)
4. Ericsson, A., Karlsson, J.: Measures for benchmarking of automatic correspondence algorithms. J. Math. Imaging Vis. 28(3), 225–241 (2007)
5. Heimann, T., Meinzer, H.P.: Statistical shape models for 3d medical image segmentation: A review. Medical Image Analysis 13(4), 543–563 (2009)
6. Heimann, T., Meinzer, H.P., Wolf, I.: A statistical deformable model for the segmentation of liver ct volumes. In: Heimann, T., Styner, M., van Ginneken, B. (eds.) MICCAI 2007 Workshop Proceedings: 3D Segmentation in the Clinic: A Grand Challenge, pp. 161–166 (2007)

7. Heimann, T., Wolf, I., Meinzer, H.P.: Optimal landmark distributions for statistical shape model construction. Medical Imaging 2006: Image Processing 6144(1), 518–528 (2006)
8. Kotcheff, A.C.W., Taylor, C.J.: Automatic construction of eigenshape models by genetic algorithm. In: Duncan, J.S., Gindi, G. (eds.) IPMI 1997. LNCS, vol. 1230, pp. 1–14. Springer, Heidelberg (1997)
9. Lorenz, C., Berg, J.v.: A comprehensive shape model of the heart. Medical image analysis 10(4), 657–670 (2006)
10. Rueda, S., Gil, J.A., Pichery, R., Alcañiz, M.: Automatic segmentation of jaw tissues in CT using active appearance models and semi-automatic landmarking. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4190, pp. 167–174. Springer, Heidelberg (2006)
11. Schenk, O., Gärtner, K.: On fast factorization pivoting methods for sparse symmetric indefinite systems. Electronic Transactions on Numerical Analysis 23, 158–179 (2006)
12. Seim, H., Kainmueller, D., Heller, M., Lamecker, H., Zachow, S., Hege, H.C.: Automatic Segmentation of the Pelvic Bones from CT Data Based on a Statistical Shape Model. In: Botha, C., Kindlmann, G., Niessen, W., Preim, B. (eds.) Eurographics Workshop on Visual Computing for Biomedicine, pp. 93–100. Delft, Eurographics Association, The Netherlands (2008)
13. Styner, M., Rajamani, K., Nolte, L.P., Zsemlye, G., Székely, G., Taylor, C., Davies, R.: Evaluation of 3d correspondence methods for model building. In: Taylor, C.J., Noble, J.A. (eds.) IPMI 2003. LNCS, vol. 2732, pp. 63–75. Springer, Heidelberg (2003)
14. VOXEL-MAN Group. Voxel-man organ atlas. University Medical Center Hamburg-Eppendorf (2008)
15. Zhen, Y., Barbu, A., Georgescu, B., Scheuering, M., Comaniciu, D.: Fast automatic heart chamber segmentation from 3d ct data using marginal space learning and steerable features. In: 11th International Conference on Computer Vision (ICCV), pp. 1–8 (2007)

The Persistent Morse Complex Segmentation of a 3-Manifold[⋆]

Herbert Edelsbrunner[1,2,3] and John Harer[4,5]

[1] IST Austria (Institute of Science and Technology Austria)
[2] Dept. Comput. Sci., Duke Univ., Durham, North Carolina, USA
[3] Geomagic, Research Triangle Park, North Carolina, USA
[4] Dept. Math., Duke Univ., Durham, North Carolina, USA
[5] Program Comput. Bio. Bioinf., Duke Univ., Durham, North Carolina, USA

Abstract. We describe an algorithm for segmenting three-dimensional medical imaging data modeled as a continuous function on a 3-manifold. It is related to watershed algorithms developed in image processing but is closer to its mathematical roots, which are Morse theory and homological algebra. It allows for the implicit treatment of an underlying mesh, thus combining the structural integrity of its mathematical foundations with the computational efficiency of image processing.

1 Introduction

The extraction of shape information from density functions is an important topic in medical imaging. We formalize this problem as the construction and simplification of the Morse complex of the function. Before discussing this formalization and our algorithm, we discuss our motivation and briefly survey the extensive related literature.

Motivation and prior work. Medicine uses a variety of technologies to non-invasively obtain images of the anatomy of a subject. Examples are magnetic resonance imaging, X-ray and other types of computed tomography, ultrasound, and X-ray projectional radiography. They provide information about the anatomy expressed in terms of intensity or density. To make sense of the data, we need image segmentation algorithms that extract shapes by delineating their boundaries. This is a necessary first step for a multitude of medical tasks, including the quantification of tissue, the diagnosis, and the study of anatomic structure.

The segmentation of a three-dimensional image can be approached in a variety of ways. Today, many different algorithms are in use, each with its own strengths and weaknesses. We refer to Clarke *et al.* [4] and Pham, Xu and Prince [19] for surveys of segmentation algorithms in medical image analysis. In an attempt to organize the body of prior work, we distinguish between *direct* and *indirect* approaches to segmentation. In the direct approach, an algorithm generates shapes using local considerations based on the density function. An example is *thresholding*, in which a small number of real

---

[⋆] This research was partially supported by Geomagic, Inc., and by the Defense Advanced Research Projects Agency (DARPA) under grants HR0011-05-1-0007 and HR0011-05-1-0057.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 36–50, 2009.

thresholds is used to partition space into the preimages of intervals between contiguous thresholds [21]. Another example is *region growing*, in which seed points are grown into regions using local expansion criteria [10]. A third example works by *spitting and merging* regions [12]; see [14] for an extension to three-dimensional magnetic resonance images. The recently most successful of the indirect approaches are *deformable models* [16] or *level set methods* [23], in which boundary surfaces are computed through numerical solutions to differential equations. Shapes can also be 'learned' using *artificial neural networks* [11] or other adaptive structures developed within artificial intelligence. Sometimes, an *atlas* representing an idealized or expected segmentation is available, and the problem reduces to *registering* the atlas with the image [13].

The prior work most similar to ours is on the *watershed method*, which is another direct approach to segmentation. It is easiest explained by the analogous process of flooding a landscape in which water seeps in from every local minimum. As the water rises, it fills up a basin around each minimum and when two basins meet, it erects a dam that follows the line along which the water meets. Algorithms in image processing that construct the dams defining the segmentation can be found in [3,25]. The concept itself is however older within image processing [2,9]. A survey of watershed algorithms can be found in [20]. The algorithm of Vincent and Soilles [25] has been extended to three-dimensional magnetic resonance images by Sijbers *et al.* [22]. The authors of the latter paper cope with over-segmentation by applying a diffusion filter [26] eliminating many spurious local minima before running the watershed algorithm. They also merge regions in a postprocessing phase, using a statistical test to guide the process.

Results. The main contribution of this paper is a new segmentation algorithm. It takes the direct approach and falls into the class of watershed algorithms, constructing regions by following the gradient of the density function. The algorithm has a short list of distinguishing characteristics based on old and new work in differential and algebraic topology:

1. The basic segmentation of space is achieved by constructing what we call the Morse complex of the density function [7,24].
2. We coarsen the complex without altering its mathematical properties using homological persistence to guide the process [8].
3. We find a compromise between structural integrity, numerical accuracy, and computational efficiency in which none of the three measures makes substantial sacrifices.

The two-dimensional version of our algorithm has been implemented and used to decompose 2-manifolds describing mechanical shapes [6]. The success of that algorithm, even for rather complicated shapes, motivates us to extend the algorithm to three dimensions. Similar to its two-dimensional counterpart, our algorithm works for general 3-manifolds and not just for $\mathbb{R}^3$. It is most efficient when the data permits an implicit representation of the triangulation representing the 3-manifold.

Outline. Section 2 presents background from differential topology and from combinatorial topology. Section 3 explains the segmentation algorithm. Section 4 discusses the simplification of the segmentation. Section 5 presents plans for the future.

2 Background

In this section, we review the mathematical background needed for our algorithm. We discuss Morse functions, piecewise linear maps, and persistent homology.

Morse functions. Letting $f : \mathbb{M} \to \mathbb{R}$ be a smooth function on a 3-manifold, we get the derivative at a point $x \in \mathbb{M}$ as a linear map from the tangent space to the reals, $\mathrm{d}f_x : \mathrm{T}\mathbb{M}_x \to \mathbb{R}$. The point x is *critical* if $\mathrm{d}f_x$ is the zero map, otherwise, it is *regular*. A *critical value* is the image $f(x)$ of a critical point x. Assuming local coordinates at x, the *Hessian* is the matrix $H(x)$ of second-order partial derivatives. It distinguishes *non-degenerate* critical points, for which the Hessian is invertible, from *degenerate* ones, for which $\det H(x) = 0$. In this paper, we are interested in using critical point theory for smooth functions merely as a guiding intuition, so we focus on the generic case of *Morse functions* defined by the following two requirements:

I. All critical points are non-degenerate.
II. All critical points have unique function values.

The Morse Lemma asserts that a non-degenerate critical point x permits local coordinates such that $f(x_1, x_2, x_3) = f(x) \pm x_1^2 \pm x_2^2 \pm x_3^2$ in a neighborhood of x [15,17]. This implies that non-degenerate critical points are isolated and, if $\mathbb{M}$ is compact, any Morse function on $\mathbb{M}$ has only finitely many critical points. The number of minus signs in the above expression is independent of the chosen local coordinates and equals the number of negative eigenvalues of the Hessian. It is referred to as the *index* of x and distinguishes *minima* (index 0), 1-*saddles* (index 1), 2-*saddles* (index 2), and *maxima* (index 3) from each other. We get a local picture by drawing a small sphere around a critical (or regular) point x, and intersecting it with the level set to separate the region below x (the set of points y with $f(y) < f(x)$) from the region above x (points y with $f(y) > f(x)$). Non-degenerate critical points have the characteristic local pictures shown in Figure 1.

Critical points are often used to study the topology of the manifold, by sweeping $\mathbb{M}$ in the direction of increasing function value and monitoring the *sub-level set* defined as $\mathbb{M}_a = \{x \in \mathbb{M} \mid f(x) \leq a\}$, where $a \in \mathbb{R}$ is the value of the current level. As a increases, we observe topology changes whenever a passes a critical value. The changes depend on the indices of the corresponding critical points and accumulate to give $\chi = \#\mathrm{min} - \#\mathrm{sad}_1 + \#\mathrm{sad}_2 - \#\mathrm{max} = 0$ for the *Euler characteristic* of $\mathbb{M}$.

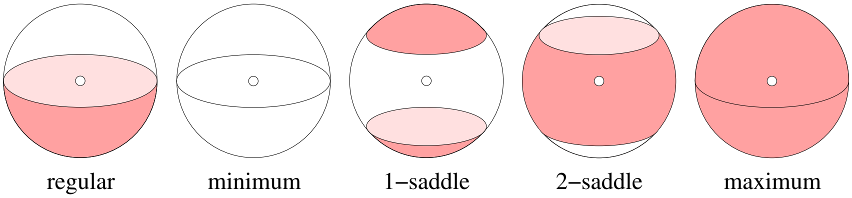


Fig. 1. Local pictures of regular and critical points of a Morse function on a 3-manifold. The level set separates the shaded region below x from the white region above x.

Morse complexes. Assuming a Riemannian metric on $\mathbb{M}$, we can use local coordinates to define the *gradient* as the vector $\nabla f(x)$ of first-order partial derivatives at x. If we follow the flow defined by the gradient, we trace out an *integral curve* $\gamma : \mathbb{R} \to \mathbb{M}$ that begins at a critical point, the *origin*, $\operatorname{org} \gamma = \lim_{t \to -\infty} \gamma(t)$, and ends at another critical point, the *destination*, $\operatorname{dest} \gamma = \lim_{t \to \infty} \gamma(t)$, without containing either. Two regular points either belong to the same or to two disjoint integral lines, which implies that the integral lines together with the critical points partition the manifold. We can now form a coarser partition by merging all integral lines with the same destination. Specifically, the *descending manifold* of a critical point x is

$$D(x) = \{x\} \cup \{y \in \operatorname{im} \gamma \mid \operatorname{dest} \gamma = x\}.$$

A minimum cannot be the destination of any integral line; its descending manifold is therefore just a point, the minimum itself. The descending manifold of a 1-saddle consists of two integral lines approaching the 1-saddle from opposite sides in the direction of the eigenvector with negative eigenvalue. Together with the 1-saddle, these two integral lines form an open 1-manifold whose endpoints are the origins of the two curves, which may or may not be the same. Similarly, the descending manifold of a 2-saddle is an open disk and that of a maximum is an open ball. Symmetrically, we define the *ascending manifold* of a critical point as the point x itself union all integral lines with origin at x. A Morse function is *Morse-Smale* if

III. the descending and ascending manifolds intersect transversally.

Being Morse-Smale is a generic property of smooth functions. Property III implies that the descending manifolds are the *cells* of a complex, in the sense that the cells partition $\mathbb{M}$ and the boundary of each cell is a union of lower-dimensional cells in the complex. We call this the *Morse complex* of the function.

Homology. The *p-th homology group* is an algebraic representation of the collection of p-dimensional holes of a topological space. Formally, it is the group of p-dimensional cycles divided by the group of p-dimensional boundaries, $\mathsf{H}_p = \mathsf{Z}_p / \mathsf{B}_p$ [18]. The *p-th Betti number* is the rank of this group, $\beta_p = \operatorname{rank} \mathsf{H}_p$, and can be interpreted as the number of p-dimensional holes. The only possibly non-zero Betti numbers a 3-manifold can have are for the dimensions $p = 0, 1, 2, 3$. The Euler-Poincaré formula asserts that

$$\chi = \beta_0 - \beta_1 + \beta_2 - \beta_3, \tag{1}$$

which should be compared with (3). As suggested by the two equations, we can make index-p critical points responsible for the p-dimensional holes counted by the p-th Betti number. It is convenient to use reduced homology groups [18, Chapter 1] to rationalize the classification of critical vertices. The ranks of these groups are the same as their non-reduced counterparts, except for $\tilde{\beta}_0 = \beta_0 - 1$ if the space is non-empty and $\tilde{\beta}_{-1} = 1$ if the space is empty. Table 1 gives the relation between the type of a vertex and the reduced Betti number of its lower link.

Piecewise linear manifolds. From now on, we consider the case in which the 3-manifold, $\mathbb{M}$, is the underlying space of a simplicial complex K and that $f : \mathbb{M} \to \mathbb{R}$ is obtained

Table 1. For simple vertex types, the dimension of the non-zero reduced Betti number of the lower link, if any, is one less than the index of the vertex

type	index	$\tilde{\beta}_{-1}$	$\tilde{\beta}_0$	$\tilde{\beta}_1$	$\tilde{\beta}_2$
regular		0	0	0	0
minimum	0	1	0	0	0
1-saddle	1	0	1	0	0
2-saddle	2	0	0	1	0
maximum	3	0	0	0	1

by piecewise linear extension of function values specified at the vertices of K. Thinking of f as the approximation of a smooth function, we introduce corresponding concepts. First we need some definitions. The *star* of a vertex u is the set of simplices that contain the vertex, $\mathrm{St}\, u = \{\sigma \in K \mid u \leq \sigma\}$. The *link* of u is the set of faces of simplices in the star that do not contain u, $\mathrm{Lk}\, u = \{\tau \leq \sigma \in \mathrm{St}\, u \mid \tau \not\in \mathrm{St}\, u\}$. Since K triangulates a 3-manifold, every vertex star has the topology of an open ball and every vertex link has the topology of a sphere. The *lower star* and *lower link* are the subcomplexes induced by the vertices below u:

$$\mathrm{St}_{-}u = \{\sigma \in \mathrm{St}\, u \mid u \neq v \leq \sigma \Rightarrow f(v) < f(u)\};$$
$$\mathrm{Lk}_{-}u = \{\tau \in \mathrm{Lk}\, u \mid v \leq \tau \Rightarrow f(v) < f(u)\}.$$

Assuming the vertices in K have pairwise different function values, the level set that passes through u intersects the link in a (not necessarily connected) curve that avoids all vertices and decomposes the link into two (not necessarily connected) regions, one below and the other above u. The region below u contains the lower link and has the same homotopy type, which can be proved by establishing a deformation retraction from the region to the lower link. Guided by the smooth case, we call u a *regular vertex*, a *minimum*, a 1-*saddle*, a 2-*saddle*, and a *maximum* if its lower link has the homotopy type of the corresponding shaded region in Figure 1. A vertex that does not fit this classification has $\tilde{\beta}_0 + \tilde{\beta}_1 \geq 2$, with $\tilde{\beta}_0 + 1$ the number of components in the lower link and $\tilde{\beta}_1 + 1$ the number of components in the symmetrically defined upper link. We thus think of it as the simultaneous embodiment of a $\tilde{\beta}_0$-fold 1-saddle and a $\tilde{\beta}_1$-fold 2-saddle.

Our assumption of vertices with pairwise different function values implies that every simplex of K belongs to a unique lower star. In other words, the lower stars partition K. Indexing the vertices in the order of increasing function value, $f(u_1) < f(u_2) < \ldots < f(u_m)$, we may form a nested sequence of subcomplexes,

$$\emptyset = K_0 \subset K_1 \subset K_2 \subset \ldots \subset K_m = K, \tag{2}$$

called a *filtration* of K, where $K_j = \bigcup_{i=1}^{j} \mathrm{St}_{-}u_i$. The complex K_j has the same homotopy type as the sub-level set $\mathbb{M}_a$ for every $f(u_j) \leq a < f(u_{j+1})$, which can be proved by establishing a deformation retraction from $\mathbb{M}_a$ to K_j. The difference between two contiguous complexes is the extra lower star, $K_{j+1} - K_j = \mathrm{St}_{-}u_{j+1}$. By definition, the Euler characteristic of K_j is the alternating sum of simplex numbers. We claim it is also the alternating sum of critical point numbers,

$$\begin{aligned}\chi &= \#\mathrm{vert} - \#\mathrm{edg} + \#\mathrm{tri} - \#\mathrm{tet} \\ &= \#\mathrm{min} - \#\mathrm{sad}_1 + \#\mathrm{sad}_2 - \#\mathrm{max},\end{aligned} \tag{3}$$

where a multiple critical point is counted $\tilde{\beta}_1 - \tilde{\beta}_0$ times. Indeed, the equation holds initially, for $j = 0$, and it is maintained when we add the lower star of u_{j+1} to K_j.

Persistence. We use homology groups to count the holes in the complexes K_i ordered by inclusion as in (2). Fixing the dimension to p, we write $F_i = \mathsf{H}_p(K_i)$ for the p-th homology group of the i-th complex. The chain of inclusions among the complexes implies a chain of maps among the homology groups induced by inclusion,

$$F_0 \to F_1 \to F_2 \to \ldots \to F_m.$$

Let $f_i^j : F_i \to F_j$ be the composition of $j - i$ of the maps along the chain. The corresponding *persistent homology group* is the image of this map, $F_i^j = \operatorname{im} f_i^j$, and the corresponding *persistent Betti number* is $\beta_i^j = \operatorname{rank} F_i^j$. We use these numbers to define the life-time of individual homology classes. The description of this idea is made complicated by multiple critical points, which may simultaneously create several classes and destroy several other classes. Defining

$$\mu_i^j = (\beta_i^{j-1} - \beta_i^j) - (\beta_{i-1}^{j-1} - \beta_{i-1}^j),$$

we say that μ_i^j homology classes *live* from $f(u_i)$ to $f(u_j)$, *being born* at time $f(u_i)$ and *dying* at time $f(u_j)$. Indeed, the first difference can be interpreted as the number of classes born at or before time $f(u_i)$ that die after time $f(u_{j-1})$ but at or before time $f(u_j)$. With a similar interpretation of the second difference, μ_i^j is the number of classes born some time in $(f(u_{i-1}), f(u_i)]$ that die some time in $(f(u_{j-1}), f(u_j)]$. Since there is no activity in the open interval between contiguous critical values, we can attribute the births and deaths as stated. The *persistence* of a class counted by μ_i^j is $f(u_j) - f(u_i)$. An algorithm that pairs up births with deaths and computes the persistence of homology classes can be found in [8]. Using the integers modulo 2 as the coefficient group for homology, it takes time $\mathrm{O}(N^3)$ in the worst case for a filtration of a complex with N simplices. The experimentally observed running time is much less than cubic and seems to be close to linear in N. The persistence of the maxima and the minima can be computed faster, in worst case time close to linear in N.

3 Algorithm

In this section, we present the segmentation algorithm. After explaining its global structure, we describe its actions in the simple cases. The general case can be interpreted as an accumulation of simple cases, as given in the appendix.

Problem specification. As input to the algorithm, we assume a triangulation K of the 3-manifold $\mathbb{M}$ and a function value for each vertex. Writing u_1 to u_m for the vertices, we assume pairwise different function values, $f(u_i) \neq f(u_j)$ whenever $i \neq j$, and we justify the assumption by the use of a simulated perturbation [5, Chapter 1.4]. As explained earlier, the function values can be extended linearly over all simplices to give a continuous PL function $f : \|K\| \to \mathbb{R}$. The output is a marking of some simplices and a labeling of all unmarked simplices. Specifically, the *marking* is a map $\mu : K \to \{0, 1\}$ and the subset of *unmarked* simplices is $K^\circ = \mu^{-1}(0)$. The *labeling* is a map $\lambda : K^\circ \to [1, m]$ whose image is the set of subscripts of the maxima. The interpretation of μ and λ is as follows. The subset of *marked* simplices, $\bar{K} = \mu^{-1}(1)$, corresponds to the union of descending 0-, 1-, and 2-manifolds in the smooth category. It decomposes the rest into components $K^\ell = \lambda^{-1}(\ell)$, which correspond to the (3-dimensional) cells of the Morse complex. The simplices of each component are labeled with the subscript of the generating maximum.

Sweep from the top. Before running the algorithm proper, we sort the vertices of the triangulation and re-name them such that $f(u_1) < f(u_2) < \ldots < f(u_m)$. The algorithm sweeps the triangulation from top to bottom, in the order of decreasing function value. Recall that K_i is the union of lower stars of the vertices u_1 to u_i. Because of the chosen ordering, the algorithm explores the complement, $L_i = K - K_i$, before it explores K_i. We call a vertex in K_i *stained* if it belongs to a marked simplex in L_i. The action taken at a vertex depends on whether it is regular or critical.

```
    for $i = m$ downto 1 do
        case $u_i$ is a maximum:
            label all simplices in the star of $u_i$ with $i$
        case $u_i$ is regular:
1           construct a spanning tree of the stained vertices in the lower link of $u_i$;
2           mark the simplices in the lower star incident to the spanning tree;
3           label the other simplices in the lower star by copying from neighbors
        case $u_i$ is a 1-saddle:
            do Steps 1 to 3 for each component of the lower link
        case $u_i$ is a 2-saddle:
            construct a spanning cactus whose cycle separates the poles;
            do Steps 2 and 3 substituting the cactus for the tree
        case $u_i$ a minimum:
            mark $u_i$
    endfor.
```

A *cactus* is a tree plus a single extra edge, which thus defines a unique cycle. To construct a cactus as needed in the 2-saddle case, we first construct a spanning tree of all vertices in the lower link, we add an edge to form a cycle, and we prune the cactus while retaining the cycle and the stained vertices.

Invariants. Before discussing the cases in more detail, we make general observations and formulate properties that hold throughout the algorithm. Note first that a simplex is marked or labeled when it is first encountered, which is when the algorithm processes its highest vertex. Since K_i consists of all simplices that have no vertex with subscript higher than i, the set of marked or labeled simplices right before processing u_i is its complement, $L_i = K - K_i$. While L_i is generally not closed, it is *relatively closed* in the sense that every face of a simplex in L_i is either in L_i or in K_i. This is of course trivially true as L_i and K_i partition the entire complex. A more interesting observation is that the set of marked simplices shares the same property, that is, every face of a marked simplex in L_i is either also marked or belongs to K_i. We now state this and two other claims as invariants of the algorithm.

INVARIANTS. The following relations hold for all subscripts of vertices, i, and all subscripts of maxima, ℓ.

(i) The set $\bar{L}_i = L_i \cap \bar{K}$ is relatively closed.
(ii) The set $L_i^\circ = L_i - \bar{L}_i = \bigcup_\ell L_i^\ell$ is open.
(iii) The set $L_i^\ell = L_i \cap K^\ell$ is empty or an open ball.

Since the algorithm works without back-tracking, $\bar{L}_i$ is the set of simplices marked right before processing u_i. To explain (ii), we call a set *open* if the inclusion of a simplex implies the inclusion of its entire star. It is then not difficult to see that (ii) holds. With a little bit more effort, we can also see that each label defines an open ball, as claimed in (iii). To see what this means, we note that L_i^ℓ is the subset of already processed simplices in the set $K^\ell = \lambda^{-1}(\ell)$, which represents a cell in the Morse complex. We call L_i^ℓ an *open ball* if the union of interiors of its simplices is homeomorphic to $\mathbb{R}^3$. To know that the cells are open balls is not important to understand the algorithm, but it is important to understand the result of the algorithm. For example, it implies that a marked vertex can be isolated only if the entire Morse complex consists of a single cell and the vertex forms the boundary of that cell. In this case, K triangulates the 3-sphere and the function f has only two critical vertices, a minimum and a maximum.

Regular case. Recall that $u = u_i$ is regular iff the level set defined by u, $f^{-1}(f(u))$, intersects the link in a single closed curve decomposing the link into two open disks. The lower link is a deformation retract of one of these disks and is therefore contractible. When we process u, we mark and label the simplices in its lower star. It is convenient to represent each such simplex by its face in the lower link, which has one lower dimension; see Figure 2. We now describe the three steps taken by the algorithm.

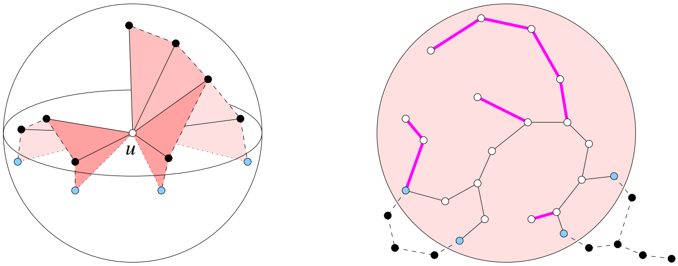


Fig. 2. Left: the star and link of the vertex $u = u_i$. The (solid) marked vertices lie in the upper hemi-sphere and the (shaded) stained vertices lie in the lower hemi-sphere. Right: the lower link of u drawn as a disk in the plane. The (solid black) spanning tree of the stained vertices is a subtree of the (solid black and shaded) spanning tree of the lower link.

`Step 1.` Construct a spanning tree within the lower link that touches all its vertices. Extract a subtree T_u that touches all stained vertices.

`Step 2.` Mark the triangles and edges in the lower star whose faces in the lower link are the edges and vertices of T_u.

The tree T_u decomposes the lower link into regions we call *bays*, each adjacent to the complement of the lower link along edges we call its *frontier*. A triangle on the other side of the frontier corresponds to a tetrahedron in the star that does not belong to the lower star and is therefore already labeled. The labels we find across the frontier of a bay are all the same.

`Step 3.` Traverse each bay of the lower link, from the frontier inwards, and copy the label found outside the frontier to the corresponding tetrahedra, triangles, and edges in the lower star.

An additional layer of detail about the three steps will be given in our description of the general case in the appendix, which subsumes the regular point as a special case.

2-saddle case. Recall that the vertex $u = u_i$ is a 2-saddle iff $f^{-1}(f(u))$ intersects the link in two closed curves decomposing it into two open disks and an open annulus, as illustrated in Figure 3, where the lower link is a deformation retract of the annulus. The three steps processing u are similar to those in the regular case, except that we begin with a cactus rather than a tree. We describe the first step in which this difference is most important.

`Step 1.` Construct a spanning tree within the lower link that touches all its vertices. Add a single edge creating a cycle that goes around the annulus, separating the two disks. Extract a cactus T_u^+ that touches all stained vertices.

By virtue of being a cactus, T_u^+ contains the cycle created by the edge added to the spanning tree. Steps 2 and 3 are the same as in the regular case.

Implementation and running time. The segmentation algorithm allows for a variety of short-cuts that make the implementation both easier and more efficient. The biggest

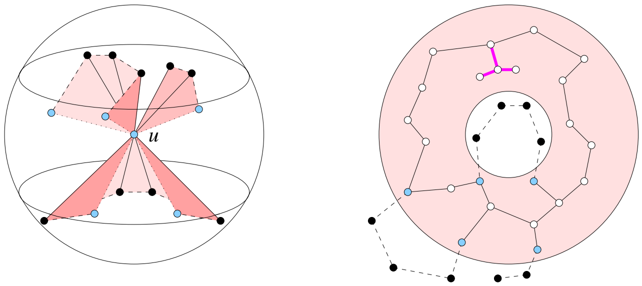


Fig. 3. Left: the star and link of the vertex $u = u_i$. The (solid) marked vertices lie on the polar caps and the (shaded) stained vertices lie on the equatorial belt that separates the caps. Right: the lower link of u drawn as an annulus in the plane. The (solid black) spanning cactus of the stained vertices is a sub-cactus of the (solid black and shaded) spanning cactus of the lower link.

gain can be expected from avoiding any explicit construction of the three-dimensional triangulation, the complex K. This is possible in the common case in which the data points form a regular grid in $\mathbb{R}^3$. However, the short-cuts do not affect the asymptotic running time of the algorithm, which we now address.

Let m, n and t be the number of vertices, edges, and triangles in K and write n_i and t_i for the number of edges and triangles in the link of the vertex u_i. The asymptotic running time of the algorithm depends on the implementation of `Step 1` and, in particular, on the construction of the spanning trees and spanning cacti. An attractive way to resolve ambiguities prefers short edges over long edges, thus constructing minimum spanning trees and cacti by always adding the shortest edge available. Every step of the construction for u_i takes time at most some constant times $n_i \log_2 n_i$, the time it takes to sort the edges by length. To take the sum over all vertices, we observe that the total number of triangles is $t = \frac{1}{2} \sum_i t_i$, because each triangle belong to two vertex links. Since every link is a triangulated sphere, we have $n_i = \frac{3}{2} t_i$ for each i and therefore $\sum_i n_i = 3t$. Furthermore, $t \leq n$ because every triangle has three edges and every edge belongs to at least three triangles. The total running time is therefore some constant times $\sum_i n_i \log_2 n_i \leq 3n \log_2 (\max_i n_i)$. For the common case in which each vertex has only a constant size star, the algorithm thus runs in time proportional to the data size.

4 Coarsening

Over-segmentation is a consequence of the algorithm creating a cell for each and every local maximum. In this section, we discuss a greedy approach to progressively coarsening the initial, fine segmentation until a desired level of resolution is reached.

Turning off maxima. A coarsening of the segmentation is achieved by turning off a subset of the maxima and treating them as regular points. The cell of an off maximum is merged into another cell, which is accomplished by turning off a 2-saddle whose descending 2-manifold would otherwise separate the cells. As a first approximation, we can understand the result by running the segmentation algorithm with some modifications in how it treats 2-saddles. Assuming an arbitrary but fixed assignment of the maxima to on and off, the algorithm takes different actions at a 2-saddle, u, depending on the status of the cells that meet at u. Let i and j be the labels of their tetrahedra (the subscripts of the maxima that generate the two cells).

`Case A.` If $i = j$ or $i \neq j$ and u_i and u_j are both on, then we treat u the same way as in the original algorithm, starting a locally separating descending 2-manifold at u.
`Case B.` If $i \neq j$ and u_j is off, then we merge the cell of u_j into that of u_i by changing the label of its simplices from j to i and suppressing the creation of a separating descending 2-manifold at u.

In summary, the algorithm determines the status of a 2-saddle (on in `Case A` and off in `Case B`) based on the status of the maxima generating the cells that meet at the 2-saddle. A drawback of this algorithm is that it fails to make the same distinction for 1-saddles and minima, and it is indeed impossible to make the appropriate assignment based on the local information available when we sweep the function from top to bottom.

Collapsing. We remedy this deficiency by making the assignment in an order that is independent of the sweep direction. To describe this, we observe that whenever we turn off a maximum, we also turn off a 2-saddle, effectively cancelling the two critical points. We implement this operation by a *collapse in the large*, which unmarks the descending 2-manifold of the 2-saddle and re-labels the cell of the maximum. A collapse in the large translates into a sequence of collapses in the small, each affecting a pair of simplices in the triangulation. For example, to remove the cell, we start the process by initializing an open ball to a tetrahedron in the star of the maximum. The ball is then expanded by collapsing its complement until the cell is exhausted. For the 2-manifold, we start the process by initializing an open disk to a marked triangle in the star of u that does not belong to any other descending 2-manifold. By construction, the triangle connecting u to the bridge completing the cycle in its link is such a triangle. This disk is expanded by collapsing its marked complement. The process either exhausts the entire descending 2-manifold or comes to a halt where the 2-manifold merges with other descending 2-manifolds.

In the process of collapsing 2-manifold/cell pairs, we may deteriorate the surrounding of a descending 1-manifold until it belongs to only one descending 2-manifold. At this moment, we collapse the 1-/2-manifold pair. Similarly, when a descending 0-manifold (a minimum) finds itself part of only one descending 1-manifold, we collapse the 0-/1-manifold pair. As before, each collapse in the large translates into a sequence of collapses in the small. The process continues until no further collapses can be applied. The result of this algorithm may depend on the sequence of collapses, for example if a 2-manifold can be collapsed from two different 1-manifolds in its boundary. We get a unique result by prioritizing the collapses. Most naturally, we would use the function value of the lower-index critical point of the defining pair as priority and perform collapses with higher priority before collapses with lower priority.

Greedily following persistence. The above procedure defines $2^{\#\max} - 1$ different segmentations, one for each non-empty subset of the maxima. To rationalize the choice, we suggest to turn off maxima greedily, following the persistence measure as introduced in [8]. This is the time-lag between when a cell is initialized and when it first meets another, older cell. We consider the union of these two cells as the continuation of the older cell. The measure of a maximum, v, is therefore $\pi(v) = f(v) - f(w)$, where w is the highest 2-saddle that separates the cell of v from the cell of a maximum higher than v. This measure can be computed in time proportional to $n\alpha(n)$ using a union-find data structure that keeps track of the components of L_i during a top to bottom sweep. Given the initial segmentation (with all maxima turned on) and the measure π at every maximum, the greedy algorithm turns off maxima in the sequence of increasing measure. After re-naming the maxima such that $\pi(v_1) < \pi(v_2) < \ldots < \pi(v_{\#\max})$ and letting $w_1, w_2, \ldots, w_{\#\max}$ be the corresponding sequence of 2-saddles, we can now state the greedy algorithm more formally.

```
for $i = 1$ to $\#\max - 1$ do
    collapse 2-manifold/cell pair of $w_i$ and $v_i$;
    repeat
        collapse 1-/2- and 0-/1-manifold pairs
    until no further collapses are possible
endfor.
```

Of course, we can halt the algorithm before arriving at a single cell, and the appropriate resolution may be chosen by the user or determined by other means.

5 Discussion

We have described a version of the watershed algorithm for segmenting a density function on a 3-manifold that uses persistence to counteract the over-segmentation, a common problem with the watershed paradigm. Many steps needed to turn this into a useful piece of software for medical data have been left unanswered. Should the algorithm be used on the raw data or will it be necessary to first apply a transformation of the data, such as a denoising, a smoothing, or a subsampling procedure? Three-dimensional medical images are usually given over a subset of $\mathbb{R}^3$, most often a cube sampled at an integer lattice of points (the centers of the voxels decomposing the cube). How severe are the artifact caused by the regular arrangement of the data points? Is it worth adding the voxel centers and use the Delaunay triangulation of the resulting body centered cubic lattice for the segmentation? It has better shaped tetrahedra than those triangulating the integer lattice and can still be treated implicitly, running the algorithm one vertex star at a time. The segmentation is defined by the two-dimensional triangulation consisting of the marked simplices. Will it be necessary to simplify this triangulation, eg. using edge-contractions running in parallel with the segmentation as described in [1]?

References

1. Attali, D., Cohen-Steiner, D., Edelsbrunner, H.: Extraction and simplification of iso-surfaces in tandem. In: Proc. 3rd Eurographics Sympos. Geom. Process., pp. 139–148 (2005)
2. Beucher, S.: Watersheds of functions and picture segmentation. In: Proc. IEEE Intl. Conf. Acoustic, Speech, Signal Process, pp. 1928–1931 (1982)
3. Beucher, S.: The watershed transformation applied to image segmentation. Scanning Microscopy Suppl. 6, 299–314 (1992)
4. Clarke, L.P., Velthuizen, R.P., Camacho, M.A., Heine, J.J., Vaidyanathan, M., Hall, L.O., Thatcher, R.W., Silbiger, M.L.: MRI segmentation: methods and applications. Magn. Reson. Imag. 13, 343–368 (1995)
5. Edelsbrunner, H.: Geometry and Topology for Mesh Generation. Cambridge Univ. Press, England (2001)
6. Edelsbrunner, H.: Surface tiling with differential topology. In: Proc. 3rd Eurographics Sympos. Geom. Process., pp. 9–11 (2005)
7. Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Morse-Smale complexes for piecewise linear 3-manifolds. In: Proc. 19th Ann. Sympos. Comput. Geom., pp. 361–370 (2003)
8. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discrete Comput. Geom. 28, 511–533 (2002)

9. Friedlander, F., Meyer, F.: A sequential algorithm for detecting watersheds on a gray level image. Acta Stereol. 6, 663–668 (1987)
10. Haralick, R.M., Shapiro, L.G.: Image segmentation techniques. Comput. Vis. Graph. Im. Proc. 29, 100–132 (1985)
11. Haykin, S.: Neural Networks: a Comprehensive Foundation. MacMillan College, New York (1994)
12. Horowitz, S.L., Pavlidis, T.: Picture segmentation by a tree traversal algorithm. J. Assoc. Comput. Mach. 23, 368 (1976)
13. Maintz, J.B.A., Viergever, M.A.: A survey of medical image registration. Med. Im. Anal. 2, 1–36 (1998)
14. Manousakas, I.N., Undrill, P.E., Cameron, G.G., Redpath, T.W.: Split-and-merge segmentation of magnetic resonance medical images: performance evaluation and extension to three dimensions. Comput. Biomed. Res. 31, 393–412 (1998)
15. Matsumoto, Y.: An Introduction to Morse Theory. Translated from Japanese by K. Hudson and M. Saito, Amer. Math. Soc. (2002)
16. McInerney, T., Terzopoulos, D.: Deformable models in medical image analysis: a survey. Med. Im. Anal. 1, 91–108 (1996)
17. Milnor, J.: Morse Theory. Princeton Univ. Press, New Jersey (1963)
18. Munkres, J.R.: Elements of Algebraic Topology. Addison-Wesley, Redwood City (1984)
19. Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation. Annu. Rev. Biomed. Engin., 315–337 (2000)
20. Roerdink, J., Meijster, A.: The watershed transform: definitions, algorithms, and parallelization strategies. Fundamenta Informaticae 41, 187–228 (2000)
21. Sahoo, P.K., Soltani, S., Wong, A.K.C.: A survey of thresholding techniques. Comput. Vision Graphics Im. Process. 41, 233–260 (1988)
22. Sijbers, J., Scheunders, P., Verhoye, M., Van der Linden, A., van Dyck, D., Raman, E.: Watershed-based segmentation of 3D MR data for volume quantization. Magn. Reson. Imag. 15, 679–688 (1997)
23. Suri, J.S., Liu, K., Singh, S., Laxminarayan, S.N., Zeng, X., Reden, L.: Shape recovery algorithms using level sets in 2-D/3-D medical imagery: a state-of-the-art review. IEEE Trans. Inform. Techn. Biomed. 6, 8–28 (2002)
24. Thom, R.: Sur une partition en cellules associée à une fonction sur une variété. Comptes Rendus l'Acad. Sci. 228, 973–975 (1949)
25. Vincent, L., Soille, P.: Watersheds of digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. 13, 583–593 (1991)
26. Yang, G.Z., Burger, P., Firmin, D.N., Underwood, S.R.: Structure adaptive anisotropic image filtering. Image Vision Comput. 14, 135–145 (1996)

Appendix

We now describe the actions taken in the general case, subsuming all types of vertices other than minima and maxima. Letting u be such a vertex, the level set $f^{-1}(f(u))$ intersects its link in $j + 1 \geq 1$ closed curves decomposing the link into $j + 2$ open regions. Using the reduced Betti numbers of the lower link, we count $\tilde{\beta}_0 + 1$ regions below u and $\tilde{\beta}_1 + 1$ regions above u so that $j = \tilde{\beta}_0 + \tilde{\beta}_1$. Each region below u retracts to a component of the lower link, and each region above u corresponds to a hole of a region below u or to the outside. Instead of one tree or cactus, we construct $\tilde{\beta}_0 + 1$ multi-cacti with a total of $\tilde{\beta}_1$ cycles. Without counting components and holes, the right number of cacti and cycles arises as a side-effect of the process that adapts the family of cacti to the topology of the lower link.

`Step 1.1.` Construct a forest T_u that contains a spanning tree within each component of the lower link, thus touching all vertices in the lower link.

`Step 1.2.` Thicken T_u to a subcomplex S_u of the lower link by iteratively adding triangles that already have two edges in S_u, until there are no more such triangles.

We may think of the thickening operation as a sequence of anti-collapses, each adding a triangle and its third edge. An anti-collapse does not change the homotopy type, which implies that the final complex S_u generated in `Step 1.2` consists of $\tilde{\beta}_0 + 1$ contractible components. We further expand these components until they cover the entire lower link. We do this by adding edges that change T_u from a family of trees to a family of cacti.

`Step 1.3.` Initialize $C_u = T_u$ and iterate until S_u is the lower link:
  `Step 1.3.1.` Add an edge of the lower link to S_u and also to C_u.
  `Step 1.3.2.` Thicken S_u by anti-collapsing triangles on both sides of the new edge until no further anti-collapses are possible.

We call the edges added in `Step 1.3.1` *bridges* because they form cycles, turning trees into cacti. Note that the thickening process guarantees that every new cycle goes around a new hole in the lower link. To continue, we shrink C_u, retaining only a minimal set of edges needed to touch all stained vertices and preserve the topological type. Call a vertex *free* if it belongs to exactly one edge in C_u.

`Step 1.4.` Extract a family of cacti by iterating the following steps:
  `Step 1.4.1.` Find an unstained free vertex in C_u.
  `Step 1.4.2.` Collapse the incident edge by removing the vertex and the edge from C_u.

`Steps 1.1` to `1.4` are illustrated in Figure 4. Given the input C_u for `Step 1.4`, the result of the sequence of collapses is unique, unless there is a component without cycles and without stained vertices. The edge collapses reduce this component to a single but arbitrary vertex in this component. The remaining two steps use the simplices in C_u to mark and the simplices in $S_u - C_u$ to label the simplices in the lower star.

`Step 2.` Mark the triangles and edges in the lower star whose faces in the lower link are the edges and vertices of C_u.

The family of cacti together with the marked edges connecting stained vertices to the outside decompose the subset of points in the link that are below u into bays. Each bay touches the curves in which $f^{-1}(f(u))$ intersects the link and thus neighbors the complementary subset of points above u. In the last step, we visit the bays and label corresponding simplices in the lower star. Call an edge *free* if it belongs to exactly one triangle in S_u.

`Step 3.` Shrink S_u back to C_u by iteratively collapsing free edges and vertices (the edge collapses undo the earlier anti-collapses and the vertex collapses repeat the collapses used to shrink the cacti):
  `Step 3.1.` Find a free edge or vertex, v in $S_u - C_u$. In either case, v belongs to a triangle in the link that does not belong to S_u. Let ℓ be the label of the corresponding tetrahedron in the star.

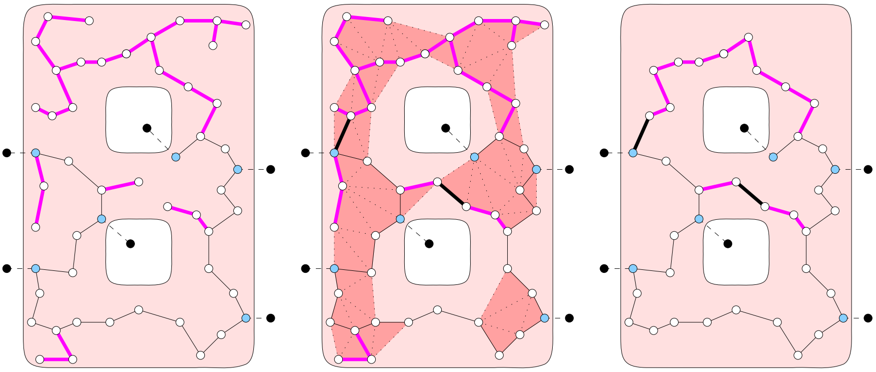


Fig. 4. Left: the level set of u intersects the link in three curves forming a double-annulus with two holes and the outside region. We see a spanning tree that touches every vertex in the double-annulus. Middle: the tree is thickened by incrementally adding the shaded triangles. When this process halts, we add a (thick black) bridge, turning the tree into a cactus, and repeat until the entire lower link is covered. Right: the cactus that spans all stained vertices extracted from the thickened complex. It contains both bridges and is a deformation retract of the double-annulus.

`Step 3.2.` Let τ be the unique triangle or edge in S_u that contains v. Label the simplices in the star that correspond to v and τ with ℓ.

`Step 3.3.` Remove v and τ from S_u.

As mentioned earlier, it is possible that a component of C_u consists of a single unstained vertex, v. In this case, `Step 3` labels all simplices in the corresponding component of the lower star, except for the edge connecting u to v, which it marks. A special case arises if there are no other components in C_u, which is the regular case. Then u itself is free and we collapse the edge by unmarking u and uv and labeling both consistent with the tetrahedra in their stars. As a side-effect, v is no longer stained.

Modelling Rod-Like Flexible Biological Tissues for Medical Training

Jian Chang, Junjun Pan, and Jian J. Zhang

National Centre for Computer Animation, The Media School, Bournemouth University,
Poole, BH12 5BB, U.K.
`{jchang,pjunjun,jzhang}@bournemouth.ac.uk`

Abstract. This paper outlines a framework for the modelling of slender rod-like biological tissue structures in both global and local scales. Volumetric discretization of a rod-like structure is expensive in computation and therefore is not ideal for applications where real-time performance is essential. In our approach, the Cosserat rod model is introduced to capture the global shape changes, which models the structure as a one-dimensional entity, while the local deformation is handled separately. In this way a good balance in accuracy and efficiency is achieved. These advantages make our method appropriate for the modelling of soft tissues for medical training applications.

Keywords: Medical training, surgery simulation, deformation, Cosserat rod, computer animation.

1 Introduction

Virtual simulation can replicate the teledisplay and instrumentation of laparoscopic surgery and is attracting increasing attention from both the research community and the medical practitioners. As a training methodology it can reduce the costs and risks and has been adopted widely by the surgical community. A training system based on virtual simulation provides the trainees with flexible access to training sessions, whereas traditionally the training sessions are limited by the availability of patients and training posts. The most essential element of such a system is to model the behaviour of the tissues in a proper manner in order to produce convincing visual and haptic feedback to the user. Slender shapes represent a group of organs and tissues in the human body. Efficient simulation of such structures provides a useful framework which lays the foundation for virtual surgery applications involving such objects. Our work discussed in this paper focuses on modelling slender biological tissues which can be presented with a simple rod model.

In the human body, many tissues are of long, thin, rod-like shapes, and we can take the advantage of modelling them as one-dimensional entities to efficiently simulate their behaviours. Examples of such tissues include blood vessels, muscle fibres, small intestines, rectum, ligaments, tendons, and hair. Though the one dimensional rod-like object has been well studied in engineering, such as the analysis of the beam frames to improve the safety of structures in civil engineering, the calculation of the

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 51–61, 2009.

deformation for the above biological tissues remains a challenging task. For instance, the tissue may exhibit large deflexion that invalidates the algorithms which were derived based on the assumption of small deformation. The finite element method and mass spring method have been widely adopted in surgery simulation. They can provide accurate results, but they are generally inefficient in modelling the rod-like object if using volumetric discretization. Other classic methods based on the centreline of a rod, such as the Cosserat rod model [1], are designed to capture the global deformation, which provide little information about the local deformation. In order to capture the local deformation, the finite element method and the mass spring systems are often the only options which unfortunately are computationally expensive for real-time simulation.

In this paper, we propose to model the rod-like biological structure with two separate layers: global deformation and local deformation. To our knowledge, Barr [2] was the first who developed the idea in computer graphics, where global and local shape changes were dealt with separately. In our method, the Cosserat rod model is adopted to model the global deformation along the centre line, while the local deformation is characterised by the deformation of a given cross-section of the tissue. Large global deformation, such as twisting and bending, is well handled with the Cosserat rod model. The local stretch of the surface is then approximated with our local deformation module.

In general, the biological tissues are not isolated in the body. They are often attached to each other, which restricts their movement, i.e. they are usually subject to physical constraints. For instance, Raghupathi et al. [3] proposed to model the mesentery which connects the blood vessel and the intestine with a mass spring sheet. In this paper, the influence of such constraints can be integrated into the Cosserat model with an additional elastic energy item.

Our contributions in this paper are as follows: (1) we have developed a general framework of global and local deformations of long flexible rod-like biological tissues; (2) we have introduced a quasi-static Cosserat rod model for fast simulation of the global deformation of a rod with additional constraints; (3) we have established a local deformation model which performs in an efficient manner.

2 Related Work

There exists a large body of research in the area of soft tissue simulation [4, 5, 6, 7, 8]. Due to the complexity of soft biological tissue, it is extremely challenging to simulate their natural behaviour accurately, especially when subject to interventions with surgical instruments.

Most of the current approaches are based on either the mass spring system [9, 10] or the finite element method [4, 11, 12]. The mass spring system uses discretized mass points in the space to represent the volume or surface. Such mass points are connected to each other with springs to capture the dynamics. The finite element method discretizes the object into many small elements, e.g. tetrahedrons. A problem is solved by finding the numerical solution of governing partial equations in an element-wise manner. The finite element method provides accurate results but it is computationally expensive.

Thin slender rod-like biological tissues have been studied in a wide range of disciplines. In computer animation, hair modelling has attracted a great deal of attention [13, 14], which can be used in character rigging. In applications of virtual surgery training, some researchers reported the development on modelling intestine with the mass spring system [3, 15]. In [16] a centreline model was introduced for long thin rod-like structures. By filtering out local changes of the cross-section shape, it greatly reduced the degrees of freedom of the problem and hence enabled the problem to be solved efficiently.

Cosserat rod can capture the large bending and twisting deformation of one dimensional rod-like object, which records the motion of points on the centreline and the rotation of a local coordinate frame attached to each individual point. It was used to model the deformation of cable like instrument in engineering [17]. Linn et al. [18] outlined a method to model rod deformation with Kirchhoff model (a special case of Cosserat model) for virtual reality applications. Several recent works [19, 20, 21, 22] in computer animation provided good reference of Cosserat rod model in detail. Despite its efficiency the Cosserat rod model's inability to handle local deformations often renders it unsuitable for high fidelity surgery training applications, as local deformation can provide useful feedback to the surgeons. In this paper, a quasi-static model of Cosserat rod is introduced to construct global deformations. Different from the previous approaches [20, 21], our quasi-static model drops the dynamic items and simplifies the solving process to allow fast simulation. In order to overcome the drawback of lack of local deformations, in this paper we model the deformation of local regions separately, whose superposition with the global deformation provides convincing looking of the soft tissues.

3 Cosserat Rod Mechanics

The Cosserat rod model, which describes the dynamics of a long thin rod-like object, is a well established model in both traditional mechanics [1] and computer graphics [20, 21].

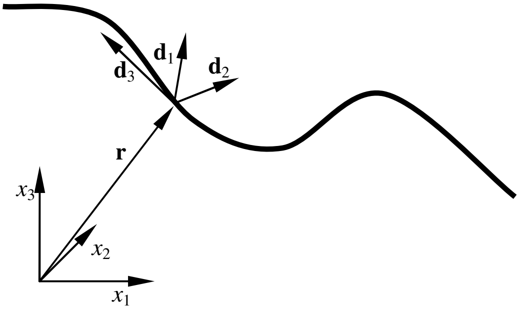


Fig. 1. Cosserat rod model

With the Cosserat rod model, a rod is represented as a space curve and its cross section is shrunk to one single point on the curve. Such curve is either selected as the neutral axis or as centroid axis (we will call this the curve centreline in the context). A local coordinate frame is attached to each point on the curve which indicates the position and orientation of the cross section.

As shown in Fig. 1, given $\mathbf{r}(s)$ as a point on the selected curve, where s is the curve length ranging from 0 to L, we denote the coordinate frame by three unit vectors $\mathbf{d}_1$, $\mathbf{d}_2$and $\mathbf{d}_3$. These three vectors are also named as directors, which are orthogonal to each other. In particular, $\mathbf{d}_3$ is defined to be along the tangent of the beam centreline, where the other two directors, $\mathbf{d}_1$ and $\mathbf{d}_2$, represent two orthogonal material lines in the cross-section perpendicular to $\mathbf{d}_3$.

$$\mathbf{d}_3 \times \frac{\partial \mathbf{r}}{\partial s} = 0; \qquad \mathbf{d}_1 \times \mathbf{d}_2 = \mathbf{d}_3 \tag{1}$$

With the definition of Darboux vector $\boldsymbol{\omega}$ as

$$\frac{\partial \mathbf{d}_i}{\partial s} = \boldsymbol{\omega} \times \mathbf{d}_i \quad i = 1, 2, 3; \tag{2}$$

we can write the torsion (τ) and the flexures (κ_1, κ_2) of the rod as

$$\boldsymbol{\omega} = \kappa_1 \mathbf{d}_1 + \kappa_2 \mathbf{d}_2 + \tau \mathbf{d}_3 \tag{3}$$

Apart from the bending and twisting deformation measured by torsion and flexures, a rod can have stretching and shearing deformations, which are described by another measure $\mathbf{v}$,

$$\mathbf{v} = \frac{\partial \mathbf{r}}{\partial s} = v_1 \mathbf{d}_1 + v_2 \mathbf{d}_2 + v_3 \mathbf{d}_3 \tag{4}$$

where v_1 and v_2 denote the shear deformation of the cross section which is the tilting of the cross-section, and (v_3-1) denotes the stretch which is the length change of the beam. In practice, we often omit the influence of any shear deformation, that is $v_1 = 0$ and $v_2 = 0$.

In a surgical operation, the maneuver of the subject (the organ) is relatively slow, and a sudden change of speed rarely occurs. Therefore, we can omit the influence of the inertial force and model the deformation in a quasi-static fashion, unlike the previous work [20, 21] where the dynamic effect is considered. In our method, the overall potential energy is written as

$$\Pi = U - W \tag{5}$$

where $U = \frac{1}{2}\int [B_1(\kappa_1 - \bar{\kappa}_1)^2 + B_2(\kappa_2 - \bar{\kappa}_2)^2 + GJ(\tau - \bar{\tau})^2 + EA(v_3 - 1)^2] ds$ is the elastic energy of the rod and $W = \int \mathbf{f} \cdot (\mathbf{r} - \bar{\mathbf{r}}) ds$ is the work by the external force $\mathbf{f}$. Values with note ($\bar{\ }$) are initial state defined on the rest shape. B_1 and B_2 represent the bending stiffness, E is the Young's modulus, A is the area of cross section, G is the shear modulus and J is the polar moment of inertia of the cross sectional area. Minimization of the potential energy gives out the solution of a rod in equilibrium.

3.1 Discretization of the Rod

To find the numerical solution, we approximate a continual rod with n+1 vertices $\mathbf{x}_{0,}$ $\mathbf{x}_1$, …, $\mathbf{x}_\mathrm{n}$ and n straight edge $\mathbf{e}_{0,}$ $\mathbf{e}_1$, …, $\mathbf{e}_{\mathrm{n}-1.}$ The ith edge $\mathbf{e}_i$ connected point $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. The local coordinate system is defined on each individual edge after discretization. For edge $\mathbf{e}_i$, we can have

$$\mathbf{d}_3^i = \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|}\,; \quad \mathbf{d}_1^i \times \mathbf{d}_2^i = \mathbf{d}_3^i \tag{6}$$

With the above notation, we can define the discrete curvature and torsion on vertex $\mathbf{x}_i$ as:

$$\kappa_1^i = \frac{2(\mathbf{d}_3^{i-1} \times \mathbf{d}_3^i)}{(\|\mathbf{e}_{i-1}\| + \|\mathbf{e}_i\|)^2} \cdot (\|\mathbf{e}_{i-1}\| \mathbf{d}_2^{i-1} + \|\mathbf{e}_i\| \mathbf{d}_2^i)$$

$$\kappa_2^i = -\frac{2(\mathbf{d}_3^{i-1} \times \mathbf{d}_3^i)}{(\|\mathbf{e}_{i-1}\| + \|\mathbf{e}_i\|)^2} \cdot (\|\mathbf{e}_{i-1}\| \mathbf{d}_1^{i-1} + \|\mathbf{e}_i\| \mathbf{d}_1^i)$$

$$\tau^i = \frac{2(\mathbf{d}_1^{i-1} \times \mathbf{d}_1^i)}{(\|\mathbf{e}_{i-1}\| + \|\mathbf{e}_i\|)^2} \cdot (\mathbf{e}_{i-1} + \mathbf{e}_i) \tag{7}$$

Here the directors $\mathbf{d}_1^i$, $\mathbf{d}_2^i$, $\mathbf{d}_3^i$ are defined on the ith edge and the curvature(κ_1^i, κ_2^i) and torsion τ^i are defined on the ith vertex. And the stretch on vertex $\mathbf{x}_i$ is defined as

$$v_3^i = \frac{\|\mathbf{e}_{i-1}\| + \|\mathbf{e}_i\|}{\|\bar{\mathbf{e}}_{i-1}\| + \|\bar{\mathbf{e}}_i\|} \tag{8}$$

with note ($\bar{}$) denoting initial state defined on the rest shape.

3.2 Biological Tissues with Constraints

In real anatomy, rod-like structures are embedded into surrounding tissues or fixed at certain position with ligaments and/or fat tissues. For example, the rectum is surrounded by the pelvic skeletal structure. To remove a low rectal cancer requires cutting the rectum near the anal verge and the surgeon has to separate the colon near the anal verge around the perimeter. To model the effect of such constraints, we added springs to the Cosserat rod model as shown in Fig. 2.

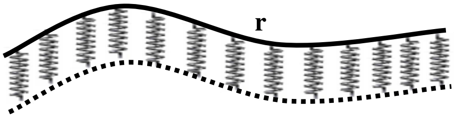


Fig. 2. Model the constraint of surrounding tissues

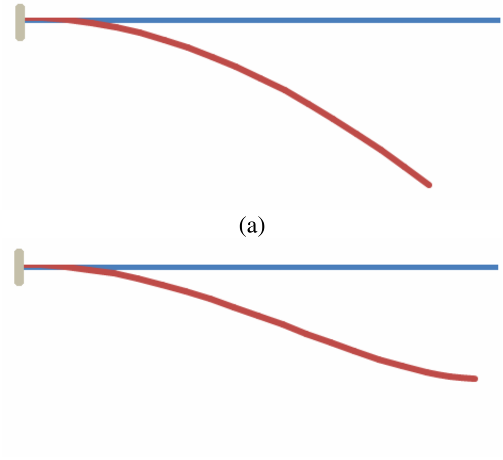
With the additional springs, we can rewrite the elastic energy U in equation (5) as

$$U = \frac{1}{2}\int [B_1(\kappa_1 - \bar{\kappa}_1)^2 + B_2(\kappa_2 - \bar{\kappa}_2)^2 + GJ(\tau - \bar{\tau})^2 + EA(v_3 - 1)^2 + K(\|\mathbf{r} - \bar{\mathbf{r}}\|)^2]ds \quad (9)$$

where the last item describes the contribution of the spring constraints. K is the spring stiffness distribution of the constraints.

If an operation involves separating part or all of the organ from its supporting tissue, we can model the effect of cutting by altering the distribution of stiffness K. $K(s)$ equals to 0 means that the rod has no connection to its surrounding tissue at point $\mathbf{r}(s)$.

Fig. 3 shows results of a uniform elastic rod under gravity, fixed on one end. The top is the result for rod without springs and the bottom is the result for rod with spring constraints.



(a)

(b)

Fig. 3. Deformation of a rod without or with spring constraint

4 Local Deformation Modelling

The Cosserat rod model represents the motion of centreline and rotation of cross section, which reflects the global changes. However, as the cross section has shrunk into one point in geometry, the shape of cross section, which relates to the local deformation caused by the intervention of surgical instruments, is missing from the Cosserat model.

The reaction force on the instrument can be calculated with the Cosserat rod model by consideration of the global deformation along the centreline. But the local deformation of a soft tube-like structure cannot be ignored in some cases. For example, in the resection operation removing a rectum cancer, poking or pushing the bowel causes visible deformation of the tube wall. Previously, such deformation had to be modelled with either the mass spring system or the finite element method, which increases the complexity of computation and modelling. Furthermore, when the geometric mesh of the mass spring system or the finite element method is coarse or distorted, the local deformation may be filtered out and not shown correctly.

Here, we select a curve C resembling the cross-section at intervention point and deform it to create a local deformation. Being a manifold of a circle, the curve C can be parameterised with angle θ from 0 to 2π as $\mathbf{R}(\theta)$. We assume that the deformed C takes an in-plane deformation only. Therefore the deformed shape of the curve is characterised by minimising the potential energy, $\Pi = U - W$.

The elastic energy is defined as

$$U = \frac{1}{2}\int [B(\kappa - \bar{\kappa})^2 + EA(v_3 - 1)^2] \frac{d\|\mathbf{R}\|}{d\theta} d\theta \tag{10}$$

where the first item presents the energy of bending, and the second item denotes that of stretching. Only one bending item appears in an in-plane deformation and there is no twisting. We write the work by external forces as follows,

$$W = \mathbf{F} \cdot (\mathbf{R}_f - \overline{\mathbf{R}}_f) - \int \frac{\mathbf{F}}{L_c} \cdot (\mathbf{R} - \overline{\mathbf{R}}) \frac{d\|\mathbf{R}\|}{d\theta} d\theta \tag{11}$$

where values with note ($\bar{\ }$) are initial values defined on the rest shape, $\mathbf{F}$ is the projection of the exerted force in the cross section at point $\mathbf{R}_f$, and a body force of $-\mathbf{F}/L_c$ is distributed along the curve to balance it, where L_c is the length of curve C. An additional constraint is added which ensures the mass centre of the curve unchanged during the deformation.

Once curve C is deformed, its deformation is propagated to its neighbouring mesh points. Firstly, before deforming the vertices of the edges which C comes across are selected (see Fig. 4(a)). For one of the selected vertices, $\mathbf{p}$, we can find the closest point on the curve as $\overline{\mathbf{R}}_p$ with distance d. Vertex $\mathbf{p}$ is moved to the new position $\mathbf{p}'$ with

$$\mathbf{p}' = \mathbf{p} + (1 - d/h)^2 (\mathbf{R}_p - \overline{\mathbf{R}}_p) \tag{12}$$

where h is a control parameter, which we set to $0.56\sqrt{A}$. A is the area of cross-section C. New points are then added to propagate the deformation further. These newly added points have edge connections to the previous selected points (see Fig. 4(b)). Given a new added point $\mathbf{p}$ which connected to the previous selected k points $\mathbf{p}_1$, $\mathbf{p}_2$… $\mathbf{p}_k$, its new position is given as

$$\mathbf{p}' = \mathbf{p} + \frac{1}{k}\sum_{i=1}^{k} (1 - e_i/h)^2 (\mathbf{p}'_i - \mathbf{p}_i) \tag{13}$$

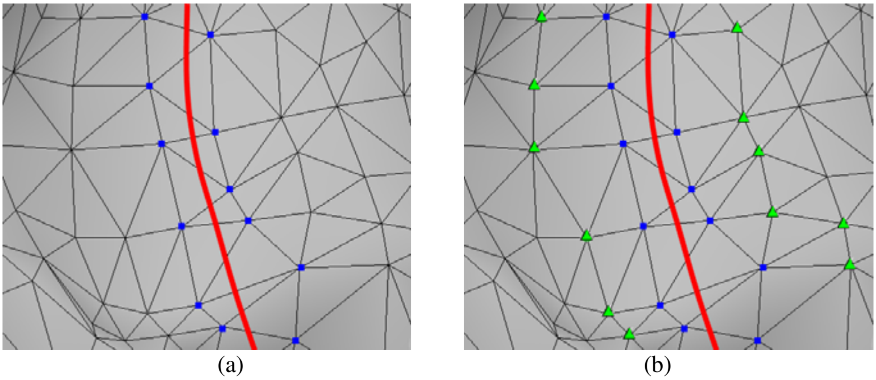

(a) (b)

Fig. 4. (a) Selecting mesh vertices (blue squares) which the curve C (red curve) comes across. (b) New added vertices (green triangles) which connect to the previous selected vertices (blue squares).

where e_i is the edge length of $\mathbf{p}_i$ and $\mathbf{p}$. This process then iterates until $\|\mathbf{p}' - \mathbf{p}\|$ is less than a given threshold $0.01\sqrt{A}$.

5 Results

A Phantom Omni device is used to control the drag and poke actions at a point on the rod-like structure. As shown in Fig. 5, the user can input the 3D coordinates of a selected point on the mesh to mimic the intervention of the instrument in surgery and the Phantom Omni device provides force feedback according to the input.

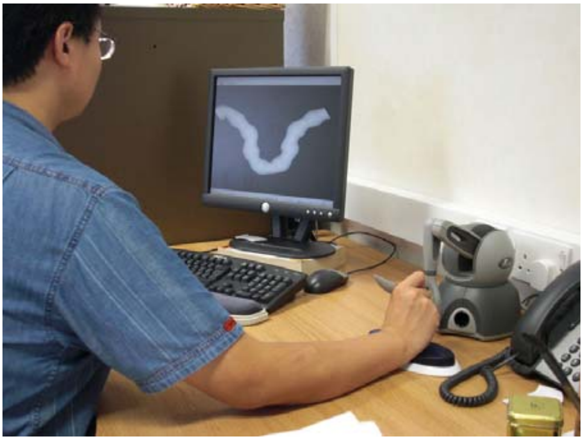
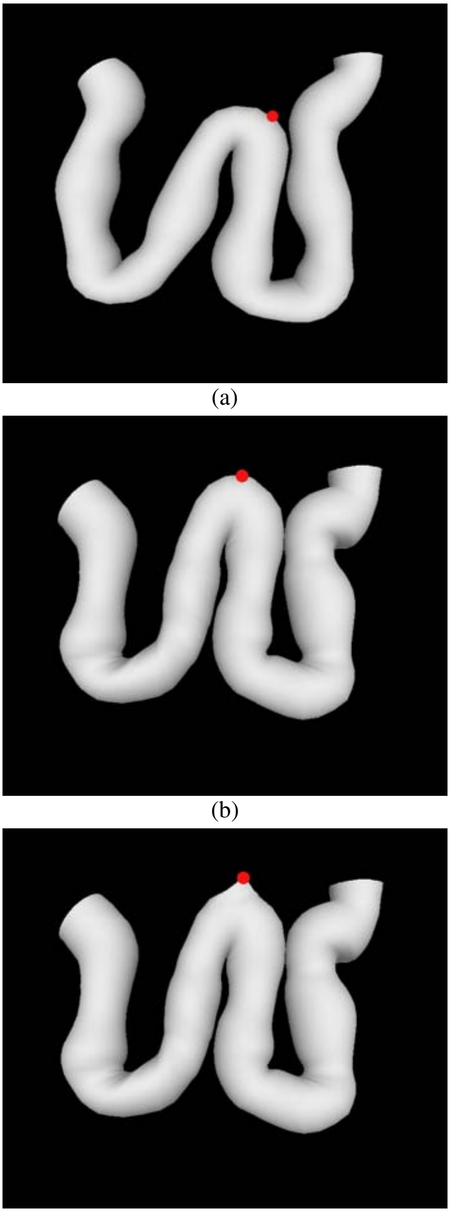


Fig. 5. A Phantom Omni device was used to get haptic output



(a)

(b)

(c)

Fig. 6. (a) a section of intestine before deformation; (b) the section of intestine deformed globally when being dragged upwards; (c) a local deformation added after global deformation

Fig. 6 shows the results without (Fig. 6(b)) or with (Fig. 6(c)) local deformation, which are simulation of a pull operation on a section of a bowel structure. It is observed that the local deformation enhanced the visual fidelity in the simulation.

6 Conclusions

We have presented a method to model one dimensional rod-like biological tissues, which handles both global and local deformations plausibly in an efficient manner. Our strategy is to formulate them with two separate layers. Global deformations of such tissues like bending and twisting are simulated with the Cosserat rod model, while local deformations are handled separately. The cross-section where the drag/poke action is exerted is extracted to generate local deformations, which are also represented with an enclosed curve subject to the given loads. The approximated local deformation propagates to the neighbouring vertices in a geometric manner. Our approach provides a good balance between visual fidelity and computational efficiency. However, if higher accuracy of the local deformation is necessary to an application, one can always use a more accurate and a slower approach, such as nonlinear finite element computation.

The main benefit of using a one dimensional entity to present the physics of the object is efficiency. Fewer degrees of freedom are required to discretize a curve than a surface or a volume in order to capture the change of physical variables. When modelling a slender structure, our approach is more desirable than the traditional finite element method or mass spring system with volumetric discretization.

We have not considered the influence of self-collision in this implementation. It is possible to include known contact algorithms, e.g. the method in [3], into the prototype. In the future, we would also like to extend this work to consider the nonlinear material properties and to model the cutting operation, which requires changes of the mesh topology locally as well as changes of the physical properties of the object accordingly.

References

1. Antman, S.: Nonlinear Problems of Elasticity. Springer, New York (1995)
2. Barr, A.H.: Global and local deformations of solid primitives. In: SIGGRAPH 1984, pp. 21–30 (1984)
3. Raghupathi, L., Grisoni, L., Faure, F., Marchal, D., Cani, M., Chaillou, C.: An intestinal surgery simulator: real-time collision processing and visualization. IEEE Trans. Visual. Comput. Graphics. 10(6), 708–718 (2004)
4. Bro-nielsen, M.: Finite element modeling in surgery simulation. Proc. of the IEEE 86(3), 490–503 (1998)
5. Cotin, S., Delingette, H., Ayache, N.: Real-time elastic deformations of soft tissues for surgery simulation. IEEE Trans. Visual. Comput. Graphics 5(1), 62–73 (1999)
6. Delingette, H., Ayache, N.: Hepatic surgery simulation. ACM Commun. 48(2), 31–36 (2005)
7. Kim, Y., Lee, K., Kim, W.: 3D virtual simulator for breast plastic surgery. Comput. Animat. Virtual Worlds 19(3-4), 515–526 (2008)
8. Comas, O., Taylor, Z.A., Allard, J., Ourselin, S., Cotin, S., Passenger, J.: Efficient nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework SOFA. In: Bello, F., Edwards, E. (eds.) ISBMS 2008. LNCS, vol. 5104, pp. 28–39. Springer, Heidelberg (2008)

9. Mollemans, W., Schutyser, F., Cleynenbreugel, J., Suetens, P.: Fast soft tissue deformation with tetrahedral mass spring model for maxillofacial surgery planning systems. In: Barillot, C., Haynor, D.R., Hellier, P. (eds.) MICCAI 2004. LNCS, vol. 3217, pp. 371–379. Springer, Heidelberg (2004)
10. Hong, M., Jung, S., Choi, M., Welch, S.W.: Fast volume preservation for a mass-spring system. IEEE Comput. Graph. Appl. 26(5), 83–91 (2006)
11. Bathe, K.J.: Finite Element Procedures, 2nd edn. Prentice-Hall, London (1996)
12. Wu, W., Heng, P.A.: A hybrid condensed finite element model with GPU acceleration for interactive 3D soft tissue cutting. Comput. Animat. Virtual Worlds 15(3-4), 219–227 (2004)
13. Bertails, F., Audoly, B., Cani, M., Querleux, B., Leroy, F., Lévêque, J.: Super-helices for predicting the dynamics of natural hair. In: SIGGRAPH 2006, pp. 1180–1187 (2006)
14. Ward, K., Bertails, F., Kim, T., Marschner, S.R., Cani, M.: A Survey on hair modeling: styling, simulation, and rendering. IEEE Trans. Visual. Comput. Graphics 13(2), 213–234 (2007)
15. France, L., Lenoir, J., Angelidis, A., Meseure, P., Cani, M., Faure, F., Chaillou, C.: A layered model of a virtual human intestine for surgery simulation. Med. Image Anal. 9(2), 123–132 (2005)
16. Huang, P., Gu, L., Zhang, S.: Real-time simulation for global deformation of soft-tissue using deformable centerline and medial representation. In: Harders, M., Székely, G. (eds.) ISBMS 2006. LNCS, vol. 4072, pp. 67–74. Springer, Heidelberg (2006)
17. Grégoire, M., Schömer, E.: Interactive simulation of one-dimensional flexible parts. Comput. Aided. Des. 39(8), 694–707 (2007)
18. Linn, J., Stephan, T., Carlsson, J., Bohlin, R.: Fast simulation of quasistatic rod deformations for VR applications. In: Progress in Industrial Mathematics at ECMI 2006, Part 2, pp. 247–253 (2007)
19. Pai, D.: Strands: Interactive simulation of thin solids using Cosserat models. Comp. Graph. Forum 21(3), 347–352 (2002)
20. Spillmann, J., Teschner, M.: CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In: SCA 2007, pp. 63–72 (2007)
21. Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., Grinspun, E.: Discrete elastic rods. In: SIGGRAPH 2008, Article 63 (2008)
22. Chang, J., Shepherd, D., Zhang, J.J.: Cosserat-beam-based dynamic response modelling. Comput. Anim. Virtual Worlds 18(4-5), 429–436 (2007)

Using Musculoskeletal Modeling for Estimating the Most Important Muscular Output – Force

Mark de Zee[1,2] and John Rasmussen[2]

[1] Center for Sensory-Motor Interaction, Dept. of Health Science and Technology,
Aalborg University, Denmark
`mdz@hst.aau.dk`
[2] AnyBody Research Group, Dept. of Mechanical Engineering,
Aalborg University, Denmark
`jr@me.aau.dk`

Abstract. This paper presents a method and a set of detailed musculoskeletal models for estimation of individual muscle forces for a given movement based on inverse dynamics. The basic methodology and the construction of the human body model library are presented and examples of prediction and validation of muscle forces in the mandible and the neck are given. Furthermore it is demonstrated how musculoskeletal models give insight into the motor control from a functional point of view. The paper concludes that motor control changes and biomechanics are dependent on each other and musculoskeletal models can help us to understand changes in muscle coordination and muscle force for different kinds of situations.

Keywords: Musculoskeletal modeling, Inverse dynamics, Optimization, Neck, Mandible.

1 Introduction

Motor control studies have traditionally been focused on electrophysiological signals like action potentials and EMG. Strangely enough the most important functional output of a muscle, namely the force, is quite often not considered. One of the reasons is probably that something as trivial as a single muscle force is almost impossible to measure in-vivo and the only viable possibility for estimating muscle forces is to make use of computational modeling of the musculoskeletal system based on the laws of physics.

This paper presents a method for estimation of muscle forces by advanced simulation of the mechanics of the human body and illustrates the method by means of two application examples.

2 Methods

2.1 Simulation Methods

A brief review of the attempts to simulate the human body as a mechanical system is best initiated with the observation that the methods traditionally fall into one of two

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 62–70, 2009.

categories, inverse dynamics and forward dynamics, which, as the names indicate, are opposite approaches.

In inverse dynamics, the motion and the external loads on the body are assumed known, and the purpose of the computation is to determine the internal forces. When the "internal forces" are mere joint moments and joint reaction forces, inverse dynamics is in most cases a straightforward procedure involving the solution of a system of linear equilibrium equations. However, for the purpose of computing individual muscle forces, inverse dynamics is haunted by the so-called redundancy problem: not enough equilibrium equations are available to determine all the muscle forces. Infinitely many different sets of muscle forces, of which the central nervous system (CNS) instantly chooses one, can therefore produce the identified joint moments. This is due to the fact that we have more muscles than strictly necessary to drive most motions. Constructing an algorithm to determine the activation of each muscle therefore entails guessing the motives behind the CNS's function. We are able to repeat movements with considerable precision so many researchers believe that the control of muscle forces must be based on some rational criterion.

Assuming that muscles are recruited according to an optimality criterion, we are faced with the task of selecting the right one. Let us briefly state the mathematical form of the inverse dynamics problem:

$$\text{Minimize} \qquad G(\mathbf{f}^{(\mathrm{M})}) \tag{1}$$

$$\text{Subject to} \quad \mathbf{Cf} = \mathbf{d} \tag{2}$$

$$f_i^{(\mathrm{M})} \geq 0, \qquad i \in \{1, \ldots, n^{(\mathrm{M})}\} \tag{3}$$

where G is the objective function of the recruitment strategy stated in terms of the muscle forces, f(M), and minimized with respect to all unknown forces in the problem, f=[f(M) T f(R) T]T, i.e., muscle forces, f(M), and joint reactions, f(R). Eq. (2) is the dynamic equilibrium equations, which enter into the optimization problem as constraints. C is the coefficient-matrix for the unknown forces, and the right-hand side, d, contains all known applied loads and inertia forces. The non-negativity constraints on the muscle forces, Eq. (3), state that muscles can only pull, not push.

Surveys of suggestions for the objective function G can be found in the literature [12]. Most of the reasonable criteria are functions of the normalized muscle forces, f(M)i/Ni, where Ni is some measure of the muscle strength at each muscle's current working conditions. It has been shown [10] that many of the criteria are asymptotically equivalent to a minimum fatigue criterion, which after some reformulation takes the following form:

$$\textit{Minimize} \quad \beta + \varepsilon_1 \sum_i \frac{f_i}{N_i} + \varepsilon_2 \sum_i \left(\frac{f_i}{N_i} \right)^2 \tag{4}$$

$$\textit{Subject to} \quad \mathbf{Cf} = \mathbf{d}$$

$$\frac{f_i}{N_i} \le \beta, i \in \left\{1,..,n^{(M)}\right\} \tag{5}$$

$$f_i^{(\mathrm{M})} \ge 0, \qquad i \in \left\{1,...,n^{(\mathrm{M})}\right\} \tag{6}$$

Here, β is an artificial, so-called bound variable that serves the purpose of simultaneously reducing all the muscle activities in the problem, and the factors ε_1 and ε_2 control penalty terms that stabilize the numerical properties of the problem and resolve indeterminacies among submaximally activated muscles.

The principal advantage of inverse dynamics for investigations is that its computational efficiency allows for realistic models of the human body comprising hundreds of muscles as shown in Figure 1. The main disadvantage is that muscle activation dynamics is not taken into account.

2.2 The Muscle Activity Envelope

The min/max formulation, Eqs. (4) through (6), is as originally noted by An et al [1] effectively a minimum fatigue criterion, i.e. the muscles are recruited to postpone fatigue of the highest relatively loaded muscle as far as possible. The physiological effect of this presumption is that the muscle activations in the model, i.e. fi/Ni, tend to form groups at different activity levels. In particular, the level formed by the muscles with maximum activity at a particular time step is usually shared by multiple synergistic; muscles with low load levels come to the aid of highly loaded muscles, and consequently many muscles will share the same activity level and contribute to carrying the load corresponding to their individual strengths.

This forms a distinctive envelope of muscle activity, which we shall henceforth call "the muscle activity envelope". The linearity of the reformulated min/max criterion guarantees that the optimization problem, Eqs. (4) through (6) is convex and the solution to the problem therefore is the global optimum. This means that no other muscle recruitment strategy can reduce the envelope further. Since the muscle activity envelope represents the maximum muscle activation in the model it can be interpreted as the percentage of maximum voluntary contraction necessary to carry the imposed load in the prescribed posture. This makes the envelope an interesting measure for ergonomic design optimization, in the sense that design changes leading to a decrease in the envelope level will be perceived as less fatiguing. The muscle activity envelope therefore serves as a convenient and simple design criterion encapsulating the combined load on all the muscles in the model.

2.3 Musculoskeletal Model

The musculoskeletal model is built with the AnyBody Modeling System ver. 4.0 (AnyBody Technology, www.anybodytech.com, 2009), which is described by Damsgaard et al. [4] in detail. The musculoskeletal model is based on the public domain AnyScript Model Repository (www.anyscript.org).

The body models comprise (i) an arm and shoulder complex with morphology according to Van der Helm [13] with 114 muscle units on each side, (ii) a lumbar spine model comprising sacrum, all lumbar vertebrae, and a total of 158 muscles [6], (iii) a rigid thoracic spine section, (iv) a cervical spine model comprising 136 muscles [7], (v) a pelvis and lower extremity model with a total of 326 muscles arranged according to Klein Horsman et al [9], and (vi) a mandible model [5]. Totally, the model comprises some 750 individual muscle units, and as such it is a very detailed description of the human musculoskeletal system. The anthropometrical dimensions of the model roughly correspond to a 50th percentile European male.

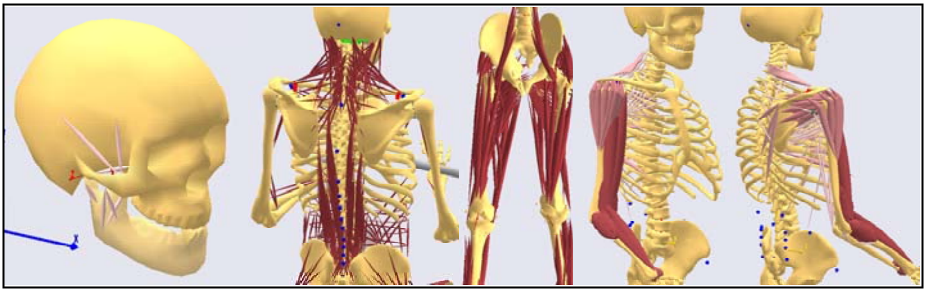


Fig. 1. Selected details of the musculoskeletal model

The model segments are rigid bodies with mass properties corresponding to the bone mass plus the contribution of the soft tissue that can reasonably be attributed to each bone. Joints are idealized frictionless constraints between the segments. The model uses standard joints such as spherical joints for the hips and hinge joints for the knee in addition to specially developed joints to represent for instance the scapula floating on the thorax.

Muscles in the model are strings spanning the distance between origin and insertion points through via points and wrapping over obstacle surfaces on the way. Wrapping is handled as a shortest path contact mechanics problem.

The model mechanics is a full three-dimensional Cartesian formulation including inertial body forces, which in the present static problem is comprised solely by gravity.

Integral validation of whole-body musculoskeletal models is very difficult to conduct. However, the parts of the whole-body model were evaluated separately. For the lumbar spine model a comparison was made [6] with in vivo intradiscal pressure measurements of the L4-5 disc available from the literature [15]. The lower extremity model has been validated against pedalling experiments giving good results in the prediction of muscle activations and pedal forces. The shoulder model is based on an already validated model developed by Van der Helm [13].

3 Neck Example

The neck is an interesting structure, because the cervical spine is a complex mechanical system with many degrees of freedom. The multiple muscles of the cervical spine act to support and move the head in three-dimensional space. Work-related neck pain is a common problem in society, and changes in motor control of the cervical spine have been documented in people with neck pain [2]. For example it has been shown that patients with chronic pain show a decreased activity in the deep cervical flexors and an increased activity in the superficial flexor muscles during a craniocervical flexion test [8]. This kind of very relevant studies is based on EMG measurements. However, the possible mechanics behind a change in motor control cannot be taken into account. In this example we shall show the potential of musculoskeletal modeling based on inverse dynamics for increasing our understanding of the change in motor control we see in patients with chronic neck pain.

3.1 The Neck Model and Simulation Protocol

A three-dimensional musculo-skeletal model of the cervical spine (See Figure 2) based on inverse dynamics was built using the AnyBody Modeling System (AnyBody Technology A/S, Aalborg, Denmark). The model is equipped with 136 muscle fascicles and most of the parameters in the model are based on work by Van der Horst [14].

In the simulation the model was forced to create a flexion moment of 30 % of maximum voluntary contraction (MVC). Two simulations were performed corresponding to a non-painful condition and a chronic pain condition. The min-max algorithm [10] was used to solve the muscle recruitment problem in order to calculate the muscle forces necessary for producing this flexion force in the two simulated conditions. In the chronic pain condition the longus colli and longus capitis were forced to

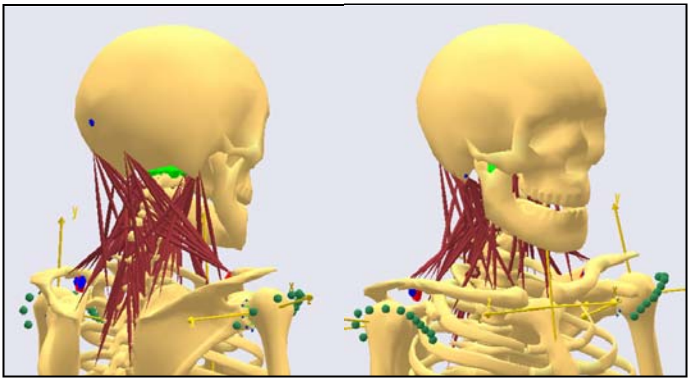


Fig. 2. The cervical spine model. The red lines indicate the various muscles included in the model.

provide a force 55 % lower than in the non-painful condition. The compensatory mechanisms were then predicted by redoing the same simulation as in the non-painful condition.

3.2 Results and Discussion

The simulations showed that a 55 % inhibition of the longus colli and longus capitis resulted in an increase of 17 % of activity in the sternocleidomastoid. This is in agreement with the experimentally data reported by Falla et al [8].

This simulation does not explain how chronic neck pain can lead to an inhibition of the deep cervical flexors. It does explain how this inhibition leads to an increased activation of the superficial flexors. The result suggest that the central nervous system counteracts the inhibition of the deep cervical muscles by choosing a muscle recruitment strategy of the remaining muscles, which minimizes fatigue while maintaining equilibrium in the system.

The musculoskeletal model of the neck gives rise to new hypothesizes by taking into account the mechanical laws, something, which is not considered before. Of course a hypothesis like this should be further supported by more experimental data as well.

4 Mandible Example

The human mandible with its two temporomandibular joints (TMJ) is a complex biomechanical system. The mandible can move with six degrees of freedom due to the special construction of the TMJ. The central nervous system (CNS) has the challenge to control all these degrees of freedom for a variety of functions like chewing, clenching, speech, swallowing. Moreover the requirements of these tasks are often a combination of high forces and precision. From a clinical perspective many dentist are interested in the loading of the TMJ, because temporomandibular disorders are common. The loading of the TMJ depends on how the CNS activates the different muscles for a given task. Still an important parameter as TMJ load is impossible to measure and musculoskeletal modeling is the only feasible method to get an estimate for this parameter. De Zee et al [5] validated a generic mandible model with the use of EMG measurement for several tasks.

An important treatment modality for correcting and reconstructing major dentofacial deformities is distraction osteogenesis [11]. Distraction osteogenesis (DO) is a gradual incremental bone lengthening technique that requires very precise treatment planning. Such a macroscopic change in the geometry of the mandible will lead to a change in muscle coordination and load transfer to the TMJ. The changes in TMJ loading as a result of DO has been estimated previously using the finite element method [3]. This method, however, does not take into account the change in muscle coordination. In this example we shall show the potential of musculoskeletal modeling based on inverse dynamics for estimating the change in muscle coordination and TMJ loading after mandibular distraction osteogenesis.

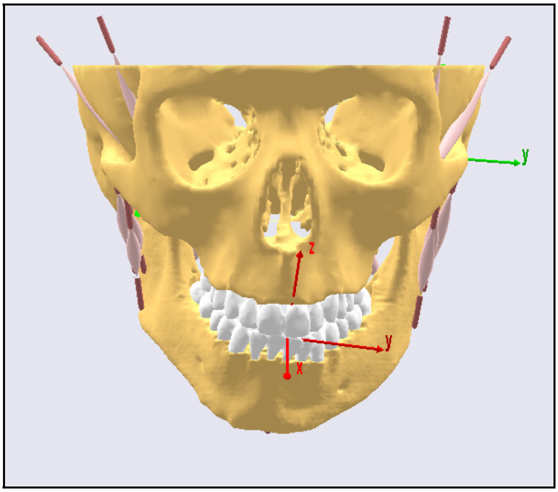


Fig. 3. A graphical representation of the musculoskeletal mandible model of a real patient in need of distraction osteogenesis. The right ramus needs to be lengthened by 15 mm.

4.1 The Mandible Model and Simulation Protocol

The presented model is patient-specific and was based on CT-scans of a real patient with unilateral hypoplasia of the right mandibular ramus due to juvenile idiopathic arthritis, which means that the right side of the mandible was much shorter than the left side. The model was equipped with 24 musculotendon actuators. For a graphical representation of this patient model see Figure 3. In reality and therefore also in the model the inclination of the articular eminence on the affected side is smaller (more flat) than the inclination on the healthy side. Moreover the peak isometric force of the masseter, medial pterygoid, and lateral pterygoid on the affected side were weaker by 17, 3 and 6 percent, respectively. These values were based on MRI scans [3].

The input in the model was a clenching force of 150 N on the central incisors both before and after DO. Corresponding to the real patient the right ramus of the model was lengthened by 15 mm. In the results we will focus on the TMJ loading and the muscle force of the lateral pterygoid. This muscle is an important muscle in controlling the forward motion of the mandible.

4.2 Results

Before DO the loading on right TMJ was 124 N and on the left side 131 N. The estimated force in the right lateral pterygoid was 30 N and 18 N on the left side.

After DO the loading on right TMJ was 108 N and on the left side 120 N. The estimated force in the right lateral pterygoid was 54 N and 93 N on the left side.

4.3 Discussion

The results show that after DO, which is a significant change of geometry, the muscle coordination has to change in order to produce the same amount of clenching force. This in itself is not surprising; however, one would never be able to quantify this change without using musculoskeletal modeling. The whole change of geometry and the muscle activations lead to a lower TMJ loading for the given clenching force.

In the future it is anticipated that, with patient-specific musculoskeletal mandible models, one can assess the change of muscle coordination and the mechanical effects on the TMJ of before and after different types of surgery.

5 Conclusions

The two examples illustrate that motor control changes and biomechanics are really interdependent variables. One may argue that this is of course what one would expect. However, in the motor control literature there is not a tradition to couple changes in motor control, expressed in electrophysiological signals like EMG, together with the functional forces within the body. The methodology presented in this work in combination with detailed musculoskeletal models can help us to understand changes in muscle coordination and muscle force for different kinds of situations. To be able to estimate muscle forces is very fundamental, because in the end the muscle forces make our body move.

An important remaining question is how the predicted forces can be validated? This question, however, is not limited to this particular paper but remains a constant throughout such modeling techniques and will be a very important research topic in the future.

Acknowledgments. The neck model was developed with contributions from RBM A/S and the help of Karen Søndergaard, Eva Søndergaard, Deborah Falla and Dario Farina. The mandible model was developed with contributions from the Villum Kann Rasmussen foundation and the help of Michel Dalstra, Paolo M. Cattaneo, Peter Svensson, Thomas Klit Pedersen and Birte Melsen. All contributions are gratefully acknowledged. This work is supported by the 3D Anatomical Human (MRTN-CT-2006-035763) project funded by the European Union.

References

1. An, K.N., Kwak, B.M., Chao, E.Y., Morrey, B.F.: Determination of muscle and joint forces: a new technique to solve the indeterminate problem. Journal of Biomechanical Engineering 106, 364–367 (1984)
2. Buckle, P.W., Devereux, J.J.: The nature of work-related neck and upper limb musculoskeletal disorders. Applied Ergonomics 33, 207–217 (2002)
3. Cattaneo, P.M., Kofod, T., Dalstra, M., Melsen, B.: Using the finite element method to model the biomechanics of the asymmetric mandible before, during and after skeletal correction by distraction osteogenesis. Computer Methods in Biomechanics and Biomedical Engineering 8, 157–165 (2005)

4. Damsgaard, M., Rasmussen, J., Christensen, S.T., Surma, E., de Zee, M.: Analysis of musculoskeletal systems in the AnyBody Modeling System. Simulation Modelling Practice and Theory 14, 1059–1070 (2006)
5. de Zee, M., Dalstra, M., Cattaneo, P.M., Rasmussen, J., Svensson, P., Melsen, B.: Validation of a musculo-skeletal model of the mandible and its application to mandibular distraction osteogenesis. Journal of Biomechanics 40, 1192–1201 (2007a)
6. de Zee, M., Hansen, L., Wong, C., Rasmussen, J., Simonsen, E.B.: A generic detailed rigid-body lumbar spine model. Journal of Biomechanics 40, 1219–1227 (2007b)
7. de Zee, M., Falla, D., Farina, D., Rasmussen, J.: A detailed rigid-body cervical spine model based on inverse dynamics. Journal of Biomechanics 40, S284 (2007c)
8. Falla, D.L., Jull, G.A., Hodges, P.W.: Patients with neck pain demonstrate reduced electromyographic activity of the deep cervical flexor muscles during performance of the craniocervical flexion test. Spine 29, 2108–2114 (2004)
9. Klein Horsman, M.D., Koopman, H.F., van der Helm, F.C., Prose, L.P., Veeger, H.E.: Morphological muscle and joint parameters for musculoskeletal modelling of the lower extremity. Clinical Biomechanics 22, 239–247 (2007)
10. Rasmussen, J., Damsgaard, M., Voigt, M.: Muscle recruitment by the min/max criterion – a comparative numerical study. Journal of Biomechanics 34, 409–415 (2001)
11. Swennen, G., Schliephake, H., Dempf, R., Schierle, H., Malevez, C.: Craniofacial distraction osteogenesis: a review of the literature: Part 1: clinical studies. International Journal of Oral and Maxillofacial Surgery 30, 89–103 (2001)
12. van Bolhuis, B.M., Gielen, C.C.: A comparison of models explaining muscle activation patterns for isometric contractions. Biological Cybernetics 81, 249–261 (1999)
13. van der Helm, F.C.: A finite element musculoskeletal model of the shoulder mechanism. Journal of Biomechanics 27, 551–569 (1994)
14. van der Horst, M.J.: Human head neck response in frontal, lateral and rear end impact loading - modelling and validation. Technische Universtiteit Eindhoven, Eindhoven, The Netherlands (2002)
15. Wilke, H., Neef, P., Hinz, B., Seidel, H., Claes, L.: Intradiscal pressure together with anthropometric data–a data set for the validation of models. Clinical Biomechanics 16, S111–S126 (2001)

Computer Assisted Estimation of Anthropometric Parameters from Whole Body Scanner Data

Christian Lovato[1], Umberto Castellani[2], Simone Fantoni[2], Chiara Milanese[1],
Carlo Zancanaro[1], and Andrea Giachetti[2,3]

[1] Dipartimento di Scienze Morfologico-Biomediche, Università di Verona
[2] VIPS lab, Dipartimento di Informatica, Università di Verona
[3] CRS4, Sardegna Ricerche, Edificio 1, 09010 PULA, CA - Italy
andrea.giachetti@univr.it
http://www.univr.it

Abstract. In this paper we describe a tool for the semi-automatic evaluation of anthropometric parameters from whole body scanner data. It is based on a user friendly interface allowing the interactive visualization of the mesh and the use of specifically designed measurements tools able to obtain standardized anthropometric estimates that can be compared with manually acquired data. The "virtual" measurements procedures implemented exploit mesh processing algorithms specifically designed or customized for the task. The system also performs an automatic pose estimation and mesh segmentation providing automatically other pose-independent anthropometric parameters. In order to evaluate the system, anthropometric measurements performed with classical tape-based techniques have been compared with those simulated on the acquired mesh.

Keywords: Anthropometric measurement, Whole-body scanner, Mesh processing.

1 Introduction

The use of 3D scanners to acquire anthropometric data is a great opportunity in order to obtain large datasets of human body measurements. Several national and international initiatives, like the CAESAR project [21], have been recently realized to collect measurements performed on 3D scanner data, and several groups are involved in research aimed at using this kind of data for the design of clothes and other objects that must be adapted to the user size.

The current generation of whole-body scanners, despite some limitations, is able to create sufficiently accurate representations of the human body. However, the output of the scanners is not a set of measurements, but a cloud of points (or a triangulated mesh derived from these points) and post processing is necessary to extract useful numbers for statistical analysis and also to compare data with tape-based measurements.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 71–83, 2009.

Relevant research efforts are being spent in order to automatically detect feature points on meshes and to perform automatically measurements and even if, in some cases, the procedures do not appear very robust to changes in pose, results are often promising, showing that the automatic analysis of body scanner data can be useful for many applications.

However, in order to use the body scanner to perform studies requiring statistical analysis or follow-up studies on single subjects as commonly done in the clinical practice, it is necessary to develop specific tools and protocols that can be used to perform a semi-automatic measurement of the parameters of interest, making the collection of them easy but robust, allowing the use of automatic methods when possible, but also enabling the user to perform manual correction of landmarks positioning and interactive measurements.

In this paper we describe a tool with these features, based on an user friendly interface and mesh processing tools enabling a Virtual Reality simulation of the manual measurements usually done in the clinical practice. Due to the use of a preprocessing pipeline including skeletonization and pose estimation, other anthropometric parameters (e.g. limb lengths and related girths) can be provided automatically by the software.

Preliminary experiments performed to validate the anthropometric estimates demonstrate that simulated tape based measurements are quite close to those manually acquired, while the automatic measurement cannot be compared directly with the manual measurements and appear not too accurate for some of the subjects due to the difficulty in matching feature positions. More advanced processing methods able to locate surface features will be used in future versions.

The paper is organized as follows: Section 2 describes the current whole body scanner technologies and the scanner used in our tests, Section 3 presents an overview of different approaches for automatic or assisted measurement of anthropometric measurements on scanner data, Section 4 presents our system, describing the mesh processing methods and the user friendly interface. Section 5 presents experimental results, comparing standard manual measurements with those simulated on our system and analyzing the reliability of automatic measurements obtained with our mesh processing pipeline.

2 Whole-Body Scanners

Whole-body scanners are becoming widely used for anthropometric studies and several manufacturers are now able to provide devices able to capture most of the body surface in a sufficiently small time to avoid relevant motion artifacts. These devices are usually developed using laser or Moiré-fringing-based technologies. In the former a laser stripe is projected onto the body surface and images are acquired by several cameras: in this way 3D points representing the body shape can be recovered by triangulation. Examples of this kind are the scanners developed by Cyberware [2], Hamamatsu [4], Vitronic [8] or Human Solutions [3].

Fig. 1. Left: Breuckmann bodyScan in action. Right: example of mesh with associated texture. It is possible to notice holes in shadowed and reflective regions.

In the latter, one or more projectors create light patterns on the body surface and 3D points are estimated by observing the pattern deformations on the body surface with a set of cameras. Examples of this kind are the scanners developed by Textile and Clothing Technology Corporation [6], Telmat[5] inSpeck [5], and the product used for our experiments, i.e. the bodySCAN developed by Breuckmann GmbH [1].

Breuckmann bodySCAN typical acquisition consists of about 400.000 points with a precision from 0.2 mm to 1.4 mm, with a time acquisition of 2.5 or 5.5 seconds, depending on the acquisition mode. In our experiments we found that this acquisition time is sufficient to obtain sufficient quality for the subsequent processing.

The scanner output consists of a triangular mesh obtained by the registration and merging of the points acquired by the different cameras. Meshes are, however, not very clean, presenting various types of defects like holes, non manifold edges, bad shaped triangles and outliers, so a pre processing step is mandatory.

Meshes nodes include also grayscale information. This is extremely important in order, for example, to acquire landmarks position useful for automatic or computer assisted measurements by exploiting natural or added texture (i.e. skin markers).

Figure 1 shows the acquisition setup and an example of textured mesh. It is possible to see holes and inaccuracies caused by occlusions and reflective materials (i.e. hair).

3 Related Work

Methods to extract anthropometric features in an interactive or automatic way have been proposed by several authors. In [19] methods for computer assisted estimation of values used in the garment industry are presented. 3D scanners are in some cases provided with software tools able to extract automatic or semi-automatic sets of measurements. A comparison of data provided by some of these tools, showing capabilities but also limitations and inconsistencies is presented in [22].

Scanner-based anthropometry and body composition have been validated against underwater weighing and tape measurement [13] and correlated with metabolic syndrome [15]. Body scanners typically show precision of measurement within 0.048 cm [20] and mean difference with manual up to 0.10 cm [12] as well as intraclass correlation coefficient between methods of > 0.99 for several circumferences [13].

In the CAESAR project, Robinette et al. [10] developed a semi-automatic method to locate landmarks based on the placement of markers on the scanned body and the user assisted location of markers on scanner images. Landmarks are then labeled and used for the measurement estimation. Ben Azouz et al. [9] developed a method to locate automatically, using features known as spin images, the landmarks used in the CAESAR project and validated the results with manual placement). Results, however, due to the intrinsic complexity of the problem (many features are located manually by palpation), are not always accurate. Leong et al [17] computed automatically a set of 21 landmarks and 35 feature lines from scanner data, with a method based on the processing of depth maps (and therefore pose dependent).

4 System Description

The goal of this work is to develop a dedicated software tool that can be used to collect and validate anthropometric measurements performed on scanner data. The main differences between our approach and others found in literature is that we want to simulate the tape based manual procedures in order to validate the scanner based measurements and to avoid the inaccuracy of the automatic procedures. Being the interactive procedure time consuming we try to facilitate the measurements exploiting some information derived by a preliminary mesh processing described in Section 4.1. The automatic mesh processing provides also a series of automatic measurements, (lengths, girths), that are still not directly used for a reliable parameter evaluation. We plan to improve the accuracy and the number of automatically extracted parameters in future releases of the tool.

4.1 Automatic Mesh Processing: Skeletonization, Segmentation and Stick Figure Estimation

Acquired meshes are automatically processed in order to extract information useful to facilitate the interactive measurement process and also to directly estimate anthropometric values without human intervention. The automatic procedure is summarized in Figure 2

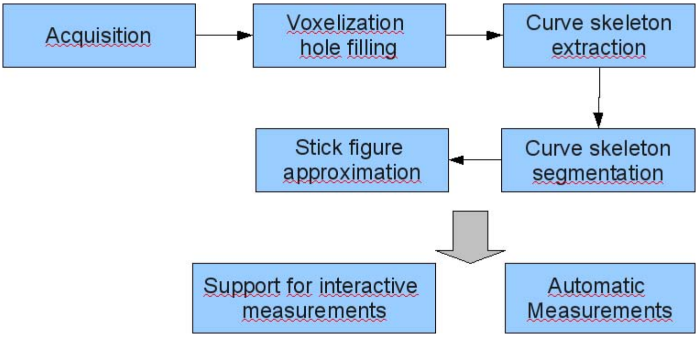


Fig. 2. The processing pipeline providing support for interactive measurements and automatically estimated values

To compute body sections perpendicular to arms, legs or trunk, it is necessary to evaluate a skeletal approximation of the body. In this way it is possible to automatically obtain a ”directional” information related to each surface point that is useful to define sectioning planes and to compute local diameters and section lengths, and also to segment head trunk and limbs, and labeling their parts with simple heuristics, as shown in [18].

The first step consists of voxelizing the mesh volume and closing holes using the method described in [16]. Then a skeletonization procedure is performed. The skeletonization algorithm applied is based on a preliminary extraction of a skeletal tree of branches performed with voxel coding method (the number of branches is a parameter of the algorithm). The tree is then its centered with an active concur approach. Skeleton branches undergo an evolution driven by elastic forces keeping the curve smooth, a force moving points along the gradient of distance from border function and another one trying to center the points in directional sections along the three axes. The method, adding post processing step removing superimposed branches and adjusting point sampling, provides good results (i.e. a precisely centered path along the tubular parts like head/neck and limbs), with reasonable computational cost (see Figure 3) and no free parameters to be tuned (force coefficients are not critical and do not need to be changed after the initial setting). To segment and characterize the different parts of the body, during the skeleton extraction, scalar values (distance from border, average diameter and eccentricity of perpendicular section and curvature) are computed on skeleton points. From the analysis of these values on the extracted skeletal tree, we are able to segment and label the regions corresponding to head, neck, trunk, arm, forearm, hand, leg, thigh, foot (see [18] for details). Left and right side are distinguished from angles (leg/thigh or thigh/foot).

The following step in our mesh analysis pipeline consists of creating a stick figure approximating the curve skeleton extracted. This approximation is extremely useful for several reasons:

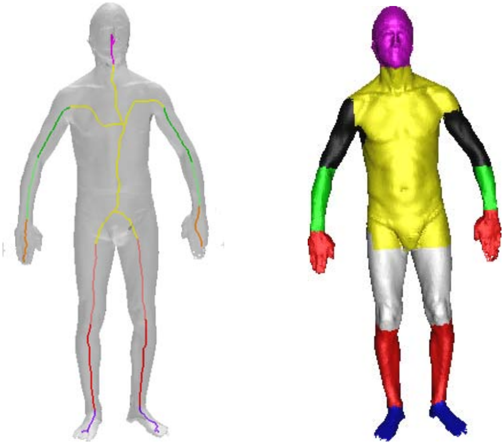


Fig. 3. Left: curve skeleton extracted and labeled superimposed to the original mesh. Right: labeled mesh.

- it provides an estimate of the pose of the subject, that can be used, for example to fit automatically deformable models representing human shapes like those described in [11] or to help the recognition of anatomical landmarks.
- it provides a robust estimate of the axis of each part of the body that we use to estimate girths.
- it directly provides anthropometric values that can be collected (i.e. lengths of arms, forearms, legs, etc.)

Stick figure approximating the extracted skeleton is initialized on feature points extracted on the curve-skeleton. Being the feature location not always precise, we make the pose estimation more robust by applying an optimization procedure forcing symmetry and including in the error function distance from data, distance from original features and average body proportions. The optimization procedure is realized through a non linear least square minimization algorithm. An example of the output of this procedure is represented in Figure 4.

4.2 User Interface and Semi-automatic Measurements

The measurement tool developed consists of a user friendly interface (see Figure 5 developed using VTK libraries [7], used to display the acquired triangulated mesh, to select on a sort of electronic record measurements names, placing on the mesh the required markers for each measurement. Measurements supported are 11 girths (Head, Neck, Arm, Forearm, Wrist, Chest, Waist, Gluteal, Mid-thigh, Calf, Ankle) and 14 lengths/distances (Radiale-stylion,Midstylion-dactylion,Iliospinale h., Trochanterion h., Throcanterion-tibiale lat., Tibiale lat. h., Foot length, Biacromial, Biiliocrestal, Transverse chest, A-P chest depth, Biepicondilar humerus, Wrist breadth, Bimalleolar breadth.)

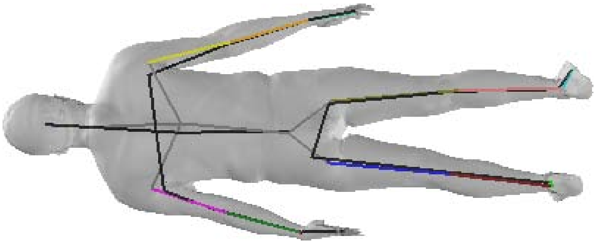


Fig. 4. The fit of the stick figure over the curve skeleton obtain head, trunk and limbs size and orientation reducing possible errors in the skeletal feature location. In the example, the black structure is the initial stick estimated from the skeletal features, while the color structure is the result of the optimization.

In order to perform all the standard measurements required it is necessary to give the possibility of selecting points on the surface and estimate from one or more points selected values:

- 3D coordinates of selected points
- Distances between selected points
- Geodesic paths joining two or more points
- Sections, possibly perpendicular to a skeletal line, with corresponding lengths and areas

The first two measures are straightforward. The computation of geodesic paths joining a set of landmarks on the original mesh can create some problems due to the fact that the meshes obtained from the bodyScan are not clean and watertight. In order to obtain accurate measurements we therefore developed an algorithm to approximate geodesic distances on the mesh that can handle holes and discontinuities in the body surface. It is based on the computation of distance maps computed on the surface: when the mesh presents borders, the map at each border point is updated as the minimum between the current value due to the node propagation, and the sum of values at other points of the same contour border and the Euclidean distance between points.

Finally, the evaluation of girths requires often the extraction of section planes perpendicular to the limbs (or neck or trunk) direction. In order to be more accurate than considering the z direction and standard posture, we use, for some of the measurement, the information extracted with the skeleton/stick figure. This should make the system more robust against changes in the standard pose of the subject with respect to other methods.

With the user interface developed it is easy to simulate interactively the measurement procedures usually done manually on the subjects. It is sufficient to select the name of the standard value on an electronic record, select the landmarks related with that measurement clicking on the model surface on the interface, check the measurement path drawn in the scene and saving the value if everything is correct.

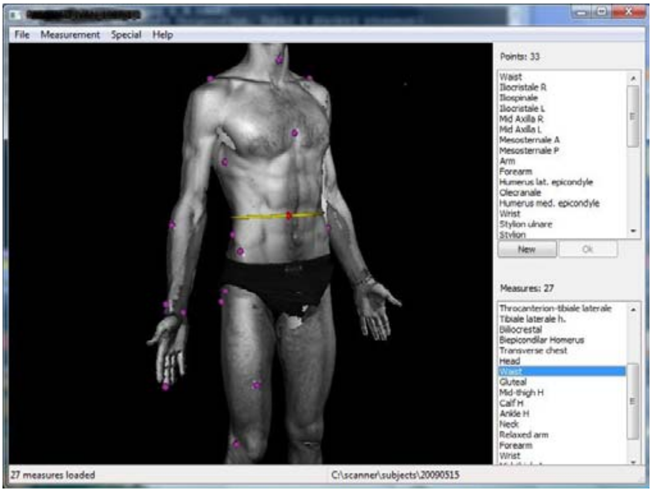


Fig. 5. User interface of the computer assisted measurement system

For each simulated measurement, the interface shows the selected points on the skin, and all the lines, geodesic paths and sections used for the value estimation. During the process, users can always change landmark positions, and, if the measures depend also on the automatically extracted section planes, it is possible to correct manually the plane orientation. Figure 6 shows examples of measurements performed on the user interface.

In our experiments, all the landmarks used for the anthropometric measurements are marked on the skin of the subject drawing a with a dermographic pen before the acquisition. The placement of such landmarks is useful for several reasons:

- it allows the offline measurement on the mesh of values landmarks that cannot be located using image based or geometrical features. Relevant measurements, in fact, are based on landmarks placed by physically touching the subject.
- it allows an easier semi-automatic placement of the points during the computer based measurements
- reference points can be used to validate automatically the landmark detection algorithms based on surface features.

During the measurement process, the user should click on the mesh near the visible cross. An automatic centering procedure can be activated to center the landmarks position near the cross center.

We plan, in future releases of the tool to implement also an automatic extraction and labeling of the landmarks exploiting the mesh segmentation derived from the curve skeleton analysis.

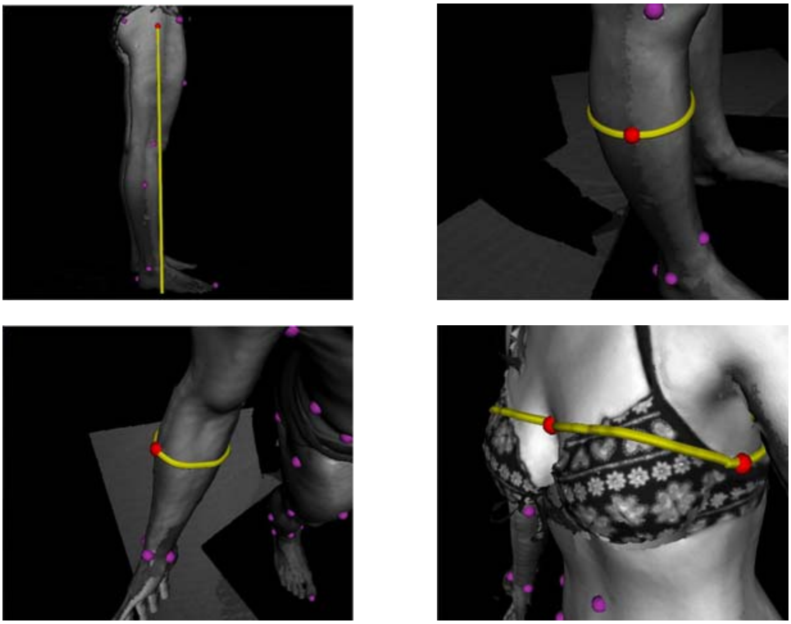


Fig. 6. Examples of measurements performed interactively with the user interface. Landmarks positioning or section plane angles can be interactively modified.

4.3 Automatic Measurements

The skeleton/stick figure extraction clearly directly provides a set of anthropometric measurements that are not corresponding to the manual ones, but can be useful for an automatic evaluation of the body size. We considered a set of 9 automatic measurements derived from the previously described analysis (see Figure 7). They are lengths or distances between nodes of the stick figure and girths computed at extracted skeletal locations, that can be related to biacromial distance, arm and forearm lengths, thigh and leg lengths, wrist, ankle, thigh and neck girths. We plan to improve in future versions the automatic processing by finding and recognizing using context information (i.e. the available mesh labeling) a set of geometrical features that can be used to estimate meaningful anatomical information.

5 Experimental Results

In our test experiments 12 young, healthy subjects (6 males and 6 females) underwent manual anthropometry according to standard procedures [23]. Landmarks were labeled with a dermographic pen and the representative set of measurements described in Section 4 was taken in duplicate with a Holtain anthropometer (Holtain Ltd, U.K.); all measurements were taken at the nearest mm by the same operator. The same landmarks were easily recognizable in the scanned

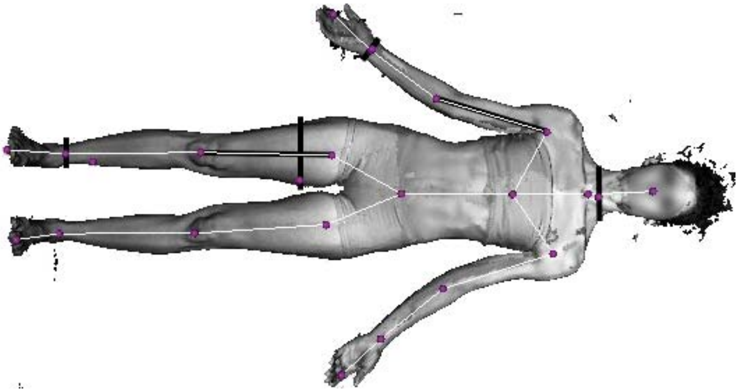


Fig. 7. Automatic measurements locations (black lines) derived by the curve skeleton/stick figure extracted (white lines)

subject and were used for digital measurement. All digital measurements were performed by the same operator.

Figure 8 shows a plot representing average absolute values of the difference between manual and simulated measurements with associated standard deviation for the set of measurements described in Section 4.2. It is possible to see that some of the values are very close and with low variance (calf, ankle, foot, biacromial), while other present more relevant differences (head, neck, chest), mainly due to the fact that the manual measurements actually change the body shape due to tissue deformation. We need, however, to perform tests on a huge number of subjects and compare the difference with inter-operator differences in manual and computer assisted procedures in order to have a clear indication of which are the measurement that are comparable and which are different and why.

Fully automated measurements did not appear completely reliable. The comparison of the most similar manual measurements reveals larger errors: this is due to the fact that in some cases the location of the implicitly extracted anatomical landmarks (endpoints of the stick figure) is not perfectly accurate, so a manual correction of these outliers should be provided. We plan therefore to include surface information in the optimization procedure in order to locate more accurately these points.

It is however interesting the fact that automatical measurements derived by our pipeline are pose invariant. To test this fact we estimated the parameters extracted on the same subject in 4 different poses (see Figure 9).

Girths computed on the limbs have a low variance ($< 4\%$), lengths are more problematic, due to the limited accuracy in the segment endpoint location in some cases. We plan to remove these outliers with improved algorithms.

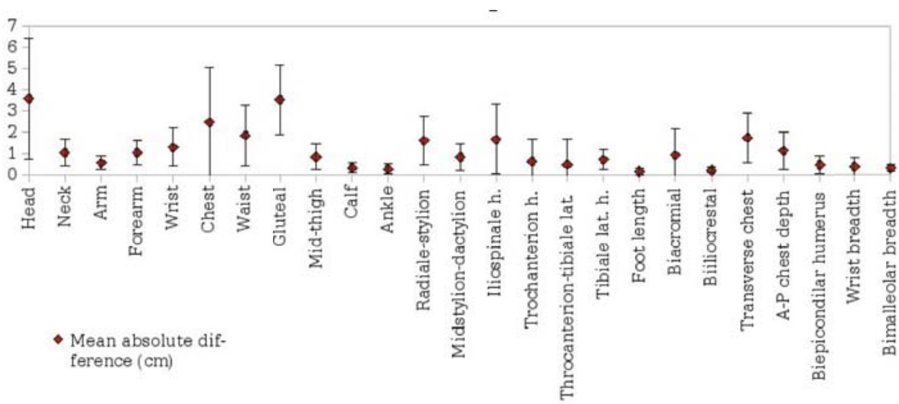


Fig. 8. Average absolute difference between manual and simulated measurements. Error bars represent the standard deviation of the difference While some measurements are almost perfectly matching, others presents a non negligible systematic error probably due to difficulty in the manual procedure (e.g. incorrect axis orientation in girths) or body deformation when touching the subject.

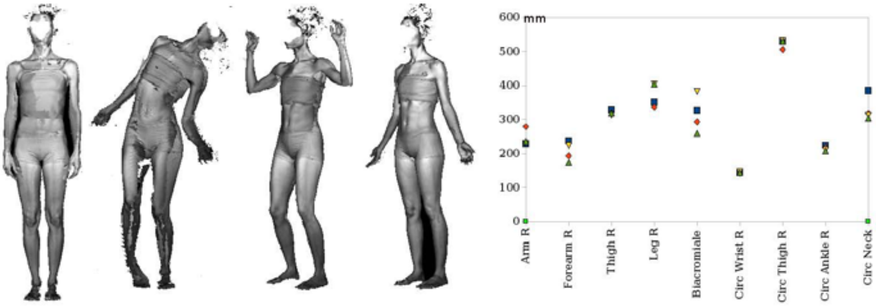


Fig. 9. Left: Same subject in different poses. Right: automatically extracted parameters for the different meshes.

6 Discussion

We presented a tool for the computer assisted evaluation of anthropometric parameters after a body scanning performed with a structured light-based device. Main features of our system are: the use of a pre processing pipeline able to estimate the pose and labeling body parts, the use of a simple user interaction to select and modify landmarks, the use of a smart algorithm to estimate girths from geodesic paths connecting landmarks and the automatic estimate of a series of parameters from the mesh processing.

Preliminary experiments performed on a set of subjects demonstrated that the computer assisted simulation of most measurements can be considered a good approximation of the true values, we need, however, to perform more experiments in order to have a complete validation of the measurement system and a more accurate analysis on the shape modifications introduced by the manual procedures. Other studies, in fact, reveal non negligible differences between manual and automatic measurements: in [14] using a $[TC]^2$ scanner the average mean difference between manual measurement and automatically located photonic measurement was 1.3 ± 4.0 cm and 5.7 ± 2.6 cm for waist and hip, respectively (on 1800 subjects).

We plan to complete the measurement tool with a larger and more accurate set of automatically extracted parameters, based on the search of surface landmarks driven by the rough pose estimation and mesh segmentation already developed.

Acknowledgements

This work is partially supported by the EU Marie Curie Program under the 3D ANATOMICAL HUMAN project (MRTN-CT-2006-035763) and by the University of Verona (Joint Project AnthropoScan).

References

1. Breuckmann gmbh web site, `http://www.breuckmann.com` (last visit on 07/2009)
2. Cyberware inc. web site, `http://www.cyberware.com` (last visit on 07/2009)
3. Hamamatsu photonics web site, `http://www.hamamatsu.com` (last visit on 07/2009)
4. Human solutions web site, `http://www.human-solutions.com` (last visit on 07/2009)
5. Inspeck inc. web site, `http://www.inspeck.com` (last visit on 07/2009)
6. Textile and clothing technology corporation web site, `http://www.tc2.com` (last visit on 07/2009)
7. Visualization toolkit web site, `http://www.vtk.org` (last visit on 07/2009)
8. Vitronic inc. web site, `http://www.vitronic.de` (last visit on 07/2009)
9. Azouz, Z.B., Shu, C., Mantel, A.: Automatic locating of anthropometric landmarks on 3d human models. In: 3DPVT 2006: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT 2006), Washington, DC, USA, pp. 750–757. IEEE Computer Society Press, Los Alamitos (2006)
10. Burnsides, D., Boehmer, M., Robinette, K.: 3-d landmark detection and identification in the caesar project. International Conference on 3D Digital Imaging and Modeling, 393 (2001)
11. Hasler, N., Stoll, C., Sunkel, M., Rosenhahn, B., Seidel, H.-P.: A statistical model of human pose and body shape. In: Dutr'e, P., Stamminger, M. (eds.) Computer Graphics Forum (Proc. Eurographics 2008), Munich, Germany, March 2009, vol. 2 (2009)
12. Heuberger, R., Domina, T., MacGillivray, M.: Body scanning as a new anthropometric measurement tool for health-risk assessment. International Journal of Consumer Studies 32, 34–40 (2008)
13. Wang, J., Gallagher, D., Thornton, J.C., Yu, W., Horlick, M., Pi-Sunyer, F.X.: Validation of a 3-dimensional photonic scanner for the measurement of body volumes, dimensions and percentage body fat. Am. J. Clin. Nutr., 809–816 (2006)

14. Wells, J.C., Cole, T.J., Bruner, D., Treleaven, P.: Body shape in american and british adults: between-country and inter-ethnic comparisons. Int. J. Obes., 152–159 (2008)
15. Lin, J.D., Chiou, W.K., Weng, H.F., Fang, J.T., Liu, T.H.: Application of three-dimensional body scanner: observation of prevalence of metabolic syndrome. Clin. Nutr., 1313–1323 (2004)
16. Ju, T.: Robust repair of polygonal models. ACM Trans. Graph. 23(3), 888–895 (2004)
17. Leong, I.-F., Fang, J.-J., Tsai, M.-J.: Automatic body feature extraction from a marker-less scanned human body. Comput. Aided. Des. 39(7), 568–582 (2007)
18. Lovato, C., Castellani, U., Giachetti, A.: Automatic segmentation of scanned human body using curve skeleton analysis. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2009. LNCS, vol. 5496, pp. 34–45. Springer, Heidelberg (2009)
19. Pargas, R.: Automatic measurement extraction for apparel from a three-dimensional body scan. Optics and Lasers in Engineering 28, 157–172 (1997)
20. Treleaven, P.C.: Sizing us up. IEEE Spectrum, 28–31 (2004)
21. Robinette, K.M., Daanen, H.: The caesar project: A 3-d surface anthropometry survey. In: International Conference on 3D Digital Imaging and Modeling, p. 380 (1999)
22. Simmons, K.P.: Body measurement techniques: a comparison of three-dimensional scanning and physical anthropometric methods. Ph.D dissertation, North Carolina State University (2001)
23. Lohman, T.G., Roche, A.F., Martorell, R. (eds.): Anthropometric Standardization Reference Manual. Human Champaign, Human Kinetics Books (1988)

A Physiological Torso Model for Realistic Breathing Simulation

Remco C. Veltkamp and Berry Piest

Dept. Computer Science, Utrecht University
`Remco.Veltkamp@cs.uu.nl`

Abstract. For the convincing modeling of virtual humans, realistic breathing is an important aspect. This paper is about the simulation of breathing, based on anatomical and physiological principles. We have built a torso, including a thorax, a deformable belly, and muscles. This results in a model with two independent breathing systems, namely abdominal and chest breathing, which gives a high degree of realism in simulating breathing. The generated spirograms, diagrams of lung volume changes, are comparable to normal human ones according to a medical expert. The generated videos show convincing breathing sequences at various frequencies. The model itself is made available for public use.

1 Introduction

That non-verbal communication the most important aspect of communication is, is a statement dating back to 1972 [6]. While verbal communication is based on spoken words, the non-verbal communication is based on body language: pose, gestures, facial expressions, and breathing, as well as a number of non-visual aspects.

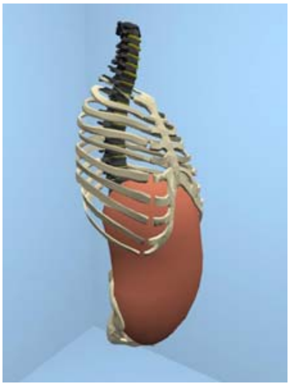


Fig. 1. Frame from the breathing animation

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 84–94, 2009.

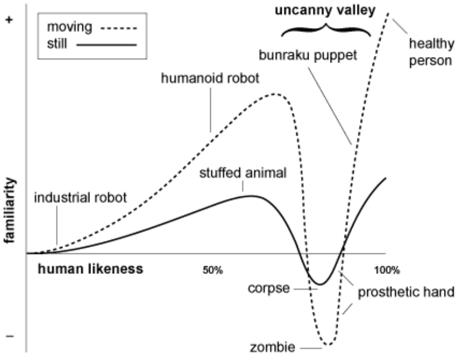


Fig. 2. The Uncanny Valley [9]

In virtual reality, gaming, simulation, animation, and medical applications, the modeling and animation of the human torso received less attention than many other aspects of character modeling such as lip synchronization, eye movement, and facial expressions. The game engine Source, developed by Valve Corporation, and used in games such as Half Life 2, is an example of a game engine that support such aspects. Using the FacePoser SDK tool, it synthesizes an animation of a spoken sentence, including lip movement, facial expressions, and head movement. Yet, the movement of the chest, caused by breathing is not well developed and supported. Figure 1 shows one frame from our video of animated chest movement.

The human visual system is very good in making a distinction between natural and unnatural animations, especially of humans. In fact, when a character is modeled that has a high degree of realism, but is not completely realistic, it is considered robot like, rather than human. This phenomenon is called the 'Uncanny Valley' [9], see figure 2. It describes the feeling people have when seeing a robot. Initially, a more human like robot is considered more pleasant. However, when the robot gets close to realistically human, people find the robot scaring and repelling. Suddenly, the little discrepancy between almost and fully realistic gives a strong negative feeling. In contrast, animated toys and monsters are not considered unrealistic, such as in movies like Toy Story, Monsters & Co., and The Incredibles.

1.1 Related Work

In the past, characters that didn't do anything specifically, were put on hold during the animation, yielding a frozen pose. This is very unrealistic, so subtle arbitrary movements of arms and body were introduced [4].

A lot of research has already been done on the simulation of muscles, although more on the deformation than on the way they exhibit force. Muscles are often modeled as mass-spring systems [10] or finite element systems [3]. Alternatively, [13] models muscles only those structures that have a visual effect on the appearance of the body, which is more similar to an artistic approach than a physiological approach.

Simulation of the abdominal breathing is done by [11]. It is done by constraint resolution on a physical model. Accuracy of elastic spring meshes was researched by [5]; the conclusion was that an exact solution is not possible. Modeling an incompressible free form solid was performed by [12].

Breathe Easy [16] is an anatomy based physiological model of the human torso built from rigid and deformable parts. The simulation is done using ODE, the Open Dynamics Engine [14]. They simulate four frequencies of breathing: slow deep breathing, calm breathing, normal breathing, and panting. The spirograms shown in [16] are not realistic according to a lung physiologist at the Utrecht Medical Centre [15]. For example the shown tidal volume of 60 ml at a frequency of 15 normal breathings per minute, is far too small for a normal person. Their accompanying video [2] shows little if any abdominal breathing, which is unrealistic, because abdominal breathing is the most efficient way of breathing.

Different types of breathing model are developed in the medical image domain, for medical applications such as multi modal image registration, see for example [8].

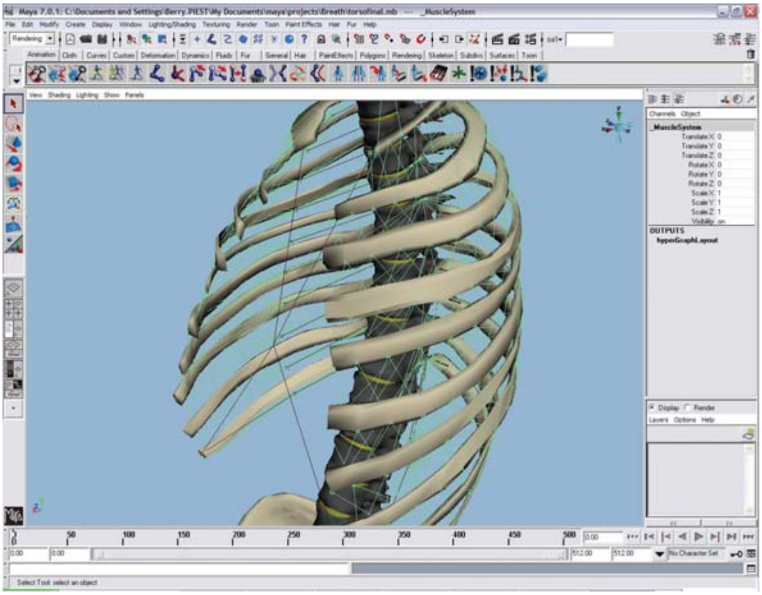
1.2 Contribution

Because breathing is an essential distinction between humans and robots, a realistic modeling of breathing is an important aspect of making animation of virtual humans convincing. In this paper we present a torso model and breathing simulation, based on anatomical and physiological principles. We have built a torso, including a thorax, a deformable belly, and muscles. In contrast to previous work, our model has two independent breathing systems, namely abdominal and chest breathing, which gives a high degree of realism in simulating breathing. The model, built in Alias Maya and Maya Embedded Language scripts, is made availabe for public use.

The generated spirograms, diagrams of lung volume changes, are comparable to normal human ones according to a medical expert. The generated videos show convincing breathing sequences at various frequencies. See figure 1 for a frame from one of the videos of animations that accompany this paper.

2 A Physiological Torso Model

In order to alleviate the shortcomings of previous models, we have designed a new model that better combines the abdominal and chest breathing, resulting in more natural spirograms and animations. We have built our model with Alias Maya and MEL (Maya Embedded Language) scripting. As an initial basis we used a torso from 3D Cafe [1]. We have divided this polygonal model into its constituent parts: ribs, dorsal vertebra, hip, shoulder blade, and collarbone, which are made rigid, so that constraints can be applied to them. We have assigned physiology based masses to these objects as provided by Zordan et al. [16]. Also, the model



_MuscleSystem
Translate X 0
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Visibility on

Fig. 3. Modeling the intercostal muscles between the ribs (green lines)

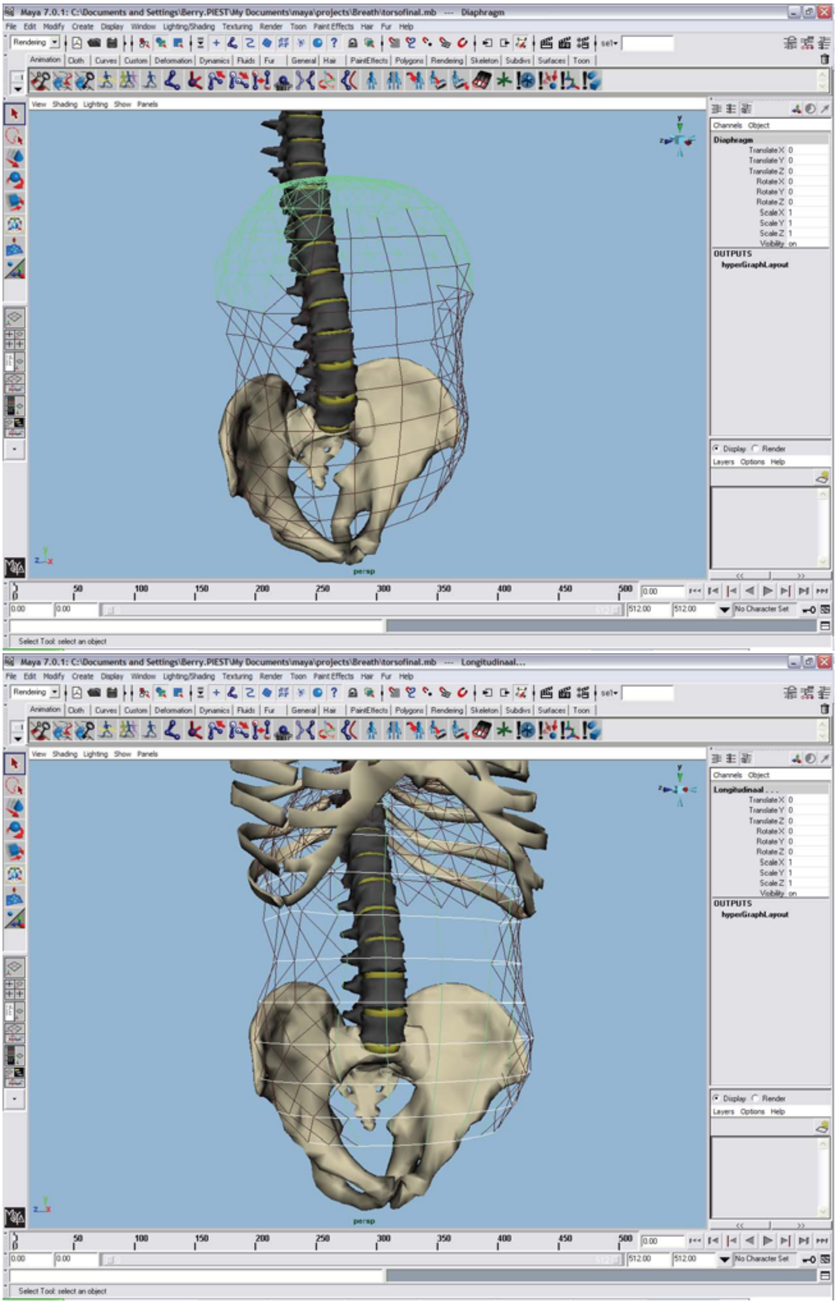
is scaled to give it a realistic size, i.e. the size of one the authors (a sternum of 220 mm, measured by a physiotherapist).

2.1 Thorax

The ribs must be able to move, and should not be fixed to the spine. Since during the breathing the ribs move up and sidewards, we don't model the joint as a ball and socket, but as two hinges. The intercostal muscles are modeled anatomically correct as follows. Since Maya springs can only connect to the center of a rigid object, boxes are attached to polygon vertices with a Pin Constraint, on top of the rib polygon. Between boxes of adjacent ribs, strings are placed crossways, see figure 3. The first rib is also attached to cervical vertebra C6. This muscle helps lifting the ribs.

2.2 Abdominal Cavity

The abdominal cavity is modeled as one volume, following the shape of the belly. At all times the belly tries to maintain the same volume. If the diaphragm is pushed, the front of the belly will expand. If the pressure is released the belly will return to its original state, because the belly muscles will relax. The surface is polygonal, the volume consist of a number of tetrahedra, equal to the number of triangles. A central point inside the cavity is the common top of all tetrahedra, the base of these tetrahedra are the triangles of the polygonal surface. The bottom and back side of the abdonimal cavity are passive, the front (belly) and



Maya 7.0.1: C:\Documents and Settings\Berry.PIEST\My Documents\maya\projects\Breath\torsofinal.mb --- Diaphragm

Diaphragm
Translate X 0
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Visibility on

OUTPUTS
hyperGraphLayout

Select Tool: select an object

Maya 7.0.1: C:\Documents and Settings\Berry.PIEST\My Documents\maya\projects\Breath\torsofinal.mb --- Longitudinaal...

Longitudinaal...
Translate X 0
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Visibility on

OUTPUTS
hyperGraphLayout

Select Tool: select an object

Fig. 4. Abdominal cavity. Top: diphragm in green. Bottom: longitudinal (green) and transverse (white) muscles. (The obliques are not shown here).

top (diaphragm) are active. A mass of 8 kilos (cf. [16]) is divided over the active vertices. To these vertices, boxes and mass-springs are attached. The force F applied by these springs are proportional to the pressure P times the area A of each triangle:

$$F = n(P \cdot A)/3, \tag{1}$$

where n is the direction vector of the force perpendicular to the triangle, the division by 3 accounts for the three vertices, and the pressure is determined by Hooke's law:

$$P \propto (V_0/V - 1), \tag{2}$$

where V_0 is the original volume, and V the current volume. For each spring the length in rest is determined; the initial distance between the vertices is set to this length, to make the model go back to its original shape.

The diaphragm is a curved plate, where muscles radiate from the center. A relaxed diaphragm rests on top of the bowels. When the diaphragm contracts, it becomes flatter, pushing away (forward) the bowels, enlarging the chest cavity, giving more room for the lungs.

The diaphragm is modeled as the top of the polygonal surface of the abdominal cavity. In order to make the diaphragm move like a plate, vertices in a square layout are connected by side and diagonal strings, see figure 4 (top). The abdominals, the belly muscles, are modeled anatomically correct, so the strings follow the direction of the actual muscles: longitudinal, transverse, and oblique, see figure 4 (bottom).

These muscles work autonomously, trying to preserve the pressure in the abdominal cavity. Another force is needed to stretch them. During simulations, for each active vertex a force on the mass point is calculated with formula (1).

2.3 Rythmogenesis

We have grouped the muscles into groups, so that they don't have to be steered individually. For a normal respiration, we use a timed vital capacity, the volume of breath per minute, of 6 litres per minute. This can be achieved by a breathing frequency of 12 times per minutes, and a tidal volume of 500 ml. This means that per breath, which lasts for 5 seconds, 500 ml is refreshed. For an animation of 40 frames per second, a full breath takes 200 frames.

Some muscles, such as the abdominals, are elastic, and need some time to return to a rest situation. For both abdominal and chest breathing we need a periodic function that are synchronized. We have used sinusoidal functions. When the abdominal breathing sine function is larger than zero, we let the diaphragm contract. The contraction duration leaves sufficient time to return to a rest situation.

To model the chest breathing, three group of muscles must be controlled, the outer intercostals, the inner intercostals, and the longitudinal muscles. During inhalation, the outer intercostals contract, so that the ribs lift. Meanwhile, the diaphragm contracts, so that the pressure in the belly increases, and the belly moves forward. As a result, the abdominal cavity volume increases, giving an

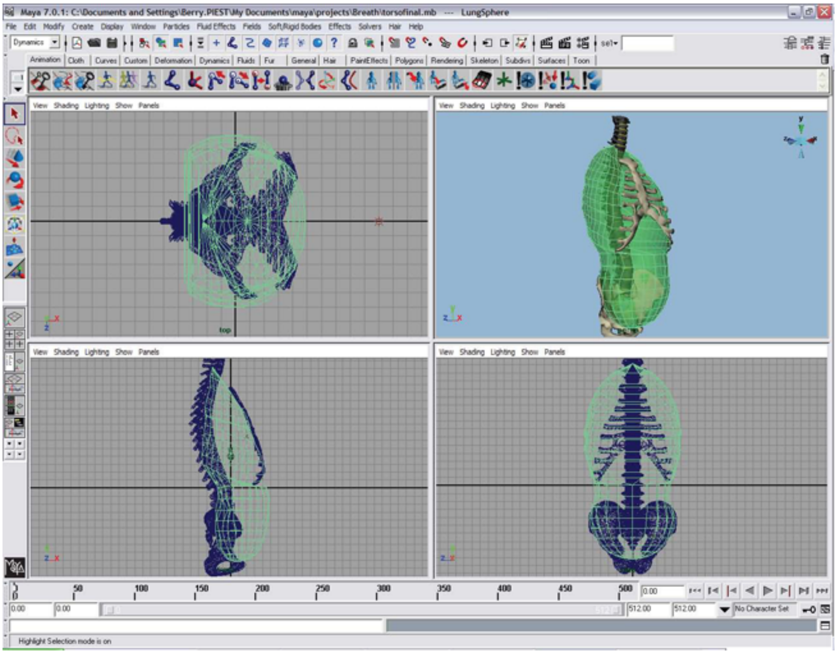


Fig. 5. The torso volume

under-pressure in the lungs, leading to an inflow of fresh air. While breathing out, the tension in the outer intercostals is released, the inner intercostals slowly contract, and the diaphragm relaxes, so that the bellow moves back, and the air is pushed out of the lungs. During the activation, a new length of each muscle is calculated as the initial length times the value of the sine function times the fraction the muscle can shorten. The derivative of the sine function also controls the speed of contraction.

2.4 Volumes

The volume of the abdominal cavity is the total volume of its constituent tetrahedra. Similarly, the volume of the whole torso is the total volume of its constituent tetrahedra, see figure 5. The lung volume is then the torso volume minus the volume of the abdominal cavity.

3 Breathing Simulation

We have simulated different types of breathing at a number of frequencies: abdominal breathing at 12 times per minutes, and combined abdominal and chest breathing at 12, 15, 20, and 24 times per minute. Figures 6, 7, and 8 each show 3 curves in a single graphic: the height of the sternum (blue) to illustrate the

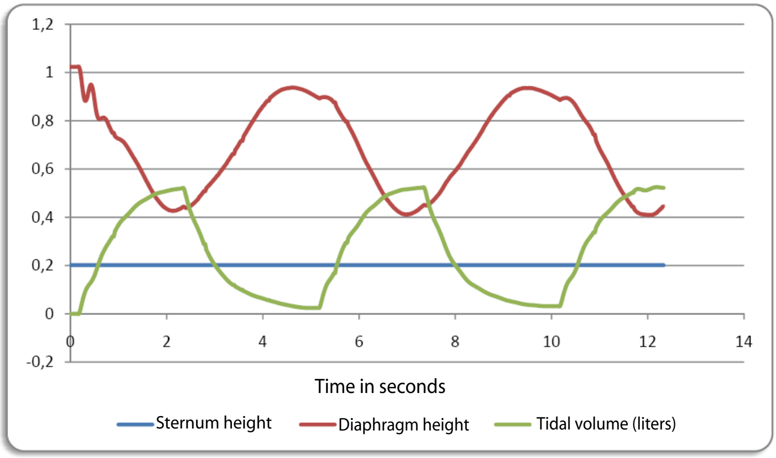


Fig. 6. Slow breathing, 12 times per minute, abdominal breathing only

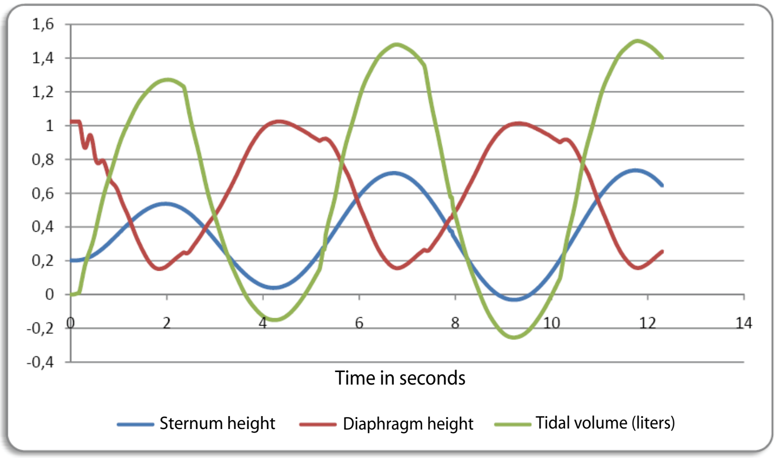
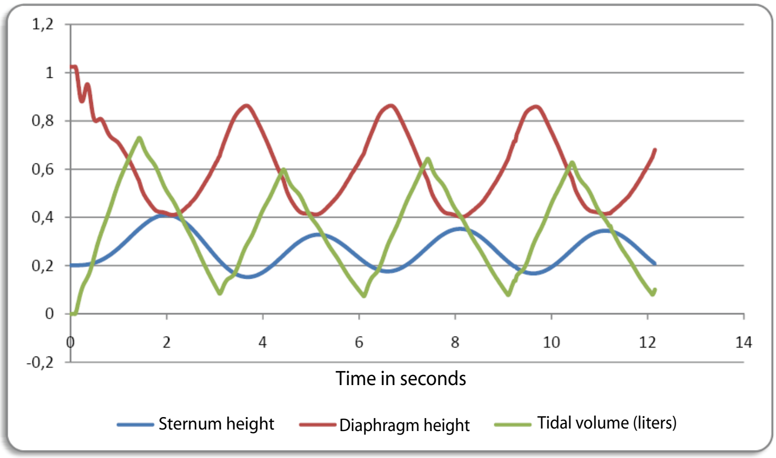


Fig. 7. Slow breathing, 12 times per minute, combined abdominal and chest breathing

moving of the thorax, the height of the diaphragm (red), and the tidal volume (green). The vertical axis is multifunctional, for the blue and red lines it is the height of the sternum and diaphragm in the model coordinates, the green one is the breath volume which is relative to the initial torso volume, which is in liters.

Figures 6 and 7 show spirograms of slow breathing, at a frequency of 12 times per minute. In the case of abdominal breathing (figure 6), the thorax remains still, and the sternum is not moving. The tidal volume is about 500 ml, which results in a timed vital capacity, the volume of breath per minute, of about 6 litres. In combination with chest breathing (figure 7), the sternum is also moving. The resulting slow, deep breathing gives a much higher timed vital capacity of



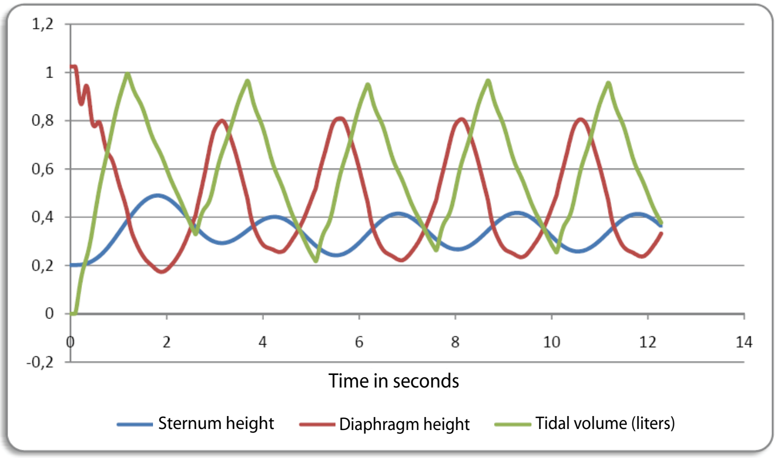


Fig. 8. Fast breathing. Above: 20 times per minute, below: 24 times per minute.

21 litres. It is clear that the tidal volume is minimal when the diaphragm is high and the sternum is low, and the tidal volume is maximal when the sternum is high and the diaphragm is low.

Figure 8 shows the spirograms for higher frequencies: 20 and 24 times per minute. Because of the higher frequency, the sternum does not move as much as with slow breathing, the blue line has a smaller amplitude than in figures 6 and 7. As a result, the timed vital capacity is 11 and 17 litres respectively. Because the torso is actively inhaling and exhaling, the muscles are actively used instead of passively. In slow breathing the thorax returns to the initial state passively, which uses the muscles' spring like effect for returning to its rest lenght. For fast

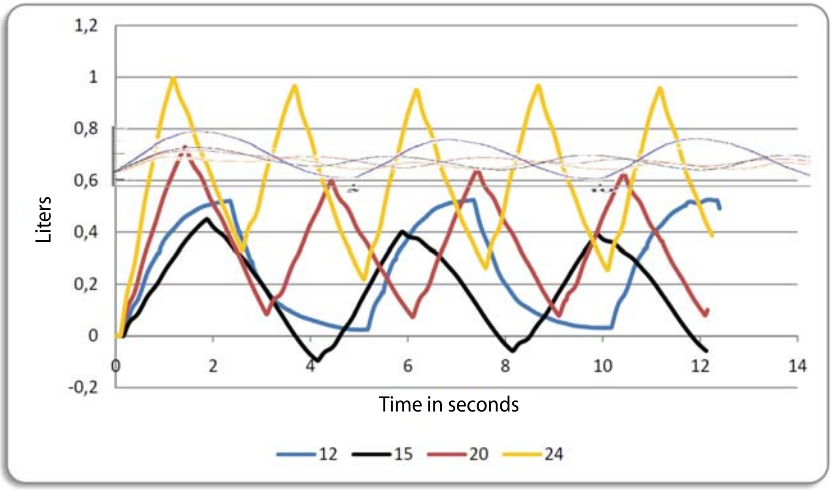


Fig. 9. Comparing our tidal volumes (large amplitudes) with other results (small amplitudes) at varying frequencies: 12, 15, 20, and 24 times per minute

breathing, the return to the initial state of the torso uses force to help returning to that state, which is why the graphs look triangular.

Looking at the physiology literature, these figures are realistic [7]. In order to make a direct comparison with [16], we overlay their results with our simulated tidal volume at different breathing frequencies: 12, 15, 20, and 24 times per minute, see figure 9. While the tidal volumes from [16] lie in the range 0.6-0.8 litres, our volumes have a much more realistic range for a normal person [7]. Also the curve shape is more realistic.

For animations of the simulated breathings, see the videos at `http://www.cs.uu.nl/groups/MG/gallery/Breathing/`

4 Concluding Remarks

Our model is more realistic than previous models, because the breathing in rest is based on abdominal breathing only, and at slightly higher frequencies on a combination of abdominal and chest breathing. Even though there is much variability between individuals, figure 9 shows that the tidal volumes of our model is much more realistic than previous models. The big advantage of our system is that abdominal and chest breathing are controlled independently. As a result, our spirogram values are realistic for normal persons.

We have modeled the torso in Maya, it can be used in any other Maya environment. The complete model is available at `http://www.cs.uu.nl/groups/MG/gallery/Breathing/`, and can be used by anyone for further modeling realistic avatars. An even more precise model would allow to simulate other actions such as blowing, whistling, and speaking.

Acknowledgment

This work is partially supported by the the FP7 project FOCUS-K3D 2007-214993 and the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

References

1. 3D Cafe, `http://www.3Dcafe.com`
2. Breathe Easy videos, `http://graphics.cs.ucr.edu/projects/simulatedBreathing/simulatedBreathing.html`
3. Chen, D., Zeltzer, D.: Pump it up: computer animation of a biomechanically based model of muscle using the finite element method. In: Proceedings of SIGGRAPH, pp. 89–98 (1992)
4. Egges, A., Giacomo, T.D., Thalmann, N.M.: Synthesis of realistic idle motion for interactive characters. In: Dickheiser, M. (ed.) Game Programming Gems, vol. 6 (2006)
5. Gelder, A.V.: Approximate simulation of elastic membranes by triangulated spring meshes. Journal of Graphics Tools 3, 21–42 (1998)
6. Mehrabian, A.: Nonverbal Communication. Aldine Atherton (1972)
7. Mines, A.H.: Respiratory Physiology. Raven Press (1993)
8. Moreno, A., Chambon, S., Santhanam, A.P., Brocardo, R., Kupelian, P., Rolland, J.P., Angelini, E., Bloch, I.: Thoracic ct-pet registration using a 3d breathing model. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 626–633. Springer, Heidelberg (2007)
9. Mori, M.: Energy. Translated by Karl F. MacDorman and Takashi Minato 7(4), 33–35 (1970)
10. Nedel, L., Thalmann, D.: Real time muscle deformations using mass-spring systems. In: Computer Graphics International (1998)
11. Promayon, E., Baconnier, P., Puech, C.: Physically-based model for simulating the human trunk respiration movements. In: Troccaz, J., Mösges, R., Grimson, W.E.L. (eds.) CVRMed-MRCAS 1997, CVRMed 1997, and MRCAS 1997. LNCS, vol. 1205, pp. 379–388. Springer, Heidelberg (1997)
12. Rappaport, A., Sheffer, A., Bercovier, M.: Volume-preserving free-form solids. IEEE Trans. Vis. Comput. Graph. 2, 19–27 (1996)
13. Scheepers, F., Parent, R., Carlson, W., May, S.: Anatomy-based modeling of the human musculature. In: Proceedings SIGGRAPH, pp. 163–172 (1997)
14. Smith, R.: Open Dynamics Engine (2006), `http://ode.org`
15. Zanen, M.D.P.: Lung physiologist Utrecht Medical Center, personal communication
16. Zordan, V.B., Celly, B., Chiu, B., DiLorenzo, P.D.: Breathe Easy: Model and control of simulated respiration for computer animation. Graphical Models 68, 113–132 (2006)

Evaluating the Impact of Shape on Finite Element Simulations in a Medical Context

Lars Walczak[1], Frank Weichert[1,*], Andreas Schröder[2], Constantin Landes[3],
Heinrich Müller[1], and Mathias Wagner[4]

[1] Department of Computer Science VII, Dortmund University of Technology, Germany
frank.weichert@tu-dortmund.de
[2] Department of Mathematics, Humboldt-Universität zu Berlin, Germany
[3] Maxillofacial and Facial Plastic Surgery, Johann Wolfgang Goethe-University Medical Center, Frankfurt, Germany
[4] Department of Pathology, University of Saarland Medical School, Homburg Saar, Germany

Abstract. Competing concepts exist regarding surgery for instance of the cleft lip and palate to date. In order to support the surgeon to predict the possible outcome of a variety of the approaches a promising procedure are morphology-based finite element simulations at histological scale. It however can be a challenge to generate volume meshes that are applicable to the mathematical modeling of three-dimensional spatial modifications. In this study we discuss the variation of the segmentations by different anatomy experts with respect to shape, analyze the associated reconstructions by the finite element method and compare them among one another. The gist of the study is that an exact segmentation is fundamental precedent for a simulation and minor deviations in shape may arise deviations in a finite element simulation.

Keywords: Finite Element Simulation, Impact of Shape, Cleft Lip and Palate, Mesh Generation, Systems Biology.

1 Introduction

Recent developments in computer-based simulation methods which are used within medical practice, e.g. surgical planning, usually rely on the combination of detailed anatomy information based on segmented images, the three-dimensional reconstruction of the region of interest and the numerical simulation of the deformation behavior. This study discusses the impact of interobserver segmentation variability, exemplified by a viscerocranial striated skeletal muscle, musculus orbicularis oris (syn.: MOO). Cleft lip and palate constitutes a birth defect involving disruptions in multiple developmental steps such as growth, differentiation, elevation, and fusion. The resulting deformity may differ from its non-clefted counterpart in skeletal muscle fiber distribution, blood perfusion and

---

[*] Corresponding author.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 95–109, 2009.

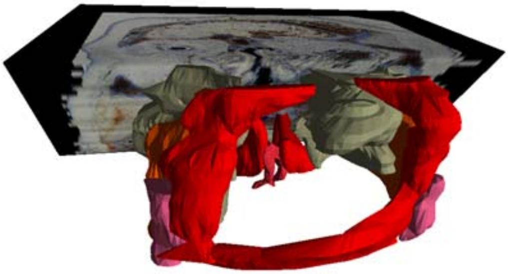


Fig. 1. 3D-representation of segmented and unsegmented structures in the soft palate. A local interpolation algorithm supporting the three-dimensional resampling of images of histological serial sections has been used to obtain a 3D image stack. Direct volume rendering techniques were then applied to generate a 3D voxel model with a superimposed clipping box allowing to visualize segmented structures (front) along with the stack (back). Anatomical structures can be distinguished by color.

nerve supply [1,2]. Micro-morphology has therefore considerable influence on any corrective concept. Contemporary operative lip plasty protocols however, tend to neglect a possible micro-morphologically correct modeling of muscular structures in favor of detailed skin reconstruction [3]. This may permit symmetry at rest but distortion and marked asymmetry upon lip movement. Micro-anatomical workup of more cases is required to clarify individual variation. Computer-assisted mathematical modeling may be applied to analyze force distribution and the resulting vectors of deformation upon striated muscular activity in both, a physiological and a clefted scenario (cf. figure 1). Postoperative results of different reconstruction methods can be compared with unaltered force interaction and vector distributional scenarios. A more physiological postoperative force distribution in silico may therefore enable the surgeon to set stages for a more harmonic lip movement in vivo. Moreover, the extent and necessity of exact muscular velopharyngeal reconstruction for physiological force distribution shall be found out and also to what extent surgical velar lengthening is necessary for normal velopharyngeal closure. Despite the concept that intravelar veloplasty should be carried out according to what is seen in non-clefted anatomy [4], there are reconstruction methods that render the patients with rather non-physiological sites [5]. The physiological consequences of each method have not been sufficiently clarified as the body of literature provides conflicting statements.

The orbicularis internus muscle of the lips represents circular striatal skeletal musculature. The number of MOO fiber insertions into the skin can be postulated to modify the level of the labial skin and therefore lip thickness bilaterally [6]. This is in part thanks to the fact that fibers of the MOO enter the upper lip from one side and decussate in the midline to splay out and insert into contralateral parts of the skin situated lateral to the philtral groove which is corresponding to the more compact median decussation of the fibers. Lip thickness is reduced by the relative absence of MOO fiber insertions into the skin which is coherent with the finding that the medial borders of the muscle insertion zones are found in the philtral ridges. It is noteworthy that the anterior projection of the pars marginalis component of the MOO constitutes the link between cutis and vermilion as hypoplasia and disorientation of MOO fibers have been seen in cases of cleft lip (International Statistical Classification of Diseases and Related Health Problems, 10th Revision, ICD-10: Q36.0, Q36.1, and Q36.9) [7].

Bearing this in mind, we ask whether a computer representation (a mesh) of the MOO (or any other muscle model at that instance) has to reflect the anatomical observations in a finite element simulation and to which degree. For an overview of existing mesh generation methods we refer to survey papers and references therein [8] or to the monographs [9,10]. Speaking of the finite element method, an important and often underestimated component is the construction of suitable meshes, which admit high resolutions of the geometry, reflect to topology as well as lead to stable numerical schemes. Here, we vary segmentations of the MOO by different investigators (i.e., interobserver variability) and by Fourier descriptors and analyze the associated reconstructions by the finite element method. The results are compared with those obtained from an idealized model of the muscle, a torus.

As the first main part of this study part 2 briefly covers general problems encountered while obtaining the source material and the segmentation. A method for constructing volume meshes based on level set approximations of the stacked polygonal segments of the regions of interest is presented in sections 3.1 and 3.2. We introduce this modeling within a general mathematical framework of energy minimization and non-linear variational problems (section 3.3). The finite element method is included in this framework (section 3.4). Numerical results are discussed with respect to the shape of the generated meshes in the finite element approach (section 4).

2 Materials

All calculations were performed on abstract representations of typical viscerocranial striated skeletal muscles of a 20^{th} century cadaveric male or female homo sapiens sapiens embryo or fetus of Caucasian origin and an average Carnegie Stage beyond 20. Details on the preparatory basis of the current study have been previously published by the authors [11]. In brief, digitalized images of serial sections of prenatal viscerocrania underwent manual segmentation and subsequent 3D reconstruction of the region of interest. All layers that contained structures of interest which could not be marked accurately (e.g., due to artifacts) were excluded from the data set.

We exemplarily reconstructed the MOO in case without cleft malformation to assess the impact of interobserver segmentation variability on the results of 3D reconstruction and finite element simulation. Manual and interactive segmentation by placing control points on the boundaries of the objects of interest may result in highly complex polygons that are hard to reconstruct. But is this level of detail necessary or not? To model this aspect, the polygons were smoothed by removing high frequencies using Fourier descriptors [12].

3 Methods

Based on the given segmented data sets the sections 3.1 and 3.2 describe the mesh generation process, sections 3.3 and 3.4 define the material model and derive the numerical approximation with finite elements.

3.1 Level Set Approximation

3D reconstruction requires a mesh which must approximate the template's geometry and preserve topology. The matched contours of all adjacent cross sections have to be connected to a surface mesh. This becomes challenging if the contours have complicated geometries or if a contour is connected to more than one contour on the adjacent cross section. Many Algorithms fail to tile complex geometries and cannot handle branches in the segmentation [13]. Our approach addresses both problems. The manually segmented contours are initially approximated using a level set.

To provide a suitable finite element conformal tetrahedral mesh on the basis of polygonal segments the level set approach is used to construct an implicit surface on a rectilinear grid [14]. This implicit representation is used to guide an octree subdivision process (see section 3.2). The surface is determined by time evolving an initial guess through a physically motivated convection model that is described by the following level set equation:

$$\frac{\partial \phi}{\partial t} = \nabla d(\boldsymbol{x}) \cdot \nabla \phi. \tag{1}$$

The zero level set of the signed distance function $\phi : \mathbb{R}^3 \mapsto \mathbb{R}$ represents the desired isocontour while $d(\boldsymbol{x})$ is an unsigned distance function that is calculated from the polygonal segments with Sethian's fast marching method [15]. Calculating the distance field is the first step towards the determination of the implicit surface. In brief, this algorithm resembles Dijkstra's method for calculating shortest paths in graphs and orders grid node distances by adding them to a heap. The initial heap entries are identified as the grid nodes surrounding the polygonal segments. Here the exact distances to the nodes are calculated, the other grid node distances are initialized with infinity. The grid node (i, j, k) with the smallest distance (the heap top) and its direct neighbors $(i \pm 1, j \pm 1, k \pm 1)$ are examined. The distances of the neighbors are updated if the distance value stored in the node is larger than the distance of the heap top and the distance between the grid nodes added together. The heap top is taken out of the heap and all updated neighbors are added to the heap. The algorithm continues with the new heap top and terminates in case the heap is empty.

In the convection model (equation 1), d serves as a potential field that attracts the contour. The initial estimate that is necessary to speed up the calculation is constructed by marching a bounding volume of the structure on the distance contours of d inward towards the polygons. This represents the second step towards finding the contour and is implemented with a fast tagging algorithm [14] that marks every gridnode as *external*, *internal* or *boundary*. At the point where the distance contours break open the marching stops, these grid nodes are tagged as *boundary* whereas previously visited grid nodes are tagged *external*. All other nodes are marked *internal*. A second run of the fast marching method starting with the *boundary* points constructs the initial guess for ϕ which is an approximate offset to the real surface. All distance values of *internal* nodes are negated. In turn, as third and final step, the preliminary surface is evolved with the level set equation (1). For solving equation (1) the spacial discretization is calculated with the third order accurate Hamilton-Jacobi ENO upwind scheme [15], for time discretization we use Euler's method.

3.2 Mesh Generation and Deformation

The resulting implicit surface ϕ is used to adaptively refine an octree structure in multiple iterations. The octree consists of what is commonly referred to as bcc tetrahedra (body-centered cubic, a crystal structure) that posses favorable properties such as subdivision invariance. In addition their quality measures [16] resemble equilaterals [17]. The refinement process iterates through all tetrahedra and samples the level set to examine whether a portion of the zero isocontour is covered which can be easily decided by examining the signs of the sample points. If a sign change occurs the tetrahedron is regularly refined, i.e. the tetrahedra is replaced by eight similar tetrahedra. This produces so-called hanging nodes which are handled in our finite element system by introducing additional constraints. Tetrahedra with all sampling values greater 0 are located outside the implicit surface and can be removed.

To optimize the approximation of the surface the generated mesh is pushed towards the zero isocontur $\phi(\boldsymbol{x}) = 0$ of the level set. This deformation is modeled by a mass-spring system which obeys Newton's second law of motion and Hooke's Law. Altitude springs [17] are added to the model to effectively prevent element collapse. These springs connect vertices with their opposite faces in a tetrahedron. External forces that act upon the outer volume mesh nodes of the system are generated in the following way: The force direction equals the negative mesh normal of the node, the magnitude is gained through the level set and is set to $|\phi|$. The system is integrated in time with the 4th order Runge-Kutta method. After the mesh generation ends the chosen striated muscle is reconstructed as a tetrahedral mesh with good quality measures (cf. figure 2).

3.3 Modeling by Hyperelasticity

Soft tissue can be described as an aggregation of viscoelastic, inhomogeneous objects that allow considerable deformation. Neglecting the time-dependency of the material behavior, suitable models for soft tissue are given by hyperelastic material models. One of the most commonly used approaches is given by transversely isotropic hyperelastic material models [18,19,20,21]. Hyperelastic material models are originally developed for rubber or rubber-like material. With regard to large deformation properties and (nearly) incompressibility, hyperelastic material models provide a reasonable framework for modeling of soft tissues on a macroscopic scale.

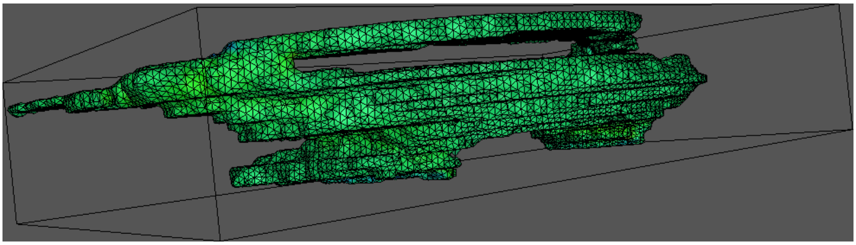


Fig. 2. Finite element mesh for the musculus orbicularis oris

The general proceeding. According to the principle of minimum potential energy, a body or a muscle is in a state of stable, static equilibrium whenever the total potential energy $\mathcal{F}$ is a minimum. This is characterized by a displacement field u fulfilling $\mathcal{F}(u) = \min_{v \in V} \mathcal{F}(v)$ where V denotes a suitable linear space of admissible displacements. The total potential energy for hyperelastic material is given by $\mathcal{F}(v) := \int_\Omega \rho W(x, \nabla v(x))\, d\Omega - \int_\Omega f v\, d\Omega - \int_{\Gamma_N} s v \cdot n\, d\Gamma$, cf. [22], [23], where $\Omega \subset \mathbb{R}^3$ describes the body of the soft tissue in an undeformed (or reference) configuration with density ρ. We imply that deformation can only be caused by a volume force f acting on Ω or by a surface force s that acts on the boundary part Γ_N with outer unit normal vector n. We assume that the volume force f describes the muscle contraction by tension in the upper and lower part of the muscle to its middle. The use of a volume force in such a way is, of course, a strong idealization of muscle contraction. A physically reasonable model for active muscle contraction can be found, e.g., in [18]. The connection to non-soft or essentially less soft tissue (e.g. bones, cartilage) is included via Dirichlet boundary conditions on a boundary part of the muscle, so that the displacements are zero on that boundary part. No boundary conditions are defined for the remaining boundary, which enforces natural (or Neumann) boundary conditions. The functional W is called the strain energy functional. The presence of x in W reflects possible inhomogeneities or anisotropies of the material. For isotropic material, W only depends on the gradient ∇v (or, more precisely, on certain invariants which are explained later).

The stationary condition of the minimization problem for the solution u reads

$$\forall v \in V: \; a(u; v) = \ell(v) \tag{2}$$

with the semilinearform $a(u; v) := \int_\Omega T_{ij}(x, \nabla u(x)) v_{i,j}\, d\Omega$, the linearform $\ell(v) := \int_\Omega f v\, d\Omega + \int_{\Gamma_1} s v \cdot n\, d\Gamma$ and $T_{ij} := \partial W / \partial v_{i,j}$ as the first Piola-Kirchhoff stress tensor. As usual Newton's method is applied to determine a solution of (2),

$$\begin{aligned} &\text{Find } \hat{u}^t \in V: \\ &\qquad \forall v \in V: \; a'(u^t; \hat{u}^t, v) = \langle \ell, v \rangle - a(u^t; v) && (3) \\ &u^{t+1} := u^t + \hat{u}^t. && (4) \end{aligned}$$

with the semi-bilinear form $a'(u;\hat{u},v) := \int_{\Omega} E_{ij\alpha\beta}(x,\nabla u)\hat{u}_{\alpha,\beta}v_{i,j}\, d\Omega$ and $E_{ij\alpha\beta} := \partial T_{ij}/\partial u_{\alpha,\beta}$. Incompressible material behavior is characterized by the condition, that the volume is not changed during the deformation process. This leads to introducing an additional constraint $J(\nabla u) := \det(F(\nabla u)) = 1$ with the deformation gradient $F_{ij} := \delta_{ij} + u_{i,j}$. Here, δ_{ij} is Kronecker's delta function.

The Mooney-Rivlin approach. Many approaches consider the soft tissue as an isotropic rubber-like material and use strain energy functionals developed for this kind of material, cf. [19] for an overview. One of the most important material models for rubber-like materials is given by the Mooney-Rivlin approach [24,25,26,27,28].

In order to capture the incompressibility constraint, a volumetric-isochoric split of the strain energy functional W is usually performed by defining $F_{\text{vol}} := J^{1/3}\mathbf{1}$ and $\tilde{F} := J^{-1/3}F$. Since there holds $\det(F_{\text{vol}}) = J$ and $\det(\tilde{F}) = 1$, only the deformation due to F_{vol} makes a contribution to the volume change. Using the right Cauchy-Green deformation tensor $\tilde{C} := \tilde{F}^{\top}\tilde{F}$ and its principal invariants $\tilde{I}_1 := \text{tr}(\tilde{C}) = \tilde{C}_{ii}$ and $\tilde{I}_2 := \frac{1}{2}(\tilde{I}_1^2 - \text{tr}(\tilde{C}^2))$ we define the volumetric-isochoric split of the strain energy functional as $W(\tilde{I}_1,\tilde{I}_2) = \kappa W_{\text{vol}}(J) + W_{\text{iso}}(\tilde{I}_1,\tilde{I}_2)$ with a bulk modulus κ, cf. [24]. The isochoric strain energy contribution in the Mooney-Rivlin approach is given by $W_{\text{iso}}(\tilde{I}_1,\tilde{I}_2) := c_1(\tilde{I}_1 - 3) + c_2(\tilde{I}_2 - 3)$ with some material constants c_1 and c_2. Reasonable definitions of the volumetric part describing the strain energy of the volume change, are $W_{\text{vol}} := \frac{1}{2}(J-1)^2$ or $W_{\text{vol}} := \frac{1}{2}(\ln(J))^2$, cf. [29].

3.4 The Finite Elemente Method

The general proceeding. Material models as described above do not yield analytical solution functions in general. For this reason, numerical methods must be applied in order to obtain an approximate solution. The finite element method (FEM) is the most commonly used scheme for numerically solving the variational equations such as (2) or (3)-(4), respectively.

In conform finite element approaches the linear space V is exchanged by a finite dimensional subspace $V_h \subset V$. Using an adequate basis $\{\varphi_i\}_{i=1}^n$ of V_h, we define $u_h^t := x_l^t\varphi_l$ and $\hat{u}_h^t := \hat{x}_l^t\varphi_l$ with $x^t, \hat{x}^t \in \mathbb{R}^n$. Newton's method based on V_h is then given by the linear equation $K(u_h^t)\hat{x}^t = L(u_h^t)$ and the update $x^{t+1} = x^t + \hat{x}^t$ with a starting vector $x^0 \in \mathbb{R}^n$, stiffness matrix $K_{kl}(u_h^t) := a'(u_h^t;\varphi_l,\varphi_k)$ and load vector $L_k(u_h^t) := \ell(\varphi_k) - a(u_h^t;\varphi_k)$. To enlarge the area of convergence Newton's method is usually modified by introducing a damping factor.

In the finite element approach, the basis $\{\varphi_i\}_{i=1}^n$ of V_h is usually constructed via a mesh $\mathcal{T}$ which is a (regular) decomposition of Ω in tetrahedrons or hexahedrons. The subscript h indicates the maximum diameter of the mesh elements. Piecewise linear or trilinear basis functions are given by the relation $\varphi_i(P_j) = \delta_{ij}$, where $\{P_j\}_{j=1}^n$ represents inner mesh points. The matrix $K(u_h^t)$ and the vector $L(u_h^t)$ are usually approximated by using a suitable quadrature rule Q on a reference element $\hat{T}$ (unit simplex or unit cube) which leads to the approximations

$\tilde{K}(u_h^t) \approx K(u_h^t)$ and $\tilde{L}(u_h^t) \approx L(u_h^t)$. In the finite element method, the assembling of $\tilde{K}(u_h^t)$ is executed by passing $T \in \mathcal{T}$ and determining element matrices through

$$\begin{aligned}\int_T E_{ij\alpha\beta}(x, \nabla u_h^t)\varphi'_{\alpha,\beta}\varphi''_{i,j}\, d\Omega \\ \approx Q(E_{ij\alpha\beta}(\Phi_T, (\nabla u_h^t) \circ \Phi_T)\varphi'_{\alpha,\beta} \circ \Phi_T\, \varphi''_{i,j} \circ \Phi_T \,|\det \nabla \Phi_T|) \quad (5)\end{aligned}$$

with $\varphi', \varphi'' \in \{\varphi_i\}_{i=1}^n$. Here, $\Phi_T : \hat{T} \to T$ is an appropriate affine and linear transformation.

Finite elemente meshes and their influence on the numerical error. In view of (5), the evaluation is strongly dependent on the evaluation of the gradient of the transformation $\nabla\Phi_T$ and, therefore, on the geometry properties of the mesh elements. It should be expected that the use of an improper mesh has a deep influence on the sensitivity to numerical disturbances (e.g. rounding errors) of the resulting linear equations.

The disturbed equation reads $(\tilde{K}(u_h^t) + \delta\tilde{K}(u_h^t))(\hat{x}^t + \delta\hat{x}^t) = \tilde{L}(u_h^t) + \delta\tilde{L}(u_h^t)$ with the disturbances $\delta\tilde{K}(u_h^t) \in \mathbb{R}^{n\times n}$ and $\delta\tilde{L}(u_h^t) \in \mathbb{R}^n$. The absolute numerical error is denoted by $\delta\hat{x}^t \in \mathbb{R}^n$. It is well-known, that the relative numerical error $\|\delta\hat{x}^t\|/\|\hat{x}^t\|$ can be estimated by using the spectral matrix condition

$$\kappa_2(\tilde{K}(u_h^t)) := \frac{|\lambda_{\max}(\tilde{K}(u_h^t))|}{|\lambda_{\min}(\tilde{K}(u_h^t))|} \quad (6)$$

where $\lambda_{\max}(\tilde{K}(u_h^t))$ is the largest eigenvalue and $\lambda_{\min}(\tilde{K}(u_h^t))$ is the smallest eigenvalue of the symmetric matrix $\tilde{K}(u_h^t)$. The general disturbance theorem reads

$$\frac{\|\delta\hat{x}^t\|}{\|\hat{x}^t\|} \le \frac{\kappa_2(\tilde{K}(u_h^t))}{1-\mu}\left(\frac{\left|\delta\tilde{K}(u_h^t)\right|}{\left|\tilde{K}(u_h^t)\right|} + \frac{\left|\delta\tilde{L}(u_h^t)\right|}{\left|\tilde{L}(u_h^t)\right|}\right) \quad (7)$$

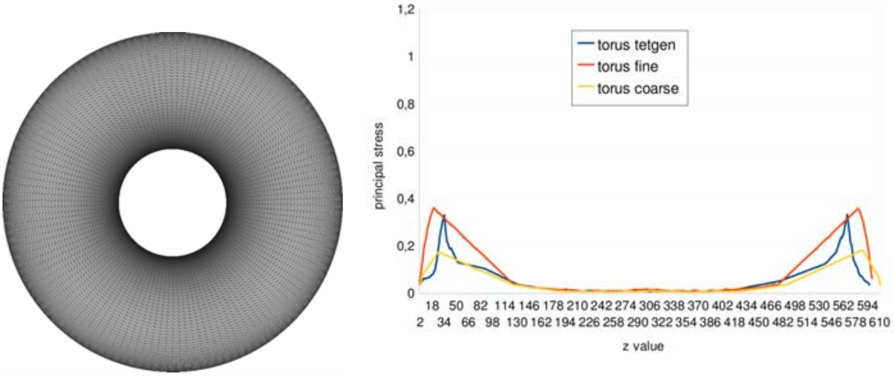
where $\mu := \kappa_2(\tilde{K}(u_h^t))\|\delta\tilde{K}(u_h^t)\|/\|\tilde{K}(u_h^t)\| < 1$ and $\|\cdot\|$ is the euclidian vector norm or the spectral matrix norm, respectively. In view of (7), a large spectral matrix condition implies that even a small disturbance of the data can lead to huge effects on the solution of the linear equation, which means that the solution process is unstable. Using (5), one can show that $\kappa_2(\tilde{K}(u_h^t)) \le Ch^{-2}$ asymptotically holds where the constant C depends on the quotient of the circumcircle and incircle of the mesh elements [30]. This, in fact, confirms our expectation of the influence of $\nabla\Phi_T$ on the sensitivity to numerical disturbances. In conclusion, it is therefore reasonable to consider the spectral matrix condition $\kappa_2(\tilde{K}(u_h^t))$ as a suitable measure for the applicability of a finite element mesh.

4 Results

Using the described method we have reconstructed all cleft lip and palate relevant structures. Here we care about the musculus orbicularis oris (MOO) and we present results encountered while reconstructing and simulating the MOO, either in its original segmentation or the smoothed variant. Two mesh resolutions (coarse and fine) are analyzed as well. A visualization of the cleft lip and palate region can be seen in figure 1. At the current stage of our studies, no fiber directions or activation of muscle contraction (as, e.g., described in [18]) are part of our modeling. We are mainly interested in the quality of the presented mesh generation algorithm for the use in finite element simulations for soft-tissue and in the impact of the segmented shapes as well as the level of detail in the segmented shapes on the simulation. We have prepared results for three different segmentations, each one with and without smoothing, and two mesh resolutions each. As a reference model, we use a torus, representing a idealized shape of the MOO, also coarse and fine. To compare the quality of the mesh generation algorithm based on the generated tori a reference mesh was generated with TetGen [31]. In order to minimize the overall-complexity, we use the described isotropic hyperelastic material model which is reasonable for many biological soft tissues and study the passive deformation behavior.

The simulation is based on a series of the finite element meshes, one of which is shown in figure 2. The series was generated by the mesh generation algorithm introduced in section 3.1. The constitutive law modeling the deformation behavior of the muscle is based on the isotropic hyperelastic Mooney-Rivlin approach as presented in section 3.3.

In order to get a better understanding of hyper-elastic deformation and the influence of the finite element mesh as well as the segmentation, a highly idealized configuration was also considered where the geometry of the MOO is reduced to a torus (cf. figure 3a). In figure 3, several simulation results of the torus' deformation based on different meshes were compared. On the one hand, a tetrahedral mesh based on the torus (cf. figure 3a) was tetrahedralized with TetGen [31]. This mesh serves as a reference mesh. On the other hand, two meshes (coarse and fine) were generated by the presented mesh generation algorithm. The resulting maximum principal stress curves can be found in figure 3b. The maximum principal stress is the largest eigenvalue of the three real eigenvalues of the symmetric Cauchy stress tensor, which is given by $\sigma_{ij} := \rho J^{-1} F_{ik} (\partial W / \partial F_{jk})$. The use of all finite element meshes yield symmetric deformation results as expected. Comparing the two mesh generation methods, the results based on the presented mesh generation algorithm are more symmetric as in the reference mesh created by the TetGen. Additionally, a comparison of the spectral matrix conditions is given in table 1. With respect to the different vertex count in all of the meshes, the comparisons convey that the spectral matrix condition yielded by the presented mesh generation algorithm is better compared to the spectral matrix condition yielded by the TetGen mesh. The number of unknowns for fine torus mesh is roughly a third compared to the mesh generated by TetGen, whereas the condition number is almost one fifth. Moreover, TetGen cannot be used to construct meshes for our medical application, since there is no initial surface mesh, because we start with stacked polygons. Additionally, taking influence e.g. on the vertex count is not directly possible and the geometric properties



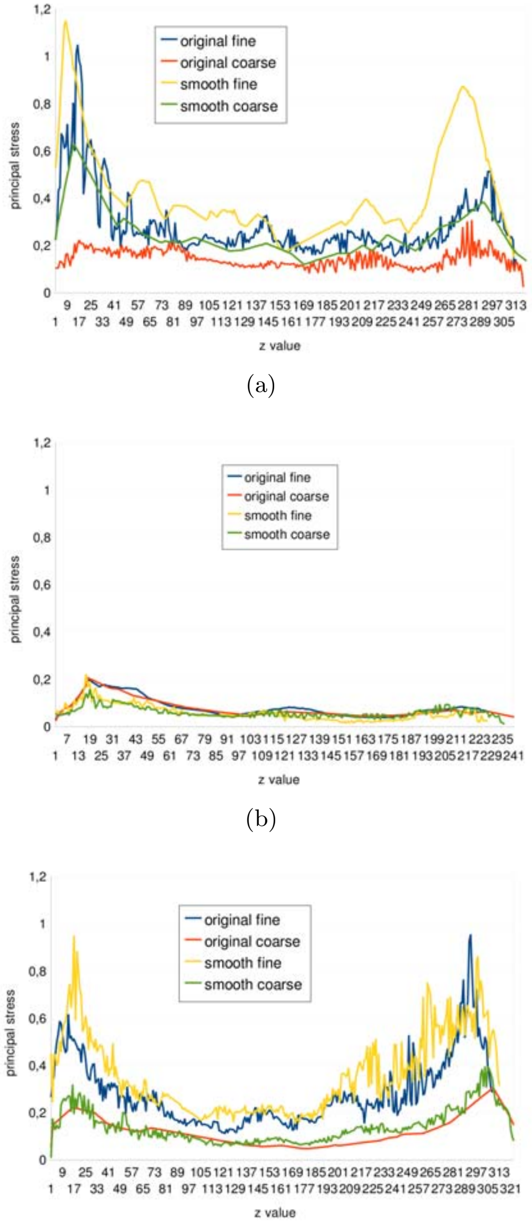
(a) (b)

Fig. 3. Torus: (a) A surface mesh of a torus (b) Simulation results for the tetrahedral mesh of the torus (a) generated by TetGen [31] (blue, 231, 614 tetrahedra), generated by the presented algorithm fine (red, 64, 679 tetrahedra), generated by the presented algorithm coarse (yellow, 14, 703 tetrahedra). The finite element method yields symmetric maximum principal stress curves as expected.

Table 1. Number of unknowns and spectral matrix condition for the musculus orbicularis oris and a reference torus. Data set describes the investigator, original stands for the unaltered segmentation, smooth for the altered variant.

Data set	Finite element mesh	Unknowns	Spectral matrix condition
“A”	original fine	117, 978	3.14035 $E7$
“A”	original coarse	23, 418	3.39126 $E6$
“A”	smooth fine	112, 656	6.92955 $E7$
“A”	smooth coarse	24, 027	5.41920 $E6$
“C”	original fine	77, 202	8.60515 $E6$
“C”	original coarse	15, 699	7.82299 $E5$
“C”	smooth fine	80, 064	2.41594 $E6$
“C”	smooth coarse	15, 993	1.24151 $E5$
“G”	original fine	103, 257	2.64039 $E7$
“G”	original coarse	20, 373	1.07328 $E6$
“G”	smooth fine	100, 608	6.21062 $E7$
“G”	smooth coarse	19, 401	9.23560 $E5$
torus	TetGen	156, 645	5, 239, 266
torus	fine	51, 474	1, 138, 141
torus	coarse	11, 898	241, 679

(like e.g. normed aspect ratiosor minimal dihedral angles [16]) of the individual elements are slightly inferior. On the other hand, the torus approximation of the tetrahedral mesh generated by TetGen is exact, whereas the presented mesh generation algorithm is only a close approximation. This does not matter much in our case, because the original shape of a segment is not known.



(a)

(b)

(c)

Fig. 4. Maximum principal stress values for several meshes originating from different investigators. Each segment (in its original form or a smoothed variant) was approximated at two resolutions (coarse with approximately 20k-30k tetrahedra, fine with approximately 100k-150k tetrahedra). Only the fine meshes exhibit stress peaks near the locked boundary whereas the coarse meshes stay relatively flat in comparison. (a) data set "A" (b) data set "C" (c) data set "G".

In figures 4a-c, finite element simulation results for the MOO (cf. figure 2) using the presented mesh generation algorithm are depicted. The muscle is assumed to be partially connected to e.g. cartilage or bones at its top and bottom side. In the figures, the maximum principal stress is shown which indicates regions of high stress intensity. As we can see, regions of high stress intensity are concentrated where the muscle is thought to connect to non-soft tissue. We saw the same result for the tori. In medical practice, such a simulation should enable the surgeon to decide where a surgical intervention is possible or where it should be avoided (see the discussion in section 5).

Comparing the original and smoothed meshes, one can see mixed results. At first, for the coarse meshes, it is impossible to fully approximate the level set surface because of the sheer size of the tetrahedra compared with the feature size or because of the modeling of the mass-spring system. The altitude springs effectively prevent further approximations at one point in time. The result is an offset to the level set which results in a worse approximation compared with the fine counterparts, e.g. the mouth opening cannot be fully resolved.

Looking at the condition curves in figure 4, the typical shapes created by analyzing the tori at coarse and fine resolutions can only be observed in some of the cases. In addition, the typical shape of the curves can only be found in the high resolution meshes (with one exception). This gives a strong hint that a certain mesh resolution is required to obtain useful information for medical practice. What is interesting is the observation that the extrema are of higher magnitude compared to the tori. This aspect must be influenced by the different geometry compared to the tori. It is unclear why data set “C” does not exhibit the typical curve at all. Analyzing normed aspect ratios or minimal dihedral angles that are normally used to evaluate mesh element quality [16] does not lead to a final conclusion as the quality statistics of all generated meshes do not differ enough to be significant. This provides a strong hint that the geometry strongly influences the results and should therefore be captured as close as possible.

On a recent Intel Core 2 Quad processor (2.83GHz, 8GB RAM) the surface reconstruction by the level set method on a $700 \times 600 \times 100$ uniform grid for a typical MOO consisting of approximately $8,800$ controlpoints takes about 25 minutes to complete. An additional 15 minutes are needed for the meshing algorithm with subsequent deformation ($>$ 100k tetrahedra), while the simulation itself takes about 30 minutes. In our prototypical implementation is room for improvement of these numbers, e.g. the level set could be defined on a sparse grid or on an octree.

5 Discussion

Simulations of the MOO alone can explain lip function only in part as the philtral form is getting additional contributions from both the superior levator labii muscle and the musculus nasalis [6]. The results of the present study suggest at least in the case of cleft lip surgery that operative techniques that respect the circular nature of the MOO in its natural uniformity [32] may be preferred over those that do not. Future finite element simulations of the MOO should however take into account that the upper levator muscle is descending as far medially as the philtral ridge to insert into the vermilion border lateral to the median groove while the nasalis muscle is inserting into the philtral ridges superiorly both contributing to labial motility [6]. Integrating these structures and their neighbors into a model that is coherent with finite element simulations would represent a more holistic approach which is associated with computational cost way beyond mainstream personal computers of to date, especially if real time functionality (one of the main stepping stones toward virtual surgery) is desired.

Virtual surgery allows surgeons to envision numerous scenarios of possible surgical options, and should allow them to accordingly predict the postoperative course. Appropriate simulation environments require solid objects to deform at interactive rates, with mesh deformation properties that correspond to real materials. Simulation of reconstructive surgery of cleft lip and palate should therefore not only be based on expertise in normal and pathological labial, palatal, and velopharyngeal anatomy. It should also include an accurate representation of the biophysical properties of the tissue types involved. This study was therefore designed to offer a mesh generation algorithm providing high physical and visual accuracy and analyzing the influence of the segmentation or smoothing on the numerical results.

Looking at medical implications after a future verification of the results, such a simulation should enable the surgeon to decide where a surgical intervention should be possible or where it should be avoided. In principle, color-coded stress patterns and concentrations of the stress values give a strong hint that such an analysis is indeed possible and a certain amount of detail is needed in the segmentation.

This can be seen in light of drawbacks associated with the segmentation process. In some layers, the differentiation of anatomical structures was considered a challenge, which may have resulted in errors that have considerable impact on the shape of the striated muscle model. As the determination of the relative height of a particular layer could only be estimated, the spatial relations between layers may not have been exact. Moreover, the mechanical process of material preparation introduced distortions that could not be eliminated. All these artifacts in plane and space are sources for uncertainty and resulting in difficulties to build an exact computer model of the muscle. The quantitative data provided by the present study is therefore not fully comparable with data from the body of literature. The present study may, however, demonstrate that a tetrahedral mesh approximation may suffice to run a finite element simulation of a striated muscle like the MOO. Studies based on a less problematic input may generate more appropriate results.

High monetary cost of surgery and possible complication management suggests the need for the acquisition of virtual surgery modules geared towards promoting knowledge of the optimal solution for each individual patient. Expanding the scope of computer sciences to tackle challenges in simulation will lead to the generation of virtual surgery scenarios for example with realistic glistening effects, view-dependent texture mapping, real-time smoke and bleeding simulation. The underlying models are still too complex to simultaneously run soft tissue simulation, collision detection, and haptic device support in a real-time performance. Sophisticated techniques are needed for the simulation of deformations resulting from muscular agonist-antagonist interaction during virtual surgery. In future studies, it needs to be evaluated if the algorithm introduced in the present study might be considered a step in that direction.

References

1. Delaire, J.: Le cheilo-rhinoplastie primaire pour fente labiomaxillaire congenitale unilateral. Rev. Stomatol. 76, 193 (1975)
2. Millard, D.: Cleft craft: The evolution of it's surgery. the unilateral deformity I (1976)
3. Randall, P., LaRossa, L.W.D.: The importance of muscle reconstruction in primary and secondary cleft lip repair. Plast. Reconstr. Surg. 64, 316–323 (1974)
4. Kriens, O.: An anatomical approach to veloplasty. Plast. Reconstr. Surg. 43, 29–41 (1969)
5. Sader, R., Zeilhofer, H., Dietz, M., Bressmann, T., Hannig, C., Putz, R., Horch, H.: Levatorplasty, a new technique to treat hypernasality: anatomical investigations and preliminary clinical results. J. Cranio-Maxillofac Surg. 29, 143–149 (2001)
6. Latham, R., Deaton, T.: The structural basis of the philtrum and the contour of the vermilion border: a study of the musculature of the upper lip. J. Anat. 121, 151–160 (1976)
7. Mulliken, J., Pensler, J., Kozakewich, H.: The anatomy of cupid's bow in normal and cleft lip. Plast. Reconstr. Surg. 92, 395–403 (1993)
8. Bern, M., Plassmann, P.: Mesh generation. In: Handbook of Computational Geometry, pp. 291–332. Elsevier Science, Amsterdam (2000)
9. George, P.L.: Automatic mesh generation. Application to Finite Element Methods (1991)
10. Frey, P.J.: Mesh generation. Hermes Science Europe Ltd (2000)
11. Landes, C.A., Weichert, F., Geis, P., Wernstedt, K., Wilde, A., Fritsch, H., Wagner, M.: Tissue-plastinated vs. celloidin-embedded large serial sections in video, analog and digital photographic on-screen reproduction: a preliminary step to exact virtual 3d modelling, exemplified in the normal midface and cleft-lip and palate. J. Anat. 207, 175–191 (2005)
12. Sarfraz, M.: Object recognition using fourier descriptors: Some experiments and observations. In: International Conference on Computer Graphics, Imaging and Visualisation, pp. 281–286 (2006)
13. Müller, H., Klingert, A.: Surface interpolation from cross sections. In: Focus on Scientific Visualization, pp. 139–189. Springer, Heidelberg (1993)
14. Zhao, H.K., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: IEEE Workshop on Variational and Level Set Methods (VLSM 2001), p. 194 (2001)
15. Osher, S., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Applied Mathematical Sciences. Springer, Berlin (2002)
16. Shewchuk, J.: What is a good linear finite element? Interpolation, Conditioning, Anisotropy, and Quality Measures (2002)

17. Bridson, R., Teran, J., Molino, N., Fedkiw, R.: Adaptive physics based tetrahedral mesh generation using level sets. Engineering with Computers 21, 2–18 (2005)
18. Martins, J.A.C., Pires, E.B., Salvado, R., Dinis, P.B.: A numerical model of passive and active behavior of skeletal muscles. Comput. Methods Appl. Mech. Eng. 151(3-4), 419–433 (1998)
19. Martins, P., Natal Jorge, R., Ferreira, A.: A comparative study of several material models for prediction of hyperelastic properties: Application to silicone-rubber and soft tissues. Strain 42, 135–147 (2006)
20. Van Loocke, M., Lyons, C., Simms, C.: The three-dimensional mechanical properties of skeletal muscle: Experiments and modelling. In: Prenderga. St., et al. (eds.) Topic in Bio-Mechanical Engineering, Trinity Centre for Bioengineering & National Centre for Biomedical Engineering Science (2004)
21. Weiss, J.A., Maker, B.N., Govindjee, S.: Finite element implementation of incompressible, transversely isotropic hyperelasticity. Comput. Methods Appl. Mech. Eng. 135(1-2), 107–128 (1996)
22. Kikuchi, N., Oden, J.: Contact problems in elasticity: A study of variational inequalities and finite element methods. In: SIAM Studies in Applied Mathematics. SIAM, Society for Industrial and Applied Mathematics, Philadelphia (1988)
23. Le Tallec, P.: Numerical methods for nonlinear three-dimensional elasticity. In: Ciarlet, P.G., et al. (eds.) Handbook of numerical analysis. Volume III: Techniques of scientific computing (Part 1). Numerical methods for solids (Part 1). Solution of equations in $\mathbb{R}^n$ (Part 2), pp. 465–622. North-Holland, Amsterdam (1994)
24. Crisfield, M.A.: Nonlinear finite element analysis of solids and structures: Advanced topics, vol. 2. Wiley. xiv, Chichester (1997)
25. Jankovich, E., Leblanc, F., Durand, M., Bercovier, M.: A finite element method for the analysis of rubber parts, experimental and analytical assessment. Comput. Struct. 14, 385–391 (1981)
26. Chen, J., Satyamurthy, K., Hirschfelt, L.: Consistent finite element procedures for nonlinear rubber elasticity with a higher order strain energy function. Comput. Struct. 50(6), 715–727 (1994)
27. Tabaddor, F.:Rubber elasticity models for finite element analysis. Comput. Struct. 26, 33–40 (1987)
28. Peng, S.H., Chang, W.V.: A compressible approach in finite element analysis of rubber-elastic materials. Comput. Struct. 62(3), 573–593 (1997)
29. Rueter, M.: Error controlled adaptive finite element methods in large strain hyperelasticity and fracture mechanics. Inst. für Baumechanik und Numerische Mechanik, Hannover (2003)
30. Hackbusch, W.: Theorie und Numerik elliptischer Differentialgleichungen. Teubner Studienbücher: Mathematik. B. G. Teubner, Stuttgart (1996)
31. Si, H.: Tetgen (2008), `http://tetgen.berlios.de`
32. Sebastian, G., Stein, A.: Regional approaches to the reconstruction of the lip region. Facial. Plast. Surg. 13, 125–135 (1997)

MotionLab: A Matlab Toolbox for Extracting and Processing Experimental Motion Capture Data for Neuromuscular Simulations

Anders Sandholm, Nicolas Pronost, and Daniel Thalmann

Ecole Polytechnique Fédérale de Lausanne, Virtual Reality Laboratory
Switzerland
{anders.sandholm,nicolas.pronost,daniel.thalmann}@epfl.ch

Abstract. During the last years several neuromuscular simulation platforms have been developed, from commercial tools to open-source based solutions to pure in-house solutions which are not public available. A major problem in running neuromuscular simulations in any of these tools are that there exist an infinitive number of motion capture laboratory setups. In this paper, the authors present a Matlab toolbox that provides a solution for this problem. The toolbox can easily be setup for any new experiment with minimal changes to the simulation environment. The toolbox provides powerful pre-simulation features such as filtering, markers evaluations and calculations of new marker setups. After that a simulation has been performed, results can be read back and several post-simulation features are available to validate the simulation.

Keywords: Motion capture, Force platform, Gait, Neuromuscular simulation, EMG analysis, Filtering, Kalman.

1 Introduction

Today the field of neuromuscular simulations is used to understand the underlaying dynamics of living beings movement, from gait research, treatment of patient with gait problem [7], to teaching physicians and developing ergonomic furniture [6]. During the last years several platforms have been developed, from commercial tools [12,3] to open-source based solutions [2] and pure in-house solutions which are not public available. The expansion of this field has also allowed the accessibility to neuromusculoskeletal models that are capable to describe different levels of complexity [13,14,3,8]. This development of more detailed models and simulation tools has given researchers and physicians better and more powerful tools to create advanced simulations and even to execute different *what if* scenarios or to evaluate different simulation results before the physical treatment has even started. Today there also exist several motion capture and force plate systems which produce and present data in different ways, some compatibles with only a few or specific tools. Comparability issues may also occurred between labs and experiments where laboratory coordinate system or number

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 110–124, 2009.

of force plates often required different setup protocols. This gives researchers a great freedom to carry out experiments and simulations in their own lab, but sometimes appears to be very problematic for changing the lab setup or sharing data with other groups/laboratories. In this paper, the authors present a Matlab toolbox that provides a solution for these problems. Currently the toolbox supports the open source platform OpenSim [3] but the toolbox can be easily extended to support other platforms. The toolbox can be setup for any new experiment with minimal changes to the simulation environment. Moreover, new models can be used in new simulations and input data can be easily shared in the C3D format [9] along with a few Matlab files. The toolbox also provides powerful pre-simulation features such as filtering, markers evaluations and calculations of new marker setups. After that a simulation has been performed the results can be read back into the toolbox and several post-simulation features are available to validate the simulation such as matching evaluation of muscular excitations against EMG patterns or plotting normalized parameters to a specific joint angle.

2 Data Acquisition

2.1 Motion Capture

Motion capture is a technique where high speed digital cameras are used to capture the 3D motion performed by a subject. There are several systems that can be used but most commonly the subject is fitted with either passive or active markers. The passive markers are often covered with an infrared reflective material and then attached to the subject on predefined anatomical landmarks. During the motion capture each camera produces infrared flashes and the reflected light from the markers is captured by the camera. In an active marker motion capture system, the subject is fitted with small lights sources, often LEDs, and the cameras do not use a synchronized flash, instead they capture the direct light from the LEDs. The two systems operate well for motion capture, but the use of active markers involves a lot of wiring, which could hinder the subject to perform a natural movement. The use of passive markers is simpler, but requires more complex hardware and calibration/optimization of the synchronized flash system. To capture a motion in three dimensions, a minimum of three cameras have to see each tracked marker. In Figure 1 a motion capture is shown, here the subject has been fitted with passive reflective skin markers, the subject is also standing on two force plates and is fitted with surface mounted electromyography (EMG) sensors.

Due to the large range of different motion captures that are performed, there is not a single protocol to follow. Parameters such as marker protocols, sampling frequencies, how large space the motion take (for example crouch is confined to a small volume, while running needs a larger volume), speed of the motion, direction of the motion and the used equipments must be taken into account to setup the musculoskeletal simulation. However, most motion capture session can be divided up into six steps.

Fig. 1. Static motion capture

Fig. 2. Motion capture of crouch

Fig. 3. Motion capture marker setup defined in this study

- Studio setup, number of cameras, coordinate systems.
- Force plate setup, number of force plates, coordinate systems.
- Calibration of the motion capture/force plate system.
- Capture of movement performed by the subject.
- Remove ghost markers
- Post-processing of data, calculating trajectories from the tracked markers.

Even if the same motion capture protocol is used, due to the calibration of the motion capture system and individual differences between the subjects, to filter the data and to change the coordinate system are often required before the motion capture can be used in a simulation.

Marker set: Before any motion capture can be performed, marker sets have to be defined so the system will be able to capture the desired motion. Several well known marker sets currently exist such as the gait analysis sets from Helen Hayes Hospital (HHS) [10] and from Cleveland Clinic (CCS) [11].

The CCS uses clusters of three markers to determine the knees and ankles coordinate systems. These clusters are on the lower portion of the lateral sides of thighs and shanks. This ensures that only a small amount of skin movements will be recorded for these markers.

The HHS was developed for video analysis of the lower extremity kinematics in order to minimize the motion capture time for the patient and reduce the number for tracked trajectories. To determine the lower limbs kinematic, a 10 cm wand is attached to each thigh and shank and records the three-dimensional motion of each leg. In this paper, we defined a new motion capture marker set that combines properties from the CCS and HHS and which support state of the art neuromuscular models and simulation platforms. The marker set can be seen in Figure 3. It contains 4 clusters, each contains 4 markers attached to a single plate. The plates are fixated to the subject's thighs and shanks, additional skin markers are attached to the pelvis, thighs, shanks and feet. Enough markers have been used to create a redundant system for each degree of freedom. The upper body contains a minimum of 3 markers on torso and head, depending on joint specification between pelvis-torso and torso-head. The recommended protocol starts with a static trial where the subject stands in a neutral pose where all markers are visible to the cameras. Using this trial the neuromuscular model can be scaled to represent the subject. Then, the skin markers on the thighs and the shanks can be removed because only the clusters, which contain less skin artifacts, will be used in the simulation. However, the authors recommend that all markers are kept during the motion capture, so if a cluster plates moves, a new static trial can directly be captured. Using this marker set a subset of markers can be selected so both the CCS and the HHS can be produced.

Force plates: A force plate is an instrument that measures the force and moment applied on it. Placed on the ground, it can so measure the ground reaction of someone or something walking, standing, running ... on it (see Figure 1. There are many different types and brands of force plates that are used in motion analysis laboratories. The main brands today are AMTI and Kistler. Both of these platforms measure the ground reaction force (GRF) but the AMTI measure also the ground reaction moment (GRMo). GRF and GRMo are most commonly sampled at 2000Hz and are described in a local coordinate system for each force plate.

Electromyography: Electromyography (EMG) is a method for recording and evaluating the activation of muscles. It detects the electrical potential generated by muscle cells when these are activated and at rest. There are two main kinds of EMG, surface EMG (see Figure 1) and needle EMG which measure the muscle potential inside a muscle. EMG has been used in several applications, such as clinical use in diagnosis of neurological and neuromuscular problems. In neuromuscular simulations, EMG signals can be used either to drive a simulation, so called forward dynamic or be used as a validation data where the simulated muscular excitation pattern has to be correlated to the captured EMG signal. In the presented framework, both surface EMG and internal EMG are supported.

Motion capture lab setup: A major problem while exchanging motion capture data today is that most of the motion analysis labs have different setups/coordinate systems. Even if the same marker set and protocol are used, there can be differences in sampling frequencies, coordinate systems and even differences between different brands of EMG, force platforms and motion capture systems. In Figure 4 a typical motion lab setup is showed with the coordinate system of the motion capture and force plate displayed. Due to the fact that three coordinate systems are used, one for the motion capture and one for each force plate, the data have to be translated into the coordinate system of the neuromuscular model (see Figure 4). In this paper, the authors will use the model coordinate system to show how the motion capture can be translated using a rigid body transformation into model coordinate system and how the GRF and GRMo are calculated into the motion capture coordinate system and then translated again into the model coordinate system.

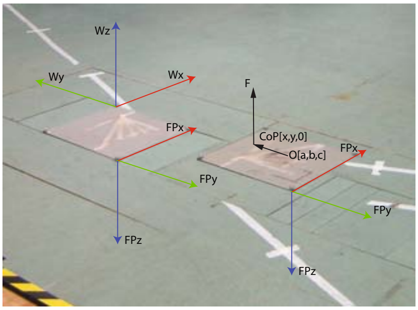


Fig. 4. Laboratory coordinate system

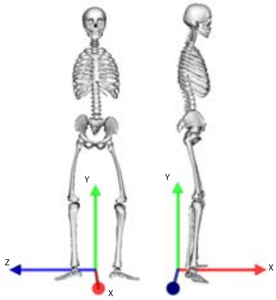


Fig. 5. Neuromuscular model coordinate system

3 OpenSim Environment

OpenSim [3,2] is an open-source platform for modeling, simulating, and analyzing neuromusculoskeletal systems. Since its creation, hundreds of people have begun to use this platform in a wide variety of applications, including biomechanics research, medical device design, orthopedics and rehabilitation science, neuroscience research, ergonomic analysis and design, sports science, computer animation, robotics research, and biology and engineering education. OpenSim technology makes it possible to develop customized controllers, user-defined analysis, contact models and neuromusculoskeletal models among other things. So the audience involves people from biomechanics scientists and clinicians to developers who need tools for modeling and simulating motion and forces for neuromusculoskeletal systems. Such bio-simulations consist of several steps in order to analyze subject-specific models and motions (see Figure 6).

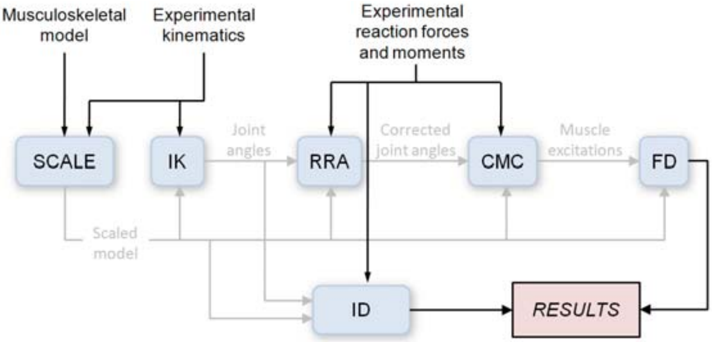


Fig. 6. Steps for generating a muscle-driven simulation of a subject's motion with OpenSim

3.1 Workflow

OpenSim guides users through steps to create a dynamic simulation. As input, OpenSim usually takes a dynamic model of the musculoskeletal system and experimentally-measured kinematics, reaction forces and moments.

Scale: In a first step, a generic model is scaled to match the anthropometry of an individual subject. The dimensions of each body segment are scaled based on relative distances between pairs of markers obtained from a motion-capture system and the corresponding virtual marker locations in the generic model. During this process mass properties, muscle fiber lengths and tendon slack lengths are also scaled. In addition to scaling the model, this step can be used to adjust the locations of virtual markers so that they better match the experimental data.

Inverse Kinematics (IK): In the second step, an IK problem is solved to determine the coordinate values (joint angles and translations) that best reproduce the raw marker data obtained from motion capture. The solver is formulated as a frame-by-frame weighted least-squares problem that minimizes the differences between the measured marker locations and the model's virtual marker locations, subject to joint constraints [4].

Residual Reduction Algorithm (RRA): Due to experimental error and modeling assumptions, measured ground reaction forces and moments are often inconsistent with the model kinematics. The RRA is thus applied to make the model coordinates from IK more dynamically consistent. Indeed, in the absence of experimental and modeling errors, no residual force has to be added to relate the gravitational acceleration of the segments and the ground reaction forces. In practice, this is never the case. The RRA consists of two passes : (i) to alter mass centers of the model (mainly torso) so that excessive leaning in the left-right or fore-aft directions is corrected, and (ii) to alter the kinematics of the model to be more consistent with the ground reaction data for example by applying residual forces and moments on the six degrees of freedom between the model and the world.

Computed Muscle Control (CMC): The purpose of CMC is to compute a set of muscle excitations that produces a coordinated muscle-driven simulation of the subject's movement. CMC uses a static optimization criterion to distribute forces across synergistic muscles and proportional-derivative control to generate a forward dynamics simulation including the state equations representing the activation and contraction of the muscles.

Forward Dynamics (FD): The next step consists in performing a forward dynamics simulation from the muscle excitations computed by the CMC. As in CMC and RRA, a 5th order Runge-Kutta-Feldberg integrator is used. In contrast to CMC, which used PD controllers in a closed-loop system to ensure tracking of the desired trajectories, the FD simulation is an open-loop system. That is, it blindly applies the recorded actuator controls with no feedback or correction mechanism to help ensure accurate tracking.

Inverse Dynamics (ID): The inverse dynamics process derives generalized forces (e.g., net forces and torques) at each joint responsible for a given movement. Given the kinematics (e.g., states or motion) describing the movement of a model and a portion of the kinetics (e.g., external loads) applied to the model, an inverse dynamics analysis can be performed. Classical mechanics mathematically expresses the mass-dependent relationship between force and acceleration with equations of motion. The ID solves these equations to yield the net forces and torques at each joint which produce the movement.

4 Matlab Toolbox

The design of the OpenSim MotionLab Matlab toolbox was done so that, with a minimum of setup time, data from a motion capture can be used in an OpenSim simulation. The toolbox can also be used to generate the required OpenSim files which then can be loaded into the OpenSim GUI. The MotionLab toolbox main features are i) reading motion capture data from C3D files, ii) calculating ground reaction force, iii) translating motion capture data into model coordinate system, iv) applying noise reduction of simulation data, v) processing EMG data and processing/plotting simulation results.

Before the toolbox can be used, the path to the toolbox has to be added to the Matlab system path and C3DServer from Motion Lab Systems [9] has to be installed. After the desired motion capture has been performed, the marker trajectories have to be labeled and exported into a C3D file.

Because of the wide varieties of laboratories and motion capture setups when it comes to number of force plates, coordinate system orientations, marker sets and EMG setups, the MotionLab toolbox requires a Matlab file session_setup.m. This file describes each motion capture session such as which marker set has been used, number of force plates, translation between the different coordinate systems and which analog channels are used.

In the following example the authors will denote Matlab variables as *this* and Matlab function calls as `this`.

4.1 Toolbox Setup

The session_setup.m contains several parameters that describe the motion capture session, the authors will outline the most important parameters and what they describe.

session.FP.prefix = ’FP’,’FP’,’FP’,’FP’,’FP’,’FP’;
session.FP.suffix = ’Fx’,’Fy’,’Fz’,’Mx’,’My’,’Mz’;

The session.FP describe the channel names of the AMTI force plate, in this example two force plates are used and Matlab will extract the channels FP1Fx, FP2Fx, FP1Fy, FP2Fy ... FP2Mz.

session.mcDirection.fp2Model(direction,:) = [-1 -3 -2];
session.mcDirection.mc2Model(direction,:) = [-1 -3 -2];

The mcDirection describe the rigid body transformation between the motion capture and the model, and between the force plate and the model. These transformations depend on which direction the model faces, here given as an index *direction*. If the subject would face -Wy direction we can extract the translation from figure 4 and 5, the force plates x-direction correspond to the -x-direction in the model, force plate y-direction correspond to the -z-direction in the model and force plate z-direction correspond to -y-direction in the model. Using these transformations, data from the motion capture and the force plates can be translated into the coordinate system of the model.

```
session.EMGChannels{1}= {’TA’ ,’Tibialis anterior’, {’tib_ant_r’}};
session.EMGChannels{2}= {’SOL’ ,’Soleus’, {’soleus_r’}};
```

The EMGChannels specify the name of the analog EMG channels that are stored in the C3D file, the corresponding muscle name and the corresponding actuators in the OpenSim model, here showing two channels.

```
session.markers(1)=struct(’name’,{’TRX’},’ikWeight’,{5},’ikUsed’,{1});
session.markers(2)=struct(’name’,{’RAM’},’ikWeight’,{25},’ikUsed’,{1});
session.markers(3)=struct(’name’,{’LAM’},’ikWeight’,{25},’ikUsed’,{1});
```

The *markers* specify the markers that are used in the current motion capture. First the name, RTPL, then how large weight the marker should be in the IK. Some marker sets use different markers to scale the model and to drive the IK, to do this the parameter ikUsed can be used to disable a marker for the IK process.

When the session_setup.m file has been created, the toolbox can be used to extract the motion capture data and carry out a simulation. Either the toolbox is used through the Matlab command window, or used in a simulation script/function. In both case some initial variables have to be set:

```
c3dFileDynamic = ’Crouch_0004.c3d’;
c3dFileStatic = ’REF_02.c3d’;
```

The *c3dFileDynamic* and *c3dFileStatic* specify the dynamic and static motion capture files, if no *c3dFileStatic* file is specified the first frames of the dynamic trial will be used to scale the model.

```
osimModelFile = ’2392_3DAH.osim’;
```

The *osimModelFile* variable specifies which model will be used in the simulation. The toolbox assumes that the model exists in the same directory as the C3D files.

4.2 Motion Capture Data

Before we can extract the motion capture data, a C3DStruct has to be created. This structure contains all the information from the session_setup.m, the model file, which direction the model faces and which C3D file should be extracted. In the following example the static data will be used, but the dynamic motion capture is extracted in the exact same way.

```
C3dStruct = extractC3d(c3dFileStatic,type, direction);
```

The function takes three arguments, the *c3dFileStatic* is the C3D file name, *type* specifies if it is a 'static' or 'dynamic' trial and *direction* specifies which direction the subject faces: Wx, -Wx, Wy or -Wy.

The toolbox has now all the information to extract the motion capture and to convert it into the model coordinate system:

```
motionData = extractMarkers(C3dStruct);
```

This will extract all markers from the C3D file that are specified in the *session.markers* and translate them into the model coordinate system.

A major problem during motion capture is markers that have gaps in their trajectories. Most common motion capture softwares can interpolate small marker gaps, but when the gaps become larger than 10 frames it is hard to perform accurate interpolations. These gaps can create inaccurate IK solutions and therefore each trajectory should be checked so no gaps exists. In our MotionLab toolbox this is done in the function:

```
[motionData,C3dStruct] = checkMarkers(motionData, C3dStruct);
```

which checks each trajectory and removes the trajectory if it contains a gap.

Filtering: A major problem with motion capture is that all signals contain noise. It is therefore important to carry out filtering to minimize the noise, so a more stable IK solution can be reached. The problem is to determine how much smoothing is needed. Not enough and the IK solution will contain noise, too much smoothing and important translation/rotation will be smoothed out. It is necessary to analyze the data before so the right type of smoothing is carried out. In the MotionLab toolbox two different smoothing techniques are implemented:

```
[motionData,C3dStruct] = lpMCSmoother(motionData, C3dStruct,cutHz);
[motionData] = kalmanMCSmoother(motionData, noise, order);
```

The `lpMCSmoother` performs a low-pass filtering on the motion capture data, this type of filter is a very robust way to smooth motion capture data. The main drawback with low-pass filters is that it is hard to choose a cutoff

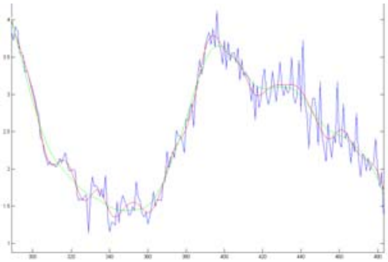


Fig. 7. Blue graph: unfiltered motion capture data. Red: Kalman smoothed (experimental noise ratio = 1 / 1000, order = 3). Green: Low-pass filtered (cutoff frequency = 6Hz).

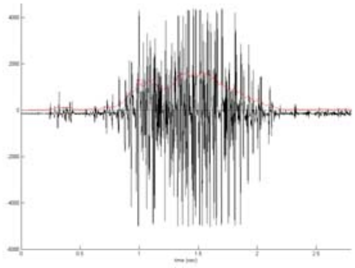


Fig. 8. Black: unfiltered EMG signal. Red: high-pass (10Hz), rectified, low-pass (6Hz) EMG signal.

frequency that removes the noise but preserves the desired motion. More complex smoothing algorithms are needed, one of them is the Kalman smoothing [1,5] that is a recursive filter based on linear dynamical system. The Kalman smoother,`kalmanMCSmoother`, requires three parameters, the motion capture data (*motionData*), the experimental noise ratio (*noise*) and the order of the filter (*order*), the drawback is that the Kalman smoother is less robust and requires more computation time. In figure 7 the differences between unfiltered motion capture data, Kalman smoothed and low-pass filter (cutHz = 6Hz) are shown. Both the low-pass and the Kalman smooth the data, but where the low-pass filter smooths out small changes in the kinematics the Kalman smoother preserves them.

In the final step the marker data is saved to the OpenSim file format [2], this is done by calling the function:

```
motionData = saveMarkers(C3dStruct, motionData);
```

4.3 Ground Reaction Force

To be able to run the RRA and the CMC tools, OpenSim uses the ground reaction forces which are captured with the motion capture session. The problem is that most commercial motion capture software are not able to process the force platform data, instead they save the raw signal to the C3D file, still in the coordinate system of the force platform. We so need to calculate the Center of Pressure (CoP) along side the force and moment in the motion capture coordinate system. To be able to calculate the center of pressure the toolbox requires the exact coordinates of each force plates stored in the C3D file, ordered from the top left corner and then clockwise. The toolbox also requires that the force platform is calibrated so that forces and moments are given in the true origin and not in the geometric origin of the force plate. To calculate the CoP and reaction force, we assume that the true origin exists at **O** located at **(a,b,c)** (see figure 4) and that the force plate is oriented in the (Wx, Wy-plane) which means that the z component of the center of pressure will always be equal to **0**. The moment around the true origin can then be written as equation 1 :

$$\mathbf{M} = [x-a, y-b, z-c] \times [F_x, F_y, F_z] + [0, 0, T_z] \tag{1}$$

Equation 1 can be rewritten as equation 2 :

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 0 & c & y-b \\ -c & 0 & -(x-a) \\ -(y-b) & x-a & 0 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ T_z \end{bmatrix} \tag{2}$$

And then into equation 3 :

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} (y-b)F_z + cF_y \\ -cF_x - (x-a)F_z \\ (x-a)F_y - (y-b)F_x + T_z \end{bmatrix} \tag{3}$$

From equation 3, we extract the Center of Pressure (x, y, T_z) shown in equation 4, (where $z = 0$);

$$\begin{aligned}
x &= -\frac{M_y + cF_x}{F_z} + a \\
y &= \frac{M_x - cF_y}{F_z} + b \\
T_z &= M_z - (x - a)F_y + (y - b)F_x
\end{aligned} \tag{4}$$

In the MotionLab toolbox, calculation of the CoP and translation of the data into model coordinate are implemented in:
`[GRF, CoP, GRMo, GRMx] = extractKinematics(C3dStruct);`
The function extracts the force plate information from the C3D file specified in the *C3DStruct*, calculates center of pressure by using equation 4 and then translates the force data into the model coordinates using the translation vector stored in *C3dStruct.transVec(i).fp2model* where i is the direction the motion is performed in *Wx, -Wx, Wy, -Wy.* The force data is returned as a structure and saved to an OpenSim file named *keySta.osimFile*_kinematic.mot in the simulation directory *C3dStruct.dirPath.*

4.4 Electromyography

The use of EMG can be a powerful tool, but the signal in its raw form needs some pre-processing before it can be used (see figure 8) in inverse dynamics or be used to verify simulation results. The MotionLab toolbox currently supports several filtering, such as high-pass, low-pass, Kalman and band-pass filtering. The toolbox can also normalize and rectify an EMG signal. The EMG tool is called through the function:
`[rawEMG, filteredEMG] = extractEMG(C3dStruct, filter, input);`
The processed EMG data is returned to the workspace and saved to an OpenSim file named *keySta.osimFile*_EMG.mot in the simulation directory *C3dStruct.dirPath.*

4.5 Calling OpenSim from Matlab

After that the motion capture, force plate and EMG data have been extracted, the toolbox can generate the required files to execute the different simulation phases in OpenSim.

The first phase is always to perform a scale of the generic model to a specific subject. This is done by calling `setupScale(C3dStruct)` which generates the scale setup file and then executes the scale tool. Similarly functionality exists for the IK process. The function `setupIK(C3dStruct)` generates the IK setup file along side with the IK_Task.xml file which contains the weight for each marker specified in *session.markers.* Similarly functionality exists for CMC, RRA and ID. In `setupRRA(C3dStruct)` and `setupCMC(C3dStruct)` the pelvis center of mass is first extracted from the model file and inserted in the RRA_Actuators.xml and CMC_Actuators.xml file before the setup file is generated and the tool executed. The inverse dynamics is called in the same manner through `setupID(C3dStruct)` and the forward dynamics through `setupFD(C3dStruct)`. Each tool checks so the prerequisites are meet, for example that a scaled model exists for the IK tool and that the IK has been executed before the RRA tool is executed. If a tool does not reach a solution the toolbox raises an error and halts further function executions.

4.6 Post-processing Simulation Result

After a simulation has been carried out the MotionLab toolbox can extract the simulation result back into Matlab:

```
result = extractResult(C3dStruct,'type');
```

where type defines which result should be extracted: 'ik', 'cmc',...,'fd'. This enables Matlab to be used as a powerful tool to carry out validation of simulations and plotting/correlation of variables.

5 Application: A Crouch Motion

In this section the authors will show an example where the MotionLab toolbox was used to extract motion capture data, to carry out a simulation and to plot data from a neuromuscular simulation. This example is not intended to be used in any medical or scientific conclusion, only as a MotionLab toolbox example.

The subject was a healthy male, early 30 with no neuromuscular medical history. Motion captures were performed using a Qualisys motion capture system. Two AMTI force plates were arranged so the subject could place one foot on each force plate. The labs coordinate system was arranged as in figure 4. First, a static reference frame was captured where the subject was standing upright without performing any movement. The subject was then asked to perform a crouching movement from a standing position without lifting his heels. After the motion capture Qualisys Track Manager was used to remove ghost markers, to label each marker and to export the motions into the C3D format.

The session_setup.m file was created as in section 4.1 and saved in the same directory as the static and dynamic C3D files.

The MotionLab toolbox can be used in two ways, either in matlab script/ functions or as we will do in this example, by calling each function from the Matlab Command Window. The first task is to extract the information from the static pose, to save the data and to call the OpenSim scaling tool. For more details about each MotionLab tool see section 4.

```
staticC3d = extractC3d(c3dFileStatic,'static', '-Wy');
staticMotionData = extractMarkers(staticC3d);
[staticMotionData,staticC3d] = checkMarkers(staticMotionData, staticC3d);
staticMotionData = kalmanMCSmoother(staticMotionData, 1000, 3);
staticMotionData = saveMarkers(staticC3d, staticMotionData);
```

If no errors has occured all the necessary information has been extracted from the C3D file and saved in the OpenSim format, we can now execute the scale function by calling `runScale(staticC3d);`. This will save the setup file needed to carry out the scale and execute the OpenSim scaling tool. The OpenSim scale

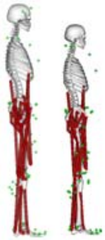


Fig. 9. Left: Unscaled model, Right: Scaled model

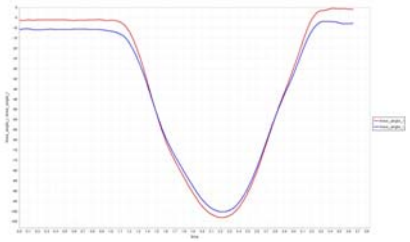


Fig. 10. Red: right knee angle, Blue: left knee angle after Inverse kinematic

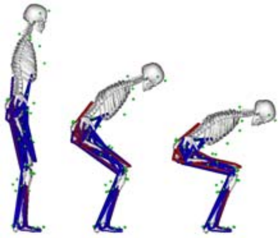


Fig. 11. Visualization of muscle activation during a crouch motion, blue: rested, red: activated

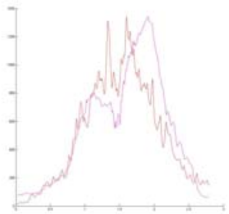


Fig. 12. Red = High-pass (10Hz), rectified, low-pass (6Hz) EMG signal. Purple: CMC muscle activation for vastus medialis.

tool will generate a new OpenSim model that will describe the motion captured subject (see figure 5) where a unscaled generic model and a scaled subject-specific model are shown.

When a scaled model has been generated it is now time to extract the dynamic motion.

```
dynamicC3d = extractC3d(c3dFileDynamic,'dynamic', '-Wy');
dynamicMotion = extractMarkers(dynamicC3d);
[dynamicMotion,dynamicC3d] = checkMarkers(dynamicMotion,dynamicC3d);
[dynamicMotion] = kalmanMCSmoother(dynamicMotion, 1000, 3);
dynamicMotion = saveMarkers(dynamicC3d, dynamicMotion);
[GRF, CoP, GRMo, GRMx] = extractKinematics(dynamicC3d);
```

The final command will extract the ground reaction force and calculate the correct center of pressure using equation 4. All the desired information is then extracted to carry out the IK operation, this is done by calling `runIK(dynamicC3d);`. This command will save a setup file for the OpenSim IK tool and in the last stage call the tool. The resulting left and right knee angles can be seen in figure 10. All other tools can in the same way be called from the MotionLab toolbox, in this example we choose to call the RRA and the CMC tool. The RRA tool is called by using the function `runRRA(dynamicC3d)`.

For more information about RRA see section 3.1. To calculate muscle activation and muscle forces the CMC tool is called using `runCMC(dynamicC3d)`, see figure 5. The CMC solution can then be read back into MotionLab and plotted/analyzed by using the function `result = extractResult(C3dStruct,'CMC')`. In figure 12 CMC muscle activation and corresponding filtered EMG are shown for the muscle vastus medialis.

6 Conclusion

We have in this paper described the functionality and the principal considerations behind the MotionLab Matlab toolbox ; and shown how it can be setup for a motion capture session with minimum alterations. The authors have also shown that the toolbox is fully interlinked with the OpenSim platform and its data formats. We believe that the toolbox can be a powerful tool for providing functionality, such as advanced filtering and analyzes of both simulated parameters and motion capture signals such as EMG. Further work include more advanced filtering, plotting and visualization and support for other simulation platforms such as Anybody and Simm.

Acknowledgments. This work is supported by the 3D Anatomical Human project (MRTN-CT-2006-035763) funded by the European Union.

References

1. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME–Journal of Basic Engineering 82, 35–45 (1966)
2. Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T.: Guendelman E and Thelen DG, OpenSim: Open-source Software to Create and Analyze Dynamic Simulations of Movement. IEEE Transactions on Biomedical Engineering 54(11), 1940–1950 (2007)
3. Delp, S.L., Loan, J.P., Hoy, H.G., Zajac, F.E., Topp, E.L., Rosen, J.M.: An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. IEEE Transactions on Biomedical Engineering 37(8), 757–767 (1990)
4. Lu, T.W., O'Connor, J.J.: Bone position estimation from skin marker coordinates using global optimization with joint constraints. Journal of Biomechanics 32, 129–134 (1999)
5. De Groote, F., De Laet, T., Jonkers, I., De Schutter, J.: Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis. Journal of Biomechanics 41(16), 3390–3398 (2008)
6. Rasmussen, J., de Zee, M.: Design Optimization of Airline Seats. SAE International Journal of Passenger Cars - electronic and electrical systems 1(1), 580–584 (2008)
7. Fox, M.D., Reinbolt, J.A., Õunpuu, S., Delp, S.L.: Mechanisms of improved knee flexion after rectus femoris transfer surgery. Journal of Biomechanics 42(5), 614–619 (2009)
8. Klein Horsman, M.D., Koopman, H.F.J.M., van der Helm, F.C.T., Poliacu Prose, L., Veeger, H.E.J.: Morphological muscle and joint parameters for musculoskeletal modelling of the lower extremity. Clinical Biomechanics 22(2), 239–247 (2007)

9. Motion Lab Systems (July 2009), http://www.c3d.org/
10. Kadaba, M.P., Ramakrishnan, H.K., Wootten, M.E.: Measurement of lower extremity kinematics during level walking. Journal of Orthopaedic Research 8(3), 383–392 (1990)
11. Castagno, P., Richards, J., Miller, F., Lennon, N.: Comparison of 3-dimensional lower extremity kinematics during walking gait using two different marker sets. Gait & Posture 3(2), 87 (1995)
12. Damsgaard, M., Rasmussen, J., Christensen, S.T., Surma, E., de Zee, M.: Analysis of musculoskeletal systems in the AnyBody Modeling System 14(8), 1100–1111 (2006)
13. Zhang, L.-Q., Wang, G.: Dynamic and static control of the human knee joint in abduction-adduction. Journal of Biomechanics 34(9), 1107–1115 (2001)
14. Holzbaur, K., Murray, V., Delp, S.: A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. Annals of Biomedical Engineering 33(6), 829–869 (2005)

Predicting Missing Markers in Real-Time Optical Motion Capture

Tommaso Piazza[1], Johan Lundström[1], Andreas Kunz[2], and Morten Fjeld[1]

[1] t2i Lab, CSE Chalmers University of Technology, Rännvägen 6B
412 96 Göteborg, Sweden
`{piazza,fjeld}@chalmers.se`
`www.t2i.se`
[2] ICVR, Swiss Federal Institute of Technology, Tannenstrasse 3
8092 Zurich, Switzerland
`kunz@iwf.mavt.ethz.ch`
`www.icvr.ethz.ch`

Abstract. A common problem in optical motion capture of human-body movement is the so-called missing marker problem. The occlusion of markers can lead to significant problems in tracking accuracy unless a continuous flow of data is guaranteed by interpolation or extrapolation algorithms. Since interpolation algorithms require data sampled before and after an occlusion, they cannot be used for real-time applications. Extrapolation algorithms only require data sampled before an occlusion. Other algorithms require statistical data and are designed for post-processing. In order to bridge sampling gaps caused by occluded markers and hence to improve 3D real-time motion capture, we suggest a computationally cost-efficient extrapolation algorithm partly combined with a so-called constraint matrix. The realization of this prediction algorithm does not require statistical data nor does it rely on an underlying kinematic human model with pre-defined marker distances. Under the assumption that human motion can be linear, circular, or a linear combination of both, a prediction method is realized. The paper presents measurements of a circular movement wherein a marker is briefly lost. The suggested extrapolation method behaves well for a reasonable number of frames, not exceeding around two seconds of time.

Keywords: Extrapolation, linear approximation, nonlinear approximation.

1 Introduction

Motion capture is the process of recording movement and translating sampled data onto a digital model. It is typically used in military, entertainment [1], sports [2], and medical applications [3]. For instance, in real-time computer-aided surgical systems with graphics, image-forming devices can provide closed-loop feedback between the real environment and the tracking system [17]. However, the captured data can also be transformed into characters and modified completely during post-production. In this paper we focus on improving the usage of real-time tracked data of human-body movement. Hence, we propose an algorithm that requires neither a simplified geometry [19] of a human (kinematics) nor a huge amount of statistical data.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 125–136, 2009.

Motion capture can either be inertial, mechanical, magnetic, or optical. In optical motion capture - which is the focus of this work - cameras are used together with active or passive infrared-reflecting markers. Among the advantages of optical motion capture is that the performer feels free to move since no cables connect the body to the equipment. Also large physical spaces can be covered, more performers can be included, and clean, detailed data is yielded [15]. Deficits of optical tracking are that it is prone to light interference and that reflective dots can be blocked by performers or other structures, causing temporary loss of data, also called occlusion. In order to overcome this problem, various approaches to close the data gaps in which no measurements are available were proposed. While some approaches only work off-line, others have real-time capability. These approaches typically use statistical data or some simplified rigid body assumptions for approximating human behavior [4].

The following section presents related work in the area of off-line and real-time methods, followed by a section showing the methodological basis for our experimental investigations. Then come two empirical sections each showing the experimental set-up and presenting gathered data from a circular movement. This is followed by a short section on applications in the area of human-body movement, and a final conclusion and outlook section.

2 Related Work and Requirements

Some researchers make use of knowledge about the tracked objects. Molet et al. [16] suggest a method to achieve real-time conversion of sensor measurements from anatomical human rotations. Their method requires only simple calibration, allows for sensors slippage, and takes advantage of knowledge of the type of motion being performed. As we aim for a method that can be benchmarked with various types of motion, we prefer not to incorporate knowledge about those types of motion into our method. This is the first requirement (R1).

Our method should be sufficiently generic to work with most real-time motion capture systems. We are aware that some researchers offer related solutions utilizing a specific camera. Aliaga et al. [18] suggest the Occlusion-Resistant Camera (ORC) to acquire active environments despite the presence of so-called moving interfering occluders. While real-time solutions to the missing marker problem may include the camera, we do not deem camera-integration to be feasible at this stage of our project. That is, our method should be sufficiently generic to work with most real-time motion capture systems. This is the second requirement (R2).

Most companies make real-time motion capture systems with passive targets and reflective markers. The markers are illuminated by infrared light synchronized with high-speed 2D digital imaging systems that can simultaneously image multiple targets. In the work presented here, we used a Qualisys Motion Capture System. Since the issue of missing markers is relatively well researched, a number of methods have been proposed. However, most of these methods are either off-line procedures to be used in post-processing, or they have unsatisfactory prediction accuracy.

Some typical off-line methods interpolate data using linear or non-linear approaches [5] [6] [7]. However, these approaches cause noticeable latencies and cannot be used in real-time applications. Another off-line approach is presented in [8], in which a geometric skeleton structure is automatically generated in order to bridge existing gaps in the measurement series. Herda et al. [9] describe a real-time method in which an anatomical human model is used to predict the position of the markers. However, this model is too complex for envisioned integration into the human modeling and simulation programs, e.g.'Jack' [10], which already offer a sophisticated anatomical human model. Hornung et al. [11] rely on the fact that the markers on a limb are separated by fixed distances. Finally, [4] uses Kalman filters for predicting marker positions, which are relatively simple and take measurement noise into consideration. However, they require statistical data about the noise and its covariance. As we consider the use of Kalman filters to be a well-examined approach, we will not research it any further here.

3 Methodology

In principle, there are two approaches to closing the gaps resulting from missing marker positions: interpolation and extrapolation. For the interpolation, special algorithms such as Catmull-Rom splines [12] must be used since they go through all measurement points being used as control points. However, such an interpolation requires future measurements also, otherwise it can only be used in a post-processing step after all data is available. Extrapolation, on the other hand, does not depend on future information, but the accuracy of an extrapolated point strongly depends on how it is determined from the existing measurements.

In this paper, we present a method of extrapolating a missing point based on previously measured points assuming the most common motion behaviors, such as linear or circular movement. Using a so-called matching factor (see equations 11–12) and the acquired measurements, our algorithm determines whether a missing point should be on a circular or linear extrapolated curve. After describing the algorithm, the paper concludes with first measurement results indicating the performance of the algorithm.

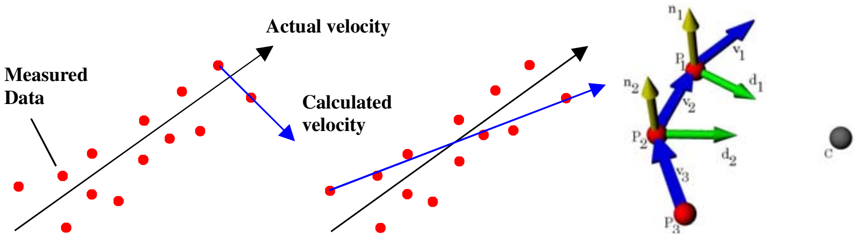
3.1 Prediction

All prediction is typically based on the Taylor series:

$$x(t+\Delta t)=x(t)+\dot{x}(t)\cdot\Delta t+\frac{1}{2}\ddot{x}(t)\cdot\Delta t^2+\frac{1}{6}\dddot{x}(t)\cdot\Delta t^3+\ldots \tag{1}$$

The further the prediction goes into the future, the more derivatives are available. Typically, only the position is available at discrete points in time, and thus the derivatives can only be obtained by a numerical differentiation. However, the higher derivatives become too noisy to be used without any additional filtering, so we use position and velocity only. To further improve accuracy, we do not utilize consecutive measurements, but, rather, from a distance of 30 frames in the system's backlog (Fig. 1). This means that the velocity is calculated over the last 250 ms.

However, using equation (1) with the terms for position and velocity would only allow a linear extrapolation. In order to predict more complex motions as well, a simple but effective approach was used. The assumption is that for short time intervals – and thus also for short periods in which a marker is not visible – a human motion can be approximated by linearly combining two predefined movements: linear and circular. Curvatures are also possible, but these are not covered by the truncated Taylor series.



Figs. 1 (left and center) and 2 (right). Fig. 1. Velocity calculated from the last two measured points (left) and from the beginning and the end of a data set (center). Red dots represent measured coordinates (positions) of a marker over time, while the arrow marked "actual velocity" shows the actual path of the marker. When using measured coordinates with a very short difference in time, the calculated velocity can differ from the true velocity greatly (left). To improve the accuracy of the calculated velocity, measured coordinates with a greater time difference were used (center). Fig. 2. Calculation of the center of rotation (C).

Instead of estimating a line and a circle that best describe the path of the marker using the least square methods, a missing marker's position is calculated using three evenly spaced samples from the tracking system's backlog. Index 1 denotes the latest (newest) frame in the backlog, Index 2 the intermediate one, and, Index 3, the older one. P_1, P_2, and P_3 denote the positions, while $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$ denote the velocities.

Linear case. The algorithm approximates a line into the historical data that would be extended to predict future coordinates. For this case, it is assumed that a missing marker will move with the same velocity as the latest/newest one. Thus, it is:

$$P = P_0 + \vec{v}_1 * dt \tag{2}$$

Where:

$$P_0 = (P_1 + P_2 + P_3)/3 \tag{3}$$

Circular case. For a circular movement, four variables must be known:

1. Center of rotation (Center of the circle)
2. Axis of rotation (normal of a plane that is defined by two vectors)
3. Radius of the rotation
4. Rotational velocity ω

Assuming that the motion is non-linear, the vectors $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$ are not parallel (Fig. 2). Each pair of vectors defines a plane with the normal vectors:

$$\vec{n}_1 = \vec{v}_1 \times \vec{v}_2 \quad \text{and} \quad \vec{n}_2 = \vec{v}_2 \times \vec{v}_3 \tag{4}$$

Better approximations to the actual velocities in P_1 and P_2 are $(\vec{v}_1 + \vec{v}_2)/2$ and $(\vec{v}_2 + \vec{v}_3)/2$. Using these, we can find two vectors, $\vec{d}_1$ and $\vec{d}_2$, both pointing approximately towards the center of rotation (Fig. 2).

$$\vec{d}_1 = \left(\frac{\vec{v}_1}{2} + \frac{\vec{v}_2}{2} \right) \times \vec{n}_1 \quad \text{and} \quad \vec{d}_2 = \left(\frac{\vec{v}_2}{2} + \frac{\vec{v}_3}{2} \right) \times \vec{n}_2 \tag{5}$$

The center of rotation, C, can be found at the point where the two vectors intersect. Using the line distance algorithm, we find two points, H and Q that approximate C:

$$C=(H+Q)/2 \tag{6}$$

Finally, the radius, r, of the rotation is obtained by:

$$r=|P_1-C| \tag{7}$$

Linear or circular case? In this next step, it has to be determined how a line or a circle approximates the measured points in the backlog. Using the least square method, the resulting errors for the line or the circle are calculated.

In the case of a line, the deviation is the shortest distance between one of the samples, P_1, P_2, or P_3, and the line, $P_0+\vec{v}_1*t$ as found using the following equation:

$$d_i=\left|(P_i-P_0)-\left(\frac{\vec{v}_1}{|\vec{v}_1|}*(P_i-P_0)\cdot\frac{\vec{v}_1}{|\vec{v}_1|}\right)\right|,\quad i\in[1,\ 3] \tag{8}$$

From this, the 'line error' can be calculated as:

$$e_L=\sum_{i=1}^{3}\left[P_i-P_0-\frac{\vec{v}_1}{|\vec{v}_1|}*(P_i-P_0)\cdot\frac{\vec{v}_1}{|\vec{v}_1|}\right]^2 \tag{9}$$

In the case of the circle, the resulting 'circle error' is:

$$e_C=\sum_{i=1}^{3}\big[|P_i-C|-r\big]^2 \tag{10}$$

To find out which approximation fits best, a matching factor, α, is defined to be:

$$\alpha=\frac{e_L}{e_L+e_C} \tag{11}$$

Note that:

$$0\le\alpha\le1 \tag{12}$$

Also, $\alpha=0$ can be interpreted as '100% linear' and $\alpha=1$ as '100% circular'.

Estimate of position. Let P_1 denote the last known position from the backlog, and let P' denote the extrapolated position that we want to find. The linear estimate is:

$$L'=P_1+\vec{v}_1*dt \tag{13}$$

In order to find the circular estimate, we calculate the angular velocity, ω.

$$\vec{c}_i=P_i-C \tag{14}$$

The vector:

$$\vec{c}_N = \frac{P_1 - C}{|P_1 - C|} \tag{15}$$

gives the direction, P_1, relative to C. Next, an approximation of the rotation axis can be calculated by:

$$\vec{n} = \frac{\vec{n}_1 + \vec{n}_2}{|\vec{n}_1 + \vec{n}_2|} \tag{16}$$

Now, the angular velocity, ω, can be calculated by:

$$\omega_1 = a\cos\left(\frac{\vec{c}_2 \cdot \vec{c}_1}{|\vec{c}_2||\vec{c}_1|}\right) \tag{17}$$

$$\omega_2 = a\cos\left(\frac{\vec{c}_3 \cdot \vec{c}_2}{|\vec{c}_3||\vec{c}_2|}\right) \tag{18}$$

$$\omega = \left(\frac{\omega_1 + \omega_2}{2}\right) * \operatorname{sgn}\left((\vec{c}_N \times \vec{n}) \times \vec{v}_1\right) \tag{19}$$

sgn stands for the sign function. The cross product results in a tangent vector of the circle. The dot product in combination with the sign function shows whether the rotation orientation is clockwise or counterclockwise.

Now, the circular estimate can be found using quaternions [14]:

$$q = \left(\cos\frac{\omega}{2}, \vec{n}\sin\frac{\omega}{2}\right);\ \ p = (0, P_1 - C);\ \ p' = (0, C' - C) = q\,p\,\overline{q} \tag{20}$$

where q, p, and p' are quaternions, $\overline{q}$ denotes the quaternion conjugate of q, and C' is the circular estimate. Finally, the estimated position P' is the result of a linear interpolation between L' and C' using α:

$$P' = L'(1 - \alpha) + C' \cdot \alpha \tag{21}$$

To further improve the predicted results, a corrector is introduced. It consists of a constraint matrix (CM) (or inter-marker distance matrix) which stores the minimal and maximal recorded pairwise distances between all markers. All markers (n) are identified by an index 1 … n. The record of two markers, say *a* and *b*, can be found by looking up the corresponding row and column. Note that the entry (a, b) describes the same relation as entry (b, a) and thus the matrix is symmetric.

Another optimization was to remove all correlations between markers that have large differences between their minimum and maximum measured distances. This was done by setting *min* and *max* to negative and positive infinity, while the difference *max – min > threshold*. An appropriate value of this threshold is 10 to 20 centimeters.

Hence, the matrix stores all rigid relations (like a marker on the elbow compared to one on the shoulder) and ignores all non-rigid relations.

4 Measurements without Constraint Matrix (WOM)

The system is built upon the assumption that the trajectories are either linear, circular, or a combination of both. The prediction is a linear combination of both, as shown in (21). After an experiment with a linear test, we set up a circular test. The data presented here is based on the circular test. The setup consisted of the Qualisys Motion Capture System [13]. The frequency of the cameras was 120 frames per second (FPS).

4.1 Linear and Circular Setup

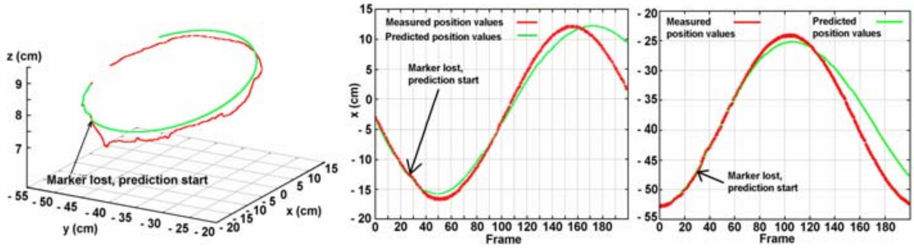
The linear test used a sloped groove, in which a spherical marker was rolled, thus providing an almost linear trajectory (Fig. 3). At one point along the groove, the marker was hidden from the camera in order to apply our algorithm. Note that the sloped groove also allowed some unwanted movements to the side in the y-direction.

We also realized a setup to measure a circular motion of the marker. The marker was hidden from the camera at a certain point of the circle (Fig. 4). Due to the simplified setup, the movement is not an ideal circular motion in the horizontal plane only.

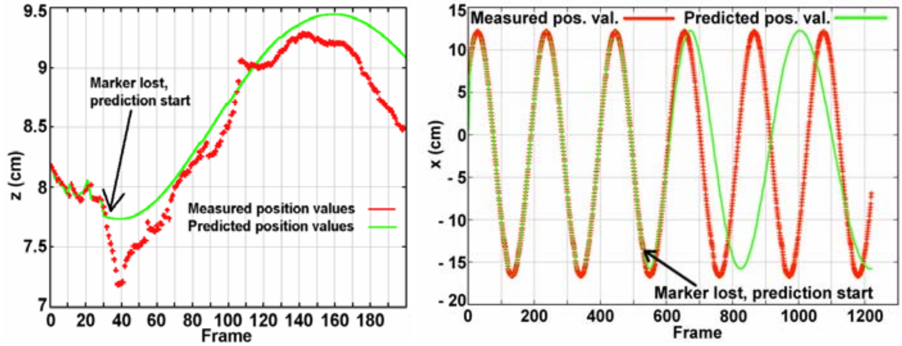
Figs. 3 (left) and 4 (right). Setups for the linear and circular measurements.

4.2 Measurement Results

To examine the performance of the proposed algorithm for a typical case, we show the results for the (almost) circular movement of a single marker. Data is predicted as soon as the marker disappears. The prediction is based on the 30 latest frames stored in the system's backlog. To determine how many frames the prediction algorithm is useful for, the prediction continues even after the marker reappears. While most of the graphs show a reduced data set, i.e. only frames 500 – 700 (Figs. 5 – 8). Fig. 9 shows the complete data set. The marker disappeared at frame 531 (or 31) and re-appeared in the next frame. The prediction algorithm was then triggered. Before the missing data frame, predicted data is a copy of measured position values. The three-dimensional plot in Fig. 5 shows that when the marker disappears, the prediction estimates a fairly circular movement. This indicates that the α-coefficient used to combine linear and circular movement is close to, but not exactly equal to one due to an imperfect setup. While an absolutely circular movement of the marker corresponds to a constant frequency, a linear movement corresponds to a frequency equal to zero. Thus, an elliptic movement is a superposition of a constant frequency and a constant component. Due to the recursive character of the prediction, the constant value is superimposed again



Figs. 5-7 (left to right measured (red) and predicted (green) data, arrow is where marker disappeared). Fig. 5. Circular movement; shown in three dimensions. 200 frames at 120 FPS are shown. Fig. 6. x-axis of the circular movement. The arrow indicates frame 31, where the marker disappeared. Fig. 7. y-axis of the circular movement.



Figs. 8 (left) and 9 (right): Measured (red) and predicted (green) data. Fig. 8. z-axis of the circular movement. Fig. 9. x-axis of the circular movement over the complete experiment. The marker disappeared at frame number 531.

and again. This explains the observed growing cycle duration (Figs. 6 – 8), which becomes even clearer in the complete data set (Fig. 9). In the horizontal plane (Figs. 6 – 7), the prediction algorithm works well for 150 frames corresponding to 125 ms of missing data. Along the vertical axis (Fig. 8), it becomes evident that the α-coefficient correctly mediates between the linear and circular estimation of the movement.

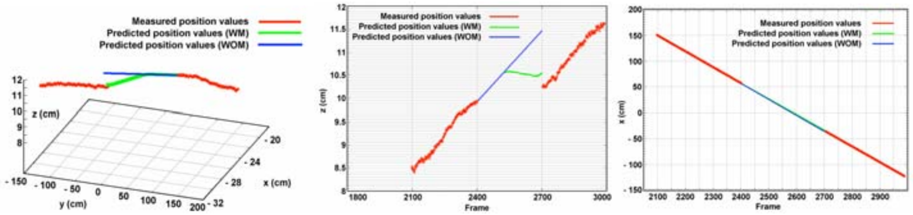
For the results shown here, without the constraint matrix, the algorithm continued to work also after the missing marker reappeared.

5 Measurements with Constraint Matrix (WM)

We developed a filter that stores min/max values of distances sampled between each (n) marker and all other (n-1) markers. This gives an n x n matrix, where each cell contains a min-max tuple of two floats, initially set to min = inf. and max = 0. For each frame and each marker visible, the matrix is updated. This matrix then serves as a filter in the case of a missing marker. After a first estimation of the missing marker's position, this estimate can be adjusted using the matrix. For each missing marker in a frame, any other (n-1) marker is used as a reference marker if the difference (here: 10 cm) between the min and max distance is below a predefined threshold. If it is above, the markers are considered to be independent; hence the constraint is marked to be invalid. For each reference marker, the estimated position of the missing marker is successively adjusted towards each reference marker. If a reference marker itself is a missing marker, the adjustment is halved.

5.1 Linear Setup

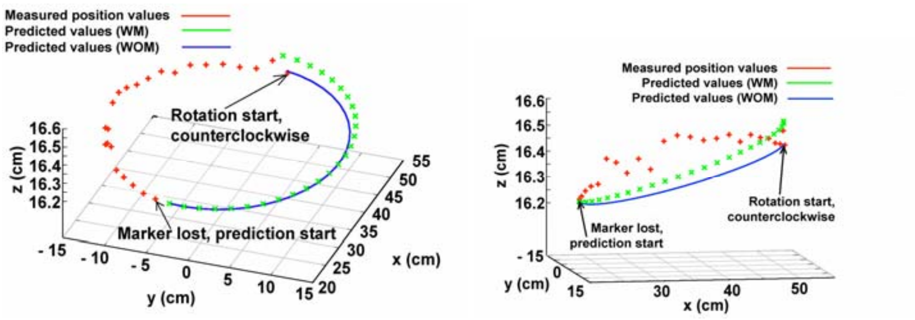
Figs. 10, 11, and 12 are generated from the data collected in the case of an almost linear movement. Fig. 10 shows the movement from the right to the left. The movement was generated using a car carrying three markers. There was a marker on each side and on the top. The latter disappeared at frame 2403 and reappeared at frame 2703, being lost for a total of 327 frames, which is an unusually long time. In Fig. 11, at frame 2524 the two prediction modes start to output significantly different values. While the prediction without the constraint matrix (WOM) gives a steep rising trajectory based upon the last 30 frames (see straight line in Figure 11), the prediction with the matrix (WM) is constrained by the other visible markers (see curved line in Figure 11). Once the missing marker reached the maximum allowed position by the constraints, it re-approached the other markers. This allows detecting and also abrupt changes in the movement and correcting the trajectory (see Figure10). This behavior cannot be seen in Fig. 12 since the car moved in a more ideal straight line along the x-axis rather along the other axes.



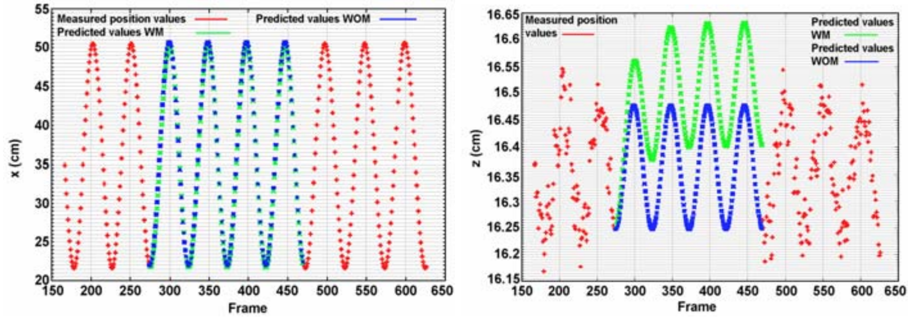
Figs. 10-12 (left to right, measured (red) and predicted (green) data). Fig. 10. Linear movement in all three dimensions: the direction of movement is from right to left. Fig. 11. Linear movement along the z-axis. Fig. 12. Linear movement along the x-axis.

5.2 Circular Setup

In this test, with its set-up shown in Figure 4, two markers are used. The first marker was placed in the center of rotation while the second was placed on the end of one of the spinning arms. At frame 274 the second marker was lost and the prediction started. The second marker reappeared at frame 470, making for a gap of 196 frames. In Figs. 13 and 14, one complete spin can be observed. In Figs. 15 and 16, the complete circular experiment is depicted. The predicted values with (WM) and without (WOM) the constraint matrices are shown. In Fig. 15, the behavior of both prediction modes seems accurate. Then, in Fig. 16, the behavior along the z-axis shows how the



Figs. 13 (left) and 14 (right). Measured (red), predicted (WM) (green), and predicted (WOM) (blue) data. Fig. 13. Circular movement. Fig. 14. Same as in Fig. 13, but from a different angle. Now it is more evident where the predictions (WM) and (WOM) start to differ.



Figs. 15 (left) and 16 (right). Measured (red), predicted (WM) (green), and predicted (WOM) (blue) complete circular trajectory data. Arrows indicate start of rotation (right) and marker disappearance (left). Fig. 15. x-axis, algorithms (WM)/(WOM) behave correctly. Fig. 16. z-axis, prediction with (WM) affected by central marker, vibrating along z-axis.

prediction mode with the constraint matrix (WM) is influenced by the position of the central marker, which vibrated along the z-axis.

6 Human-Body Movement

The proposed method (WM) was also applied to a less predictable movement such as that of a walking person. Here, seven markers were attached to one side of a person's shoulder, elbow, hand, hip, knee, ankle, and foot. The missing marker was chosen to be the one on the ankle in the moment in which the foot approaches and points to the ground and the knee moves forward to generate the successive step. The marker is supposed to move downward until the foot has touched the ground and then stay still until the foot is lifted from the ground and moved forward. The prediction was accurate enough to fake a visually satisfactory human kinematic for the ankle.

7 Conclusions and Outlook

While we avoided making use of knowledge about the tracked object (R1), we have still not tested whether the results scale other experimental platforms (R2). The results showed that the suggested algorithm (WOM) delivers valuable data for up to 150 ms of occlusion. For longer periods, prediction may decrease in accuracy. This is due to the α-coefficient that combines linear and circular movements (11). Hence, an initially small linear or circular component may increasingly cause predicted data differences. However, under normal conditions it is quite unusual that markers disappear for a longer period of time. Thus, the proposed algorithm copes with most of the missing marker problems. To further improve the prediction, a constraint matrix (WM) was introduced, storing all inter-marker distances that are above a certain threshold. Now, missing markers can be predicted more easily. If no reference marker is visible, the matrix can not correct the predicted trajectory for the missing markers and the only effect is to keep all missing markers in a geometrical context to each other.

References

1. Menache, A.: Understanding Motion Capture for Computer Animation and Video Games. Morgan Kaufmann Publishers Inc., San Francisco (1999)
2. Hashiguchi, J., Nivomiya, H., Tanake, H., Nakamura, M., Nobuhara, K.: Biomechanical Analysis of a Golfswing using Motion Capture System. Proceedings of the Annual Meeting of Japanese Society for Orthopaedic Biomechanics 27, 325–330 (2006)
3. Broeren, J., Sunnerhagen, K.S., Rydmark, M.: A Kinematic Analysis of a Handheld Stylus in a Virtual Environment: A Study in Healthy Subjects. Journal of NeuroEngineering and Rehabilitation 4(13) (2007)
4. Aristidou, A., Cameron, J., Lasenby, J.: Real-Time Estimation of Missing Markers in Human Motion Capture (2008)
5. Wiley, D., Hahn, J.: Interpolation Synthesis of Articulated Figure Motion. IEEE Computer Graphics and Applications 17(6), 39–46 (1997)
6. Nebel, J.: Keyframe Interpolation with Self-collision Avoidance. In: Eurographics, pp. 77–86. Springer Computer Science, Heidelberg (1999)
7. Rose, C., Cohen, M.F., Bodenheimer, B.: Verbs and Adverbs: Multidimensional Motion Interpolation. IEEE Computer Graphics and Applications 18(5), 32–40 (1998)
8. Rhijn, A., Mulder, J.: Optical Tracking and Automatic Model Estimation of Composite Interaction Devices. In: Proceedings of IEEE Virtual Reality Conference, pp. 135–142 (2006)
9. Herda, L., Fua, P., Plänkers, R., Boulic, R., Thalmann, D.: Skeleton-based Motion Capture for Robust Reconstruction of Human Motion. In: IEEE Proceedings of Computer Animation, p. 77 (2000)
10. Siemens: Jack Human Modeling and Simulation, `http://www.plm.automation.siemens.com/en_us/products/tecnomatix/assembly_planning/jack/index.shtml` (accessed 4/2/2009)
11. Hornung, A., Sar-Dessai, S.: Self-calibrating Optical Motion Tracking for Articulated Bodies. In: Proceedings of IEEE Virtual Reality Conference 2005, pp. 75–82 (2005)

12. Catmull, E., Rom, R.: A Class of Local Interpolating Splines. In: Barnhill, R.E., Reisenfeld, R.F. (eds.) Computer Aided Geometric Design, pp. 317–326. Academic Press, London (1974)
13. Qualisys Inc., `http://www.qualisys.com/` (accessed 5/2/2009)
14. Hanson, A.: Visualizing Quaternions. In: ACM Siggraph 2005 Course (2005)
15. Furniss, M.: Motion Capture. Massachusetts Inst. of Technology Comm. Forum (2005)
16. Molet, T., Boulic, R., Thalmann, D.: A Real-Time Anatomical Converter for Human Motion Capture. In: Boulic, R., Hegron, G. (eds.) Eurographics Workshop on Computer Animation and Simulation 96, pp. 79–94 (1996)
17. Welch, G., Foxlin, F.: Motion Tracking: No Silver Bullet, but a Respectable Arsenal. IEEE Computer Graphics and Applications, 24–38 (2002)
18. Aliaga, D.G., Xu, Y., Popescu, V.: Occlusion-Resistant Camera Design for Acquiring Active Environments. In: IEEE Computer Graphics and Applications, pp. 68–78 (2007)
19. Hornung, A., Sar-Dessai, S., Kobbelt, L.: Self-calibrating Optical Motion Tracking for Articulated Bodies. Proceedings of IEEE Virtual Reality 2005, 75–82 (2005)

Motion Analysis of the Arm Based on Functional Anatomy

Charles Pontonnier[1] and Georges Dumont[1,2]

[1] IRISA/BUNRAKU,
Campus de Beaulieu,
F-35042 Rennes Cdex, France
[2] ENS Cachan Antenne de Bretagne,
Campus de Ker Lann,
F-35170 Bruz, France

Abstract. This article presents a biomechanical model of the right arm, developped with respect to the functional anatomy of the human. This model is developped as a motion analysis tool. The main application issued from this model is the muscle forces estimation of the flexion/extension and internal rotation of the forearm joints. This estimation is based on an inverse dynamics method, improved with additionnal constraints such as co-contraction factor between flexors and extensors of a joint. The article first presents the related work, then presents the biomechanical model developed, and at last some results obtained for a sample movement of the arm.

1 Introduction

The proper design of working stations has a direct impact on the working conditions. Ergonomics takes an important place as a factor of productivity. Indeed, musculoskeletal troubles are often associated with the working stations in the industry, damaging the health of the worker and decreasing the productivity. In France, the increase of musculoskeletal troubles at work is very consequent since the addition of the table 57 in September 1991 into the professional diseases legislation. This table mainly includes the articular troubles, particularly the troubles associated to the upper extremities. These musculosketetal troubles represent 75 percent of the professional diseases in France [1]. The purpose of this article is the presentation of a biomechanical model of the upper extremity that could be usable to decrease the musculoskeletal troubles. One of the solutions to improve the working conditions is to estimate and analyze muscle forces developed by a human during its work tasks. This estimation is a good way to improve the ergonomics [2], because of the tools that can be used to perform the analysis [3]. The analyzing tool presented with the biomechanical model is a muscle forces estimation method.

The reminder of this article is structured as follows : first we present the related work that is the basis of our study. Next we present our biomechanical model and its analysis tool in three steps, as described above. Our results present

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 137–149, 2009.

estimated muscle forces involved in flexion/extension and internal rotation of the forearm joints for a captured motion. We discuss about these results with respect to the literatture and others biomechanical models. At last, we conclude on the method and its limits and we present the future fields that we want to explore.

2 Related Work

This section is divided in two parts. First we present several biomechanicals models that present interesting methods and results. We detail the performances of these models in order to show the relevance of the model we have developped. The second part presents several works that are used as a basis for our method of estimation of the muscle forces.

2.1 Biomechanical Models for Motion Analysis

Currently, several biomechanical models are developped around the world. All of them present different advantages and drawbacks, reaching different goals.

Delp and his collaborators have developed an open-source platform named OpenSim [4]. This platform allows the user to develop dynamic simulations on a musculoskeletal system with motion capture data, using an original algorithm (Computed Muscle Control) based on muscles excitations to provide muscle forces. The computation time is very long (10 minutes for 3 seconds of simulation). An upper extremity model developed by Holzbaur and al. in [5] is available on this platform. Figure 1 shows this upper extremity model. The model allows the user to develop motion analysis from motion capture data.

Yamane and al. [6] have designed a complete musculoskeletal system representing human including 366 muscles driving 155 degrees of freedom. It computes muscle forces from motion capture involved in a walk in 1 second per frame, so it can not be used to drive real-time application. It is based on a physiological and physical approach, including an optimisation algorithm.

Chalfoun has developed a simulation toolkit for the forearm and the hand [7]. The musculoskeletal system is very realistic, using a complete muscular topology

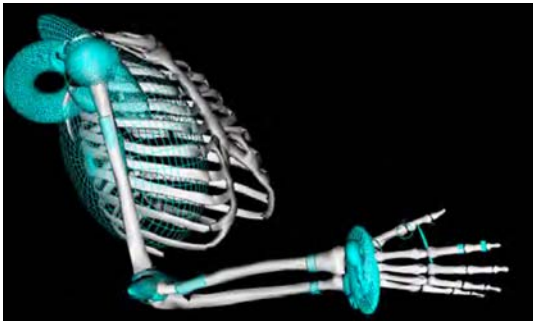


Fig. 1. Upper Extremity model designed with OpenSim [5]

of the hand and the forearm. It provides good results that have been compared to EMG data. This platform does not provide any interactive tool, the motion has to be modelled to obtain any result.

Maurel developed a musculoskeletal model for the shoulder [8]. This model is very realistic and has precise information about muscular topology of the arm and shoulder kinematics. The information about muscular topology are interesting in order to compute muscle moment arms in our estimation.

These musculoskeletal systems have a great interest in our study, and are good support for our work. The main advantage of these models are their realism. The results obtained with OpenSim and the Yamane's model are sufficiently validated to be used as a comparison. Nevertheless, we want to have muscle forces data to apply ergonomic tools on them, so we have to develop a faster and simpler technique, in order to implement it in real time. This is the main drawback of these models, they are too heavy and slow to be used dynamically.

2.2 Muscle Forces Estimation

There are two main approaches for studying muscle forces involved in human motion, the first is the forward dynamics approach and the second is the inverse dynamics approach. Buchanan presented these two methods in detail [9]. This section deals with the advantages and drawbacks of the two methods.

Fig. 2. Forward dynamics approach for estimating muscle forces involved in human motion [9]

Figure 2 presents the forward dynamics method. This method is divided in five different steps. This first step is based on electromyogramms (EMG) data acquisition and is also the main advantage of the forward dynamics method. The second step is the estimation of the muscle force by a muscular activation model that is directly applied muscle by muscle. The lack of knowledge on muscular modeling and muscular parameters [10] is the main limitation of the method (third step). The model has to be fitted in order to estimate muscle forces that are dynamically consistent. Signal processing related to the EMGs treatment constitutes the second limitation. The signal obtained from the electrodes presents various noises (activity of other muscles near the measured muscle), so the signal has to be processed with many arbitrary elements (filters, optimization loops). At last, the instrumentation is quite heavy because of the number of electrodes that should be equal to the number of muscle activities that are to be measured. The fourth and fifth steps are checking steps that allow comparing muscle forces obtained from the forward dynamics method with positions obtained, for example, with motion capture data.

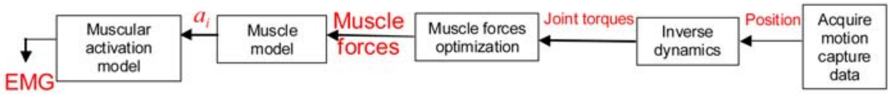


Fig. 3. Inverse dynamics approach for estimating muscle forces involved in human motion [9]

Figure 3 presents the inverse dynamics method for studying human movement. This method is divided in different steps from right to left (these steps are very close to the forward dynamics method steps) . The inverse dynamics method uses kinematics data (such as motion capture data) in order to compute muscle forces by using an inverse dynamics method. The second step is an inverse dynamics step. The main advantage is that muscle forces are issued from mechanical equations end thus are dynamically consistent. The third step is an optimization step.

The first drawback of this method is caused by the redundancy of the muscular topology for each joint. Generally, many muscles are involved in one joint motion, and several muscles are working on two or three different joints. This redundancy implies the use of optimization methods to obtain muscles forces from joint torques data. The redundancy implies another important limitation: the muscle forces issued from the computation do not take into account the muscle co-contraction. Without any information on muscular antagonism, many muscles do not have the activity that they are supposed to have. The method that we propose in this article takes the co-contraction into account by using an additional constraint in the optimization formulation. At last, the inverse dynamics method could be slow because of this optimization step. This problem is an important limitation in a virtual reality approach that implies interaction with a real human and so that has to deal with real-time applications. We propose a solution to this problem in the last part of the article. The fourth and fifth steps are checking steps that allow comparing the estimated muscle force to EMG measures. The instrumentation for the experimentation is mainly a motion capture system that is well adapted to be an input for virtual reality application.

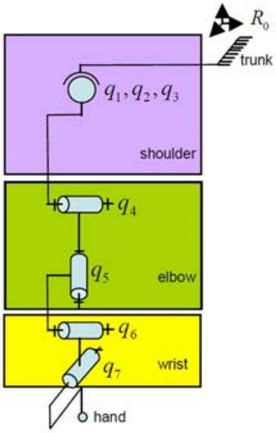
3 Biomechanical Model of the Right Arm

Our model is divided in three distinct parts. In this section we first present the kinematical model we have developed. Next we present the dynamical model developed to obtain joint torques from joint positions, velocities and accelerations issued from the inverse kinematics computation. At last we present the musculoskeletal model and the method developed to estimate muscle forces from the two precedent models.

3.1 Kinematical Model

This kinematical model is based on functional anatomy [11]. Figure 4 presents the model and Table 1 presents the joint limits.

In order to analyse a motion, we use motion capture data (issued from a VICON©system) and an inverse kinematics algorithm that computes the joint positions associated to the motion. The motion capture gives us position data for each segment of the model, we have developed an inverse kinematics method based on a succesive replacement of each segment one by one. We compute the inverse kinematics algorithm for each segment, following the chain segment by segment downward (see Figure 4) from the shoulder to the wrist.



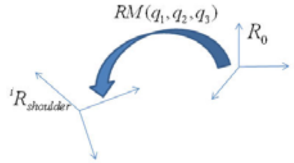
Our study is based on the elbow motion and the following joints in the kinematical chain. We consider the shoulder as a spherical joint. We do not take into account the constitution of the joint; we only use it to place the arm in the good position and orientation. For our range of motion, the spherical approximation is sufficient in order to rebuild the motion. In a further development we may use the Maurel's model [8]. The elbow is modeled as a gimbal joint that allows the flexion/extension and the internal rotation of the forearm. At last, the wrist is represented by two rotations that correspond to the flexion/extension and to the adduction/abduction of the hand.

Fig. 4. Kinematical model of the right arm

Table 1. Joint functions and bounds of the kinematical model of the right arm

Joint	Function	Bounds(°)
q_1	Flexion/Extension of the arm	[0,180]
q_2	Abduction/Adduction of the arm	[-60,90]
q_3	Internal rotation of the arm	[-90,30]
q_4	Flexion/Extension of the forearm	[0,150]
q_5	Internal rotation of the forearm	[0,165]
q_6	Flexion/Extension of the hand	[-80,90]
q_7	Abduction/Adduction of the hand	[-50,20]

To obtain joint positions, we resolve a reduced system of equations issued from the equality between the real orientation matrix computed from markers positions and a combination of the rotation matrix issued from the kinematical model. For example, in order to orient and place the shoulder, we will solve a reduced system decribe below in figure 5. The main advantage of this method is its computation time, which is near 1 ms for one frame treatment.



$$RM(q_1,q_2,q_3) = [rot(\overrightarrow{y_0}, q_1)] * [rot(\overrightarrow{x_1}, q_2)] * [rot(\overrightarrow{z_2}, q_3)]$$
$${}^iR_{shoulder} = [{}^i\overrightarrow{x}_{shoulder}, {}^i\overrightarrow{y}_{shoulder}, {}^i\overrightarrow{z}_{shoulder}]$$
$$RM(q_1,q_2,q_3) = {}^iR_{shoulder}$$

Fig. 5. Definition of orientation of the shoulder for the frame i, where $RM(q_1, q_2, q_3)$ is the rotation matrix issued from the combination of the rotation angles q_1,q_2 and q_3 defined above. ${}^i\overrightarrow{x}_{shoulder}$,${}^i\overrightarrow{y}_{shoulder}$ and ${}^i\overrightarrow{z}_{shoulder}$ are the three vectors constituting the direct orthonormal basis associated to the shoulder for the frame i.

The results obtained with this method allow us to compare the position of the reconstructed markers and the position of the real markers issued from the motion capture data. The main error is near 1%, which represent in our application a maximal distance between real and reconstructed markers near from 1,5 mm.

3.2 Dynamical Model

The dynamical model that we have developed is realized by using the Matlab© Simmechanics toolbox. The right arm is modelled as an articulated system of rigid bodies. The inertia parameters are defined for each motion capture subject with respect to the De Leva tables [12] and are automatically scaled with a precomputing algorithm that we have developed [13]. The rigid bodies are connected with perfect mechanical joints with respect to the kinematical model (figure 4).

This model allows us to compute the joint torques involved in the motion. A numerical derivation is applied to the joint positions obtained from the inverse kinematics step. The simulation applies the Newton laws of dynamics to the system and thus computes the joint torques. These torques are then used to compute the muscle forces.

A visualization of the motion is associated to this model. This model is constructed according to the kinematical model defined in figure 4 and is an assembly of a geometric definition of each segment [14].

3.3 Musculoskeletal Model

The musculoskeletal model is designed in two steps. First we modelize the muscles and attach them to the skeleton. At last we define an optimization algorithm to compute muscle forces.

Muscular Topology and Functional Anatomy: To obtain a fine analysis of the forces developed by the muscles during the motion, we have to properly define their topological situation on the target skeleton, their role (in which functional joint of the arm they are involved) in the motion, and at last their properties and capabilities. We have chosen to define the muscles as viscoelastic actuators.

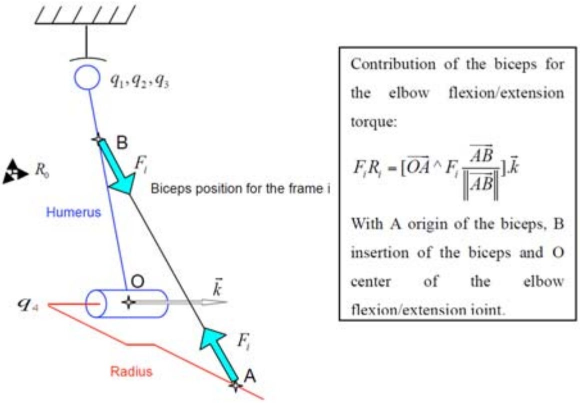


Fig. 6. Computation of the contribution of the biceps for the elbow flexion/extension torque

This approach is commonly accepted in the biomechanical community. We can cite for example the work of Zajac [10]. This model is useful to define the cost function to minimize and the constraints associated that allow us to estimate muscle forces, as it is described further.

At each frame we rebuild the muscle origins and insertions in the main coordinate system to obtain the moment arms related to each joint. Figure 6 shows how the contribution of the biceps is computed for the flexion/extension of the elbow motion. The muscle is considered as a mechanical actuator and the force developed by the muscle is oriented in the main direction of the muscle. Furthermore, muscles work only in contraction. The origins and insertions of the muscles are defined on the basis of Clinically Oriented Anatomy [11]. We use the topological information issued from this book to define our own origins and insertions with respect to the dimensions of the target skeleton. The origins and insertions are scaled on the basis of the humerus and radius length. Our model does not provide any warping point or via point in the muscle action lines, such as in Holzbaur's work [5], so that could be a further improvement if our results are not sufficient with the actual definition of the muscles. The computation of the muscle moment arms leads to the definition of equilibrium constraints between the computed torques issued from the inverse dynamics step and the muscle contribution for each joint, as it is described in the next section.

Table 2 summarizes the adimensonal coordinates of the muscles involved in the flexion/extension and the internal rotation of the forearm. Muscles with multiple origins and insertions are modeled with only one origin and one insertion. In order to use ergonomics tools [3], we only define one force for one muscle.

Muscle Forces Estimation: As previously mentioned, the estimation of muscle forces with an inverse dynamics method is a redundant problem. The equilibirum of one joint leads only to one equation for many muscle forces. These equations express that the torques obtained from the inverse dynamics are equal

Table 2. Adimensional origins and insertions of muscles

Muscle	Origin (adimensional)	Insertion (adimensional)
Biceps	$[-0.085 \;\; -0.043 \;\; 0.043]_{^{i}R_{shoulder}}$	$[-0.045 \;\; 0.045 \;\; -0.099]_{^{i}R_{elbow}}$
Triceps	$[0.043 \;\; -0.021 \;\; 0.128]_{^{i}R_{shoulder}}$	$[0.062 \;\; 0 \;\; -0.037]_{^{i}R_{elbow}}$
Brachialis	$[-0.037 \;\; 0 \;\; -0.531]_{^{i}R_{shoulder}}$	$[-0.05 \;\; -0.025 \;\; -0.124]_{^{i}R_{elbow}}$
Brachioradialis	$[-0.043 \;\; 0 \;\; -0.798]_{^{i}R_{shoulder}}$	$[0 \;\; -0.099 \;\; -0.943]_{^{i}R_{elbow}}$
Anconeus	$[0.043 \;\; 0.085 \;\; -0.957]_{^{i}R_{shoulder}}$	$[0.037 \;\; 0.025 \;\; -0.124]_{^{i}R_{elbow}}$
Pronator teres	$[0.043 \;\; 0.106 \;\; -0.957]_{^{i}R_{shoulder}}$	$[0.05 \;\; 0.025 \;\; -0.298]_{^{i}R_{elbow}}$
Pronator quadratus	$[0.037 \;\; 0.05 \;\; -0.868]_{^{i}R_{elbow}}$	$[0.037 \;\; -0.074 \;\; -0.868]_{^{i}R_{elbow}}$
Supinator	$[-0.198 \;\; 0 \;\; 0.05]_{^{i}R_{elbow}}$	$[0 \;\; 0.099 \;\; -0.55]_{^{i}R_{elbow}}$

to the sum of the contributions of each muscle related to the joint. The most common and natural solution to this problem is to minimize a cost function under non-linear constraints that represents the equilibrium equations, the physiological limits and functions of the muscles and the muscular topology of the arm. Our work is mainly focused on the design of these constraints, whose have to be chosen and developed to perform the most realistic muscle forces estimation.

The choice of the cost function is not obvious. Several works are studying different cost functions to estimate realistic muscle forces. The main work that we have used to make our choice is Challis' work [15]. The chosen cost function is the normalized sum of the squared muscle stresses. This choice is strongly related to the hypothesis that biomechanical energy is minimized for the human movements. We associate constraints to the cost function in order to represent the physiology of the muscles. Equation 1 shows the chosen cost function to minimize.

$$f = \sum_{m} \left(\frac{F_i}{(F_{\max})_i} \right)^2 \tag{1}$$

In which F_i is the force applied by muscle i and $(F_{max})_i$ is the maximum isometric force developed by the same muscle. As we proposed in [16], the maximum isometric force developed by a muscle depends on the joint configuration. The work of Von Konsky [17] allows us to adjust these capabilities for the muscle involved in the flexion/extension of the forearm for each frame. The lack of knowledge about maximum capability of the muscles involved in the internal rotation of the forearm obliges us to consider a constant maximum capability for these muscles, as it is proposed in [5].

Physiologically, the muscles are activated by pair (or pair of muscle groups). In the case of an elbow flexion/extension, when the flexors (Biceps, Brachialis and Brachioradialis) are activated, the extensors (Triceps and Anconeus) are activated too because of the antagonism of the joint. To express this co-contraction phenomenon, we use an additional equality constraint:

$$h_3(F_i) = F_1 + F_3 + F_4 - \alpha_{q_4}[F_2 + F_5] = 0 \tag{2}$$

In which F_1,F_3,F_4 are respectively the forces relative to the Biceps, the Brachialis and the Brachioradialis,F_2 and F_5 are respectively the force relative to the

Triceps and the Anconeus. $\alpha(q_4)$ is the co-contraction factor for the elbow flexion/extension that is issued from an experimentation computed with OpenSim. This first study is a simple treatment of a captured extension of the elbow. We obtain a ratio between the flexors activities and the extensors activities. We name this ratio the co-contraction factor. Co-contraction factor depends on the position of the flexion/extension joint. We use OpenSim as a reference to develop this co-contraction factor. The interest of this co-contraction factor has already been discussed in a precedent article [18]. This article presents the results obtained with and without the co-contraction factor for the elbow flexion/extension. The main goal of this additional constraint is to translate the muscle behavior contained in a complete simulator like OpenSim, without any physiological behavior computation.

At last, we add the equalities constraints that represent the equilibrium between the contribution of each muscle involved in each motion and the corresponding torque issued from inverse dynamics computation :

$$h(F_i) = c - M_c * F = 0 \tag{3}$$

In which c is the vector of the computed joint torques (issued from the inverse dynamics computation), M_c is the contribution matrix, including the moment arms computed at each frame as described in the last section, and F the vector of the muscle forces.

At last the optimization problem could be summarized as follows :

$$\begin{aligned} &\text{For each frame :} \\ \min f(F) = \sum_m &\left(\frac{F_i}{(F_{\max})_i}\right)^2 \ . \\ &\text{Under constraints :} \\ h_{1,2}(F) &= c - M_c * F = 0 \ . \\ h_3(F) &= F_1 + F_3 + F_4 - \alpha_{q_4}[F_2 + F_5] = 0 \ . \\ h_4(F) &= F_6 + F_7 - \alpha_{q_5} F_8 = 0 \ . \\ g_i(F_i) &= F_i - (F_{\max})_i \leq 0 \ . \end{aligned} \tag{4}$$

4 Results and Discussion

In this section we present some results obtained with the model for a sample movement of the forearm.

Figure 7 presents the joint positions and torques issued from the kinematical and dynamical models. The considered joints are the flexion/extension and the internal rotation of the forearm. The sample movement is simple. The flexion/extension joint presents an angle between 40° and 10° (0° corresponds to a complete extension of the forearm) and the computed torque varies between 0 and 4 newton meters. Those results show that the joint torque is pretty weak during the motion. Actually, the subject does not hold any additional load (use a tool, hold a box). For the internal rotation, the variation of the angle is quite null (only four degrees) and the computed torque is about 1.10^{-3} N.m. This is due to the fact that this joint only holds the distribued loads of the forearm.

Figures 8 and 9 show the estimated forces for the muscles involved in the two joints during the movement. The muscles present an activity that is related to their order of importance. For example, the flexors of the forearm (Biceps, Brachialis, Brachioradialis) are working during the motion, and the main flexor (Brachialis) presents the main activity. The forces obtained are contained in a range of values that are realistic if we consider the joint torques associated. We can see that the co-contraction factor allows the system to compute a related activity of the extensors (Triceps, Anconeus). In the case of the anconeus, it does not represent a big activity. It is due to its minor role in the joint motion. The main limitation of these results is the lack of continuity of the computed muscle forces. Actually, the muscles are considered as viscoelastic actuators, and we have to define some additional rules in the computation in order to take it into account. The first improvement we can add to the algorithm is to obtain a better co-contraction factor. Currently the co-contraction factor is related to the joint position. Our idea is to obtain a law relying it to the joint velocity too. This improvement should allow us to improve the realism of the computed forces.

The lack of validation for these results is the most important step we have to complete in our next work. The results presented here needs to be at least compared with OpenSim results. We are currently working on OpenSim in order to obtain comparative results. We want to ensure that the addition of the co-contraction factor leads to a global behavior of the musculoskeletal system that provides accurate results for the muscle forces estimation. At last, we are currently recording some new motion data in order to improve the global results (co-contraction factor definition) and see different working cases.

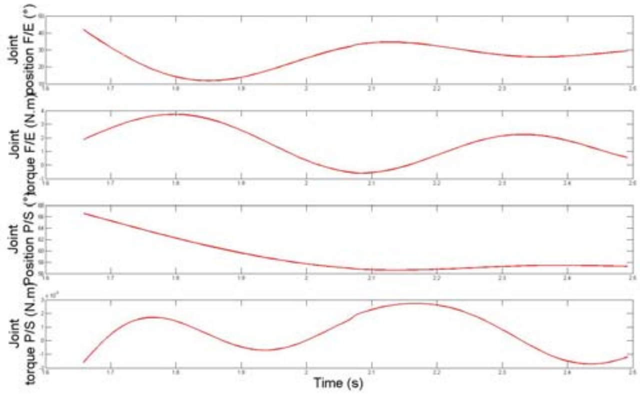


Fig. 7. Inverse dynamics results for a sample movement of the forearm

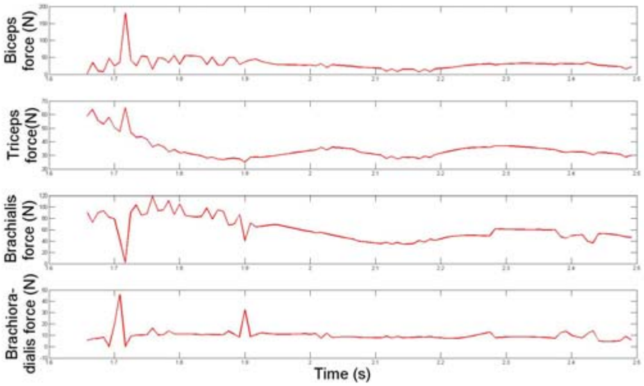


Fig. 8. Estimated muscle forces for a sample movement of the forearm

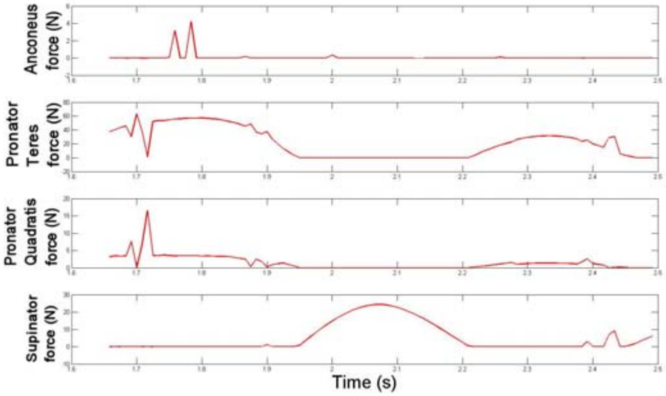


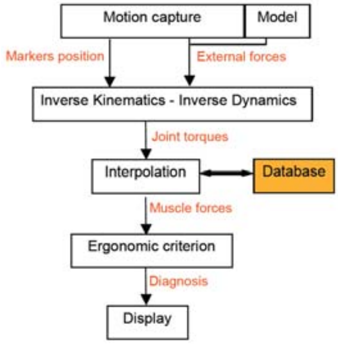
Fig. 9. Estimated muscle forces for a sample movement of the forearm

The mean computation time for a frame is 0.04 second, computed on a Pentium IV dual core 2,3GHz processor. This result corresponds to a frequency of analysis about 25 Hz. This is a good result compared to the literatture but it is not sufficient for real-time application.

5 Conclusion and Perspectives

This article presents a biomechanical model of the right arm designed in order to analyse motion for working station design. We have detailed the different parts of the model. The model presents a kinematical part, a dynamical part and at last a biomechanical part. The model is based on functional anatomy. We present some results for a sample motion of the right arm. The analysis of the flexion/extension and internal rotation of the forearm joint is the basis of the discussion about the model.Our results of muscle forces estimation are realistic, but the method presents a lack of continuity that does not correspond to the viscoelastic model [10] that is commonly accepted for the muscles. We have to improve the method to improve the results. The forces obtained are comparable to the ones presented in the litterature, but need further validation (comparison with OpenSim's results, EMG data).

At last, once the method improved, we will replace the optimization step with an interpolation step (see figure 10), that presents better characteristics in terms of computation time. This is the next step in our work.



To use the method in VR sessions, we have to reach 100 Hz for the I/O computation. The main improvement we consider is to interpolate the muscle forces instead of optimizing them online. Our work now is to compute a pool of results in order to constitute a database of muscle forces. This database, computed offline, will allow us to interpolate the muscle forces in online applications. Interpolation does not take much time to compute, so we think that we can decrease the computation time by four with this improvement of the method.

Fig. 10. Real-time application architecture

References

1. Aptel, M., Aublet-Cuvelier, A.: Prévenir les troubles musculosquelettiques du membre supérieur: un enjeu social et économique. Santé Publique 17(3), 455–469 (2005)
2. Bidault, F., Chablat, D., Chedmail, P., Pino, L.: Distributed approach for access and visibility task under ergonomic constraints with a manikin in a virtual reality environment. In: Proceedings of 10th IEEE International Workshop on Robot and Human Communication, pp. 32–37 (2001)
3. Ma, L., Chablat, D., Bennis, F., Zhang, W., Guillaume, F.: A new muscle fatigue and recovery model and its ergonomics application in human simulation. In: Proceedings of IDMME Virtual Concept, pp. 1–10 (2008)
4. Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., Thelen, D.G.: OpenSim: Open-Source software to create and analyze dynamic simulations of movement biomedical engineering. IEEE Transactions 54, 1940–1950 (2007)
5. Holzbaur, R.S., Katherine, M.M., Wendy, D.S.L.: A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. Annals of Biomedical Engineering 33, 829–840 (2005)
6. Yamane, K., Fujita, Y., Nakamura, Y.: Estimation of physically and physiologically valid somatosensory information. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2624–2630 (2005)

7. Chalfoun, J.: Prédiction des efforts musculaires dans le système main avant-bras: Modélisation, simulation, optimisation et validation. PhD thesis, Université de Versailles Saint-Quentin en Yvelines (2005)
8. Maurel, W., Thalmann, D.: A Case Study Analysis on Human Upper Limb Modelling for Dynamic Simulation. Computer Methods in Biomechanics and Biomechanical Engineering 2(1), 65–82 (1999)
9. Buchanan, T.S., Loyd, D.G., Manal, K., Besier, T.F.: Neuromusculoskeletal Modeling: Estimation of Muscle Forces and Joints Moments and Movements From Measurements of Neural Command. Journal of Applied Biomechanics 20, 367 (2004)
10. Zajac, F.E.: Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Critical Reviews in Biomedical Engineering 17, 359–411 (1989)
11. Moore, K.L., Dalley, A.F.: Clinically Oriented Anatomy, 4th edn. Lippincot Williams Wilkin (1999)
12. De Leva, P.: Adjustments to Zatsiorsky-Seluyanovs segment inertia parameters. Journal of Biomechanics 29, 1223–1230 (1996)
13. Pronost, M., Dumont, G.: Dynamics-based analysis and synthesis of human locomotion. The Visual Computer, 23 (2007)
14. Copyright ©2001 University of Brussels (ULB), `http://www.ulb.ac.be/project/vakhum`
15. Challis, H.J.: Producing physiologically realistic individual muscle force estimations by imposing constraints when using optimization techniques. Med. Eng. Phys. 19, 253–261 (1997)
16. Pontonnier, C., Dumont, G.: From Motion Capture to Muscle Forces in Human Elbow Aiming at Improving Ergonomics of Working Stations. In: Proceedings of IDMME Virtual Concept, pp. 1–10 (2008)
17. von Konsky, B.R., Zomlefer, M.R.: The sensitivity of upper limb muscle moments to variation in physiological cross-sectional area. In: Proceedings of the 18th Annual International Conference of the IEEE Bridging Disciplines for Biomedicine, vol. 5, pp. 2242–2243 (1996)
18. Pontonnier, C., Dumont, G.: Inverse Dynamics Methods Using Optimization Techniques for the Estimation of Muscle Forces Involved in the Elbow Motion. International Journal on Interactive Design and Manufacturing 78 (2009)

WAPA: A Wearable Framework for Aerobatic Pilot Aid

Xavier Righetti, Sylvain Cardin, and Daniel Thalmann

VRLAB, EPFL, Lausanne, Switzerland
{Xavier.Righetti,Sylvain.Cardin,Daniel.Thalmann}@epfl.ch

Abstract. Disorientation induced by G-forces during aerobatic flight generates difficulties for the pilots to perfectly align their aerobatic maneuver. This paper presents a modular wearable system for enhancing training of aerobatic pilots. A combination of accelerometers and a gyroscope is used to detect possible deviations compared to the optimum trajectory. The wearable system informs the user in real time about the corrections to apply via vibrotactile actuators and speech synthesis. This publication presents a work in progress in order to validate the system in simulation.

Keywords: wearables, user interaction, task guidance, vibrotactile, accelerometer.

1 Introduction

Aerobatic flying pushes the pilot's biological conditions to its limit. The main exercise in aerobatic flight is to align perfectly the airplane during the various imposed maneuvers. G-forces induced by the movement of the airplane create vestibular dysfunction [1]. The vestibular apparatus is responsible with proprioception for self representation for our body orientation in space. Deficiency of the sensory system under such circumstances leads to errors in trajectories and in severe cases to accidents.

In demanding situations such as piloting a plane, the user should focus at 100% on his task and thus cannot be distracted by displays and other information outputs. The use of wearable devices is especially tailored for this kind of applications where the user's hands, eyes and attention are actively engaged with the physical environment.

In this paper, we present an approach, which is currently a work in progress: the user will select the modules that he wants to wear depending on the functionalities he wants to use. We focus on using wearable intelligent clothing to provide extra sensory information for aerobatic pilots.

Next section will describe the current state of the art concerning the use of wearables in task intensive situations, while section three will introduce our proposition. Section 4 will describe the setup of our experiments and finally, we will present our conclusions and remarks about future work in the last section.

2 State of the Art

Various researches have been performed using vibrotactile stimulation to improve self-orientation in different environment. Some work focuses in altered gravity

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 150–158, 2009.

environment like heavy and 0 G-force for pilot and astronaut [2] [3]. Those researches show improvements of the mental representation of the body orientation in space. The use of Pneumatic and electromagnetic actuators located in a vest to give directional information about the position of different targets has been tested for correcting drift during aircraft piloting [4]. The feedback from the test pilots indicates that this form of tactile feedback reduces drastically the attention need for self-orientation perception and target localization. Furthermore, it has been proved to be intuitive and accurate. Measurements of trajectory error and response time between the use of the Tactical Situation Awareness System (TSAS), and an additional visual indicator show clear improvements using tactile feedback over visual display. Also, Joint research, done at the department of experimental psychology at the University of Oxford and the Transport Research Laboratory of Wokingham, studies the influence of vibrotactile collision warning signals in a driving simulation [5]. Their setup used two vibration actuators fixed on the user belt, one in front and one in the back. The driver is also provided with a three-state visual indicator to show if he is too close, too far or just at a good distance from the car ahead. The trial is to follow a car that will accelerate and brake at random times with disabled brake lights. This experiment shows a great improvement in response time while using the vibrotactile actuator.

Many researches have shown the potential of attaching accelerometers for sensing body motion. Typical applications include activity recognition [6] [7], gesture recognition [8], medical rehabilitation [9] and ambulatory monitoring [10]. Alternately, MIThril, a complete wearable platform [11], has been developed to facilitate the development of distributed real-time multimodal and context-aware applications. Used in conjunction with the WearARM [12], a wearable computing core, researchers are trying to build a high-performance wearable system that is truly wearable and unobtrusive. For every wearable implementation, one must always choose a tradeoff between bulkiness and performance. In our approach, we favored the “wearability” of the system over the processing power, ruling out specific applications such as mixed reality guidance where the use of computer vision requires intensive calculations. This kind of application can be seen in [13], where the authors have investigated the use of wearable computers to provide task guidance in aircraft inspection. In the preliminary results, they noticed that the user tends to over-rely on the computer and that the user’s performance is directly related to the design quality of the user interface.

3 Proposition

As we have seen in the previous section, various garments with embedded computing devices input and output systems have been developed but they are usually made for one purpose. In our case, we have envisioned to build a versatile jacket with an embedded I2C bus (cf. Fig. 1) and magnetic connectors in order to let the user keep the same garment and choose the desired functionalities according to the modules that are attached to it. This paper presents an application of versatile wearable device applied to aerobatic piloting.

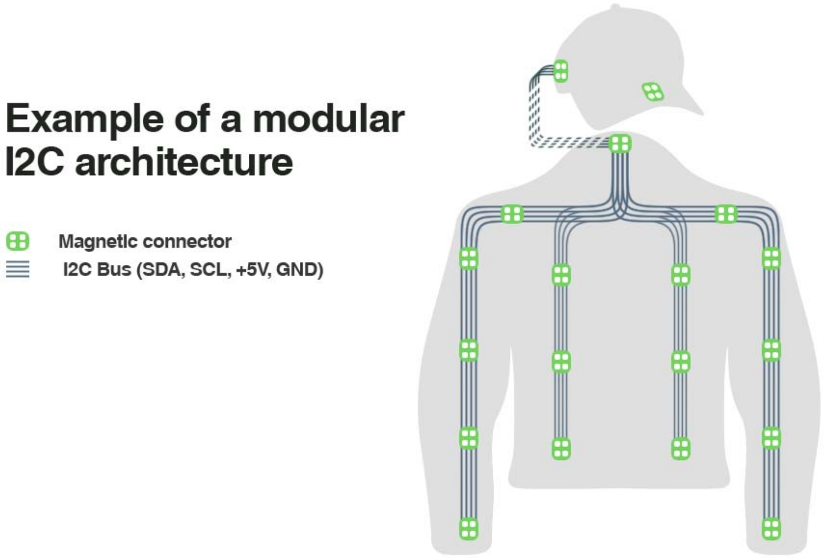


Fig. 1. Schematic view of the I2C bus embedded into a jacket

In this case, the jacket is designed to be worn by an aerobatic pilot in order to help him improve the perfect achievement of aerial maneuvers. Therefore, wearable sensors like gyroscopes, compasses and accelerometers can e used to detect the plane's attitude. We believe that sensors attached to the body could give quite accurate measures as the pilot is attached very tightly to his seat. Of course, information such as the plane's attitude is already available in the flight instrument of the aircraft. Using external sensors avoid us the need to physically link our system to the aircraft insuring that the jacket is perfectly wireless and thus avoiding the need to alter the in-flight system. Actuators, like vibrotactile devices and speakers, inform the user about eventual trajectory corrections to take into account. The communication between the pilot and his/her wearable system is achieved in such a way that neither hands nor the eyes are distracted from the piloting task.

3.1 Human-Machine Interface Scenario

The user is in control of the aircraft at a sufficient altitude and the area is clear of other aircraft or obstacles. The pilot wears our system embedded into his jacket and follows a straight path with no significant wind.

The pilot selects the aerobatic maneuver to perform by selecting the appropriate icon onto the touch screen located on his left forearm. The icon represents a standard perfect maneuver, roll or loop, computed accordingly with the current aircraft behavior.

The text to speech module confirms orally which maneuver has been selected and starts a count down from ten seconds. At the end of the countdown the pilot should initiate the maneuver while the text to speech module announces the angle at which the aircraft should be all along the realization of the maneuver.

The vibrotactile system indicates the error between its current orientation and the predicted orientation using the prerecorded standard maneuvers as a comparison. The vibrotactile system stimulates the user upper body skin by localized vibrations. A vibration on the left indicates that the user should roll more on his left; a vibration in the back indicates that he should pitch more up. The intensity of the vibration matches the size of the deviation error.

4 System Architecture

This section presents the architecture of the final system as it will be used inside the real aircraft. This setup is still actually under development and needs some additional testing before a real in-flight validation. A communication bus and power lines are embedded inside the jacket along with magnetic connectors in order to attach functionality modules in a plug-and-play manner. The modules are composed by a master module including a tactile screen for user interaction, an orientation tracker module to recover the aircraft orientation, a text to speech module for vocal commands and a set of vibrotactile modules for enabling vibrotactile stimulation of the pilot.

4.1 Communication Bus

We have designed our system in such a way that all wearable modules are connected together using the two wires I2C protocol. I^{2}C (Inter-Integrated Circuit) is a simple half-duplex[1] serial 2-wire bus developed by Philips in the early 1980's for efficient communication between different ICs (Integrated Circuits). It has been designed originally to solve many interfacing problems such as synchronization issues when designing digital control circuits and has now become a world standard. In our implementation, the I2C bus is composed of four wires: two for transmitting data (clock signal and data signal) and two for powering the modules with a 5V regulated voltage. The connectors for attaching the modules are made of a simple flexible PCB, as seen on Fig. 2. On top of it, four magnets are fixed with silver epoxy conductive glue. The magnets are arranged in a way that they repel the module if the user tries to attach it the wrong way.

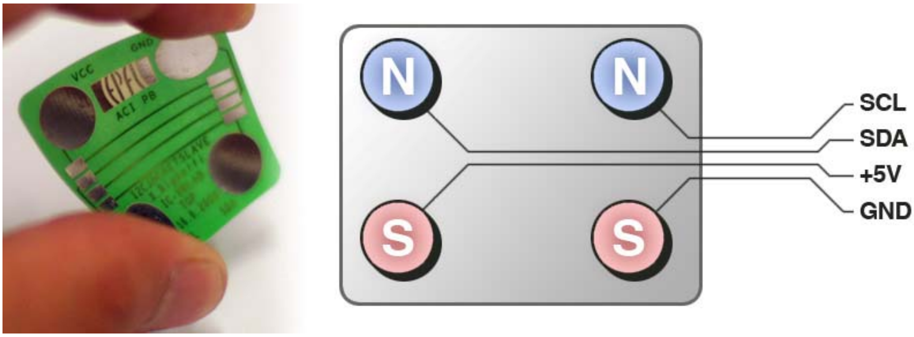


Fig. 2. Flexible connector and magnets arrangement

---

[1] Transmission of information in opposite directions but not simultaneously.

4.2 Master Module

The master module is responsible for managing the communication for all slave modules, centralizing the data coming from the sensor, processing the data and controlling the embedded actuators such as the vibrotactile modules and the text-to-speech module. It is composed of an 8-bit microcontroller (PIC18F8722), a Bluetooth adapter (Bluegiga WT11) and an I2C driver. It includes also an OLED tactile display allowing the pilot to calibrate the modules and to select the maneuver he or she wants to perform.

4.3 Slave Orientation Tracker Module

Our slave orientation tracker module uses the MTx sensor from Xsens, which is a small and accurate 3DOF Orientation Tracker. It provides drift-free 3D orientation as well as kinematic data: 3D acceleration, 3D rate of turn and 3D earth-magnetic field. This sensor allows us to compute the deviation error according to current orientation and rotation rates in comparison to the precomputed optimal trajectory for the maneuver.

4.4 Slave Text-to-Speech Module

The text-to-speech module announces the name of the maneuver to perform and count down to start it. At the end, a message is generated to inform the user about his error rate. Moreover, the module is connected with the headphone system and thus improves the interaction with the user by giving auditory feedbacks. This module can be used in two ways: sentences can be prerecorded and spoken at specific times or some ASCII text can be fed and spoken on the fly.

4.5 Slave Vibrotactile Modules

A combination of 8 vibrotactile modules allows the pilot to correct his trajectory in real time by applying vibrations on the upper body torso skin. The actuators are simple DC motor with an asymmetric mass lot originally designed for mobile phones. The position and the strength of the vibration inform the user about the eventual deviation to the optimal precomputed trajectory.

5 Experimental Setup

This part presents the current state of our development. Before putting our system on board of a real aerobatic aircraft we need to validate the principles of interaction. It is important to notice that in flight experimentations are difficult to perform due to the inherent cost of the aircraft and the availability of experimented pilots. In order to validate the principles, we developed a simulation system where the plane is replaced by a flight simulator. The goal of this setup is to be as close as possible from the real environment. Thus, the simulator offers a realistic airplane behavior and coherent piloting conditions using a throttle box and a joystick. Although this setup will not be sufficient to assess the full efficiency of the current system due to the lack of G-forces and stress pressure induced in the real situation, this system has been considered

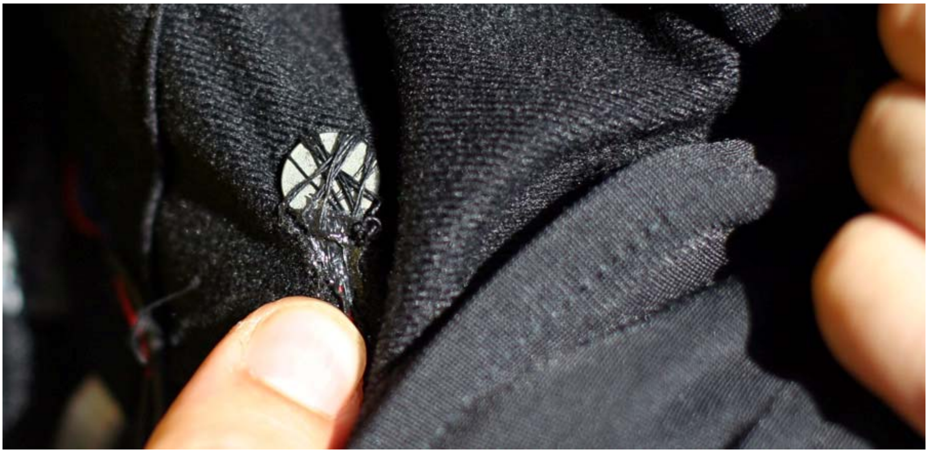


Fig. 3. One of the 32 vibration motor embedded in the jacket

sufficient enough to provide a validation platform for test and calibration of our final system.

There are two major differences between the final and the experimental. First, the experimental setup is built around the flight simulator X-plane[2]. So the orientation tracker module is disabled, since the user is not inside a moving aircraft. Instead, the orientation information is directly gathered from the flight simulator. X-plane allows the development of custom plug-ins. We have implemented a UDP server to recover the different orientation parameters. We have used the Euler angles values and their derivative. They have been compared between a “perfect maneuver” and the current airplane behavior to determine the error factor.

The second difference is the vibrotactile feedback. In order for us to find the best place to place the vibrotactile actuator modules, we used a jacket developed during a previous research [14]. This jacket includes 32 vibration motors from mobile phone technology (cf. Fig. 3). The jacket is connected as a slave module inside the wearable platform in the same manner that the single vibration actuator module. The vibrotactile jacket is worn under the modular jacket. The system is also connected to a computer that controls the simulation, recovers the orientation and trajectory of the aircraft, and accordingly generates the vibration feedback. It also includes a calibration interface for the vibrotactile actuator. This external device communicates wirelessly using Bluetooth with the master module of the wearable system. This system is only needed to tune up the vibration feedback in the simulation. For the final setup, the whole feedback generation algorithm will be implemented inside the slave vibration modules. This external calibration interface allows us to test and modify in real time different configurations for the vibration actuators with different parameters of placement, timing and intensity.

---

[2] www.x-plane.com

6 Test Procedure

In order to validate our system, we have used a sample population of persons that were asked to fly an aerobatic aircraft inside the simulator. As shown in Fig. 4, the user is seated in front of the computer screen, wearing our system and using a throttle box and a joystick to pilot the plane. During the experiment, the user has to press on the correspondent icon onto the tactile screen to select the maneuver to be performed. He waits for the countdown and starts the maneuver. We assume that every test subject is able to perform the two recommended basic maneuver which consists in a roll and a loop.

That assumption is really delicate since the performance is linked to the previous piloting experience. So we have let users familiarize themselves with the simulator for enough time for them to be able to keep a constant direction and perform the required basic maneuvers. It could require from a few minutes to several hours depending of the subject.

To determine of the influence of the wearable system from one user to another, we used only the difference between trajectories with or without the help of tactile feedback. The difference by comparison with an established standard and perfect maneuver reflects the ability of the subject to perform the task. The goal is to demonstrate if the user's performance is improved when using our wearable system on a large enough population.

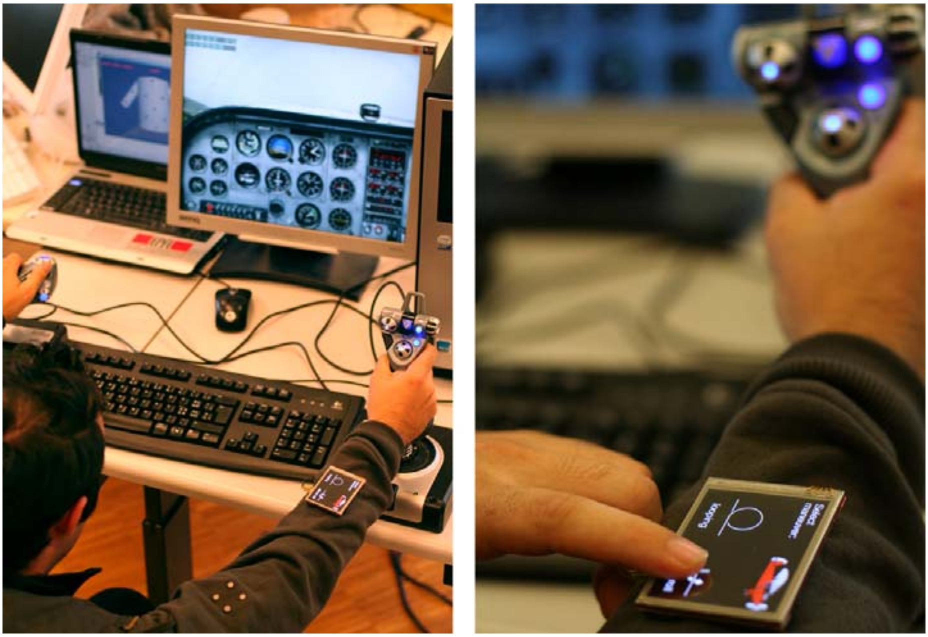


Fig. 4. Experimentation

User acceptance is also a crucial question when you want to assess the impact of a HCI-related technology. In order to recover user feedback we have developed a small questionnaire concerning their background concerning piloting task, their confidence and feeling about our wearable system. We have focused questions on self representation of the aircraft behavior, vibrotactile feedback coherence, overall system comfort and user interest in the system.

7 Discussion and Future Work

We have obtained a fully functional simulation platform and were able to perform the overall calibration of the system. The number of vibrotactile modules has been set to 8, two located on the chest, two on each arm and two in the back. This disposition allows the user to sense in an effective manner the orientation angles, roll and pitch, of the aircraft.

We have performed the experiment with a few intern subjects in order to tune up the different parameters of the simulation and the results are encouraging. We were able to detect slightly better performances while using vibrotactile feedback. The main issue we have encountered is the difficulty to perform loops and rolls correctly without much experience on the simulator. Also, the system is efficient enough for small trajectory correction but if the airplane deviates a lot from its trajectory, it becomes very difficult to give a coherent feedback.

The next step is to perform large test study in simulation to properly validate our proposition. Those tests can be a good opportunity to invite and get feedback from real aerobatic pilots. We need to ensure that the virtual model is realistic and corresponds to an aerobatic airplane that the pilots have already flown. Moreover, in order to finalize our setup for a real aerobatic experimentation, we need to manufacture the slave vibrotactile actuator modules and implement the vibration feedback inside the distributed microcontroller boards.

Acknowledgments. The work presented was supported by the EU project INTERMEDIA (38419), in the frame of the EU IST FP6 Programme.

References

1. Muller, T.U.: G-induced vestibular dysfunction (“the wobblies”) among aerobatic pilots: a case report and review. Ear Nose Throat, 269–272 (2002)
2. Traylor, R., Tan, H.Z.: Development of a Wearable Haptic Display for Situation Awareness in Altered-gravity Environment: Some Initial Findings. In: Proceedings of the 10th Symposium on Haptic interfaces For Virtual Environment and Teleoperator Systems HAPTICS, p. 159. IEEE Computer Society, Washington (2002)
3. van Erp, J.B.F., Van Veen, H.A.H.C.: A multipurpose tactile vest for astronauts in the international space station. In: Proc. Eurohaptics 2003, pp. 405–408 (2003)
4. Raj, A.K., Suri, N., Braithwaite, M.G., Rupert, A.H.: The tactile situation awareness system in rotary wing aircraft: Flight test results. In: Proceedings of the RTA/HFM Symposium on Current Aeromedical Issues in Rotary Wing Operations. RTO NATO, Neuilly-sur-Seine, France, pp. 16.1-16.7 (1998)

5. Ho, C., Reed, N.J., Spence, C.: Assessing the effectiveness of "intuitive" vibrotactile warning signals in preventing front-to-rear-end collisions in a driving simulator. Accidents Analyses and Prevention 38, 988–996 (2006)
6. Ermes, M., Parkka, J., Mantyjarvi, J., Korhonen, I.: Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. IEEE Transactions on Information Technology in Biomedicine 12, 20–26 (2008)
7. Van Laerhoven, K., Gellersen, H.W.: Spine versus porcupine: A study in distributed wearable activity recognition. Proc. of the Eighth IEEE Intl. Symposium on Wearable Computers 1, 142–149 (2004)
8. Cho, I., Sunwoo, J., Son, Y., Oh, M., Lee, C.: Development of a Single 3-axis Accelerometer Sensor Based Wearable Gesture Recognition Band. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 43–52. Springer, Heidelberg (2007)
9. Bonato, P.: Advances in wearable technology and applications in physical medicine and rehabilitation. Journal of NeuroEngineering and Rehabilitation 2 (2005)
10. Karantonis, D.M., Narayanan, M.R., Mathie, M., Lovell, N.H., Celler, B.G.: Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE transactions on information technology in biomedicine 10, 156–167 (2006)
11. DeVaul, R., Sung, M., Gips, J., Pentland, A.: Mithril 2003: applications and architecture. In: Proceedings of the Seventh IEEE International Symposium on Wearable Computers, pp. 4–11 (2005)
12. Lukowicz, P., Anliker, U., Tröster, G., Schwartz, S.J., DeVaul, R.W.: The weararm modular, low-power computing core. In: IEEE micro, pp. 16–20 (2001)
13. Ockerman, J.J., Pritchett, A.R.: Preliminary Investigation of Wearable Computers for Task Guidance in Aircraft Inspection. In: Second International Symposium on Wearable Computers, pp. 33–40 (1998)
14. Cardin, S., Thalmann, D.: Vibrotactile Jacket For Perception Enhancement. In: Proc. International Workshop on Multimedia Signal Processing, Cairns, Australia (2008)

Discriminative Human Full-Body Pose Estimation from Wearable Inertial Sensor Data

Loren Arthur Schwarz, Diana Mateus, and Nassir Navab

Computer Aided Medical Procedures, Technische Universität München, Germany
`{schwarz,mateus,navab}@cs.tum.edu`,
`http://campar.cs.tum.edu/`

Abstract. In this paper, a method is presented that allows reconstructing the full-body pose of a person in real-time, based on the limited input from a few wearable inertial sensors. Our method uses Gaussian Process Regression to learn the person-specific functional relationship between the sensor measurements and full-body pose. We generate training data by recording sample movements for different activities simultaneously using inertial sensors and an optical motion capture system. Since our approach is discriminative, pose prediction from sensor data is efficient and does not require manual initialization or iterative optimization in pose space. We also propose a SVM-based scheme to classify the activities based on inertial sensor data. An evaluation is performed on a dataset of movements, such as walking or golfing, performed by different actors. Our method is capable of reconstructing the full-body pose from as little as four inertial sensors with an average angular error of 4-6 degrees per joint, as shown in our experiments.

Keywords: Human pose estimation, Gaussian process regression, wearable inertial sensors, ambulatory motion analysis.

1 Introduction

Human motion analysis plays an important role in various fields of application, such as industrial ergonomic studies or sports engineering, but especially in the medical domain. Precise and quantitative representations of human motion can help physicians in diagnosis, treatment planning and progress evaluation. For instance, orthopedists use motion analysis to assess the status of diseases affecting the skeletal system [1]. Neurologists analyze the movements of epilepsy patients to identify the cortical locations that cause epileptic seizures [2].

Typical systems for acquisition of full-body motion work with optical tracking systems based on a stationary multiple camera setup. The person is often required to wear cumbersome markers or suits and to stay within a dedicated tracking volume. However, medical motion analysis can be most beneficial when patient movements are monitored over extended periods of time and during activities of daily life. In such cases, a pose estimation system using wearable body sensors as an input is of great advantage. Patients equipped with a few wireless

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 159–172, 2009.

and light-weight sensors on their extremities could move naturally and follow their everyday routine, while their movements are recorded. In contrast to existing ambulatory motion analysis systems that record only specific signals, such as heart rate or step count [1,3,4], a mobile pose estimation system could provide physicians with a reconstruction of full-body movements.

However, inferring full-body pose from inertial sensor data, such as acceleration or orientation of individual body segments, is a challenging problem. The sensor measurements typically exhibit high drift rates and provide only a limited number of constraints for analytical calculation of human pose, e.g. using inverse kinematics [5,6,7]. Instead, we propose to use prior knowledge about feasible human poses obtained from sample motions in a learning-based approach. In a training phase, the person-specific functional relationship between the low-dimensional sensor data and the full-body pose, acquired with a motion capture system, is learned. After training, our method allows reconstructing the pose of a person in real-time, given only the data of a few inertial orientation sensors attached to the limbs (Figure 1). The focus of our method is not a generalization across different actors, but the ability to model the particular motions of individual persons for evaluation, as in the case of medical motion analysis.

The problem of estimating human pose from low-dimensional input data can be approached using *generative* or *discriminative* learning techniques. In the former case, one searches for the joint angle configuration of a simplified skeleton model that best fits the measured input data [8,9,10,11]. This typically requires using costly iterative optimization algorithms. On the other hand, *discriminative* methods learn a direct functional relationship from input data to joint angles, allowing efficient pose prediction. Gaussian Process Regression (GPR) is a discriminative learning technique that has been applied to predict full-body pose from image-based input data, such as human silhouettes [12,13,14,15]. We propose to employ GPR for learning the direct mapping from inertial sensor data to full joint angle configurations. To the best of our knowledge, a discriminative learning-based pose estimation approach using observations from wearable sensors has not been proposed before.

In our method, person-specific GPR models are learned for each activity of interest, such as walking or particular disease-related movements. The person is required to perform sample movements using a motion capture system to generate training data. After training, the predictive means of the GPR models provide pose estimates from wearable sensor data. Additionally, in order to determine which of the learned activities a person is performing, we propose to use a multi-class Support Vector Machine (SVM) classifier, trained on the wearable sensor dataset with the known activities as class labels.

The main contribution of this paper is to demonstrate that full human body pose can be estimated based on just a few wearable inertial sensors using a discriminative learning approach. This opens the way for creating a wearable pose estimation system suitable for long-term evaluation of full-body motion during daily life. The quantitative evaluation of our method shows that the full-body pose can be estimated for the activities we considered (Section 3.1)

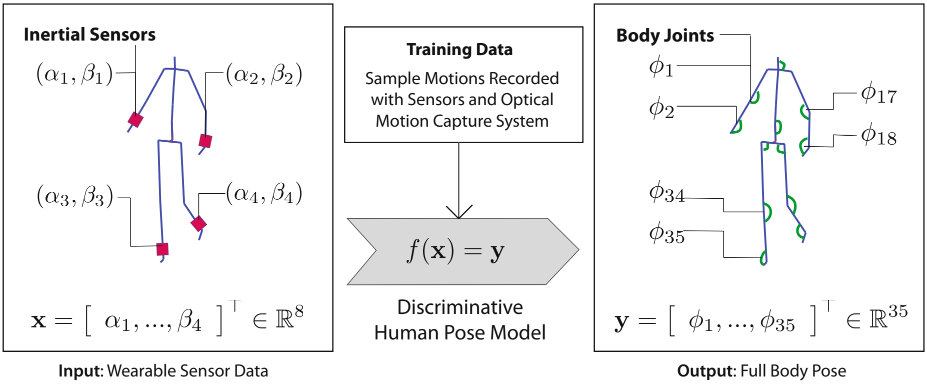


Fig. 1. Schematic of the proposed pose estimation approach. An actor is equipped with a few inertial sensors measuring orientation angles. Full body pose is predicted using a Gaussian Process Regression model that has been trained on the functional relationship between the sensor data $\mathbf{x}$ and the joint angle configuration $\mathbf{y}$.

with average errors of 4-6 degrees per joint, using four, three and even two inertial orientation sensors. This reconstruction quality is comparable to other state-of-the-art methods that use higher-dimesional input for discriminative pose estimation [12,13,16,17].

1.1 Related Work

There is a large variety of learning-based approaches for human pose estimation in the literature [12,13,14,15,16,17]. Many recently published methods are *generative* and focus on learning low-dimensional representations of human poses using so-called latent variable models [8,9,10,11]. After projecting a training dataset of poses into a low-dimensional (latent) subspace, these methods predict new poses by optimizing for the latent positions that best account for new observations. For instance, Urtasun et al. [11] use Gaussian Process Latent Variable Models (GPLVM) to learn a 3D latent space from motion capture data. Human poses are then tracked in video sequences by optimization in each frame.

Disadvantages of such *generative* pose estimation methods are their sensitivity to initialization and the high computational cost of pose inference [13]. Several authors therefore propose to directly learn a mapping from simple, low-dimensional observations to the corresponding full body poses by non-linear regression [12,17,15]. Such *discriminative* approaches have the advantage that poses can be predicted efficiently, without the need for iterative optimization.

Different types of input data have been proposed for discriminative pose estimation, such as silhouettes in 2D images [12], SIFT-features [15] or 3D voxel-representations of humans [17]. Fossati et al. [13] use trajectories of body parts in videos as an input. In contrast, our method estimates human pose from wearable inertial sensor measurements. Learning methods that have been described in the context of discriminative human pose estimation include Relevance Vector Machines [12,17], Support Vector Machines [18] and Gaussian Process Regression [13,14,15]. We use the latter technique because of its compact mathematical formulation and its computational efficiency.

Similar to our goal of reducing the dimension of the input data, Chai and Hodgins [19] reconstruct full-body pose from a low number of visual markers in an optical setup. However, the person is still constrained to stay within the camera volume. Wearable sensors allow us to acquire information on human pose without being bound to a specific location.

The goal of estimating human pose based on wearable sensors has been targeted by several authors [6,7,20], however, mostly relying on physical models, inverse kinematics and stochastic filtering. For instance, Zhou et al. [7] use accelerometers to compute the pose of one arm based on kinematic constraints. Roetenberg et al. [6] fuse inertial and ultrasonic sensor streams using a Kalman filter, allowing to track position and orientation of the four limbs and the torso. A Kalman-based approach is also described by Zhu et al. [20], where magnetic sensors are used for tracking of a single arm. Our method removes the need for complex, physical motion models, since learned prior knowledge about human movements is used to cope with limitations of low-dimensional sensor data.

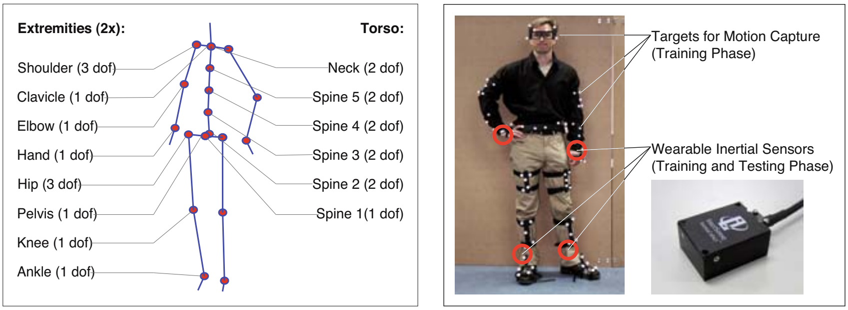
Paper Organization. In section 2, the methods we apply for pose estimation are presented. Training of human motion models is explained in section 2.3. Section 2.4 describes how our method recognizes activities from inertial sensor data. Section 2.5 explains how pose is estimated, given the learned pose models and an activity classification. A quantitative evaluation of the proposed method is given in section 3.

2 Human Pose Estimation Approach

In this paper, we propose a method to reconstruct full-body pose of an actor, given only the readings from wearable orientation sensors attached to the extremities of the actor. We propose to learn person-specific motion models for each considered activity from corresponding sets of sensor data and full-body joint angles, obtained offline by motion capture. In the testing phase, only the sensor data is used for pose estimation, allowing the person to move freely and follow their daily routine.

2.1 Method Outline

Let $\mathbf{x}_i \in \mathbb{R}^d$ be a vector containing the concatenated measurements of all inertial sensors for a single pose. Also, let $\mathbf{y}_i \in \mathbb{R}^D$ ($d \ll D$) denote a vector of joint angles that fully determines a pose in terms of the articulated body model (Figure 2(a)). The sensor measurements for a specifc activity, consisting of a sequence of M poses, are arranged in a matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]^\top$. The corresponding joint angle configurations are stored in a matrix $\mathbf{Y} = [\mathbf{y}_1 - \bar{\mathbf{y}}_n, ..., \mathbf{y}_M - \bar{\mathbf{y}}_n]^\top$. Here, $\bar{\mathbf{y}}_n$ is



(a) Articulated Body Model (b) Training Equipment

Fig. 2. a) Articulated body model consisting of five kinematic chains (arms, legs and torso). Each chain contains joints with one, two or three degrees of freedom (dof), totaling 35 dof. b) Actor equipped with reflective markers for optical motion capture and wireless inertial sensors. Markers are only worn during training.

the mean joint configuration for activity $n \in \{1, ..., N\}$ and N is the number of considered activities.

We train a GPR model $\mathcal{G}_n = \{\mathbf{X}, \mathbf{Y}, \theta\}$ for each activity, modeling the non-linear functional relationship $f_n(\mathbf{x}_i) = \mathbf{y}_i$ between the sensor readings $\mathbf{x}_i$ and the joint angles $\mathbf{y}_i$. This implies optimizing the so-called hyperparameters θ of the GPR model. In order to compute the pose estimate $\mathbf{y}_*$ from new, unseen sensor data $\mathbf{x}_*$, we first need to determine the activity n_* performed by the actor. For this purpose, we train a multi-class SVM to classify the activities directly from the sensor measurements. The full-body pose estimate is then computed from the predictive mean of the GPR model corresponding to activity n_*.

2.2 Gaussian Process Regression

This section briefly introduces the general principles of GPR. Regression methods can be used to recover an unknown functional relationship $f(\mathbf{x}) = z$, given noisy observations of the output variable $z_i \in \mathbb{R}$ at certain values of the input variable $\mathbf{x}_i \in \mathbb{R}^d$. Knowledge about this relationship allows predicting z_* for a new value $\mathbf{x}_*$. For the following paragraphs, let $\{\mathbf{x}_i, z_i\}_{i=1...M}$ be a set of training data pairs and let $\mathbf{z} = [z_1, ..., z_M]$ denote the observed function values.

By definition, a Gaussian process is a collection of random variables, any finite number of which have a joint (multivariate) Gaussian distribution [21]. One can thus model a function f with a Gaussian process, by assuming that the function values $\mathbf{z}$ are a sample from the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{K})$. Here, $\mathbf{K}$ is a covariance matrix with entries given by a covariance function k, such that $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The type of covariance function that is used, along with the hyperparameters, determines properties of the sought functional relationship $f(\mathbf{x}) = z$. In this paper, we follow the typical choice of the squared exponential covariance function [22,23]:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha^2 \exp\left(-\frac{1}{2\beta^2}(\mathbf{x}_i - \mathbf{x}_j)^2\right) + \gamma^2 \delta_{ij}, \tag{1}$$

where δ_{ij} equals 1 if $i = j$ and 0 otherwise. Here, the hyperparameters are $\theta = (\alpha, \beta, \gamma)$ [21]. γ represents the assumed noise level in the training data and β represents the length-scale. Suitable values for θ are obtained by optimization over a training dataset (section 2.3).

Adding a new pair of input and output values $\{\mathbf{x}_*, z_*\}$ to the initial distribution of function values $\mathbf{z}$, results again in a multivariate Gaussian distribution, since the function values are assumed to follow a Gaussian process:

$$\begin{bmatrix} \mathbf{z} \\ z_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k}(\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*)^\top & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right), \tag{2}$$

where $\mathbf{k}(\mathbf{x}_*)_i = k(\mathbf{x}_i, \mathbf{x}_*)$. The distribution in Equation 2 can be used to state the conditional probability $p(z_*|\mathbf{z})$ of the new output value z_* given the training data [21]. This probability, in turn, follows the Gaussian distribution $\mathcal{N}(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*))$, with mean and variance given by

$$\mu(\mathbf{x}_*) = \mathbf{k}(\mathbf{x}_*)^\top \mathbf{K}^{-1}\mathbf{z}, \tag{3}$$

$$\sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*)^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}_*). \tag{4}$$

When only a new value $\mathbf{x}_*$ is available and $z_* = f(\mathbf{x}_*)$ is unknown, the GPR estimate for z_* given the training data is the predictive mean (Equation 3). The predictive variance (Equation 4) can be seen as an assessment of prediction uncertainty. Note that prediction and uncertainty can be computed in closed form. While in the derivation above the functional relationship is assumed to be real-valued, the extension to the case of vector-valued functions is simply achieved by concatenating the one-dimensional regression result for each dimension of output space.

2.3 Learning Human Pose Models

We now describe our use of GPR to learn the functional relationship between inertial sensor data and full joint angle configurations. Given a matrix of sensor measurements $\mathbf{X}_n$ for activity n and a matrix of full joint angle configurations $\mathbf{Y}_n$, we train a GPR model $\mathcal{G}_n = \{\mathbf{X}_n, \mathbf{Y}_n, \theta\}$ by minimizing the negative log marginal data likelihood with respect to the hyperparameters θ:

$$-\log p(\mathbf{Y}_n|\mathbf{X}_n, \theta) = \frac{1}{2}\mathrm{tr}(\mathbf{Y}_n^\top \mathbf{K}^{-1}\mathbf{Y}_n) + \frac{1}{2}\log|\mathbf{K}| + c, \tag{5}$$

where c is a constant and the matrix $\mathbf{K}$ is a function of the hyperparameters (see Equation 1). During minimization, the first term fits the model to the training data, whereas the second term prevents overfitting by penalizing complex models [22]. Optimization can be done offline, for instance using standard gradient-based approaches. We use the conjugate gradient method [24].

2.4 Classification of Sensor Measurements

Since we train individual GPR models $\mathcal{G}_n$ for each activity performed by an actor, new wearable sensor data needs to be classified to one of the activities before pose estimation to choose the appropriate model. To this end, we propose to use a multi-class SVM classifier based on the standard one-against-all approach and using an RBF kernel [25]. The training data for the classifier is generated by concatenating the sensor measurement matrices $\mathbf{X}_i$ over all activities in the training dataset into a matrix $\mathcal{X}$ and supplying the known class labels for each sensor measurement in a vector $\mathcal{Y}$. We denote the SVM classification for a given input measurement $\mathbf{x}_*$ as n_*.

2.5 Pose Estimation

Once a new, unseen sensor reading $\mathbf{x}_*$ has been classified as belonging to activity n_*, we can recover the full joint angle configuration $\mathbf{y}_*$ using the corresponding GPR model $\mathcal{G}_{n_*}$. To this end, we use the predictive mean of the model, in analogy to Equation 3 (extended to the vector-valued case), which is essentially a weighted combination of the joint angle configurations in the training dataset:

$$\tilde{\mathbf{y}}_{n_*}(\mathbf{x}_*) = \bar{\mathbf{y}}_{n_*} + \mathbf{Y}_{n_*}^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}_*). \tag{6}$$

The notation $\tilde{\mathbf{y}}_{n_*}$ implies that this is only an estimate of the true, unknown joint configuration $\mathbf{y}_*$ corresponding to $\mathbf{x}_*$, as given by the model for activity n_*. The equation can be efficiently evaluated online for each incoming sensor reading, since $\mathbf{K}^{-1}$ can be precomputed.

In order to increase robustness with respect to scattered misclassifications, we extend this simple frame-by-frame classification scheme as follows. Let $\mathbf{x}_*$ be the j-th sensor measurement in the sequence. All preceding frames in a sliding window of size w are classified using the SVM classifier. The pose estimate for frame j is then a linear combination of the predictions from all models, weighted by their relative contribution within the current window:

$$\tilde{\mathbf{y}}(\mathbf{x}_*) = \sum_{n=1}^{N} \frac{\lambda_n}{w} \cdot \tilde{\mathbf{y}}_n(\mathbf{x}_*), \tag{7}$$

where λ_n is the number of frames in the current window classified as activity n. The more frames in a window are classified as a certain activity, the higher the influence of the corresponding GPR model will be. When all frames are classified identically, only one GPR model is used for pose estimation. An inherent advantage of this weighting approach is that transitions between different activities are smooth in the pose reconstruction. When the activity performed by the actor changes from one to another, the influence of both corresponding GPR models is shifted gradually (for an evaluation, see section 3.4).

3 Experimental Results

We performed several types of experiments in order to quantitatively assess our pose estimation method. We first present separate studies for the activity classification and pose estimation parts (sections 3.2 and 3.3). An evaluation of the combined approach is given thereafter (section 3.4). Finally, we illustrate the influence of the number of inertial sensors and that of the training dataset size on pose estimation results (section 3.5).

3.1 Data Acquisition

For training and evaluation purposes, we use a database consisting of motion sequences that were recorded simultaneously with the wearable sensors and a motion capture system. We considered the following generic activities: *clapping hands*, *golfing*, *jumping on one foot*, *scratching head*, *walking* and *waving hand*. Each type of motion was recorded with three different actors and six repetitions per actor. Individual sequences are between 200 and 900 frames in length.

In our experiments, we used wireless InertiaCube sensors that output orientation values (yaw, pitch and roll), filtered from an internal accelerometer, a gyroscope and a compass (www.intersense.com). We discard the yaw values, since they give the absolute orientation of the actor with respect to the earth's magnetic field, which is unnecessary for learning general properties of movements. Unless stated otherwise, we used four orientation sensors in our experiments, such that the input space is eight-dimensional ($d = 8$). Figure 2(b) shows an actor equipped with inertial sensors and markers required by the infrared tracking system (www.ar-tracking.de).

3.2 Pose Estimation Accuracy

In a first set of experiments, we evaluated the pose estimation approach assuming we know which activity is being performed in the testing data. We measured the error between the joint angles estimated with our approach and the ground truth obtained from the optical motion capture system. The acquired database of motions was used for cross-validation experiments, i.e. repeated measurements were performed for every activity and actor, each time using one of the sequences in the dataset for testing and the remaining ones for training. The error for activity n is computed per frame as

$$e_{\text{ang}} = \frac{1}{D} \sum_{i=1}^{D} \left| \, [\mathbf{y}]_i - [\tilde{\mathbf{y}}_n]_i \, \right|, \tag{8}$$

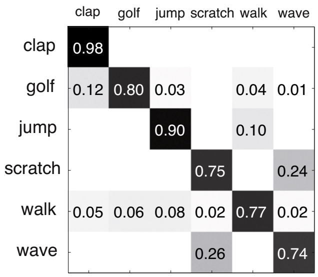
where D is the number of joints, $\tilde{\mathbf{y}}_n$ is the vector of joint angles estimated using Equation 6 and $\mathbf{y}$ is the corresponding ground truth joint configuration. The mean error over all frames in the testing sequence and the standard deviation are listed in Figure 3(a). Typical reconstruction errors lie between 4 and 7 degrees

e_{ang}	μ	σ
clap	6.23°	1.61°
golf	6.24°	0.86°
jump	4.15°	1.09°
scratch	5.80°	1.51°
walk	6.49°	0.85°
wave	4.57°	1.79°

(a) Pose Reconstruction Error

	Recall	Precision
clap	0.98	0.85
golf	0.80	0.91
jump	0.90	0.89
scratch	0.75	0.73
walk	0.77	0.84
wave	0.74	0.73

(b) Classification Results



	clap	golf	jump	scratch	walk	wave
clap	0.98					
golf	0.12	0.80	0.03		0.04	0.01
jump			0.90		0.10	
scratch				0.75		0.24
walk	0.05	0.06	0.08	0.02	0.77	0.02
wave				0.26		0.74

(c) Confusion Matrix

Fig. 3. Experimental results for pose reconstruction and pose classification, computed in a six-fold cross-validation study for 3 actors. a) Average angular error per joint per frame. b) Recall and precision for SVM-based pose classification for each activity class. c) Confusion matrix with average classification rates per combination of ground truth (rows) and prediction (columns).

per joint per frame, with standard deviations between 0.8 and 2.0 degrees. These values are comparable to the results of other state-of-the-art discriminative pose estimation methods that use higher-dimensional input data, e.g. obtained from restrictive camera setups [12,13,16,17,15]. Figure 4 visualizes the pose reconstruction result for two sample sequences, showing the predicted angles for the main joints, the corresponding ground truth and the angular error throughout the sequences.

3.3 Activity Classification Reliability

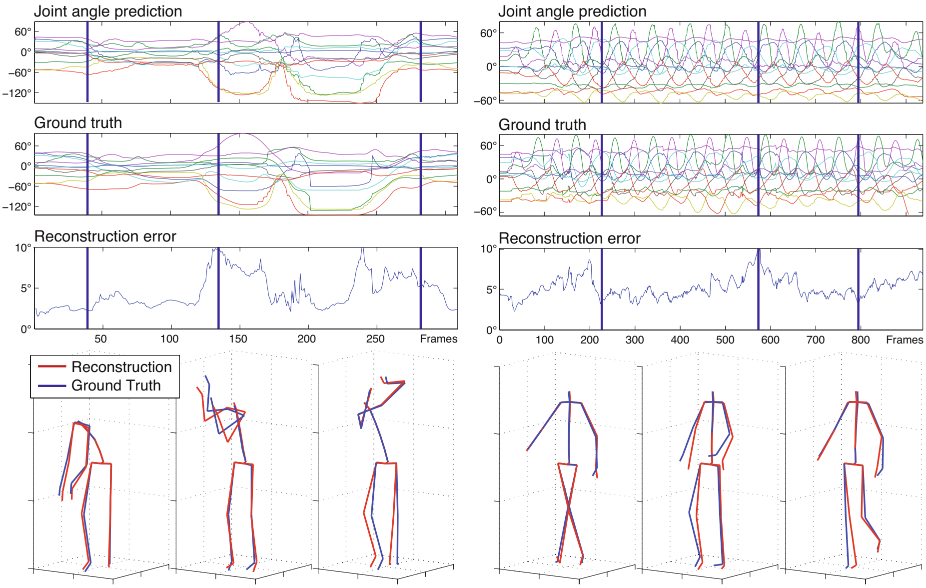
In a second series of experiments we separately evaluated the SVM-based activity classification approach. As in section 3.2, we performed a set of six-fold cross-validation studies for each acvitity in the training dataset, repeated for each of the three actors. Figure 3(b) indicates recall[1] and precision[2] for each activity class and Figure 3(c) shows the confusion matrix. Entry (i, j) of the matrix gives the relative number of frames belonging to class i that were classified as j.

The highest correct classification rates are achieved for *clapping* (98%) and *jumping* (90%), since these activities consist of repetitions of a few similar and characteristic poses. Around 20% of frames belonging to *walking* are classified as other activities. This can be explained by the multitude of poses taken during walking, some of which are similar to poses of other activities, e.g. jumping. The most significant misclassifications occur between *scratching head* and *waving*. The precision for these activities drops below 75% and the confusion matrix indicates relatively high numbers in the corresponding entries. However, the quality of the pose estimates for frames that are misclassified

---

[1] Recall: relative number of poses belonging to class n that are classified as such.

[2] Precision: relative number of poses classified as n that really belong to this class.



(a) Golfing (321 frames)

(b) Walking (945 frames)

Fig. 4. Illustration of pose estimation results for two sample sequences. The top graph shows the reconstructed angles for the main body joints over the length of the sequence. The second graph is the corresponding ground truth. Reconstruction error is plotted in the third graph. Skeletons are displayed for the frames indicated with black vertical bars (red: reconstructed pose, blue: ground truth pose).

between similar activities is not severely affected, as shown in the following section.

3.4 Combined Activity Classification and Pose Estimation

In order to demonstrate the combined performance of the activity classification and pose estimation parts, we created sequences consisting of all the activities considered in this paper performed in a row. For each of the three actors, a six-fold cross-validation study was done. As described in section 2.5, pose estimates were computed by combining the pose predictions of multiple GPR models based on the classification of frames in a sliding window ($w = 15$).

Figure 5(a)-(c) shows the results for one of the sequences. The error does not significantly increase in areas where the pose was misclassified. Averaged over all experiments in this study, over 78% of frames are correctly classified. The angular error is 5.6° per joint on average. This value is equivalent to what can be achieved in the more controlled setting with prior knowledge of the performed activity (section 3.2). The SVM-based activity classification scheme therefore significantly increases the flexibility of our pose estimation approach without having a negative effect on pose reconstruction accuracy.

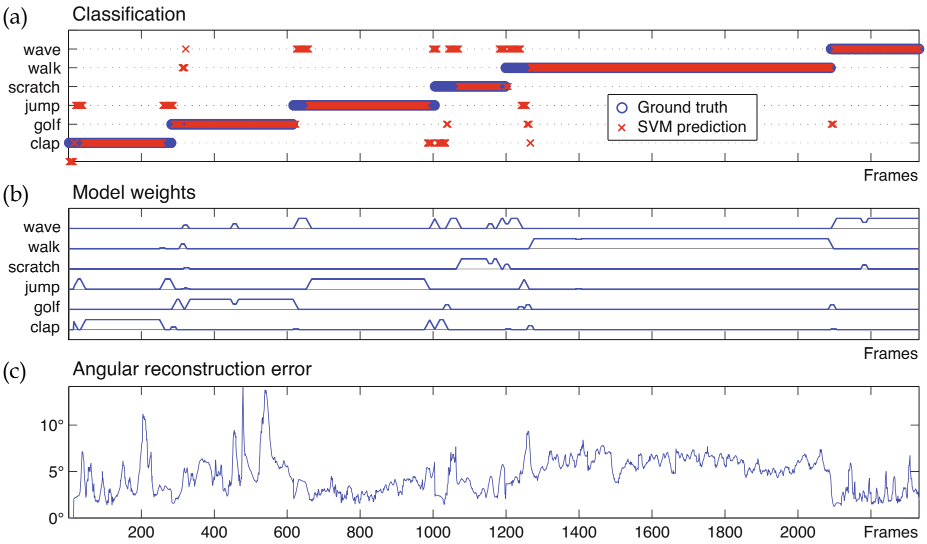
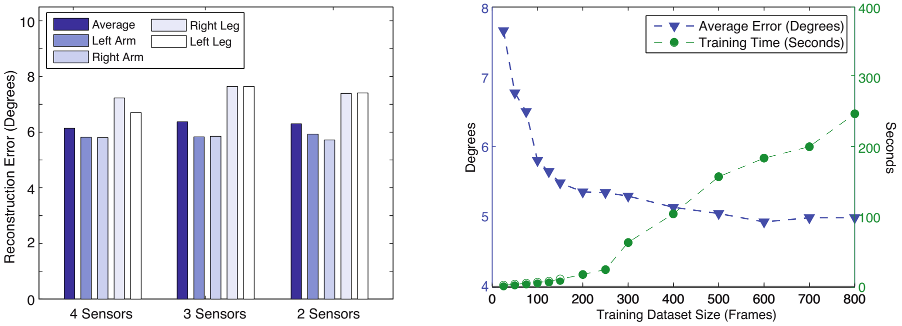


Fig. 5. Activity classification and pose estimation over a sequence of six consecutive movements. a) Predicted and ground truth activity classification over the length of the sequence (2380 frames). b) Weights between 0 and 1 for each of the learned GPR models derived from the classification in a sliding window. c) Pose reconstruction error.



(a) Varying Number of Sensors (b) Varying Training Dataset Size

Fig. 6. a) Pose estimation error for varying numbers of inertial sensors. Average errors over all joints and errors for selected body segments are reported. Walking sequences were used in a six-fold cross-validation experiment. b) Influence of training dataset size on reconstruction error and training duration. Five walking sequences of one actor were used for training, subsampled to match the designated frame count.

3.5 Number of Sensors and Training Dataset Size

Although the inertial sensors do not impose a burden on a wearer, we evaluated the ability of our method to estimate full-body pose from less than four inertial sensors. For this experiment, we only used the *walking* sequences in our dataset, since all four extremities are moved equivalently in this activity. We selected the following sensor configurations: four sensors (left wrist, right wrist, left shinbone, right shinbone), three sensors (left wrist, right wrist, left shinbone) and two sensors (left wrist, right shinbone). Figure 6(a) shows the average angular error values achieved in our cross-validation study. The error for the legs is higher than that of the arms, since apparently the variation between individual walking sequences is higher for the legs. Most remarkably, the pose estimation error does not increase for any of the configurations with less than four inertial sensors, even when only two sensors are used. The experiments show that, for activities as regular as *walking*, our method is capable of predicting 35-dimensional full-body poses from input data of 6 or even 4 dimensions (3 or 2 sensors).

The size of the training dataset influences the processing time required to train a GPR model and the achievable reconstruction quality. Training duration increases with the number of frames, but can improve the ability of the GPR model to generalize to different motion styles. In order to evaluate this behavior, GPR models for the walking motion were learned with a varying training dataset size, ranging from 25 to 800 frames. Figure 6(b) gives the cross-validation results. While it takes considerably more time to train a GPR model with a dataset size of 800 frames, the reconstruction quality does not improve above a number of approximately 500 frames. In fact, when using up to 250 frames for training, significantly shorter training times (< 40 seconds in our implementation) can be achieved by the price of just a slight increase in average angular error ($\sim 5.5°$).

4 Conclusion

We have presented a discriminative pose estimation method that allows reconstructing the full-body pose of a person, given only the limited observations of a few wearable inertial sensors attached to the person's limbs. Our method makes use of Gaussian Process Regression for learning the non-linear mapping between sensor measurements and full joint angle configurations. The system is trained on the specific movements of individual persons for each activity of interest. Since our method only requires an installed motion capture system for training, it can be applied in many settings, especially in complex, occlusive environments and outdoors. It is particulary suited for applications where movement patterns of individual persons are in focus, such as in ambulatory medical motion studies. Our experiments show that a low number of 2-4 inertial sensors are sufficient for reconstructing full-body pose for movements the system has been trained on. We plan to evaluate the feasibility of our approach in the medical scenario of motion analysis for epilepsy patients. In addition, the method will be extended in the direction of reconstructing more arbitrary movements, based on a training dataset of general human poses.

References

1. Aminian, K., Andres, E.D., Rezakhanlou, K., Fritsch, C., Robert, P.: Motion analysis in clinical practice using ambulatory accelerometry. In: Magnenat-Thalmann, N., Thalmann, D. (eds.) CAPTECH 1998. LNCS (LNAI), vol. 1537, pp. 1–11. Springer, Heidelberg (1998)
2. Cunha, J., Li, Z., Fernandes, J., Feddersen, B., Noachtar, S.: Movement quantification during epileptic seizures: a new technical contribution to the evaluation of the seizure semiology. In: International Conference of the IEEE EMBS (2003)
3. Jovanov, E., Milenkovic, A., Otto, C., Groen, P.D.: A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. Journal of NeuroEngineering and Rehabilitation 2(1), 6 (2005)
4. Najafi, B., Aminian, K., Paraschiv-Ionescu, A.: Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. IEEE Transactions on Biomedical Engineering (January 2003)
5. Luinge, H., Veltink, P., Baten, C.: Ambulatory measurement of arm orientation. Journal of Biomechanics 40(1), 78–85 (2007)
6. Roetenberg, D., Slycke, P., Veltink, P.: Ambulatory position and orientation tracking fusing magnetic and inertial sensing. IEEE Transactions on Biomedical Engineering 54(4), 883–890 (2007)
7. Zhou, H., Stone, T., Hu, H., Harris, N.: Use of multiple wearable inertial sensors in upper limb motion tracking. Medical Engineering and Physics 30, 123–133 (2008)
8. Grochow, K., Martin, S., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. ACM Transactions on Graphics 23(3), 522–531 (2004)
9. Gupta, A., Chen, T., Chen, F., Kimber, D., Davis, L.: Context and observation driven latent variable model for human pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
10. Shon, A., Grochow, K., Hertzmann, A., Rao, R.: Learning shared latent structure for image synthesis and robotic imitation. Neural Information Processing Systems (NIPS) 18, 1233 (2006)
11. Urtasun, R., Fleet, D., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: IEEE International Conference on Computer Vision, ICCV (2005)
12. Agarwal, A., Triggs, B., Montbonnot, F.: Recovering 3d human pose from monocular images. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28(1), 44–58 (2006)
13. Fossati, A., Salzmann, M., Fua, P.: Observable subspaces for 3d human motion recovery. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), March 2009, pp. 1–8 (2009)
14. Urtasun, R., Darrell, T.: Sparse probabilistic regression for activity-independent human pose inference. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (January 2008)
15. Zhao, X., Ning, H., Liu, Y., Huang, T.: Discriminative estimation of 3d human pose using gaussian processes. In: International Conference on Pattern Recognition (ICPR), pp. 1–4 (2008)
16. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Discriminative density propagation for 3d human motion estimation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2005)
17. Sun, Y., Bray, M., Thayananthan, A., Yuan, B., Torr, P.: Regression-based human motion capture from voxel data. In: British Machine Vision Conference, BMVC (2006)

18. Okada, R., Soatto, S.: Relevant feature selection for human pose estimation and localization in cluttered images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 434–445. Springer, Heidelberg (2008)
19. Chai, J., Hodgins, J.: Performance animation from low-dimensional control signals. In: International Conference on Computer Graphics and Interactive Techniques, pp. 686–696 (2005)
20. Zhu, R., Zhou, Z.: A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. IEEE Transactions on Neural Systems and Rehabilitation Engineering 12(2), 295–302 (2004)
21. Rasmussen, C., Williams, C.: Gaussian processes for machine learning (2006)
22. Ek, C., Torr, P., Lawrence, N.: Gaussian process latent variable models for human pose estimation. Machine Learning for Multimodal Interaction, 132–143 (2008)
23. Lawrence, N.: Gaussian process latent variable models for visualisation of high dimensional data. Neural Information Processing Systems, NIPS (2004)
24. Lawrence, N.: Gaussian process library (2009), `http://www.cs.man.ac.uk/neill/gplvm/`
25. Abe, S.: Support Vector Machines for Pattern Classification. Springer, Heidelberg (2005)

A 3D Human Brain Atlas

Sebastian Thelen[1], Joerg Meyer[2], Achim Ebert[1], and Hans Hagen[1]

[1] Technical University Kaiserslautern, 67653 Kaiserslautern, Germany
{s_thelen,ebert,hagen}@cs.uni-kl.de
[2] University of California, Irvine, CA 92697-2625, U.S.A.
jmeyer@uci.edu

Abstract. 3D representations of human physiology provide interesting options in the field of education. Understanding the human brain seems to be much easier when the anatomical structure is shown in the three-dimensional domain rather than in a 2D or flat projection. Seeing how the brain is 'wired' and how the different regions are connected to form circuits and complex networks requires a spatial understanding of the brain structure. Conclusions about how this structure evolved can be drawn more easily from a 3D model than from a 2D depiction of the brain.

Such 2D depictions are typically found in textbooks. Our goal is to make a brain atlas three-dimensional, so that different user groups can use the atlas to learn more about the brain and possibly make new discoveries.

In order to facilitate this, we have a developed a 3D human brain atlas, which serves as an educational tool for various types of students. The software is interactive and supports multiple user profiles, ranging from K-12 students to physicians and future brain surgeons.

We describe a method that combines high-resolution image data, large-scale volume visualization, and rendering on a distributed display cluster with a novel approach to human-computer interaction. The interaction with the atlas is accomplished by using barcodes, which are attached to various brain regions. The user can walk around in front of a large, 200 megapixel tiled display wall, which consists of $10 \times 5 = 50$ LC flat panel 30" displays and measures 23 x 9 ft. Using a camera-equipped cell phone as a universal input/output device the user scans a barcode and is then either prompted with a question to name the region and enter it on the handheld device (brain quiz), or will be provided with additional information, e.g. research documents about the selected region. The information that is provided to the user on this device over a wireless network depends on the user profile under which the user is registered and has identified himself or herself to the system.

We describe new interaction methods for large, wall-sized display systems, which enable every user to experience the visualization provided by the system either on their own or collaboratively. This new approach is different from existing methods which usually require one person to operate the system and take the lead, while others become merely observers. The system also facilitates the delivery of additional, specific information for each user based on their age group, educational background, or research intent.

Keywords: Brain Atlas, Educational System, Universal I/O Device, Tiled Displays, Large-scale Visualization, Visual Tagging.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 173–186, 2009.

1 Introduction

Our goal is to present a method for 3D display of a human anatomical brain atlas on a large, tiled display wall, and the introduction of a new, two-way interaction method for multiple users based on camera-enabled handheld display and input devices. With the introduction of smart phones, pocket PCs and multi-touch enabled cell phones such devices have become ubiquitous. Wireless communication frees the user from any kind of wire or tether. For our two-way communication, we use either IEEE 802.11 or CDMA/GSM network connectivity, which is also ubiquitous and robust now.

The main idea of the 3D Human Brain Atlas is to use a large, volumetric model of the human brain, which is computed from high resolution cross-sectional images, so-called real-color cryosections, and to employ a software which allows various user groups to interact with this large display using a handheld device. Such a universal input/output device, which was inspired by the tricorder device from the StarTrek$^{\text{TM}}$TV series, serves both as a barcode scanner and as a touch-screen device, which is both a display and a key entry device. While the large screen shows a 3D rendered image of the brain, the small screen on the handheld device provides additional information, which can be personalized for each user. Using a special barcode on the side of the screen, the user can identify herself or himself to the system in order to obtain specific information or to trigger a particular program, such as a brain quiz.

A key component in this research project is a high-resolution display which is capable of 3D rendering of volumetric brain data sets. We use a distributed rendering cluster which consists of 25 rendering nodes and 50 displays in a 10×5 configuration, providing a combined resolution of 200 megapixels. High resolution is important when displaying small barcodes, which are not supposed to obstruct the user's field of view to much. On a small desktop display, the barcodes would cover a significant part of the screen, possibly hiding significant anatomical features. On a large display, the barcodes can be displayed at a sufficient level of detail by covering only a very small percentage of the screen. As an additional benefit, multiple users can interact simultaneously with the large display, each one in their own way or working together towards a common goal.

First, we want to focus on large, high-resolution displays. In recent years, such displays have become increasingly popular due to latest advances in display and hardware technology. They combine the resolution of multiple screens and are usually driven by a cluster of rendering nodes. Two basic techniques have emerged, making use of different display technologies: *Projector-based* tiled displays consist of multiple computer projectors, often arranged in a regular grid, that have been calibrated in terms of geometry, color and luminance to form one seamless display. *Monitor-based* systems consist of a set of LC panels that have been mounted on a frame and can easily reach up to several hundred megapixels resolution. For our system we have chosen the second approach.

With respect to human-computer interaction, we try to overcome the constraints posed by traditional input devices by giving each user the opportunity for untethered interaction with the data being displayed. Barcode-like visual tags are the interface between the tiled wall and the mobile devices. The tags allow us to create new interaction metaphors that facilitate collaborative work in front of large displays and particularly address the requirements of the 3D brain atlas.

Education is the main purpose of the atlas. Our intention is to create an application that can be used for training of medical students as well as for education of fifth-graders. The atlas is supposed to give an insight in the physiology of the human brain and can be used as a *tutor system* or as an *interactive quiz*. With a target audience spanning over multiple age groups, requirements for the interactive software application are quite diverse. User groups differ in their background knowledge and their preferences, which must be taken into account when developing an interaction method.

Furthermore, the atlas should enable large user groups to interact with the visualization and with other group members all at the same time. A major benefit of tiled displays is the opportunity for collaborative interaction. However, many input devices, especially those used for desktop PCs, are not suited for large displays. Regular keyboards, for instance, are impractical to use when standing in front of a large display wall, and a regular computer mouse also becomes useless without a 2D surface to place it on. In contrast, camera-enabled cell phones are almost ubiquitous now and can serve as a universal barcode scanner and input/output device, which can provide the user with additional, personalized information or prompt the user for touch- or text-based input.

Our paper is structured as follows: Section 2 gives an overview of related work on human brain atlases. It then mainly focuses on the fields of large-scale displays, distributed visualization and interaction with display walls. Section 3 describes the tag-based interaction method and how the 3D atlas uses it to present information on the user's cell phone screen in addition to the data that is already being displayed on the wall. In section 4 we present implementation details of the 3D atlas and describe two different execution modes. Furthermore, we explain how user profiles have been implemented and how varying user requirements are being addressed by customizing the visual experience based on user input and current system status. Section 5 discusses the results and conclusion are provided in section 6.

2 Related Work

A 3D Human Brain Atlas has significant advantages over a 2D atlas. The complex structure of the brain can be understood more easily when seeing it in 3D rather than in a 2D projection. In the past, there have been various successful attempts to create digital brain atlases, both 2D and 3D. Mikula et al. [16] describe the implementation of an internet-enabled virtual brain atlas that provides high-resolution microscopy data of different species. Clients are able to query multi-resolution images of various brain parts from a server storing more than 31 terabytes of data. Besides, various 3D models can be downloaded as VRML files. ”The Whole Brain Atlas” [23] is another internet-based application focusing on the human brain and provides a multitude of 2D views to investigate the effects of brain strokes and degenerative diseases like Alzheimer's or Huntington's disease. Though not dealing with the human brain, the ”Allen Brain Atlas” [1] seeks to combine genomics with neuroanatomy by creating gene expression maps for the mouse brain. The desktop application allows users to explore the expression of approximately 20,000 genes in a fully interactive 3D environment.

In order to display a large 3D brain data set, we need a high-resolution display. Tiled high-resolution displays have been widely used to visualize large data sets. For instance, Vol-a-Tile [20] was designed to view large volumetric datasets on tiled display walls like the 30 megapixel GeoWall2 [5] built at the Electronic Visualization Laboratory (EVL). Stallion [21], the world's currently highest resolution tiled display, was built at the Texas Advanced Computing Center (TACC) at the University of Texas. Seventy-five 30 inch LC panels have been arranged in a 15×5 grid and achieve a combined resolution of 307 megapixels, which is used to run simulations and view high-resolution image content. Other installations include NASA's Hyperwall [19], consisting of 49 LC displays, and the 100 megapixel display called LambdaVision [10]. A comprehensive survey on large display installations was given by Ni et al. [18].

3D texture mapping is a common method to render volumetric data sets, and is used by the 3D atlas to visualize the human brain model. In its simplest form, the technique is limited by the fact that the original data has to fit entirely into the texture memory of the graphics card. Various optimizations have been proposed to overcome this limitation, making use of space subdivision schemes and multiresolution representations of data [9,22]. The data structure we employ is described in section 4.2 and has been proposed by Meyer et al. [14]. It has been used to implement a network-based rendering application that is able to visualize data sets between 20 MB and 76 GB size by first transferring low-resolution versions of the data via network and then gradually refine them.

In addition to the challenge of efficiently rendering large data sets, high-resolution displays strongly affect the research field of human-computer interaction. Their size allows users to naturally interact with each other and discuss the content being displayed. They have been shown to outperform standard displays in a variety of situations. Ball et al. [2] were able to proof that it is mainly the opportunity for users to physically navigate the space in front of the display (i.e., head turning, walking, pointing, etc.) that is the reason for the superiority of visualizations displayed on large screens over renderings shown on desktop displays. This fact is important when it comes to the question of how to create untethered interaction devices and has inspired us in the design of a tag-based interaction technique for the 3D atlas presented section 3.

Kerren et al. [7] present a comprehensive survey of human-computer interfaces. They identify different groups of devices, ranging from basic interaction devices (e.g., mouse, keyboard or joystick) and tracking devices (e.g., mechanical, magnetic or ultrasound) to gesture interfaces (different types of data gloves) and haptic devices featuring force feedback. Moritz et al. [17] investigate the usability of different input devices for large displays in the context of protein visualization. Their study shows that many input devices, especially those used for desktop PCs, are not suited for large displays, because they do not scale to the number of people, thus can only be controlled by a subset of users, or because they do not scale to the size of the display, i.e., they have a limited interaction radius.

For the 3D brain atlas we present an interaction method that is based on camera-equipped cell phones as universal input devices. Barcode-like tags are the interface between the atlas and mobile devices. By scanning barcodes, web-services are launched that, in their simplest form, display computed data on the cell phone screen in addition to the visualization on the large display. Ballagas et al. [3] also use visual codes to establish absolute coordinate systems on large display surfaces. The coordinate systems are used to select objects on high-resolution screens via cell phones. Madhavapeddy et al. [12] use visual tags in combination with Bluetooth for large display interaction and demonstrate their use as a pointing device in an application called ”World Map”.

3 Interaction Using Visual Tags

Collaborative interaction with high-resolution displays is limited by the type and number of input devices currently used in most installations. A single device is generally shared by multiple users, and often, one person takes the lead and controls the visualization for the entire group. True collaboration, however, means every user has their own input device that they can use to interact with the data being displayed on the screen, and both the interaction device and the display wall potentially respond to each user’s input. Untethered interaction, therefore, is the aim of the barcode technique implemented in the 3D atlas.

3.1 High Capacity Color Barcodes

A variety of two-dimensional barcodes exists (e.g., EZcode, QRCode, Maxicode and ShotCode) that mainly differs in the way information is encoded and the number of bytes that can be represented. For the atlas, we chose Microsoft’s ***H**igh **C**apacity **C**olor **B**arcodes* (HCCBs). HCCBs consist of grids of colored triangles either encoded by a 4-color or 8-color system. Microsoft’s implementation for mobile phones uses 4 colors in a 5 x 10 grid to represent 13 bytes of raw data including ReedSolomon error correction (see Fig. 1). Their barcodes encode one of the following items: URLs, free text passages, vCards and dialers. In any case, the actual data is not stored in the tag itself. Each barcode refers to an entry stored on Microsoft’s Tag Server which provides the information a client seeks. The software to identify and decode tags is available for a number of cell phones and currently free of charge. By using Microsoft’s HCCB software we avoid the trouble of dealing with cell phone specific implementation details. In fact, no programming work is required on the cell phone side at all. This frees us from particular mobile programming environments, e.g., Java, C, or, Objective C. As a result, we are able to support a large number of cell phones models.

3.2 Client-Server Architecture for Tag-Based Interaction

Tag-based interaction for the 3D atlas is based on the client-server architecture depicted in Fig. 2. All HCCBs visualized in the application have been created on the MS Tag Server. The data and type of information they encode have also been specified there. The atlas uses HCCBs to solely encode URLs, so when a user scans a barcode, the MS Server transfers the URL that has been registered with the tag to the cell phone. The URL links to a local webserver that is able to process client requests and generate HTTP output. The webserver runs a set of *Common Gateway Interface* (CGI) scripts written in Python. The name of the script to execute and additional input values are passed in the URL’s `query string`. Calling `http://ourserver/generateInfo.py?brainpart=hypothalamus`, e.g., executes

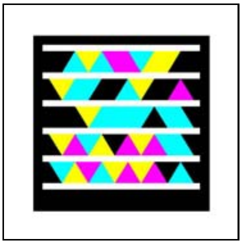


Fig. 1. HCCB barcode. Triangles of four different colors encode 13 bytes of raw data including error correction in a 5 x 10 grid.

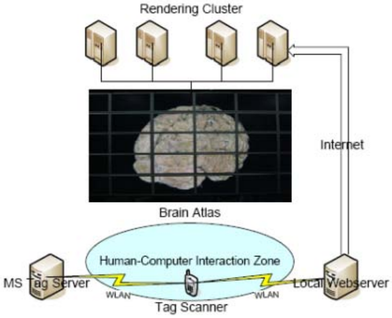


Fig. 2. Client-server architecture used in system set-up

the server script `generateInfo.py` and processes the key-value pair `brainpart=hypothalamus`. The script generates HTTP code that is being displayed on the cell phone and shows information about the indicated brain region, i.e., `hypothalamus`. Since data processing takes place on the webserver, even complicated operations can be implemented, such as accessing a connected MySQL data base. The data base, for example, can manage user groups or access privileges to certain parts of the data set. In the client-server architecture, cell phones act as typical thin clients that only display data, while data processing takes place on the webserver. All cell phones must have access to a stable Internet connection, which can either be provided by the phone company's GSM or CDMA data network or by a local wireless hotspot. Though no direct link between mobile devices and large displays is inherent in the design, computers driving the displays can query update information from the webserver. That way, a bidirectional communication between cell phones and large displays can be realized, enabling users to interact with the atlas.

3.3 What Are the Benefits?

Our tag-based interaction technique in particular contributes to the following issues: Using cell phones as universal input devices for the 3D brain atlas allows interaction metaphors to *scale* both in terms of user group and display size. The system depicted in Fig. 2 has no theoretical technical limitations and can easily handle user groups between one and several dozens of members. Furthermore, the size of a display is of no importance to our approach since the design only relies on an established network connection.

By using barcodes, we can address the issue of having to deal with user groups of different levels of expertise and preferences. Before working with the application, users sign up for a particular user group (i.e., student, instructor, K-5, undergraduate, medical, or researcher) by scanning a barcode. The interaction methods and presentation of data are then customized according to the group profile.

Information about the function of different brain parts can be comprehensive and easily fill several text pages. Displaying them on a large screen in addition to the brain model can lead to *information overload* and confusion of the user. Moreover, combining two-dimensional text with three-dimensional scenes is critical. Displaying text information on the cell phone thus is a way to avoid information clutter. Furthermore, data representation could not be customized according to the status of a user, if it was displayed on the tiled wall.

The resolution of tiled walls allows the rendering of HCCBs on a *small spatial area* in a decent quality. Barcodes can be integrated into the application without dominating a scene and still be recognizable for cell phone cameras.

4 The 3D Human Brain Atlas

The brain model that is being visualized by the atlas is calculated from a 3.57 GB histology data set of a human cadaver brain, that has been cut into 753 slices and scanned at a resolution of $1,472 \times 1,152$ pixels per slice. The resolution of the data set thus is high enough to make even smallest details, such as blood vessels, visible. We use *3D texture mapping* as a rendering method for the brain model. The data set size demands for special processing techniques in order to display it in an adequate quality. The main limitation when it comes to 3D texture-based volume rendering is the available amount of texture memory, which in our case is only about $1/10$th of the data set size per render node (i.e., the graphics card of each render node contains 256 MB video RAM). Instead of downsampling the entire brain model to fit into the video memory, we split up the computation among the nodes of the cluster and let each node display its part of the scene at the highest possible quality.

There are two execution modes for the 3D brain atlas that differ in the way people use the application for their work: *quiz mode* and *tutor mode*. In quiz mode, candidates test their knowledge of the anatomy of the human brain and receive immediate feedback from the application. In tutor mode, users deepen their knowledge by exploring the data set individually, using their cell phones as interaction devices. Both modes offer different opportunities for tag-based interaction, which will be described in section 4.3 and section 4.4.

4.1 The Cluster Display Wall

The visualization cluster (see Fig. 3) for the 3D brain atlas consists of 25 PowerMac G5 computers. They are equipped with nVidia GeForce 6800 and nVidia Quadro FX4500 graphics cards, each possessing 256MB of video RAM. The 25 compute nodes drive fifty 30 inch Apple Cinema Displays that have been arranged in a 10×5 grid. Together they form a 200 megapixel tiled display wall. Each screen has a native resolution of $2,560 \times 1,600$ pixels. The wall can display scenes at a maximum resolution of $25,600 \times 8,000$ pixels. A designated front-end computer launches applications and synchronizes the render nodes. The current operating system is Mac OS X Tiger.

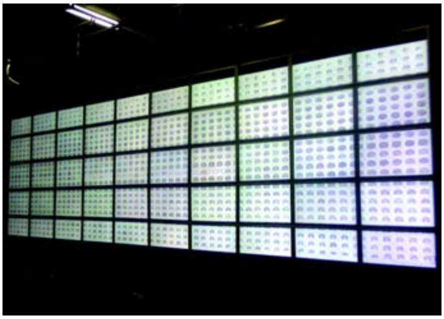


Fig. 3. 200 megalpixels tiled display cluster

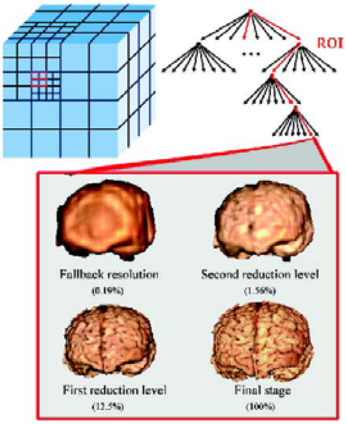


Fig. 4. Data structure

4.2 The Volume Renderer

To overcome the constraints of limited texture memory on GPUs, the volume renderer of the 3D atlas uses a dynamic subdivision scheme incorporating a multiresolution representation of data. The data structure we employ is depicted in Fig. 4 and has been introduced by Meyer et al. [14].

In an image-based out-of-core frustum clipping step, an octree helps determining those parts of a scene, that have to be visualized by a particular render node. Once the subvolume of each render node has been computed, we calculate a wavelet level, so that the subvolume fits into the video RAM of the render node with highest possible quality. Instead of just scaling the entire scene to fit the size of the wall, each of the fifty tiles displays its part of the data set in an optimum resolution.

To render an image, an opacity and color value must be assigned to each voxel of the data set. The volume renderer is able to load and apply transfer functions for color and opacity that have been generated externally. Both, regular one-dimensional and multi-dimensional transfer functions [8,15] are supported. Multi-dimensional transfer functions use additional parameters beyond the voxel intensity to assign transparency and color values. By using the gradient magnitude of a voxel, for instance, we can highlight material boundaries and interfaces between different tissues. Thereby, it becomes possible to separate different material types and to provide users with a wide spectrum of visual representations.

4.3 The Quiz Mode

In quiz mode, users test their knowledge of the physiology of the human brain. We use HCCBs to label 23 brain regions, that have to be recognized and named by test candidates. Red pins indicate the exact region to be named. A tag is attached to the end of each pin and must be scanned by the test candidate (see Fig. 5(b)). After scanning by the user and retrieving the registered URL from the MS Tag server, an input form

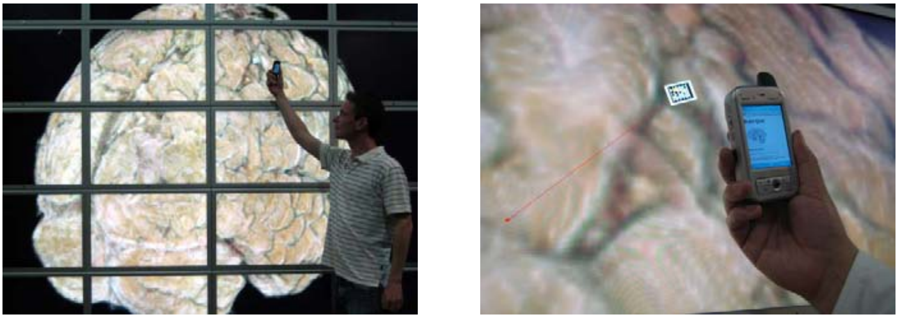

(a) (b)

Fig. 5. a) User in front of the display, b) Close-up: User prompted to enter data after scanning a tag

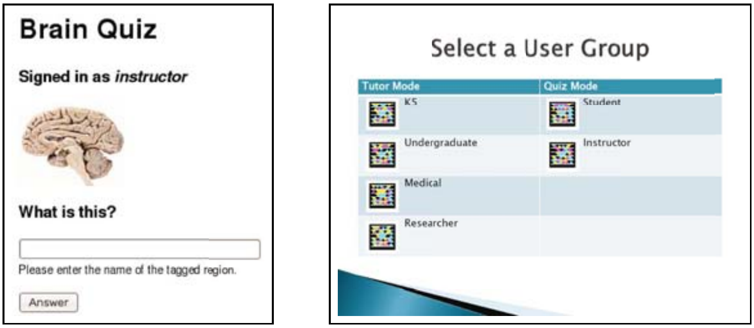
appears on the cell phone display asking to name the region. The test person's answer is encoded in the URL's `query string` and transmitted to the webserver. A CGI script analyzes the string and keeps track of all current answers. After having named all regions, a statistic is presented on the cell phone screen indicating which answers were correct and where errors occurred.

Before taking a quiz, users have to register and specify their status by scanning one of two barcodes, that are constantly being displayed in the lower left corner of the display wall. Users registered as *students*, take part in the quiz with standard privileges. Users registered as *instructors*, are able to query answers directly without filling out the input form by pressing the *answer button* at the bottom of the form (see Fig. 6(a)). While everyone is allowed to take the quiz with student privileges, only selected users can register as instructors. The script retrieves the IP address of the HTTP request and searches for it in the MySQL data base on the webserver. If the IP is found in the table of registered instructor IPs, the correct answer is transmitted upon request.

It is worth pointing out, that the application allows multiple persons to take the quiz *simultaneously*. Users independently interact in front of the display wall and groups with several dozens of members can be handled easily. This is a proof of the scalability of the tag-based interaction method used in our application. Furthermore, the example demonstrates a feature of the system, which is crucial for the atlas as an educational application in particular, and important for large display interaction in general: the ability to support multiple user groups and handle access privileges. A common problem with large display visualization is, that in many cases, the entire data set is visible for everyone. Often, one wants to restrict access to certain information (i.e., the answers to a quiz). As shown, barcodes provide a solution to this issue. In quiz mode, the 3D brain atlas distinguishes between two different user groups (i.e., students and instructors) and treats the members of each group according to their access privileges.

4.4 The Tutor Mode

In tutor mode, users may study human brain physiology and learn about the functionality of different brain regions. The mode is versatile and supports a variety of submodes,



Brain Quiz

Signed in as *instructor*

What is this?

Please enter the name of the tagged region.

Select a User Group

Tutor Mode	Quiz Mode
K5	Student
Undergraduate	Instructor
Medical	
Researcher	

(a) (b)

Fig. 6. a) Input form: Instructor can query answer directly by pressing the answer button, b) The ”user group menu” allows users to sign up for a particular group before starting to work with the tiled display

each aimed at the requirements of a particular user group. As in quiz mode, we label regions with barcodes that are then scanned using camera-equipped cell phones. Here, instead of asking for the name of a region, we provide users with relevant information about a selected brain region. The information provided to the user depends on their background knowledge and their specific profile. The following user groups are supported:

K-5. This elementary grade mode provides children with an opportunity to discover the human brain at a basic level. The information provided is mostly free from medical jargon. This version of the atlas teaches, for instance, about the parts of the brain which are in charge of processing aural, haptic and visual input, and which regions are mainly responsible for long- and short-term information memorization.

Undergraduate. For the study of the human brain anatomy and various taxonomies, we have labeled the respective regions in the data set and link barcodes to encyclopedic websites which contain detailed information and references to medical handbooks, research sites and medical studies. The information is displayed in a summary format on the handheld device and can be transferred to a public screen for more detail, as described below.

Medical. Pathological information is provided to members of this user group. Medical students need to learn which parts of the brain can be affected by diseases and how to diagnose them correctly. This is accomplished by using the same label set as before and linking the barcodes to medical reports, case studies, diagnostic and treatment methods. The information is displayed in a summary format on the handheld device and can be transferred to a public screen for more detail, as described below.

Research. Scientists often need an overview of the current literature about a particular topic. Using the same label set as before, the server script performs a literature search in a public data base and displays the results on the cell phone screen. If desired, the results can be transferred to a large, public screen for further inspection and research.

The numbers and locations of tags may vary between user groups. For K-5, for instance, we render a simplified set of 11 barcodes indicating the most important functional regions of the human brain. The number regions that may be relevant to a medical student, practitioner or researcher, however, is much higher and would most likely overwhelm a younger audience. Therefore, the 3D atlas offers a feature which allows the user to select a particular group membership. A ”user group selection” menu is displayed on a designated screen on the right side of the tiled wall, containing HCCBs that trigger a reload of the set of barcodes for the brain model. The menu is depicted in Fig 6(b).

Besides a designated screen for the ”user group selection” menu, the 3D atlas features an additional reserved screen on the side of the tiled wall, the so-called *public screen*, which enables discussions and is capable of displaying additional documents with large amounts of text, images and graphics. A public screen initially displays information that would normally only be visible on a cell phone display. However, since cell phone displays are too small for larger documents or for more than one user to see, the public screen shows the very same information in a full screen web browser. This screen can then also be used to pull up additional documents rich in information content. The idea of the public screen is to spur group discussions by making the object of focus visible to the entire group and to let users display additional information. Users decide whether they want to make the information on their cell phone public. In order to transfer content to the public screen, they have to press a ”publish button” that is always displayed at the bottom of the cell phone screen.

5 Discussion

The concept of defining user groups and providing them with different, group-specific information, which we used in the 3D Human Brain Atlas, is very flexible and general. The same software application that allows fifth-graders to deepen their knowledge about the human brain and to learn about its basic functions enables medical students to acquire detailed knowledge about the human brain anatomy and medical practitioners to research common pathology of selected brain regions. All these different views and applications are based on the same structural brain data enhanced with barcode labels and databases linked to these labels.

In order to facilitate these various ways of interacting with the 3D Brain Atlas, we have implemented several new interaction techniques, which make use of camera-enabled handheld cell phone devices that are being used as universal input/output devices. A novel aspect of this type of interaction is the fact that multiple users can interact with a large display independently or as a group. Two-way wireless communication with each user provides each user with a personalized experience. By means of a public screen users are enabled to share their information with others or to display additional documents rich in information.

Wireless communication using IEEE 802.11 WLAN or GSM/CDMA protocols frees users from tethered input or tracking devices. We provide each user with a wireless, universal input/output device, allowing for physical navigation in front of the screen wall and collaboration with other users. Currently, users can communicate directly by talking to each other and by sharing information via a public screen. It would also be possible to send data directly to another user's handheld device, which can be implemented in future versions.

One minor problem, which is inherent to the nature of tiled displays, is the fact that barcodes cannot be scanned when they are interrupted by the bezels of LC panels. This can be solved by using a seamless display as proposed by Majumder et al. [13], by using a seam-aware software as suggested by Mackinley et al. [11], or by projecting onto the bezels and making them disappear as proposed by Ebert et al. [4]. Currently, we simply rotate the data set to make sure the barcodes are displayed continuously.

6 Conclusions

In this paper we presented a 3D human brain atlas that has been implemented on a tiled display wall consisting of 50 LC panels. The atlas is based on a distributed 3D texture-based volume renderer, which is used to display a 3.57 GB histological human brain data set. The application was designed as an educational system, allowing user groups with different backgrounds or levels of expertise to explore the data set, and currently offers two execution modes: quiz mode and tutor mode.

In order to enable multiple users or user groups to interact with the application, we developed an interaction technique that was designed to overcome the limitations of traditional input devices. The method is based on camera-equipped cell phones that are being used to scan two-dimensional barcodes that are being placed in a three-dimensional scene. Scanning these barcodes with a cell phone launches a web service that allows us to implement methods to facilitate the interaction of users with each other and the large display. In this article, we introduced interaction concepts based on this idea, and we described how they can be used for customizing the presentation of information depending on the group membership status of a user or for restricting access to certain parts of the label set or information associated with the labels, such as the answers to a quiz.

We found that the system has no limitations in terms of the number of users that are supported other than the physical screen space, which in our case is a large 23 x 9 ft. wall. In fact, the system could support many more users, as the screen space extends beyond the dimensions of a single room, as is the case with distributed rendering systems at different locations [6].

Acknowledgments

This work was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) as part of the International Graduate School (International Research Training Group, IRTG 1131) in Kaiserslautern, Germany. The authors would like to thank Stephen F. Jenks, Sung-Jin Kim, and Li-Cheng Chang (University of California, Irvine), Falko Kuester and Kai-Uwe Doerr (University of California, San Diego) and the National Science Foundation for their support.

References

1. The Allen Brain Atlas, `http://www.brain-map.org/` (accessed July 24, 2009)
2. Ball, R., North, C.: The effects of peripheral vision and physical navigation on large scale visualization. In: GI 2008: Proceedings of graphics interface 2008, Toronto, Ont., Canada, pp. 9–16. Canadian Information Processing Society (2008)
3. Ballagas, R., Rohs, M., Sheridan, J.G.: Sweep and point and shoot: phonecam-based interactions for large public displays. In: CHI 2005: CHI 2005 extended abstracts on Human Factors in Computing Systems, pp. 1200–1203. ACM, New York (2005)
4. Ebert, A., Thelen, S., Olech, P.-S., Meyer, J., Hagen, H.: Tiled++ - An Enhanced Tiled Hi-Res Display Wall. IEEE Transactions on Visualization and Computer Graphics 99(1), 5555
5. The Geowall2, `http://www.evl.uic.edu/cavern/optiputer/geowall2.html` (accessed July 24, 2009)
6. HIPerSpace, `http://vis.ucsd.edu/mediawiki/index.php/Research_Projects:_HIPerSpace` (accessed July 29, 2009)
7. Kerren, A., Ebert, A., Meyer, J.: Human-Centered Visualization Environments. In: GI-Dagstuhl Research Seminar 2007. Revised Lectures (2006) Revised Lectures. LNCS, vol. 4417. Springer, Heidelberg (2007)
8. Kniss, J., Kindlmann, G., Hansen, C.: Multidimensional Transfer Functions for Interactive Volume Rendering. IEEE Transactions on Visualization and Computer Graphics 8(3), 270–285 (2002)
9. LaMar, E., Hamann, B., Joy, K.I.: Multiresolution techniques for interactive texture-based volume visualization. In: VIS 1999: Proceedings of the conference on Visualization 1999, pp. 355–361. IEEE Computer Society Press, Los Alamitos (1999)
10. Lambdavision and SAGE - Harnessing 100 Megapixels
11. Mackinlay, J.D., Heer, J.: Wideband displays: mitigating multiple monitor seams. In: CHI 2004: CHI 2004 extended abstracts on Human factors in computing systems, pp. 1521–1524. ACM, New York (2004)
12. Madhavapeddy, A., Scott, D., Sharp, D., Upton, E.: Using camera-phones to enhance human-computer interaction. In: Sixth International Conference on Ubiquitous Computing (Adjunct Proceedings: Demos) (2004)
13. Majumder, A., Brown, M.S.: Practical Multi-projector Display Design. A. K. Peters, Ltd., Natick (2007)
14. Meyer, J., Borg, R., Hamann, B., Joy, K., Olsen, A.: Network-based rendering techniques for large-scale volume data sets. In: Farin, G., Hamann, B., Hagen, H. (eds.) Hierarchical and Geometrical Methods in Scientific Visualization, pp. 283–296. Springer, Heidelberg (2002)
15. Meyer, J., Nguyen, H.T.: Direct Manipulation of Multi-Dimensional Transfer Functions using Data Clustering Analysis. In: IEEE VisWeek 2008, Visualization, pp. 38–39. IEEE Computer Society Press, Los Alamitos (2008)
16. Mikula, S., Trotts, I., Stone, J.M., Jones, E.G.: Internet-enabled high-resolution brain mapping and virtual microscopy. Neuroimage 35(1), 9–15 (2007)
17. Moritz, E., Wischgoll, T., Meyer, J.: Comparison of input devices and displays for protein visualization. Crossroads 12(2), 5 (2005)
18. Ni, T., Schmidt, G.S., Staadt, O.G., Ball, R., May, R.: A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. In: VR 2006: Proceedings of the IEEE conference on Virtual Reality, Washington, DC, USA, p. 31. IEEE Computer Society, Los Alamitos (2006)

19. Sandstrom, T.A., Henze, C., Levit, C.: The hyperwall. In: CMV 2003: Proceedings of the conference on Coordinated and Multiple Views In Exploratory Visualization, Washington, DC, USA, p. 124. IEEE Computer Society, Los Alamitos (2003)
20. Schwarz, N., Venkataraman, S., Renambot, L., Krishnaprasad, N., Vishwanath, V., Leigh, J., Johnson, A., Kent, G., Nayak, A.: Vol-a-Tile - A Tool for Interactive Exploration of Large Volumetric Data on Scalable Tiled Displays. In: VIS 2004: Proceedings of the conference on Visualization 2004, Washington, DC, USA, p. 598.19. IEEE Computer Society, Los Alamitos (2004)
21. Stallion – World's Highest Resolution Tiled Display for Open Science Deployed by Texas Advanced Computing Center, `http://www.tacc.utexas.edu/ta/ta_display.php?ta_id=100659` (accessed July 24, 2009)
22. Weiler, M., Westermann, R., Hansen, C., Zimmermann, K., Ertl, T.: Level-of-detail volume rendering via 3D textures. In: VVS 2000: Proceedings of the 2000 IEEE symposium on Volume visualization, pp. 7–13. ACM, New York (2000)
23. The Whole Brain Atlas, `http://www.med.harvard.edu/AANLIB/home.html` (accessed July 24, 2009)

Context Preserving Focal Probes for Exploration of Volumetric Medical Datasets

Yanlin Luo, José Antonio Iglesias Guitián, Enrico Gobbetti, and Fabio Marton

CRS4, Pula, Italy
`{yanlin,jalley,gobbetti,marton}@crs4.it`

Abstract. During real-time medical data exploration using volume rendering, it is often difficult to enhance a particular region of interest without losing context information. In this paper, we present a new illustrative technique for focusing on a user-driven region of interest while preserving context information. Our focal probes define a region of interest using a distance function which controls the opacity of the voxels within the probe, exploit silhouette enhancement and use non-photorealistic shading techniques to improve shape depiction.

1 Introduction

Visualization and exploration of volumetric datasets coming from MRI or CT scans, both widely used in the medical field, is a very well established research area in Computer Graphics. In the last few years, GPU-based implementations of Direct Volume Rendering (DVR) have emerged as a suitable solution for real-time interactive rendering on desktop platforms [1]. Current solutions are principally based on ray-casters and have demonstrated the capability to manage in GPU moderate-size datasets. Recent results on out-of-core techniques have shown how it is possible to overcome the limitations due to higher resolution datasets [2] [3].

Even though DVR is the most widely used and accepted technique to produce high quality images, there still exist some open issues regarding how to extend its adequacy for the abstraction of significant features from the resultant images, in particular for applications involving clinical diagnosis or medical education. Nowadays non-photorealistic and illustrative techniques are emerging in order to create a new higher layer of abstraction providing us new valuable information.

Our understanding of complex datasets is principally based on structure recognition, and our visual system reduces presented information through abstraction [4]. Therefore, we should equip medical visualization systems with a filtering mechanism for helping us in the task of efficiently abstracting the information of volumetric datasets for better understanding and analysis.

The main control mechanism when using DVR is the specification of colors and opacities. Assigning a high opacity value to a certain portion of the data may result in the occlusion of interesting structures. Furthermore, many overlapping structures may not embody the important structural details, and, in general, cluttered images are quite difficult to understand. Therefore, the challenge is the resolution of the inherent occlusion problem and particularly how to focus and enhance the rendering on a particular region of interest.

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 187–198, 2009.

Looking into human mental models, we know that the visual information gathering is not a continuous process. The human eye is capable of performing from two up to five fixations per second [5], and gathering and processing of information only occurs during an eye fixation. The eyes rest on a point of the stimulus while the focused information is visually gathered and processed [6]. A motion in the human visual field periphery causes an eye reflex by which an observer focuses on a moving object, differentiating three basic categories of motions in the three-dimensional space: translation, rotation and change of an object's shape [7].

A reasonable idea is, thus, to emphasize our region of interest by decreasing the weight of other parts which can create an occlusion effect or disturb our attention. In this paper, we present a context-preserving enhancement technique based on a focal probe model for exploring volumetric medical datasets. In our model the region of interest is defined by a distance-based function which decreases the opacity of the voxels far from the center, enabling the context-preserving effect. The presented model allows the user to translate, rotate or scale the probe according to the requirements of the region of interest. The probe guides the viewer's attention to its focus region where traditional DVR combined with illustrative techniques are used in order to render and enhance volumetric shapes.

The paper is structured as follows. In Sec. 2 we introduce GPU-based approaches for DVR, as well as non-photorealistic illustrative techniques and context-preserving volume rendering. In Sec. 3, we describe our context-preserving model based on focal probes presenting rendering results in the Sec.4 and conclusions in Sec.5.

2 Related Work

GPU accelerated volume rendering has become a very well established research area. We refer the reader to the recent book of Engel et al. [1] for a survey on GPU volume rendering. In order to solve the problems due to higher resolution datasets we employ an out-of-core based approach which is based on the management of a hierarchical multiresolution structure in conjunction with adaptive loaders. For a deeper survey on our volume rendering core technology we refer the user to the recent publication of Gobbetti at al. [2].

Our approach to efficiently abstract significant features from the resultant images coming from DVR and its GPU-based implementations, fits within the framework of illustrative visualization.

Traditional illustrative non-photorealistic rendering techniques typically mimic the style of traditional illustrations. They take advantage of the illustrators long experience in depicting complex structures or shapes in an easily comprehensible way. Ebert et al. [8] combine some physics-based illumination model with non-photorealistic techniques to enhance the perception of structure, shape, orientation, and depth relationships in a volume model. For similar purpose, different stylistic choices have been used in traditional medical illustrations, such as silhouette or contour enhancement, pen-and-ink, stippling, hatching, etc. For example, Treavett et al. [9] describe algorithms based on pen-and-ink illustration which are integrated within a traditional volume rendering pipeline. Lu et al. [10] develop a direct volume illustration system that simulates traditional stipple drawing. Nagy et.al [11] give an approach combining line drawings and direct volume rendering. Dong et al. [12] introduces volumetric hatching directly produced from medical volume data.

In this work, we are interested in techniques for emphasizing details of structures which are in focus and for reducing clutter while preserving useful context information in nearby areas.

Context preserving volume rendering focuses on solving the problem of which part of the volume must be emphasized, and what kind of rendering style is more suitable to enhance this part. At the same time, it faces the problem of how to avoid the collateral effects of losing important context information, maintaining enough visual cues for de-emphasized or less important parts. The main approaches generally rely on exploiting a way to decrease the importance of the less relevant information in favour of a region of interest. Hauser et al. [13] propose the two-level volume rendering model allowing visualization of volume data with important inner structures together with semi-transparent outer parts as context information. They integrate different rendering modes and compositing types, such as DVR, MIP, or tone shading for the inner part of the volume and contour enhancement for the outer part. Cohen et al. [14] discuss in depth how information visualization ideas can be applied to scientific visualization in an effective way. In particular, they combine the "focus and context" concept with distortion effects to improve understanding of large volume datasets. Viola et al. [15] introduce importance-driven volume rendering, where the emphasis can be automatically put on the part which has been assigned more importance. Ropinski et al. [16] propose volumetric lenses to interactively focus regions of interest, rendering the parts of the volume intersecting the lens, which is defined by a convex 3D shape, using a different visual appearance. Bruckner et al. [17] suggest to use cut-away views to focus the attention on the intersection region and ghosting views which tends to generally give a better impression of the spatial location of the object in focus.

Context-preserving in volume rendering can be also achieved by exploiting lighting intensity as an input to a function for opacity variation. Bruckner et al. [18] use this idea to reduce the opacity in rather flat regions oriented toward the light source. Parts of the volume receiving less lighting are renderered as semi-transparent silhouettes helping to preserve context information. Krüger et.al [19] present a method which enables the user to focus on a particular region, using defined parameters such as the size and location of the focus, a weight for the context, and material properties. An approach similar to ours is the one proposed by Zhou et. al [20], which emphasizes a region according to the Euclidean distance from the sampled voxel to a sphere center. Tappenbeck et al. [21] employ distance-based transfer functions, which allows to hide, emphasize or color structures based on their distance to a relevant reference structure. In our approach, we exploit the pq-distance to give a focus shape consistent with a probe defined by a superquadric function. Furthermore, we propose to combine relief shading techniques for enhacing the focus region with a better shape depiction and silhouette darkening effects. For preserving context information, we use an adaptive voxel dependent transparency and perform the clipping of voxels in regions which can create image clutter.

Silhouette enhancement is typically based on gradient or curvature estimations. Csbfalvi et al. [22] visualize object contours based on gradient information as well as on the angle between viewing direction and gradient vector using depth-shaded maximum intensity projection. Kindlmann et al. [23] employ curvature along the view direction to achieve illustrative effects, such as ridge and valley enhancement. It is also common in medical illustrations to depict shape or surface details in a way that is inconsistent with physically-realizable lighting model. Inspired by the techniques used in cartography, Rusinkiewicz et al. [24] develop a non-photorealistic rendering strategy based on multiscale local toon shading. Based on it, we develop a non-photorealistic shading model applicable to volume rendering.

3 The Context-Preserving Focal Probe Model

In our focal probe model we provide the possibility to interactively define a region of interest composed of a part in focus and its context. The focus and context information are separated by the assignment of different rendering styles, which allow to easily distinguish between the different parts. Nevertheless, the rendering styles can be smoothly blended to provide a more continuous effect.

We define the region of interest using a distance-based function which basically decrease the opacity of the samples according to its distance to the center. The focus rendering style is inspired by relief shading techniques to perform enhancement of shape details combined with a silhouette darkening effect. Meanwhile, the context style performs silhouette detection combined with a view-dependent transparency modulation effect. Finally, we clip all samples before the probe and saturate all samples behind it, in order to avoid cluttering results for rays intersecting the probe.

The presented model allows the user to translate, rotate or scale the probe size according to the requirements for an adequate visualization of the datasets.

3.1 Background

We will briefly describe the ray casting process along one viewing ray for simplicity's sake. We assume a volumetric scalar field as $f(P) \in R \; (P \in R^3)$, then denote the viewing ray direction by V and its nomalized vector by $\hat{V}$. Let P_i be the i-th sample location along a viewing ray V, f_i be the data value at P_i, g_i be the gradient at P_i and $\hat{g}_i$ be its normalized vector. We denote the gradient magnitude by $\|g_i\|$, and the normalized gradient magnitude by $\|\hat{g}_i\|$, which is $\|g_i\|$ divided by the maximum magnitude $\|g_{max}\|$. Since the gradient has noise, we filter it using an interpolation function as below

$$w_i = \text{smoothstep}(\|\hat{g}_i\|, g_l, g_h) \tag{1}$$

where smoothstep is a cubic function defined by

$$\text{smoothstep}(t,a,b) = \begin{cases} 0 & \text{if } 0 \le t < a \\ (\frac{t-a}{b-a})^2(-2(\frac{t-a}{b-a})+3) & \text{if } a \le t \le b \\ 1 & \text{if } b < t \le 1 \end{cases} \tag{2}$$

Conventional volume rendering DVR uses the front-to-back alpha blending which employs a physically motivated absorption-plus-emission optical model, computing the accumulated opacity α_i^* and color $c_i^* = (r_i^*, g_i^*, b_i^*)$ at step i with regular increment $\Delta s = \|P_i - P_{i-1}\|$ along the viewing ray V as follows

$$\begin{cases} c_i^* & = c_{i-1}^* + (1 - \alpha_{i-1}^*)\alpha_i c_i \\ \alpha_i^* & = \alpha_{i-1}^* + (1 - \alpha_{i-1}^*)\alpha_i \end{cases} \tag{3}$$

where α_i and $c_i = (r_i, g_i, b_i)$ are the opacity and color respectively at P_i, both derived from the transfer function.

The light properties make it possible to enhance objects for a variety of effects. Before accumulating, generally we use the traditional Phong model to perform a shading effect by modifying the color c_i to be the shaded color $c_{i,\text{shaded}}$ as below

$$c_{i,\text{shaded}} = \lambda_i c_i \tag{4}$$

where

$$\lambda_i = k_a + \sum_{\text{lights}} k_d(\hat{L} \cdot \hat{g}_i) + k_s(|\hat{H} \cdot \hat{g}_i|)^{k_e} \tag{5}$$

k_a, k_d and k_s are the ambient, diffuse and specular lighting coefficients. $\hat{L}$ is the normalized light vector, $\hat{H}$ is the normalized half-way vector, and k_e is the shininess constant.

After shading, we denote the opacity at P_i by $\alpha_{i,\text{shaded}}$, which basically follows the same variation as color in Eq. 4. Then we substitute c_i with $c_{i,\text{shaded}}$ and α_i with $\alpha_{i,\text{shaded}}$ into Eq. 3 performing the normal accumulation process.

3.2 Probe Shapes

Medical datasets can capture quite heterogeneous parts of the human body, varying in a wide range of physical dimensions. For a satisfactory exploration of diverse volume datasets we propose to use an exploration tool based on the superquadrics family functions. These functions provide us a flexible and relative simple solution for modelling rounded or sharp corners for a variety of basis geometric shapes (See Fig. 1).

We use a geometric distance based on the pq-norm to define the superquadric. Since it provides us with a distance function, we can use it to define a probe center.

The pq-norm for a sample P_i with x_i, y_i, z_i coordinates is defined by

$$\|P_i\|^{pq} = \|(\|(x_i, y_i)\|^p, z_i)\|^q \tag{6}$$

where

$$\|(x_i, y_i)\|^p = (|x_i|^p + |y_i|^p)^{\frac{1}{p}} \tag{7}$$

is p-norm, which is the generalization of the Euclidean metric ($p = 2$). As shown in Fig. 2, various consistent focus shapes can be obtained by using different p and q values.

Fig. 1. Some superquadrics shapes

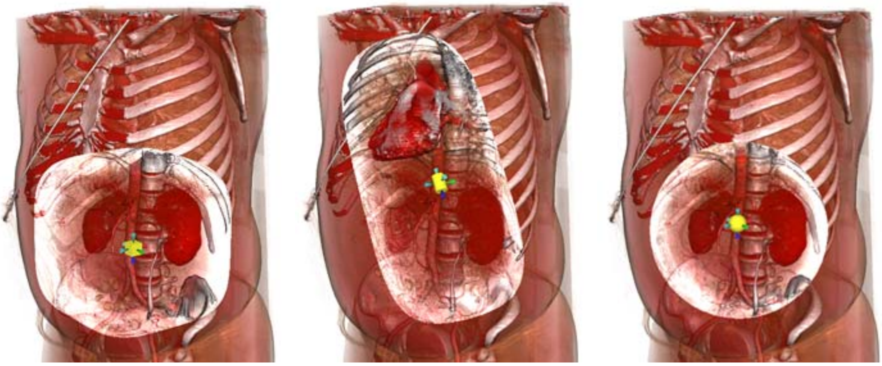


Fig. 2. Probe shapes. We have tested various probe shapes, including rounded cubic ($p = 4$, $q = 4$), cylindrical ($p = 2$, $q = 4$) and spherical ($p = 2$, $q = 2$).

3.3 Distance Based Merging of Rendering Styles

In our focal probe model we propose to employ different rendering styles corresponding with the focus and the context region. One idea could be to use a particular threshold to differentiate between the different zones within the probe, but this decision can lead to sharp transitions with a continuity loss in the resulting rendering. Instead, we propose to merge both rendering styles using a distance-based function d_i having a plateau for the focus region determined by $\rho \quad (0 \leq \rho \leq 1)$, which can be interactively adjusted by the end user of the system.

$$d_i = \begin{cases} 1 & \text{if } 0 \leq \|P_i\|^{pq} \leq \rho \\ 1 - g_\beta(\frac{\|P_i\|^{pq} - \rho}{1-\rho}) & \text{if } \rho < \|P_i\|^{pq} \leq 1 \end{cases} \tag{8}$$

where

$$g_\beta(t) = \frac{t}{e^\beta(1-t)+t} \tag{9}$$

is the schlick rational function [25]. We use $\beta \quad (\beta < 0)$ for having a smoother or sharper decreasing value of the context region.

We use different rendering styles for the focus and context regions which will be introduced later. Let $(c_{i,\text{focus}}, \alpha_{i,\text{focus}})$ and $(c_{i,\text{context}}, \alpha_{i,\text{context}})$ be the color and opacity at P_i defined by these rendering styles, our proposed blending strategy is performed by using d_i as focus fraction (See Eq. 10).

$$\begin{cases} c_{i,\text{blend}} = d_i c_{i,\text{focus}} + (1-d_i) c_{i,\text{context}} \\ \alpha_{i,\text{blend}} = d_i \alpha_{i,\text{focus}} + (1-d_i) \alpha_{i,\text{context}} \end{cases} \tag{10}$$

After blending, c_i is substituted by $c_{i,\text{blend}}$ and α_i by $\alpha_{i,\text{blend}}$ into Eq. 3 performing the normal accumulation in the rendering pipeline.

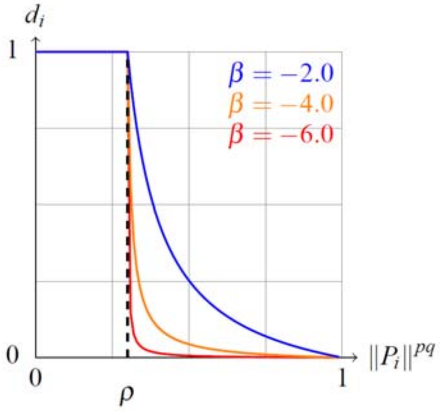


Fig. 3. Distance based function

3.4 Focus Model

Next, we propose a rendering style which performs better shape depiction for the focus region based on traditional DVR and non-photorealistic techniques (See Fig. 4). In particular, we perform a silhouette darkening effect based on the following detector

$$s_i = w_i * \text{smoothstep}(1 - |\hat{g}_i \cdot \hat{V}|, s_l, s_h) \tag{11}$$

where s_l and s_h control the silhouette sharpness. The focus color $c_{i,\text{focus}}$ and opacity $\alpha_{i,\text{focus}}$ at P_i are defined by

$$\begin{cases} c_{i,\text{focus}} = (1 - s_i) r_i c_i \\ \alpha_{i,\text{focus}} = r_i \alpha_i \end{cases} \tag{12}$$

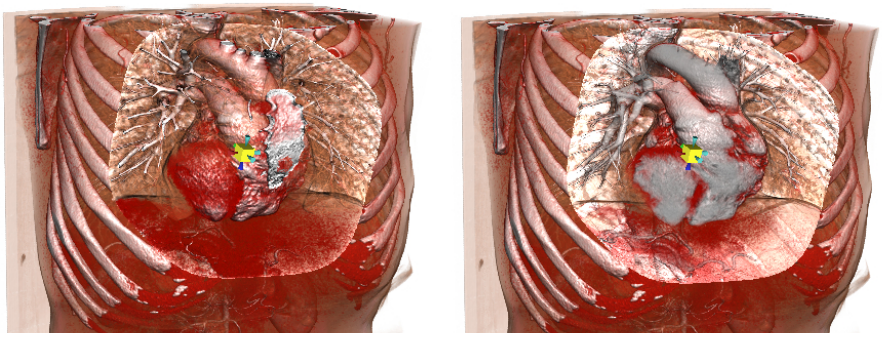


Fig. 4. Comparison between a probe with and without focus. On the left image we provide an example of a normal probe without focus. Notice how the sternum bone hides part of the heart. Besides, flesh and lung tissues create a cluttering effect as they cover part of neighbour structures. On the right image, we show the same situation using a context-preserving focal probe. In this case, heart is right focused, no cluttering is created around it. Furthermore, context information is preserved and previously hidden vessel structures are now better identified.

where r_i is a shading factor implementing an illustrative effect explained next and defined by Eq. 14. We combine the previous color darkening processing with an illustrative relief shading effect. Principles in relief shading [24] advise to ommit shadows and specular effects, exaggerate the height of ridges and valleys in the object shape and use a particular lighting effect which appears to originate as from the top of the image. Furthermore, these principles suggest to locally adjust light direction in order to homogenously light the interesting part of the volume. Finally, to support multiscale toon-shading effects, it is recommended to blend between different smoothed versions of the volume.

Applying inside our system Rusinkiewicz et al.'s techniques based on relief shading, requires a special preprocessing to produce smoothed versions of the volume dataset for multiscale-based effects. In our model we will apply similar basis concepts in order for detail enhancement without necessity of preprocessing. We compute on the fly a local version $\hat{n}_i$ of the smoothed normal, accesing samples located at positions $x_{i\pm1}$, $y_{i\pm1}$ and $z_{i\pm1}$ for a sample at P_i with x_i, y_i, z_i coordinates. Next, we make a local light adjustment at each sample, computing the new light position L_i^* at P_i based on the global light position L, and the smoothed normal $\hat{n}_i$ at P_i. For this purpose, we create a light which is perpendicular to the smoothed normal $\hat{n}_i$, exaggerating the shading effect for ridges and valleys in our volume (See Eq. 13).

$$L_i^* = L - \hat{n}_i(\hat{n}_i \cdot L) \tag{13}$$

Furthermore, as shown in Eq. 14, we use the original shading $\hat{L} \cdot \hat{g}_i$ as a basis for the lighting of the diffuse component, thus to avoid the appearance of excessively dark zones in the final rendering. Next, we sum the contribution of the P_i local light $\hat{L_i^*} \cdot \hat{g}_i$. Finally, we have altered the local lighting to produce a toon shading effect, just clamping $\hat{L_i^*} \cdot \hat{g}_i$ between 0 and 1 and multiplying the result by the toon shading factor a.

$$r_i = k_a + k_d(\hat{L} \cdot \hat{g}_i + a * \mathrm{clamp}_{[0..1]}(\hat{L_i^*} \cdot \hat{g}_i)) \tag{14}$$

3.5 Context Model

The rendering style proposed for the context region is based on view-dependent transparency and silhouette detection. The context color $c_{i,\mathrm{context}}$ and opacity $\alpha_{i,\mathrm{context}}$ at P_i are defined by

$$\begin{cases} c_{i,\mathrm{context}} = \lambda_i c_i \\ \alpha_{i,\mathrm{context}} = h(\theta) s_i \lambda_i \alpha_i \end{cases} \tag{15}$$

where

$$h(\theta) = \begin{cases} 0 & \text{if } 0 \le \theta \le \varphi \\ 1 - \frac{cos\theta - cos\varphi}{cos\phi - cos\varphi} & \text{if } \varphi < \theta \le \phi \end{cases} \tag{16}$$

Shown by the left most image in Fig. 5, θ is the angle formed by the O vector connecting the eye position with the probe center and the viewing direction V, ϕ is the angle

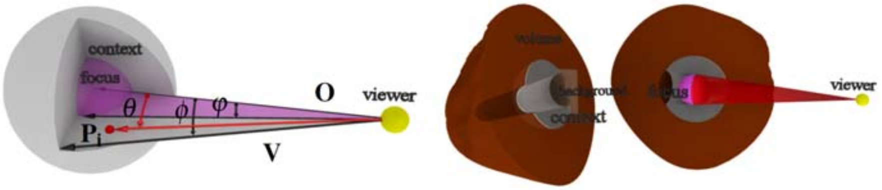


Fig. 5. Context definition for focal probes. In the left image, for a sample P_i in the context region, we define the θ angle which decides how much the context will be displayed. In the right, voxel clipping is performed in order to produce a non cluttered render image.

defining the maximum angle for rays intersecting the probe, and finally φ the angle defining rays passing through the focus region. The shading factor λ_i has been previously defined by Eq. 5. The $\alpha_{i,\text{context}}$ parameter introduces a view-depent effect, giving more opacity to voxels in the outer part of the probe and increasing the transparency as samples are closer to the probe center.

The use of volumetric probes entails the problem of possible occlusion due to the voxels situated between the eye position and the probe. This is also valid for voxels in the context region, which can impede the correct visualization of our interest region. We illustrate this problem in the right image of Fig. 5. Therefore, a reasonable strategy seems to cancel the accumulated color just before the first intersection of the viewing ray with the probe, corresponding to the red cone in the right of Fig. 5. Samples in the context region behind the probe focus are ignored, and following samples in the volume are blended with the background contributing less to the final accumulated color along the ray.

4 Results

We report on the results obtained when performing exploration of medical datasets by using our focal probe model. We have tested the proposed technique with a variety of volumetric medical datasets. We discuss the results obtained with the inspection of a $512 \times 512 \times 1559$ whole body contrast and a $512 \times 512 \times 743$ torax study, both CT[1] with $16 bit/sample$.

Regarding the focal probe model, we employ the following parameter configuration. We use $\beta = -2.0$ for having a smooth transition between the focus and context information (See Fig.3). To separate the focus and context regions, we adjust the ρ parameter in the interval $[0.4, 0.7]$. For gradient filtering we compute the w_i interpolation function with $g_l = 0.015$ and $g_h = 0.95$. Related with the non-photorealistic shading, we control the silhouette detector s_i using $s_l = 0.7$ and $s_h = 0.95$ for a fine silhouette enhancement without creating cluttering owing to low gradient values. We define the toon shading factor $a = 3.0$ to enhance lighting differences between ridges and valleys in the focus region.

---

[1] Source: Geneva University Hospital, Radiology Department.

As described in Fig. 3 the behaviour of our focal probe model establishes a center of attention, fitting in the region around of the probe center. Transparency is modulated by the d_i distance based function and its effect results in more transparency for voxels in the context region. In Fig. 6 we can appreciate the effect of an interactive incursion of a focal probe within the whole human body CT. At the beginning, the probe acts just as a clipping sphere and as soon as the center of the probe is inside the volume, the focus effect becomes visible, as well as the context-preserving effect showing contour shapes. When exploring parts of the volume which are too deep, voxel clipping becomes active.

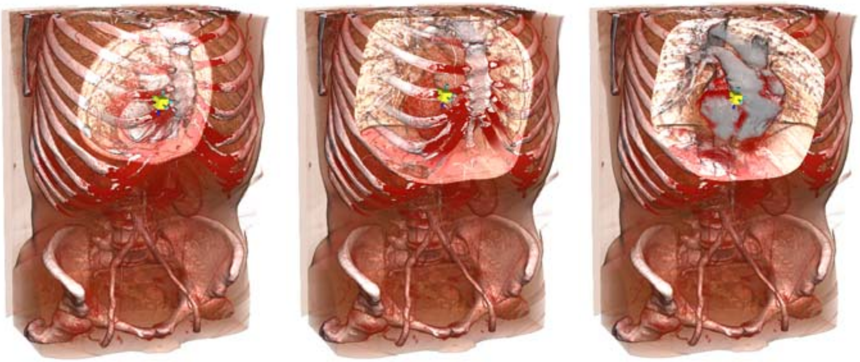


Fig. 6. Context-preserving focal probe. From left to right we increase the focal probe penetration stopping when the heart is right focussed in the center of the probe.

Using the silhouette darkening effect we can depict shape borders using a darker color and enhancing the shape over its background. In Fig. 7 we show a series of snapshots of our volume renderer varying the parameter s_h of Eq. 11, which indicates the threshold value above which a sample is considered a border shape.

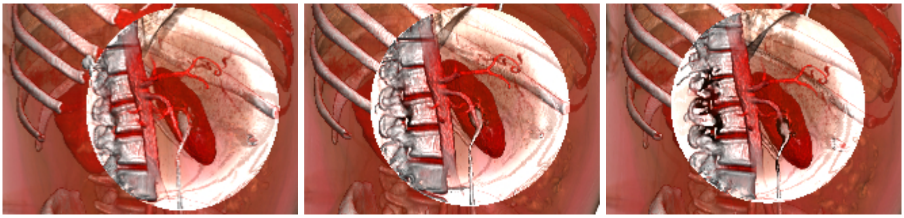


Fig. 7. Silhouette darkening effect. From left to right, no silhouette darkening $s_h = 1$, silhouette darkening with $s_h = 0.8$ and $s_h = 0.4$.

The relief shading technique, described in Sec. 3.4, allows the user to appreciate additional details within the focus region, exaggerating the lighting effect by using a perpendicular light for each voxel (See Fig. 8). In our approach we rely on a local version of the smoothed normals, and for this reason, the enhanced details are limited to local details, normally corresponding to high frequencies in the volume.

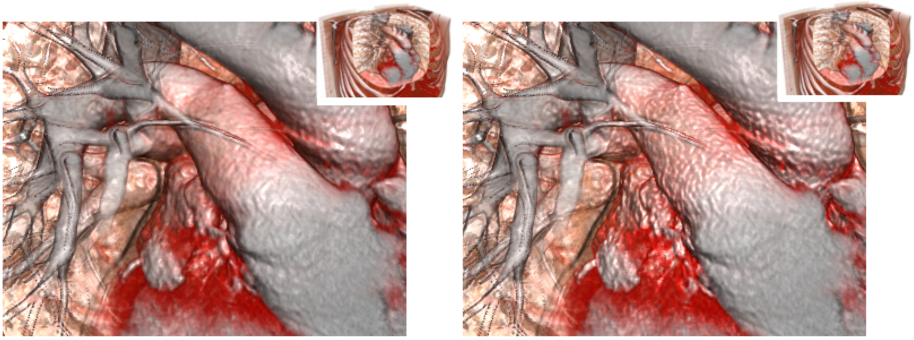


Fig. 8. Relief shading effect. On the left we show a probe focus without relief shading. On the right we show the same viewpoint with relief shading for the focus probe. Notice the how high frecuency details are highlighted.

5 Conclusion

We have described an alternative focal probe model for interactive exploration of medical volumetric datasets. The key feature of our approach relies on the enhancement of natural filtering mechanisms of humans to separate and abstract information.

Our model is based on the development of different rendering styles for the focus and context information. The superquadrics family functions are used to get consistent focus shapes. Furthermore, we propose an adaptive voxel shading, based on illustrative non-photorealistic techniques, in order to enhance shape depiction in the focus region. In addition, we use silhouette detection to convey better structure cues in the focus region and to preserve the context information in the outer part. Finally, we propose a strategy to smoothly blend both rendering styles.

A particular clipping strategy is performed for voxels in front of the probe and behind it, in order to solve the occlusion problem and to avoid the creation of a noisy background.

Acknowledgments. This work is partially supported by the EU Marie Curie Program under the 3DANATOMICALHUMAN project (MRTN-CT-2006-035763).

References

1. Engel, K., Hadwiger, M., Kniss, J., Rezk-Salama, C., Weiskopf, D.: Real-time Volume Graphics. AK-Peters (2006)
2. Gobbetti, E., Marton, F., Iglesias-Guitián, J.A.: A single-pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets. The Visual Computer 24(7–9), 797–806 (2008)
3. Crassin, C., Neyret, F., Lefebvre, S., Eisemann, E.: Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering. In: ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), pp. 15–22 (2009)
4. Hendee, W.R., Wells, P.N.T.: The perception of visual information. Springer, New York (1997)

5. Ware, C.: Information Visualization:Perception for Design, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2004)
6. Smolnik, S., Nastansky, L., Knieps, T.: Mental representations and visualization processes in organizational memories. In: Proceedings of the Seventh International Conference on Information Visualization (IV 2003), pp. 568–575. IEEE Computer Society Press, Los Alamitos (2003)
7. Thomas, R.J., Strothotte, T.: Motion enhanced visualization in support of information fusion. In: Proceedings of International Conference on Imaging Science, Systems, and Technology(CISST 2001), pp. 492–497. CSREA Press, Las Vegas (2001)
8. Ebert, D., Rheingans, P.: Volume illustration: non-photorealistic rendering of volume models. In: Proceedings of IEEE Visualization, pp. 195–202 (2000)
9. Treavett, S.M.F., Chen, M.: Pen-and-ink rendering in volume visualization. In: Visualization 2000: Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000), Washington, DC, USA. IEEE Computer Society, Los Alamitos (2000)
10. Lu, A., Morris, C.J., Ebert, D.S.: Non-photorealistic volume rendering using stippling techniques. In: Proceedings of IEEE Visualization, pp. 211–218 (2002)
11. Nagy, Z., Schneider, J., Westermann, R.: Interactive volume illustration. In: Proceedings of Vision, Modeling, and Visualization, pp. 497–504 (2002)
12. Dong, F., Clapworthy, G.J., Lin, H., Krokos, M.A.: Nonphotorealistic rendering of medical volume data. IEEE Comput. Graph. Appl. 23(4), 44–52 (2003)
13. Hauser, H., Mroz, L., Bischiand, G.I., Gröller, M.E.: Two-level volume rendering. IEEE Transactions on Visualization and Computer Graphics 7(3), 242–252 (2001)
14. Cohen, M., Brodlie, K.: Focus and context for volume visualization. In: TPCG 2004: Proceedings of the Theory and Practice of Computer Graphics 2004 (TPCG 2004), Washington, DC, USA, pp. 32–39. IEEE Computer Society Press, Los Alamitos (2004)
15. Viola, I., Kanitsar, A., Gröller, M.E.: Importance-driven volume rendering. In: Proceedings of IEEE Visualization 2004, pp. 139–145 (2004)
16. Ropinski, T., Steinicke, F., Hinrichs, K.H.: Tentative results in focus-based medical volume visualization. In: Butz, A., Fisher, B., Krüger, A., Olivier, P. (eds.) SG 2005. LNCS, vol. 3638, pp. 218–221. Springer, Heidelberg (2005)
17. Bruckner, S., Gröller, M.E.: Volumeshop: An interactive system for direct volume illustration. In: IEEE Visualization, pp. 671–678 (2005)
18. Bruckner, S., Grimm, S., Kanitsar, A., Gröller, M.E.: Illustrative context-preserving exploration of volume data. IEEE Trans. Vis. Comput. Graph 12(6), 1559–1569 (2006)
19. Krüger, J., Schneider, J., Westermann, R.: Clearview:an interactive context preserving hotspot visualization technique, vol. 12, pp. 941–948 (2006)
20. Zhou, J., Döring, A., Tönnies, K.D.: Distance based enhancement for focal region based volume rendering. In: Proceedings of Bildverarbeitung für die Medizin 2004, pp. 199–203 (2004)
21. Tappenbeck, A., Preim, B., Dicken, V.: Distance-based transfer function design: Specification methods and applications. In: Proceedings of SimVis 2006 (2006)
22. Csbfalvi, B., Mroz, L., Hauser, H., König, A., Gröller, M.E.: Fast visualization of object contours by non-photorealistic volume rendering. Computer Graphics Forum 20(3), 452–460 (2001)
23. Kindlmann, G., Whitaker, R., Tasdizen, T., Möller, T.: Curvature-based transfer functions for direct volume rendering: Methods and applications. In: Proceedings of IEEE Visualization, October 2003, pp. 513–520 (2003)
24. Rusinkiewicz, S., Burns, M., DeCarlo, D.: Exaggerated shading for depicting shape and detail. In: ACM Transactions on Graphics (Proc. SIGGRAPH), vol. 25 (2006)
25. Schlick, C.: A fast alternative to phong's specular model. In: Graphics Gems IV, San Diego, CA, USA, pp. 385–387. Academic Press Professional, Inc., London (1994)

Use of High Dynamic Range Images for Improved Medical Simulations

Meagan Leflar, Omar Hesham, and Chris Joslin

School of Information Technology, Carleton University, 1125 Colonel-by Drive, Ottawa, Canada, K1S 5B6
`{mleflar,ohesham,cjoslin}@connect.carleton.ca`

Abstract. Here we describe the use of high-dynamic range lighting techniques to improve the rendering quality of real-time medical simulation systems. Specifically we show our method of extracting the lighting information from an actual endoscopic light probe used for surgery and how we apply this lighting information into a real-time rendering system based on OpenGL and using Shaders to improve the realism of the probe's lighting in the scene; with the consideration that it is the only light source in the simulation.

Keywords: surgery simulation, endoscopic, light probe, HDRI lighting.

1 Introduction

Endoscopic surgery saves lives, reduces post-operative trauma, and considerably reduces hospital operating costs by producing more out-patients than in-patients compared to regular open surgery. Therefore the promotion of surgical procedures, skills, and skilled surgeons in this field is an important aspect of medical progress. Obviously the main issue in education in this area is access to hands-on training rather than theoretical procedures; to this end a simulated environment enables interns to perfect their art.

Conventional training has been performed through the use of animal subjects, medical donation cadavers, or using real patients and heavily supervised sessions; usually in this order of progression. Apart from the practical learning experience, the other key element in this is an evaluation of progress; i.e. are an intern's skills improving. In certain cases, a new surgeon will have had training in a specific procedure as little as four times before performing an operation on an actual patient.

In answer to this problem surgical simulators have been a promising technology for several years now; the idea that we can replace the procedure of training interns with a highly accurate simulation of any surgical situation is generally considered a dream by many.

1.1 The Advantages of Simulated Environments

In general, simulations should have several advantages: (1) they do not require the use of cadavers or the use of biological components that need to be later disposed of. (2) They can be made available around the clock and are (again) not reliant on the

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 199–208, 2009.

availability of cadavers (which are essentially either animal parts or medical donations). (3) They can simulate living tissue and given the proper physics should provide a better haptic response than anything but the real thing considering that a cadaver would quickly amortize over time and an animal organ would be composed in a slightly different way. (4) We can provide a variable environment and introduce a large range of variation; everything from simple operations, to emergency situations, to extreme cases of unusual but actual situations that are exactly repeatable for performance assessment. (5) We can make an assessment of an intern's performance as shown in the studies conducted by Adamsen et al [17] as well as Ferlitsch et al [18] that can be used to determine how well the intern is performing, where they need to improve, and how they cope under certain circumstances, and provide them with the ability to improve.

1.2 Disadvantages of Simulated Environments

On the flip side, there are several disadvantages to simulators; generally they tend to be expensive (most of the popular models are US$100K+ at the time of writing) which generally has to be justified through a daily expenditure compared against other costs. For example, a fully grown pig's liver can be used as a training organ, which costs approximately $6 per session, lasting some 20 minutes. On a daily basis, this would cost $430 / day (on a 24Hr basis); on this basis a virtual simulator would have to be used at least 1 year of continual use before the cost advantage would become apparent. However, this presents another problem. As Dongen et al [8] discuss in their research work, they found that interns required significant encouragement to use a laparoscopic simulator, mainly in the form of competition; this is despite the fact that a simulator was shown to improve the assessed performance of all interns in a fairly predictable manner. Lastly, one of the common disadvantages of a virtual simulator was realism. While current simulators are able to provide a fairly realistic environment, surgeons are still unhappy with the haptic response (including the physics), the lack of complex fluids to simulate proper blood flow, the ability to interact with complex soft surfaces (such as the colon), and the rendering/light required to create a real environment to make it look natural – endoscopic training has the advantage that it is always (in real or simulated) viewed on a 2D screen; which eliminates the need for 3D immersion (i.e. stereo).

It should be noted that, in a practical sense, for interns to become competent in the use of the endoscope and various tools they do not require a simulator; it is the great advantage of the performance assessments and the ability to examine various scenarios that drives the growing need for realistic surgical simulations.

1.3 Simulator Systems

Endoscopic simulators come in various guises; commercial systems are very predominant in this field of research. The list of systems includes: LapMentor from Simbionix [9] which has a fairly realistic simulation and is an all-in-one system; MIST from Mentice [10] provides a very simple simulator for improving hand skills; LapSim from Surgical Science [11] is a tiered system with everything from basic skill training to Laproscopic Cholecystectomies; VEST Systems VSOne [12] is a complete system mainly focused on skill training exercises; and finally LapVR from Immersion [16] is a wide ranging system with everything from basic skill training, to general surgery, to more specific procedures. These represent the bulk of the commercially available systems – each with their own advantages and disadvantages. Most of them focus on the basic concept of training a surgeon in their hand-eye coordination and improving their skills in tool use – therefore reality is not a major issue for some.

In the non-commercial sector, endoscopic simulator systems such as those introduced by Bibin et al [4] called SAILOR system, which is a desktop simulator focused on the haptic feedback and training for specific procedures. Lim et al [15] describe some of the advances in surgery simulation, specifically the use of video images to improve realism; the issue with their technique is that they appear to combine an obviously realistic video environment with a 3D model lit using a standard lighting model – this served to enable elements such as interactive cutting of the 3D model, but it did little to improve the realism of the simulation itself. Kuhl et al [16] follow along similar lines in presenting an endoscopic simulator focused on the interaction; however in their case they had more constraints in the form of a duct (representing the digestive tract). Korner et al [19] present their technique for endoscopic interaction using a haptic device, the focus of their system was on the rendering of the correct haptic feedback.

Other methods of providing surgical simulators include adapting game engines. Marks et al [7] evaluated several game engines for the purpose of surgical training; their conclusions were mainly that while they were capable of being used for training purposes, they mainly lack the ability to interact with soft tissues, which is obviously a serious consideration for surgery.

1.4 Realistic Lighting and Rendering

In terms of the lighting for a real-time system, it is very common practice to use a simple light, specifically from the system being used (e.g. OpenGL or DirectX). Lighting in these systems is uniform and depending on the setup have a linear distribution factor around the central axis (i.e. the Z-axis) of the light which is limited by the cone; however this light is based on a low dynamic range (0 – 255). In order to properly light and create a more realistic scene we need to increase the dynamic range, especially because if we are to examine the light in this range it will only have one value which is 255 (due to its obvious brightness). Debevec et al [5, 6] have shown that we can capture several sets of images of a “light probe” and combine them together in order to obtain an image which has an improved dynamic range (commonly referred to as a High Dynamic Range Image, or HDRI), providing darker regions with a range of values while in the same image doing the same for the bright areas (and everything in between). By obtaining an image of a spherical environment (either through a fisheye lens, panoramic camera, or reflective sphere), the image represents a 2D spherical map of all the light falling onto that sphere at that point – the HDRI means that all the light information is captured, rather than just the light levels that could be stored within the low dynamic range, regardless of its source (i.e. whether it is from a direct source, or from a reflection, is irrelevant). This means that using this 2D spherical map we can render a highly realistic scene for the light in that position.

Although the technique of HDRI was developed back in 1995, it was not until technology (processors and memory) were able to handle the relatively larger images that new concepts began to spring up. It is now becoming apparent that a restrictive range of 0 to 1.0 or 0 to 255 is not sufficient and everything from High Dynamic Range Videos [20] to High Dynamic Range Displays [21] are being introduced to solve other issues.

The other part of this process is High-Dynamic Range Rendering, or HDRR, as presented by McTaggart et al [14], is required to properly render the effectiveness of different light sources. While we are not producing values greater than the range capable by the display device, we can have virtual light sources that are very different from one another. For instance, where we might have two light sources one bright and another very bright, both will be represented as 1.0; however any interaction with the environment will reduce this value in the same way. Therefore two different lights will be represented and dealt with in the same manner. In order to preserve the effective light intensity of a very bright source we use light values which are not restricted to 1.0.

For a surgical endoscopy simulator, the enclosed and fixed space means that we could use a light probe as previously described to obtain the lighting information; the positioning would mean that the image would be approximately correct for the given situation and produce a relatively realistic image. The problem is more a practical one whereby placing a light probe (a reflective sphere which is photographed in order to collect high dynamic range lighting information) inside a patient would obviously cause some issues (even a small one). Therefore, rather than obtaining the lighting information through a light probe, we investigated obtaining the luminance characteristics of the single light source itself through HDRI. We have this advantage because there is effectively only a single light source inside the patient: that of the endoscopic light source. This approach is yet more beneficial for the reason that often the singular light source repositions with the camera (the endoscopic light source is in fact attached directly to the camera) therefore only this one angle of illumination (that can be derived from the photos we will acquire in section 2) is necessary.

This paper is broken down into the following way: Section 2 discusses the acquisition of the lighting information from endoscopic lighting probes; Section 3 presents the real-time rendering system, including the specific techniques that are required. Section 4 concludes the paper and includes some future work.

In this paper, we focus on the realistic lighting and rendering for endoscopic-based procedures, specifically for simulations. In this case, we are concerned with realism rather than training procedures or in vitro considerations.

2 Lighting Information Acquisition

The lights used for an endoscope are usually high brightness types such as halogen or xenon. Xenon (4500K – 5000K) arc lamps (as opposed to Xenon flash lamps used in Camera flashes) provide a light very close to natural daylight (5500K); Halogen lamps (3200K) are bright but not necessarily closer to daylight, generally being more yellow; i.e. Halogen is generally considered a bright, but lower cost alternative. More recently the introduction of Light Emitting Diode (LED) light sources that use a combination of red, green and blue lights have emerged; these lights are focused on the environmental development rather than anything related to brightness or colour correctness. Endoscope lights are fed through an optical cable and are generally attached to either a probe, or more commonly the camera itself so that where the camera is pointing the light is illuminating the scene.

The first step in the production of an accurate light source is to acquire an HDR image [1]. Obtaining an HDRI version is important because it provides us with proper light intensity values not restricted to the range 0 – 255 as with low-dynamic range images. The issue with low-dynamic range pixel values are that bright and very bright are represented as the same value (255) when in reality they are actually far apart, however to the fact that the range of light values is only 255 we cannot properly represent these values. HDRI has proven popular in the film industry for use in lighting 3D objects for integration into real-scenes by acquiring light information through light probes; HDRI assures that the full range of lighting information is provided to for the rendering system; in general that image uses the full range of HDRI, in terms of obtaining the HDRI of a light source we are specifically interested in defining the differences between the bright areas. If we were to simply take a picture of the light source we would end up with essentially an image where all the pixel values are 255 for the light and generally 0 for the remainder.

In place of this, we need to take several images of the light at different exposure settings. The goal in taking these photographs is to have the same optical view as the light itself. This is accomplished by extinguishing all other lights in the area and taking photographs directly of the light itself. Previous to this research, a few individuals have attempted to acquire the HDRI by directing the light at a translucent screen and taking the various photographs from the opposite side of the screen. However, this method is as flawed as the screen implemented to perform it – any screen has a given colour and texture that distorts the image. By acquiring the light information directly from the light itself, we bypass this undesirable effect.

We took seventeen shots using a Nikon D200 camera at different exposure values (EVs) by setting the aperture to f22 and adjusting the shutter speed accordingly as the resulting images are shown in Figure 1. In this situation (in a completely dark room with only the light source), it is essential to adjust the aperture setting on the camera upward according to the intensity of the light source being captured.

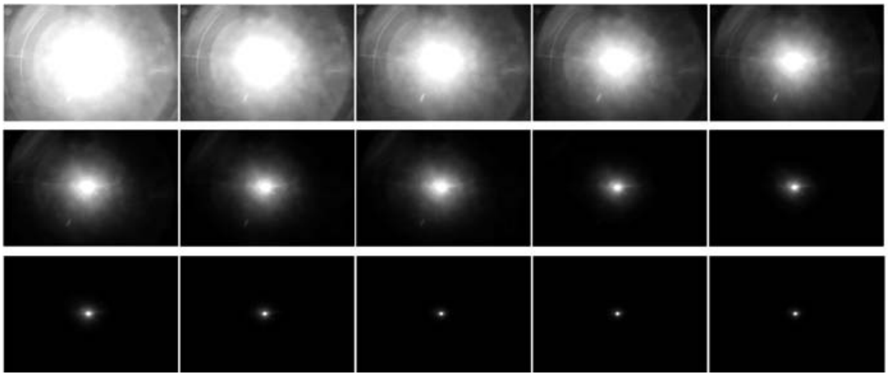


Fig. 1. A range of shots taken at different exposure settings

Merging these images together in Adobe Photoshop CS4, the final HDRI is acquired and saved as a Radiance RGBE file for compatibility purposes. This image now contains all of the exposure information from the previous nine images.

3 Real-Time Rendering Using HDRI

Having obtained a file with the HDR image (usually stored in the Radiance RGBE format) containing the detailed lighting information of the endoscopic probe, we proceeded to construct a program that demonstrates the superior quality and realism that the HDRI version of the endoscopic light source would provide a surgical simulation model.

To exhibit this, a simple 3D model of a liver was created in Autodesk Maya 2009 and imported into a real-time virtual environment. This model was then lit in two contrasting ways: with mathematically generated low dynamic range (LDR) spotlight illumination and with a spotlight light source produced using the HDRI endoscopic light as a projected texture. In both cases we applied several Shaders to the model. Figures 2 and 3 illustrate the different light sources and their effect on the virtual scene.

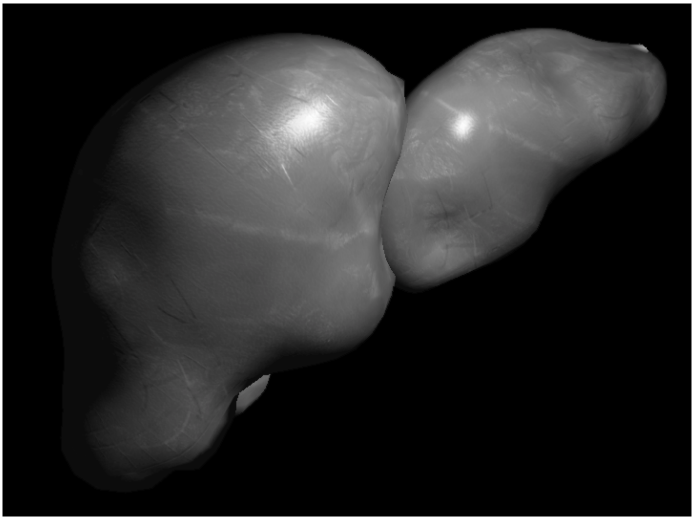


Fig. 2. A screenshot in which our virtual scene is being lit with a standard, mathematically generated LDR spotlight

This composition, albeit simplified, reflects the nature of the lighting situation of an endoscopic surgery in that the endoscopic probe is the only light source for the majority of the surgical process. The ability to accurately reproduce the exact nature of that sole light source will give the scene much more realism and ultimately enable surgeons to gain a better feel for the real surgery from the simulated experience, and therefore facilitate the training process.

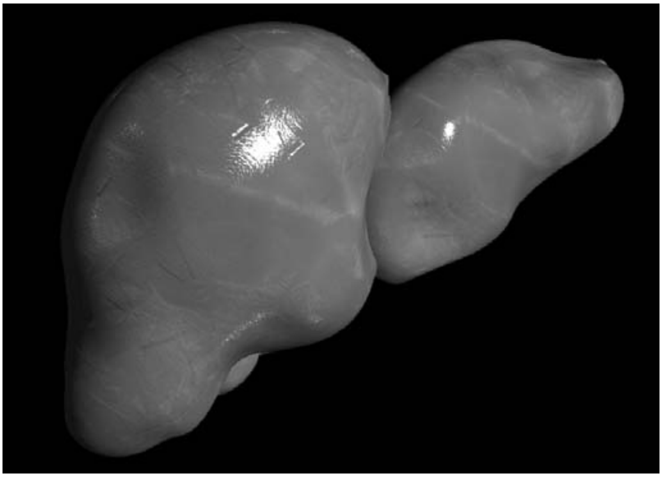


Fig. 3. The real-time virtual environment lit by an HDRI being projected as a spotlight

An alternate way to achieve a similar level of realism with regards to the nature of the light produced by the endoscopic probe during surgery would be to utilize Illuminating Engineering Society (IES) lights. IES lights have become the standard in photometric data distribution, and were originally developed to facilitate the electronic transfer of photometric information over the internet. IES lights are essentially mathematical maps of the distribution of light. Although IES lights are physically accurate, they require dynamic computations which are resource intensive and significantly reduce rendering speed. Clearly, they are far from ideal for use in a real-time system such as a surgical simulation. For this reason, HDR lighting is a superior method for the development of real-time medical simulation software.

The main challenge that accompanies working with this high dynamic range endoscopic light image is how to deal with the additional exposure data that is required to store an HDRI. The standard 8-bit buffer object is of no use in this case, as it cannot support the 32-bit floating-point HDRI file, and though there are a few high dynamic range monitors and television screens in production, most display hardware is not yet capable of properly interpreting the HDRI data. Instead of viewing an image with an exposure range that approaches the accuracy of the human eye, the user is presented with garish contrast and flat hues as the monitor misinterprets the exposure information.

The pipeline that we developed as a solution is organized to preserve the additional HDRI data until the very end of the process, at which point a tone map is applied to convert the HDRI into something that can be processed competently by the display hardware. This process has four main components:

(1) Floating-point Frame Buffer Objects (FBOs) are implemented to hold the 32-bit floating-point HDR values. FBOs also allow for real-time post-processing effects, also known as off-screen rendering. This allows one to add filters and make changes to a scene before writing it to screen. As well as being compatible for use with HDRI, using FBOs also does not decelerate the rendering process, which is helpful for real-time applications such as this one. A complication that one must keep in mind when wishing to implement this method is that FBOs are a reasonably new method and

therefore some graphics cards do not yet support it. For example, ATI graphics cards are not currently capable of running programs employing FBOs, but up-to-date GeForce graphics cards are.

(2) HDR Lighting: The HDRI endoscopic light file is loaded as a 32-bit texture. Projective Texture Mapping is then used to obtain a suitable perspective projection matrix that projects the texture in a manner that resembles a spotlight [22]. This matrix is further transformed by the transformation matrix of an abstract spotlight object, which carries position, rotation and scale information.

(3) HDR Shading: The projection matrix acquired from component (2) is then passed on to the Shader to project the HDRI texture values as lighting information onto the surface of objects. Specific algorithms for different lighting models –like Blinn, Phong and Sub-Surface Scattering- treat the HDRI projected values as a light: how it should reflect, refract, and diffuse the luminance information it is receiving from the HDRI texture projection. For our liver model, we chose a Phong Shader with Specular highlighting and Normal mapping. The Shader's operators were adjusted by enforcing a higher colour range, which float members can store, and ensuring the RGB values were not clamped to a [0, 1] range. At the pixel shadin level, we find that scaling the diffuse and specular contributions of a Phong Shader by a fraction of the HDRI value produces suitable results.

(4) HDR Rendering: The scene is first rendered in high dynamic range, and then the scene is tone mapped. With the OpenGL Shading Language (GLSL), a full screen Shader [2] is applied that adjusts the HDRI of the endoscopic light, evening out the exposure such that the HDRI may be compressed into a low dynamic range image (LDRI). With an Exposure value specified through user input or procedurally, the following tone map operator was used:

$$\text{Color} = \text{HDRI.rgb} * 2.0^{((255*\text{HDRI.exponent}) - 128)} * \text{Exposure} \tag{5}$$

We chose to encode our HDRI in Radiance RGBE format. It stores 32-bits per pixel which consist of an 8-bit mantissa for each primary, followed by a single 8-bit exponent [3].

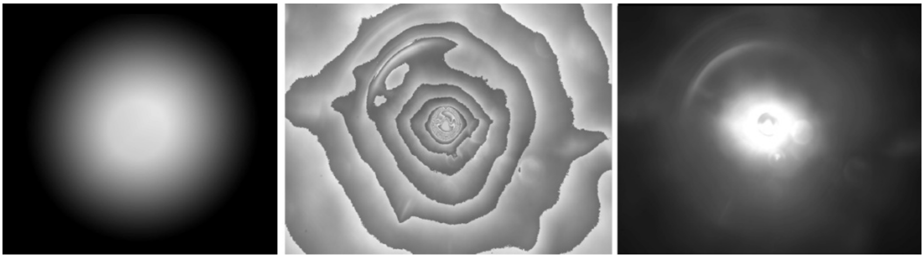


Fig. 4. From left to right: a shot of the mathematically generated LDR spotlight, the HDRI light source displayed on a LDR screen before it has been tone-mapped, the HDRI light source after it has been tone mapped. These three images illustrate the importance of tone mapping to avoid the over-exposed effect visible in the center image, and exemplify the greater level of light information that one can acquire using an HDRI as opposed to a mathematically generated LDRI.

The LDRI is then bound as a texture to a full-screen polygon and rendered to the main frame buffer, which is an integer FBO. This final step is essential because it preserves the effects that the HDRI lighting has had on the scene – because tone mapping is done after having first rendered the scene in HDRI, we benefit from having done the shading and lighting effects in HDRI - while converting the final output to a format that can be displayed properly on LDR screens.

Eventually, with the further development of HDR screens, this step will become unnecessary.

4 Conclusion and Future Work

In this paper we have demonstrated a method for capturing the lighting information from an endoscopic light source directly based on a high-dynamic range in order to improve the quality and realism of the rendering for a real-time environment system used to provide a surgical simulation system. We have shown that compared to the low-dynamic range fixed light, that we obtain a much more realistic rendering.

In terms of future work we are looking at the lens distortion and colour factors introduced by common endoscopic cameras. These factors are not naturally introduced by a standard real-time rendering camera as the equations used to translate the 3D world into a 2D space use a homogeneous conversion. However, in the real world each camera introduces a distortion of the image through the lens (as the lens is not perfect) – this is especially true of lenses used for endoscopy which have a smaller aperture and longer focal length.

Acknowledgements

This research work has been funded by a Natural Sciences and Engineering Research Council of Canada Discovery Grant. The authors would also like to thank Dr. Joseph Mamazza, Director of Minimally Invasive Surgery at The Ottawa Hospital, for his consultation on endoscopic surgery.

References

1. Block, C.: The HDRI Handbook. Rocky Nook Inc., Santa Barbara, ISBN: 978-1933952055 (2007)
2. Dickheiser, M.: Game Programming Gems 6, ISBN: 1-58450-450-1
3. Ward, G.: Real Pixels. In: Arvo, J. (ed.) Graphics Gems II. Academic Press, London (1992)
4. Bibin, L., Lecuyer, A., Burkhardt, J.M., Delbos, A.: SAILOR: a 3-D medical simulator of loco-regional anaesthesia based on desktop virtual reality and pseudo-haptic feedback. In: Virtual Reality and Simulation Technology (VRST), October 2008, pp. 97–100 (2008)
5. Debevec, P.E., Malik, J.: Recovering High-Dynamic Range Radiance Maps from Photographs. In: Proceedings of SIGGRAPH 1997, Computer Graphics Proceedings, Annual Conference Series, Los Angeles, California, pp. 369–378 (1997)

6. Yu, Y., Debevec, P.E., Malik, J., Hawkins, T.: Inverse Global Illumination: Recovering Reflectance Models of Real Scenes From Photographs. In: Proceedings of SIGGRAPH 1999, Computer Graphics Proceedings, Annual Conference Series, Los Angeles, California, August 1999, pp. 215–224 (1999)
7. Marks, S., Windsor, J., Wunsche, B.: Evaluation of Game Engines for Simulated Surgical Training. In: GRAPHITE, pp. 273-281 (December 2007)
8. van Dongen, K.W., Van der Wal, W.A., Borel Rinkes, I.H.M., Schijven, I.A.M.J., Broeders, M.P.: Virtual Reality Training for Endocopic Surgery: Voluntary or Obligatory? In: Springer-Verlag Surgical Endoscopy, vol. 22(3), pp. 664–667 (2008)
9. Simbionix - LapMentor: `http://www.simbionix.com/LAP_Mentor.html`
10. Mentice - MIST: `http://www.mentice.com`
11. Surgical Science – LapSim: `http://www.surgical-science.com/`
12. VEST - VSOne: `http://iregt1.iai.fzk.de/KISMET/VestSystem.html`
13. Immersion – LapVR: `http://www.immersion.com`
14. McTaggert, G., Green, C., Mitchell, J.: High Dynamic Range Rendering in Valve's Source Engine. In: ACM SIGGRAPH 2006 Courses, Article No. 7 (2006)
15. Lim, Y.J., Jin, W., De, S.: On Some Recent Advances in Multimodal Surgery Simulation: A Hybrid Approach to Surgical Cutting and the Use of Video Images for Enhanced Realism. In: Presence, vol. 16(6), pp. 563–583. MIT Press, Cambridge (2007)
16. Kuhl, C., Dumont, G.: Virtual Endoscopy: From Simulation to Optimization of an Active Endoscope. In: European Series in Applied and Industrial Mathematics (ESAIM), November 2002, vol. 12, pp. 84–93 (2002)
17. Adamsen, S., Funch Jensen, P.M., Drewes, A.M., Rosenberg, J., Grantcharov, T.P.: A Comprative Study of Skills n Virtual Laparoscopy and Endoscopy. In: Society of American Gastrointestinal Endoscopic Surgeons (SAGES), March 2003, pp. 229–234 (2003)
18. Ferlitsh, A., Glauninger, P., Gupper, A., Schillinger, M., Haefner, M., Gangl, A., Schoefl, R.: Evaluation of a Virtual Endoscopy Simulator for the Training in Gastrointestinal Endoscopy. In: Endoscopy 2002, vol. 34(9), pp. 698–702 (2002)
19. Korner, O., Manner, R.: Haptic Display for a Virtual Reality Simulator for Flexible Endoscopy. In: Eurographics Workshop on Virtual Environments, pp. 13–18 (2002)
20. Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High Dynamic Range Video. ACM Transactions on Graphics 22(3), 319–325 (2003)
21. Seetzen, H., Heidrich, W., Stuerzlinger, W., Ward, G., Whitehead, L., Trentecoste, M., Abhijeet, G., Vorozcovs, A.: High Dynamic Range Display Systems. In: ACM SIGGRAPH 2004, pp. 760–768 (2004)
22. NVIDIA Developer Zone - Projective Texture Mapping: `http://developer.nvidia.com`

My Corporis Fabrica: A Unified Ontological, Geometrical and Mechanical View of Human Anatomy

Olivier Palombi[1,2], Guillaume Bousquet[1], David Jospin[1,2], Sahar Hassan[1],
Lionel Revéret[1], and François Faure[1]

[1] LJK Lab, Grenoble Universités, CNRS, INRIA Rhône-Alpes, France
[2] Laboratory of Anatomy, Grenoble Universités, France

Abstract. A new anatomical database, My Corporis Fabrica (MyCF), is presented. It's based on the reference anatomical ontology FMA (the Foundational Model of Anatomy) with the possibility to coherently integrate 3D geometrical data, as well as biomechanical parameters. The purpose of this extension is to allow the user to intuitively create a patient-specific 3D representation from a formal description of anatomical entities and also to automatically export this description to test a physical simulation. The main contribution of MyCF consists in the formalization of a comprehensive database structure implemented in MySQL, linking canonical description of anatomical entities with reality-grounded instances of such description in terms of geometrical data and physical attributes. The principle of ontology modeling inherited from FMA is maintained in order to guarantee a close consistency between the two databases. A detailed illustration is given for the knee. In this example, the canonical description of anatomical entities is developed, including the location of attachment of ligaments as well as their stiffness, using an intuitive 3D GUI. Finally, an instance of the knee based on this description is automatically augmented with the canonical part information and then exported to a physical simulator. This example shows the benefit of coupling an exhaustive description of the anatomical structures with a physical simulation to study the dynamic effect of ligaments on the stability of the knee articulation.

1 Introduction

Anatomical models of human body are required in an increasing number of domains, from the teaching of anatomy to biomechanical simulation. Typically, accurate 3D representation of human anatomy is needed in application such as surgery simulator, realistic prototyping of cloth or representation of avatars in 3D virtual communities. The 3D representation of internal anatomy entities is primarily accessible from medical imaging such as CT or MRI tomography. These imaging techniques provide a raw volume and need to be segmented to identify individual anatomical entities. Even if some powerful methods have been proposed, this segmentation step is still time consuming as it cannot be fully automatic. An alternative approach to volumetric segmentation is to use 3D models designed by

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 209–219, 2009.

artists, but this involves a risk of loss in realism. Therefore, the need for a validated anatomical database of the human body linked with 3D representation is rapidly growing. This database should provide high-level information such as ontological relationships, as well as geometrical data for graphics applications and physical parameters for simulations. It should be extensible and anatomically relevant to integrate new conceptual entities, physical instances and characterization parameters. This is the goal of our approach with the design of the My Corporis Fabrica (MyCF) database. Specifically, MyCF extends FMA (the Foundational Model of Anatomy) [1] to provide additional data to create geometrical and mechanical models of anatomy. It is based on a novel database structure which considerably facilitates its extensibility. The remainder of this paper is organized as follows. In section 2, we review previous work in the domain. The design of MyCF database is presented in section 3. Implementation details are discussed in section 4. An overview of potential applications is presented in section 5. Concluding remarks are presented in section 6.

2 Previous Work

Integrating patient-specific content into computational resources requires the use of formal and logical representation of anatomical knowledge. The field of computing science focused on structuring knowledge is called *ontology*. Ontologies of anatomical knowledge has been widely performed with a high level of details and complexity. The FMA can be presented as the most sophisticated anatomical ontology available today for the scientific community [2]. FMA follows the OBO-Foundry principles [3], one such principle being that descriptions about distinct domains belong to distinct ontologies. FMA is a symbolic representation of the phenotypic structure of human body. Therefore, FMA do not contain topology , geometrical and functional data. Such information should be stored in other ontologies. The technical complexity involved in formalizing the anatomy can not handle general variability in any easy way. Specific approaches have to be imagined to address this issue. This lacking link is the key point in the emerging research field of *semantic 3D media* [4] and more specifically for medical applications [5,6].

The Digital Anatomist Information System is based on FMA to link graphical and symbolic representations of anatomy [7]. A static 3D content (pictures of 3D models) is available to illustrate the FMA's concepts. But 3D data are not completely integrated in the ontology. Starting from FMA to embed ontology into specific computational resources seems to be relevant regarding the level of compliance with modeling principles used in FMA [8]. FMA is available in several formats, independent of any specific authoring environment. FMA is implemented with an open source ontology editor and knowledge-base framework called *Protégé*. Data are stored in a *MySQL* table but direct access to query is not intuitive. A querying agent exists for FMA but the flexibility of use is restricted [9].

Only structural anatomy has been considered in FMA (OBO-Foundry principles). Topology, geometry and functions are required to build an efficient ontology for physical simulation. Strategy to extend FMA for a specific purpose is presented in [10]. An original approach, strictly limited to the muscoloskeletal system, has been proposed by Charbonnier et al. [11]. In the semantic-driven clinical examination platform, presented by Charbonnier et al., FMA is not extended. Anatomical knowledge is retrieved from FMA and functional information are combined in a second step to generate physical simulation through a specific platform. However, the main idea is similar to what MyCF intends to address.

3 Database Design

3.1 Generic Concepts

An ontology is a specification of a conceptualisation [12]. Common components of ontologies include classes, attributes and relations. Classes are sets, abstract objects, entities, concepts. Classes have attributes or properties. Relations make links between classes. Each relation has a conceptual meaning. For instance in the following sentence: "The patella is a *part of* the knee and *is a* bone organ", relations (*is_a* and *part_of*) used between anatomical entities (patella, knee and bone organ) give a formal and logical description of canonical anatomy. *Is_a* and *part_of* are also called hierarchical relationships. The key point to build an ontology for specific application is the use of associative relationships (domain-specific) which must be clearly defined [13].

The general purpose of an ontology is to provide a mean of classifying individuals. Patient-specific data are *instances* of canonical anatomical entities stored in the ontology. Three kinds of binary relations can be used to manage efficiently anatomical knowledge and individuals: $< class, class >$ seen below, $< instance, class >$ and $< instance, instance >$.

MyCF is focused on the managing of individuals through an intuitive graphic user interface for non-expert users in ontology. The goal is to allow anatomists to contribute to the canonical part of MyCF. By this way, anatomical content of MyCF will be validated by experts. On the other hand, an user of MyCF can add individual content and get correct anatomical information as terminologies, functions, localizations, interactions and so on.

MyCF is based on FMA for the ontological description of anatomy. An original MySQL schema is proposed to ease sharing and extension, as explained in the subsection 3.2 and to manage individuals. A SQL query is enough to retrieve data from one or more tables, allowing efficient implementation in a specific application.

3.2 Description of the mySQL Database

The figure 1 shows the logical structure of MyCF's database. The entity relationship diagram (ERD) can be divided in two main parts: canonical anatomical knowledge on one hand and individual data on the other hand.

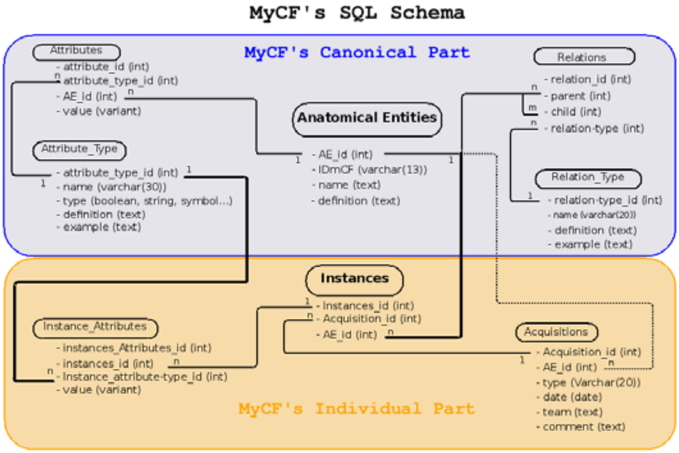


Fig. 1. Entity Relationship Diagram of MyCF's database. The canonical anatomy part is on the top and the individuals part is on the bottom. Black lines depict relationships (1:n) between tables. Two bold lines show the relationships between the canonical and the individual parts. The dotted line depicts a relationship used to force user to select an existing anatomical entity (restricted to anatomical segment) during insertion process of a new acquisition in the database.

MyCF's canonical part: All canonical classes of MyCF are stored in the main table called *Anatomical Entities*. An ontology id, used as a primary key, is given for each one. Definition and favorite name are available as well.

In an ontology, anatomical entities are organized into a hierarchy named taxonomy. A taxonomy is a directed acyclic graph satisfying some conditions [8]. Our database schema allows to represent generic graphs, including directed acyclic graphs. Each entry of *Relation_Type* defines a graph, which can be a taxonomy or a more general graph, where its nodes are contained in *Anatomical Entities* and its oriented edges in *Relation*.

In practical terms, the table called *Relation_Type* gives definition of relations. For instance, the relation *is_a* is defined here with text definition and examples. The binary relations between anatomical entities are stored in the table called *Relations* where parents and son are defined using anatomical entity ids and where the type of relation is given.

The attributes are managed also with two tables. The table *Attributes* defines the properties of *Anatomical Entities*. The table *Attributes_Type* ensures the genericity of the attributes.

With those five tables, it is possible to expand endlessly MyCF without adding tables. Furthermore, the simple organization allows users to write his own SQL queries to retrieve or update data.

MyCF's individual part: Instances are linked to anatomical entities. The 1:n relationship (bold line in figure 1) allows to have many instances of one anatomical entity. Instances have specific attributes, the position in a 3D scene for instance. The types of attributes are stored in the canonical part of the database. But the attribute's value of a given instance is saved in the table called *Instance_Attributes*. Furthermore, general information can be stored in the canonical part of the database as, for instance, the yellow color conventionally used to depict bones in anatomical drawings. If the color is not given by the user for an instance, color stored in the canonical part can be automatically retrieved. By this way, a priori anatomical knowledge can be used to complete patient-specific content. The origin of patient-specific 3D data is a set of medical images (after segmentation). 3D models coming from a given acquisition will be used together to generate a 3D scene. The table called *Acquisitions* is used to manage it.

3.3 Instances

Geometry is one of the most challenging issues in anatomy modeling. Each anatomical entity can be represented using various types of geometric models (volumetric images, surface or volume meshes, analytical surfaces, implicit surfaces, etc.) at various resolutions or levels of detail. Consequently, we can not store one single geometrical description for each anatomical entity. So, we store, for each entity, a set of instances referenced by file names. We set no limitation on the file format and the type of geometric models. We have developed plugins for Blender [14], a free and open-source geometric modeler, to edit geometrical models of MyCF, see Fig 2. It handles surface meshes and can be used to modify shapes, to displace and resize objects and to label areas such as ligament attachment places.

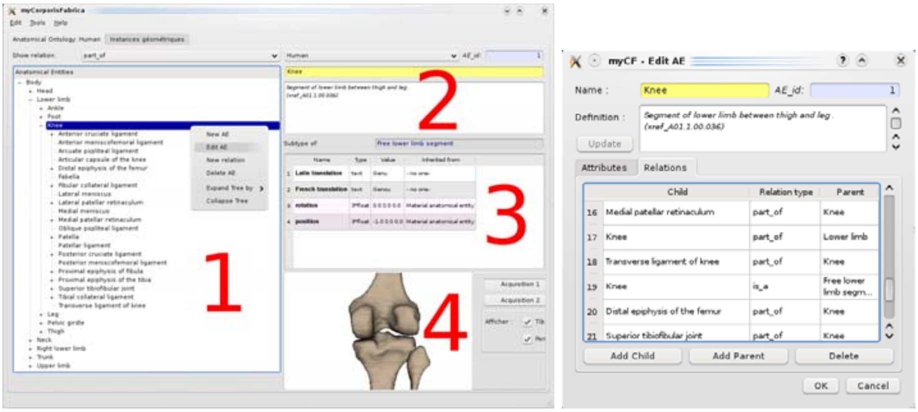
Geometry is often modeled based on medical images which represent several anatomical entities, which can be consistently viewed or simulated together. We therefore gather the anatomical entities modeled from a given set of medical images in *acquisitions*, which allows us to retrieve all of them simultaneously from the database.

4 Implementation

To ensure flexibility and extension, the canonical part of the database is described by only five tables. The ontology contains many taxonomies, as defined in section 3.2. As a consequence, many graphs are spread into this five tables and information retrieval can be tedious. To easily exploit the *MyCF* database, we propose a library called *libmyCF*, written in python. It contains classes representing the anatomical entities, their relations and attributes and some useful algorithms are present.

In the canonical part, we automatically deal with the case of the attributes inheritance. So, when the attributes of an anatomical entity are required, all the attributes of this entity are returned, plus all the attributes of its parents, defined by the relation *is_a*, if the parsed attributes are not yet defined.

Fig. 2. Blender is used to edit individual geometrical models. Blender objects are created from information stored in MyCF (import script). Positions, insertions of ligaments are editable and modifications can be saved in MyCF (exportation script). The tibial collateral ligament (MYCF:00031) is under edition. The ligament has two extremities (the proximal and the distal extremity) which are defined in MyCF.



Name : Knee AE_id: 1

Definition : *Segment of lower limb between thigh and leg. (xref_A01.1.00.036)*

	Child	Relation type	Parent
16	Medial patellar retinaculum	part_of	Knee
17	Knee	part_of	Lower limb
18	Transverse ligament of knee	part_of	Knee
19	Knee	is_a	Free lower limb segm...
20	Distal epiphysis of the femur	part_of	Knee
21	Superior tibiofibular joint	part_of	Knee

(a) (b)

Fig. 3. MyCF graphical user interface. **a)** Main window of MyCF. 1.the anatomical entities tree for the relation *part_of*, 2.information about the anatomical entity (AE), 3.attributes for this AE, 4.VTK visualization of an instance example. **b)** Edition of an anatomical entity. All the attributes and the relations of this anatomical entity are listed and editable here.

If the value of an attribute is modified and this attribute is inherited, a new attribute is created for this anatomical entity. Instance attributes are shown in the figure 3.

In the individual part, when the required instance does not exist, a new instance is created using the corresponding anatomical entity information. On the same way, when the required instance attribute is missing, it is automatically created from the attribute of the same type of the anatomical entity corresponding to the instance.

For instance, let us consider a knee modeled by two bones (femur and tibia) and a ligament, see Fig 2. During the importation of the instances into Blender, we need to know where to insert the extremities of the ligament on the bones. To do that, we just have to request the attribute named *position*. Usually, the value of this instance attribute is retrieved and the extremity is placed. Then, we can modify the position of the extremity on the bone in Blender and export this new value into the database.

The first time the extremity is imported into Blender, the instance attribute is not set in the database. When the import script treat the instance 'extremity of the ligament', the position value is requested. As there is no instance attribute, a new one is automatically created from the attribute of the anatomical entity corresponding to the instance of the extremity. Then, modifications can be done in Blender and exported.

Thus, the link with the knowledge contained in the canonical anatomical model is automically managed by the library, and plugins using MyCF information are easy to write.

5 Applications

Gathering ontological, geometrical and mechanical data consistently in a database allows new applications. We sketch some of them in this section.

5.1 Teaching

An obvious application of the database is the teaching of anatomy. MyCF can be used by students to handle anatomical knowledge within a consistent presentation. The graphical representations of anatomical entities allow a more intuitive way for knowledge-based navigation and understanding, as illustrated in Fig 3. Each anatomical entity can be displayed in 3D, while its logical relations with the other entities appear in the interface. Variability can be seen as well through 3D instances saved in the database. The Digital Anatomist Interactive Atlases integrates the FMA with 3D graphical illustrations but without real 3D interaction [15].

5.2 Mechanical Models

Beside geometry, we store mechanical parameters in the database, which allows to export an acquisition to physical simulation engines. The user can select the anatomical entities to export and the mechanical models to physically simulate them. For instance, to simulate a knee, the user can decide to export the bones and the main ligaments. Various mechanical models can be used to represent a given object, depending on the purpose of the simulation. For instance, studying the kinematics of a given joint based on bones and ligaments requires rigid bone models. Alternatively, studying bone fractures requires deformable models such as finite elements. When the rigid model is applied to a given entity, its mass and inertia matrix are automatically computed based on geometry and density. Alternatively, the user can chose to model an entity as a Finite Element Mesh with Hooke material. In this case, the same process is used to compute the mass and stiffness matrices, depending on the geometry, the shape functions as well as material parameters.

Deformable models such as finite elements require volumetric meshes. Meshes are an important issue in simulation, and we can not propose all meshing algorithms using various complexity and quality parameters. When volumetric tetrahedral meshes are available in the geometric data, they can be easily exported to finite element simulation modules. The precision and the computation time of finite element simulations heavily depend on the geometrical quality and the resolution of the volumetric meshes. Nesme et al.'s grid-based finite elements use deformable regular grids embedding objects defined by arbitrary geometrical models [16]. The grids have arbitrary resolution, which allows to trade off accuracy for speed according to the application, and the geometrical quality is not an issue since all cells are cubes. This approach allows to create finite element models for any virtual object at desired resolution.

5.3 Completion of a Model

A priori knowledge is useful to complete models where all anatomical entities are not defined. For instance, bone models may be built based on a CT-scan acquisition, whereas the connective tissues are not easily visible on the same data. In such a context, performing the mechanical simulation of a knee requires adding ligaments to the model, based on the knowledge of the ligaments properties and insertions locations on the bones.

MyCorporisFabrica allows the automatic creation of missing anatomical entities, using geometrical and material knowledge. The current implementation allows the creation of the main ligaments of the knee. The skeletal ligaments are modeled as one-dimensional mass-spring networks attached to the bones. To manage attach points, we have subdivided each ligament (using the *part_of* relation) in a proximal and a distal extremities. The areas on bones where skeletal ligaments are attached are clearly defined in the ontology. We use the relation *is_inserted_on* to inform where ligament's extremities are inserted. That exact position required for a given instance should be added manually. We're currenlty working on an automatic process to localise ligament insertions on instances using anatomical knowledge. Default values for spring stiffness and damping constants are stored in the database, as well as a default number of particles for the ligament. These values are modifiable by the user at creation time. Moreover,

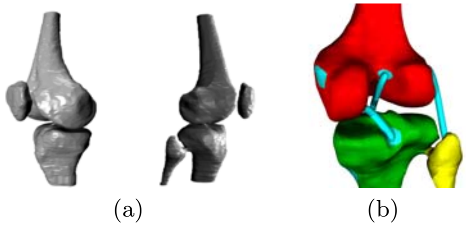

(a) (b)

Fig. 4. Completion of a model. **a)** Knee bones from an acquisition. **b)** Ligament models not visible in the acquision are added using anatomical knowledge.

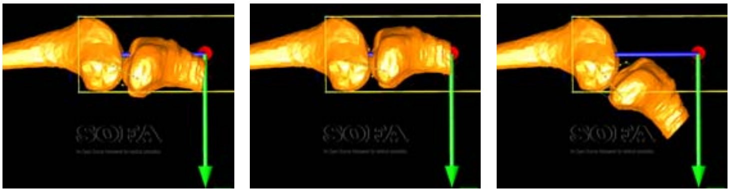


Fig. 5. Snapshots of a SOFA simulation of the model shown in Figure 4, with four ligaments

the 3D editor can be used to modify the attach points. Figure 4 illustrates this process.

Once the model is complete, it can be exported to a simulation engine, as illustrated in Figure 5. We use the SOFA simulation library [17] because it handles a variety of mechanical models and their interactions, it is free, open-source and highly customizable. However, export plugins to other simulation engines could be easily added to My Corporis Fabrica.

Currently, information encoded in digital shape is implicit. Coupling geometric information with knowledge can be a solution to get a computational access to embedded information. In the domain of patient-specific anatomy, digital shapes understanding will take a central role. Completion of a model is one of the most interesting application of anatomical ontology. A complete automatic process of completion have to deal with shapes understanding. A system to perform non-trivial segmentations of 3D surface meshes and to annotate the detected parts through concepts expressed by an ontology is presented by Attene et al. [18]. A similar approach is presented to handle multimedia content available on the Web in [19]. MyCF has to include metadata to ease shapes understanding.

6 Discussion

The main idea, which drives MyCF project, is to make available to the scientific community an open collaborative anatomical platform focused on the managing of patient specific data. Interested scientists should install and use MyCF on is own system (a MySQL server is required). We expect feedback from users. In the same time, we plan to maintain a open reference database with available 3D models. Access to MyCf as open source should facilitate further extensions. The curators of MyCF will be in charge to maintain consistency in anatomical representation. Therefore, access to manage this part of MyCF will be under restricted control.

To the best of our knowledge, MyCF is the first multidisciplinary project focused on a close collaboration around anatomical physical simulations. The anatomits can find a formal frame for anatomical knowledge with interactvie 3D visualization and nonanatomist users can manage patient-specific content through confirmed anatomical concepts.

MyCF is a brand new project. The knee is presented as a case study and has been used to built the system. We plan to increase amount of anatomical enitiies in MyCF. To do so the collaboration between users and curators must be strong and active. A collaborative web site around MyCF has been recently opened for this purpose. Readers are invited to visit *www.mycorporisfabrica.org*. In the future, MyCf should be included in the OBO Foundry (*www.obofoundry.org*).

References

1. Rosse, C., Mejino, J.L., Modayur, B.R., Jakobovits, R., Hinshaw, K.P., Brinkley, J.F.: Motivation and organizational principles for anatomical knowledge representation: the digital anatomist symbolic knowledge base. J. Am. Med. Inform. Assoc. 5(1), 17–40 (1998)
2. Rosse, C., Mejino, J.L.V.: A reference ontology for biomedical informatics: the foundational model of anatomy. J. Biomed. Inform. 36(6), 478–500 (2003)
3. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., Consortium, O.B.I., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.A., Scheuermann, R.H., Shah, N., Whetzel, P.L., Lewis, S.: The obo foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. 25(11), 1251–1255 (2007)
4. `http://www.focusedk3d.eu`
5. Magnenat-Thalmann, N., Schmid, J., Delingette, H., Guitian, J.I., Agus, M.: 3d anatomical modelling and simulation concepts. In: Eurographics 2009 Tutorial Notes (2009)
6. Webpage 3D Anatomical Human project `http://3dah.miralab.unige.ch` (accessed March 2008)
7. Brinkley, J.F., Rosse, C.: The digital anatomist distributed framework and its applications to knowledge-based medical imaging. J. Am. Med. Inform. Assoc. 4(3), 165–183 (1997)
8. Zhang, S., Bodenreider, O.: Law and order: assessing and enforcing compliance with ontological modeling principles in the foundational model of anatomy. Comput. Biol. Med. 36(7-8), 674–693 (2006)
9. Mork, P., Brinkley, J.F., Rosse, C.: Oqafma querying agent for the foundational model of anatomy: a prototype for providing flexible and efficient access to large semantic networks. J. Biomed. Inform. 36(6), 501–517 (2003)
10. Rosse, C., Kumar, A., Mejino, J.L.V., Cook, D.L., Detwiler, L.T., Smith, B.: A strategy for improving and integrating biomedical ontologies. In: AMIA Annu. Symp. Proc., pp. 639–643 (2005)

11. Charbonnier, C., Gilles, B., Magnenat-Thalmann, N.: A semantic-driven clinical examination platform. In: Surgetica 2007, Computer-Aided Medical Interventions: Tools and Applications, Chambry, France, September 2007, pp. 183–189 (2007)
12. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5(2), 199–220 (1993)
13. Smith, B., Ceusters, W., Klagges, B., Khler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A.L., Rosse, C.: Relations in biomedical ontologies. Genome. Biol. 6(5), R46 (2005)
14. `http://www.blender.org`
15. Wong, B., Rosse, C., Brinkley, J.: Semi-automatic scene generation using the digital anatomist foundational model. In: J. Am. Med. Assoc. AMIA 1999 Symp., Suppl. pp. 637–641 (1999)
16. Nesme, M., Kry, P., Jeřábková, L., Faure, F.: Preserving topology and elasticity for embedded deformable models. ACM Trans. Graph, Proceedings of SIGGRAPH 2009 (aout 2009)
17. Allard, J., Cotin, S., Faure, F., Bensoussan, P.J., Poyer, F., Duriez, C., Delingette, H., Grisoni, L.: SOFA - an open source framework for medical simulation. In: Medicine Meets Virtual Reality, MMVR 15, Long Beach, California, USA, February 2007, pp. 1–6 (2007), `http://www.sofa-framework.org`
18. Attene, M., Robbiano, F., Spagnuolo, M., Falcidieno, B.: Semantic annotation of 3d surface meshes based on feature characterization. In: Falcidieno, B., Spagnuolo, M., Avrithis, Y., Kompatsiaris, I., Buitelaar, P. (eds.) SAMT 2007. LNCS, vol. 4816, pp. 126–139. Springer, Heidelberg (2007)
19. De Floriani, L., Hui, L., Papaleo, L., Huang, M., Hendler, J.: A Semantic Web Environment for Digital Shapes Understanding.

Formal Representation of Tissue Geometric Features by DOGMA Ontology

Han Kang and Robert Meersman

Vrije Universiteit Brussel – STAR Lab, Pleinlaan 2, Gebouw G – 10,
1050 Brussel, Belgium
{han.kang,meersman}@vub.ac.be

Abstract. The mission of semantic data modelling lies in definition of the meaning of data within a context of data interrelationships. In our previous work, we created semantic models for geometric features of corporal tissues, and then developed an automatic tissue classification system. However, people who are not familiar with related background knowledge may not understand these highly specialized models. Consequently, these models cannot be shared and reused in other applications. Unlike semantic data modelling, ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is independent of specific applications. One of the ontology engineering approaches is DOGMA ontology. It consists of an ontology base which holds sets of conceptual relations and a layer of ontological commitments which holds the domain rules. In this paper, we propose to apply DOGMA ontology for formal representation of geometric features of corporal tissues, for the purpose of knowledge reusability.

Keywords: Geometric Feature, Tissue Classification, Semantic Data Modelling, Knowledge Reusability, DOGMA Ontology.

1 Introduction

Due to its bias toward the operation strategy employed by DBMS, the logical data structure of a DBMS, whether hierarchical or relational, cannot completely achieve in a conceptual definition of data. Thus, the requirement of data definition from a conceptual aspect has resulted in the development of semantic data modelling techniques. Semantic data modelling aims at obtaining more meaning of data by integrating relational concepts with more powerful abstraction concepts which are existing in the Artificial Intelligence field. It provides high level modelling primitives in order to facilitate the representation of real world situations [1].

In our previous work [2], we created semantic models for geometric features of corporal tissues, and then developed an automatic tissue classification system that combines both an image segmentation method [3] which is based on Fuzzy C-Means algorithm (FCM) [4] and these semantic models. We proposed such a combination because existing segmentation methods only partitions an image into a number of non-overlapped and constituent regions which are homogeneous with respect to some characteristics such as grey level or texture, but they cannot identify the

N. Magnenat-Thalmann (Ed.): 3DPH 2009, LNCS 5903, pp. 220–228, 2009.

physical significance of these segmented regions. Nevertheless, it may be difficult for people who do not hold related background knowledge to understand these highly specialized and expertized models. As a result, these models lack of shareability and reusability.

Different from semantic data modelling, the essential merit of ontology [5] is its independence of specific applications, i.e. ontology consists of relatively generic knowledge that can be shared and reused in different applications. One of the ontology engineering approaches is DOGMA (Developing Ontology-Grounded Methods and Applications) ontology [6]. It is composed of an ontology base which holds sets of conceptual relations and a layer of ontological commitments which holds the domain rules. The separation of ontology base from domain rules aims at enhancing the reuse and design scalability. In this paper, we propose to apply DOGMA ontology for formal representation of geometric features of corporal tissues. So that people with different needs and backgrounds can communicate and collaborate upon these medical knowledge.

2 Formal Representation of Geometric Features of Corporal Tissues

In this part, we will first introduce the foundations of DOGMA ontology. Then we will present the formal representation of geometric features of corporal tissues by DOGMA ontology.

2.1 Foundations of DOGMA Ontology

As we introduced above, DOGMA ontology has two components: ontology base and ontological commitments.

The ontology base is intended to capture domain axiomatizations. It consists of a set of binary conceptual relations [7]. The lexical rendering of a binary conceptual relation is called lexon. A lexon is formally described as <γ: Term a, Role, Term b>, where Term a and Term b are linguistic terms. γ is a context identifier which is used to bound the interpretation of a linguistic term into a concept. For each context γ and term i, the pair (γ, i) is assumed to refer to a concept. An ontology is based on many interpretation-independent plausible fact types about a universe of discourse [8]. The DOGMA ontology base is composed of them. In DOGMA, lexons are a more formal but still linguistically determined representation of propositions about a domain to be modelled. Thus, except for an intuitive linguistic interpretation, lexons have no formal interpretation or semantics [9]. In addition, it's stated that an ontology is language neutral [8], as suggested by [10], rather than language dependent as stressed by [11]. The logical (and thus unambiguous) vocabulary of a formal knowledge representation language should be clearly distinguished from potentially ambiguous natual language vocabulary expressing the intended meaning [12]. In the DOGMA framework, together with a context identifier, the language identifier plays an disambiguating role in determining which concept is linked with which word or natural language term [13]. Meta-lexons are the language-neutral and disambiguated counterpart of the language dependent lexons [8].

The commitment layer consists of a set of application axiomatizations. Particular applications commit to the ontology base through an application axiomatization. Such a commitment is called application ontological commitment. Each application axiomatization consists of: (1) a set of lexons from an ontology base; (2) a set of rules to constrain the usability of these lexons.

2.2 Formal Representation of Geometric Features of Corporal Tissues

In our previous work [2], we developed an automatic tissue classification system. It is composed of four components: image segmentation, knowledge acquisition, semantic data modelling, and rules generation. In component "image segmentation", we run a FCM-based algorithm [3] to partition an image into a number of constituent regions which are homogeneous regarding grey level. Due to this limitation, we often obtain several tissues in one segmented class and several classes corresponding to one tissue. For splitting each class into tissues and giving significance to each class, we propose to integrate qualitative medical knowledge on geometric properties of different tissues. In component "knowledge acquisition", we obtain knowledge through an online questionnaire. The experts are invited to fill this questionnaire or we search for the answers from the anatomical text books. In this questionnaire, the knowledge is expressed using linguistic description. We take human thigh as an example to show this questionnaire. For example, (1). Is spongy bone inside cortical bone? (2). Do muscle and adipose tissue border each other? From all the answers collected in the questionnaire, we concentrate on six features: (1). Relative positions between tissues. (2). Neighboring relation of each tissue. (3). Ranking order of all tissues according to their areas. (4). Number of connected components of certain tissues. (5). Tissues having similar grey level. (6). Shape of certain tissue. In component "semantic data modelling", we created semantic models for geometric features of corporal tissues. Then, in component "rules generation", we define rules according to the previous semantic models which permit to label tissues from image.

Due to the defect of semantic data modelling, these semantic models cannot be shared and reused. So we propose to apply DOGMA ontology for formal representation of geometric features of corporal tissues for the purpose of knowledge reusability. Assuming there are n tissues on a specific medical image denoted as Tissue 1, …, Tissue n, we formalize the tissue geometric features by DOGMA ontology to build an ontology base as follows.

(1) Relative positions between tissues.

<Geometric feature: Tissue i, Relative position, Tissue j>

where i, $j\in\{1, \ldots n\}$ and "Relative position" is an attribute between Tissue i and Tissue j, and it belongs to the set {above, below, left, right, above left, below left, above right, below right, inside, outside}. For example, <Geometric feature: Tissue i, below, Tissue j> means Tissue i is below Tissue j in the context of "Geometric feature".

(2) Neighboring positions between tissues.

<Geometric feature: Tissue i, Neighbors, Tissue j>

where i, $j\in\{1, \ldots n\}$. It means that Tissue i and Tissue j are Neighbors.

(3) Decreasing order of areas between tissues with one connected component.

<Geometric feature: Tissue i, Decreasing order of areas, Tissue j>

where i, $j\in\{1, \ldots n\}$. It means the area of Tissue i is greater than that of Tissue j.

(4) Numbers of connected components in tissues.

<Geometric feature: Tissue i, Has, Number of connected components>

where $i \in \{1, \ldots n\}$. It means that Tissue i has a specific number of connected components.

(5) Tissues having similar grey level.

<Geometric feature: Tissue i, Similar grey level, Tissue j>

where $i, j \in \{1, \ldots n\}$. It means that Tissue i and Tissue j have similar grey level.

(6) Shape of certain tissue.

<Geometric feature: Tissue i, Has, Shape>

where $i \in \{1, \ldots n\}$. It means that Tissue i has a specific shape. And the shape belongs to the set {triangle, circle, rectangle, …}.

After constructing the ontology base, we will work on the commitment layer. It is composed of a set of application axiomatizations. We focus on the application axiomatization of tissue classification. Then we define rules according to the lexons from the ontology base which permit to identify tissues from image. In these rules, all the terms and roles in the ontology base will be underlined.

(1) For any labelled tissue T_i, if there is only one unlabelled tissue T_j so that there is a Relative position between them, and if T_j does not have Similar grey level with any other tissues, then we can find the class corresponding to T_j by comparing positions of pixels and label it.

(2) For each unlabelled tissue T_i, if all its Neighbors of tissues have been labelled, then we can find and label T_i using connectivity analysis related to these neighboring tissues.

(3) If any pair of tissues which have Similar grey level has only one unlabelled tissue T_i, then we can rank all the unlabelled classes with one connected component in Decreasing order of areas and make equivalence between them and all ranked unlabelled tissues with one connected component.

(4) Given a specific Number of connected components, if the number of unlabelled tissues is 1, and if the number of unlabelled classes is 1, then this class is equivalent to the related tissue.

(5) For each group of tissues corresponding to one class obtained by the FCM-based image segmentation algorithm, (they have Similar grey level), if only one tissue in it has not been labelled, then we can easily separate and label it by deducing the pixels corresponding to the labelled tissues from this class.

(6) For each Shape, if the number of unlabelled tissues which have this Shape is 1, and if the number of corresponding unlabelled classes with one connected component is 1, then this class is equivalent to the related tissue.

3 Validation of the Previously Proposed System

After applying DOGMA ontology for the formal representation of geometric features of corporal tissues, we successfully applied our previously proposed system [2] to the tissue classification of human arm, forearm, thigh, and crus. We take the classification of thigh as an example to show this system. Thigh is composed of many tissues [14]. However, for simplicity, only four following tissues are considered [15]: muscle, adipose tissue, cortical bone, and spongy bone.

Fig. 1 shows one MRI image of thigh. It's in axial plane, T1-weighted, obtained from a system of Philips Intera of 1.5 Tesla. The size of voxel is 0.93 * 0.93 * 1.0 mm^3, and the coding is 12 bits. In addition, the signal has been analyzed by spin echo technique in order to minimize artifacts. We can see that there are four tissues and the background. And cortical bone and background have similar grey levels, and spongy bone and adipose tissue have similar grey levels.

Clinical interest of this study is to analyze the evolution of the volume of soft tissue at the position of mid thigh. Evidently, there is a relationship between the identified muscle surface and the strength [16]. The analysis on the evolution of soft tissues along the lower segments is also significant [17]. Other studies have shown that muscle volume in the hemiplegic subject is strongly correlated with an index of functional independence [18]. The tissue classification can also be used in biomechanics to calculate the inertial properties of body segments from MRI sequences [19].

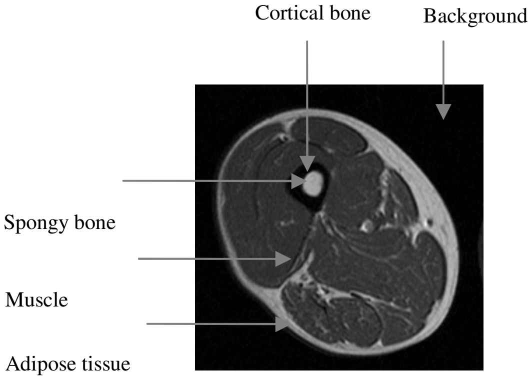


Fig. 1. A MRI image of thigh

When applying the proposed system, we run the FCM-based algorithm with 3 classes because only three main grey levels can be observed (see Fig. 2). Then, we obtain from the clustering result that cortical bone and background belong to one class C_1, adipose tissue and spongy bone to another class C_2, and muscle corresponds to C_3.

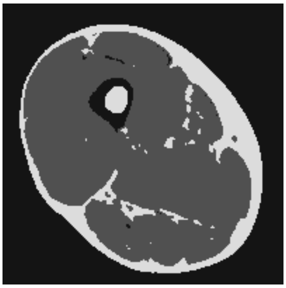


Fig. 2. Segmentation of thigh

Next, when applying the intelligent control by the above six rules, we first identify the background (see Fig. 3) because it is an object with only one connected component linking all the four bounds of image. We separate it from C_1 and then recognize cortical bone (see Fig. 4) from the remaining pixels in C_1 using Rule 5 with respect to the background. Next, we apply the Rule 2 with respect to the cortical bone for determining pixels of spongy bone (see Fig. 5) from C_2. The remaining pixels of C_2 can be then identified as adipose tissue (see Fig. 6) using Rule 5 with respect to the spongy bone. Using Rule 2 with respect to cortical bone and adipose tissue, we can label C_3 as muscle (see Fig. 7).

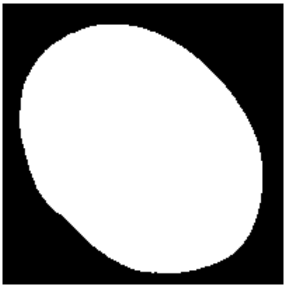


Fig. 3. Identification of background

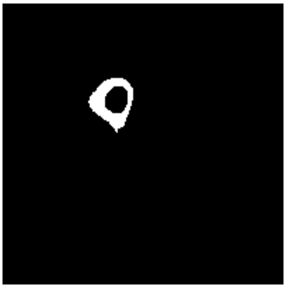


Fig. 4. Identification of cortical bone

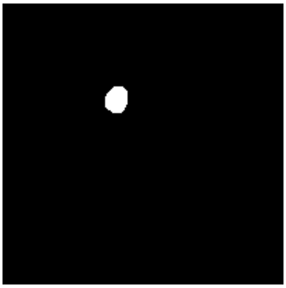


Fig. 5. Identification of spongy bone

Fig. 6. Identification of adipose tissue

Fig. 7. Identification of muscle

Using our proposed intelligent system, we effectively separated the four tissues and the background on the image of thigh. Then we reconstruct a synthetic image using the obtained five classes with following grey levels: background: 20; adipose tissue: 250; spongy bone: 150; cortical bone: 200; muscle: 60 (see Fig. 8b).

(a) (b)

Fig. 8. (a). The classification by a medical expert. (b). The classification by the proposed system.

Next, we do quantitative comparison of the classification by our system and a reference classification (see Fig. 2a) provided by a medical expert in Lille Regional University Hospital (CHRU de Lille, France). The comparison criterion (cc) is:

$$cc = \frac{A_{ij} \cap A_{refj}}{A_{ij} \cup A_{refj}} . \tag{1}$$

where A_{ij} is the set of pixels belonging to the *j*th class obtained by the *i*th method, and A_{refj} is the set of pixels belonging to *j*th class in the reference image. Table 1 shows the corresponding quantitative comparison.

Table 1. Quantitative comparison of classification.

Tissue	Result
Adipose tissue	82.58%
Cortical bone	84.87%
Spongy bone	83.98%
Muscle	95.42%

4 Conclusion

In this paper, we present the application of DOGMA ontology for formal representation of geometric features of corporal tissues which aims at knowledge reusability and shareability. We first build an ontology base for these geometric features. Then we define rules according to the lexons from the ontology base which permit to identify tissues from image. In future, we will make more efforts for exploring other tissue features and add them to the ontology base. And we will define new rules according to the new lexons in ontology base. We will also work on the quantification of the muscle/fat ratio, assessment of the muscle/fat temporal variation, and measurement of the volume of muscle, fat and bone in human legs based on the obtained results. In addition, we will validate the proposed system on other parts of human body.

Acknowledgments. This work was supported by the Marie Curie Research Training Network project "3DanatomicalHuman" within EU's Sixth Framework Programme. The project was under Contract No. MRTN-CT-2006-035763.

References

1. ter Bekke, J.H.: Semantic Data Modeling. Prentice Hall, Hemel Hempstead (1992)
2. Han, K., Xianyi, Z., Abdelmalik, T.A., Antonio, P.: Integration of Human Knowledge for Automatic Tissue Classification on Medical Images. In: 8th International FLINS Conference on Computational Intelligence in Decision and Control, Spain, pp. 447–452 (2008)
3. Chen, S., Zhang, D.: Robust Image Segmentation Using FCM with Spatial Constraints Based on New Kernel-Induced Distance Measure. IEEE. Trans. Sys. M. Cyber. 34, 1907–1916 (2004)
4. Bezdek, J.: A Convergence Theorem for the Fuzzy Data Clustering Algorithms. IEEE. Trans. Pat. Ana. Mac. Int. 2, 1–8 (1980)
5. Decker, S., Fensel, D., Harmelen, F.V., Horrocks, I., Melnik, S., Klein, M., Broekstra, J.: Knowledge representation on the Web. In: International Workshop on Description Logics, pp. 89–97, CEUR-WS.org. (2000)

6. Mustafa, J., Robert, M.: Formal Ontology Engineering in the DOGMA Approach. In: Meersman, R., Tari, Z., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 1238–1254. Springer, Heidelberg (2002)
7. Robert, M.: The Use of Lexicons and Other Computer-Linguistic Tools in Semantics, Design and Cooperation of Database Systems. In: Zhang, Y., Rusinkiewicz, M., Kambayashi, Y. (eds.) CODAS 1999, pp. 1–14. Springer, Heidelberg (1999)
8. Peter, S., Yan, T., Robert, M.: An Ontology Engineering Methodology for DOGMA. J. App. Ont. 3, 13–39 (2008)
9. Robert, M.: Ontologies and Databases - More Than a Fleeting Resemblance. In: International OES/SEO Workshop. Luiss Publications (2001)
10. Nirenburg, S., Raskin, V.: Ontological Semantics, Formal Ontology, and Ambiguity. In: 2nd International Conference on Formal Ontology in Information Systems, pp. 151–161, Ogunquit (2001)
11. Guarino, N.: Formal Ontologies and Information Systems. In: 1st International Conference on Formal Ontology in Information Systems, Trento, pp. 3–15 (1998)
12. Peter, S., De Bo, J.: Ontologies – A Revamped Cross Disciplinary Buzzwork or a Truly Promising Interdisciplinary Research Topic? J. Lin. Ant. 3, 279–292 (2004)
13. De Bo, J., Peter, S., Robert, M.: Creating a “DOGMAtic” Multilingual Ontology Infrastructure to Support a Semantic Portal. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2003. LNCS, vol. 2889, pp. 253–266. Springer, Heidelberg (2003)
14. Weir, J., Abrahams, P.H.: Imaging Atlas of Human Anatomy. Times Mirror International Publishers Limited, London (1997)
15. Hédoux, P.: Détermination de Paramèters Biomécaniques Personnalisés à Partir d’Imagerie Médicale, Application aux Sujets Hémiplégique. PhD dissertation, Université de Valenciennes et du Hainaut-Cambrésis, France (2004)
16. Rambaud, F., Pinti, A., Hédoux, P.: Correlations Between Morphological Change of the Lower Limbs and Functional Recovery of Hemiplegic Patients. J. Inno. Tec. Bio. Med. 28, 13–19 (2007)
17. Hédoux, P., Pinti, A.: Measurement and Analysis of Volume Evolution of Soft Tissues on the Longor of the Shank and the Thigh in Hemiplegic Subjects with MRI. J. Adv. Mod. Simu. Ent. 67, 116–126 (2006)
18. Maughan, R.J., Watson, J.S., Weir, J.: Strength and Cross-Sectional Area of Human Skeletal Muscle. J. Phy. 338, 37–49 (1983)
19. Cheng, C.K., Chen, H.H., Chen, C.S., Lee, C.L., Chen, C.Y.: Segment Properties of Chinese Adults Determined from Magnetic Resonance Imaging. J. Cli. Bio. 15, 559–566 (2000)

Author Index