

Path-Based Mathematical Morphology on Tensor Fields

Jasper J. van de Gronde, Mikola Lysenko, and Jos B.T.M. Roerdink

Abstract Traditional path-based morphology allows finding long, approximately straight, paths in images. Although originally applied only to scalar images, we show how this can be a very good fit for tensor fields. We do this by constructing directed graphs representing such data, and then modifying the traditional path opening algorithm to work on these graphs. Cycles are dealt with by finding strongly connected components in the graph. Some examples of potential applications are given, including path openings that are not limited to a specific set of orientations.

1 Introduction

An image can be considered as a function whose argument is a position. However, with increasing computing power and increasing sensor capabilities, it is becoming more and more common to acquire data that varies as a function of both position *and* orientation. For example: orientation scores (and similar schemes) [9, 15–17, 29, 32], flow fields, diffusion MRI data [28, 48] and seismic anisotropy data [40]. When filtering such data this additional structure can be taken into account to extract more relevant data.

Path openings [23, 24] are a class of morphological filters that are able to extract long and thin structures. They essentially allow preserving pixels that are part of a long path, while suppressing pixels that are only part of short paths (this allows these filters to be interpreted as hyperconnected filters [53]). Applied to traditional binary or greyscale images this can be useful for finding cracks, roads, fibres, and other thin elongated structures [10, 34, 37, 45, 49].

J.J. van de Gronde • J.B.T.M. Roerdink (✉)

Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen,
P.O. Box 407, 9700 AK Groningen, The Netherlands
e-mail: j.j.van.de.gronde@rug.nl; j.b.t.m.roerdink@rug.nl

M. Lysenko

Department of Computer Sciences, University of Wisconsin - Madison, 1210 W. Dayton St.,
Madison, WI 53706-1613, USA
e-mail: mikolalysenko@gmail.com

© Springer International Publishing Switzerland 2015

I. Hotz, T. Schultz (eds.), *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, Mathematics and Visualization,
DOI 10.1007/978-3-319-15090-1_6

In this work we motivate why the concept of a path is natural to consider in the context of (symmetric) tensor fields. We then explain how path openings traditionally work on directed *acyclic* graphs, and how they can be generalized to allow for cycles. This leads to an efficient algorithm, which we use to demonstrate that our method can indeed be used to produce sensible results on (tensorial) orientation scores and diffusion MRI data.

1.1 Related Work

Certain fibre tractography methods already use a graph-based approach [6, 25, 44, 47]. However, these are typically undirected graphs, and are primarily used for finding tracts (using *shortest* paths), and not for filtering the data. The use of directed graphs in this work allows us to prevent a path from doubling-up on itself unnecessarily, while still allowing very efficient filtering of the *entire* data set. Still, the graph building methods employed by these tractography methods could be a source of inspiration.

Some of the work on mathematical morphology on tensor fields by Duits et al. [18] also considered dilations along what can be considered as “paths” following the local orientation. But rather than using path openings, more traditional dilations and erosions were used (it is not immediately clear whether these operations could be used to implement path openings). And instead of using a graph-based approach, a PDE-based approach was used. Franken and Duits [20] applied a similar approach to orientation scores (although without focussing on morphological operators).

Bismuth et al. [4] developed a method to at least partially deal with curved features in a close relative of path openings, by iteratively linking up approximately linear segments. However, they only support a limited number of changes of direction, and still rely on being able to define suitable directed acyclic graphs. In contrast, here we simply forgo the requirement that the graph is acyclic. In addition, our method could be used to link up approximately linear segments with different orientations too, without the need to explicitly build chains of a certain length. It is not immediately clear how some of their other refinements could translate to our work though.

The technique recently developed by Morard et al. [38] could in theory also be adapted to compute path openings on tensor fields, as it simply finds a number of (potential) paths and then filters along those using a 1D algorithm (instead of implicitly filtering along *all* paths). However, we would still need a suitable method for finding paths. Also, it is not yet clear how well it approximates the “true” path opening.

2 Definitions and Notation

2.1 Mathematical Morphology and Openings

Mathematical morphology is a framework for non-linear image processing based on (algebraic) lattices [22]. A lattice is a partially ordered set such that every two elements have a uniquely defined least upper bound and a uniquely defined greatest lower bound. These are called the join/supremum (\vee) and meet/infimum (\wedge), respectively. Typically, one works with lattices of images, using a partial order that compares images in a pixel-wise fashion: an image is less than or equal to another image if every pixel value in it is less than or equal to the corresponding pixel value in the other image. We will work mostly with the lattice of graphs, in which one graph is less than or equal to another graph if it is a subgraph of the other graph.

Central in the theory of mathematical morphology is a categorization of operators based on the properties they have. For example, an operator is called an erosion if it distributes over taking the meet ($\varepsilon(a \wedge b) = \varepsilon(a) \wedge \varepsilon(b)$). Another important class of operator is the *opening*. An (algebraic) opening is an operator from a lattice to another (possibly the same) lattice that is [22, 41]:

- increasing* a smaller input implies a smaller output,
- anti-extensive* the output is less than or equal to the input, and
- idempotent* the output is a fixed point of the operator.

So more input means more output, you cannot get more out than you put in, and applying an opening twice does the same as applying it once. If an opening depends on some parameter to control its “strength”, it is often possible to produce an intermediate data structure called an *opening transform* that contains information on which pixels to keep for what parameter values.

2.2 Tensors

We consider a tensor field to be an image whose pixel values are tensors based on the (local) tangent space. For example, even though a vector can be considered a tensor, we do not consider a colour image to describe a tensor field, as a colour vector does not describe a direction in the image domain.

We will only concern ourselves with real, symmetric tensors based on (tangent spaces of) Euclidean spaces. Most crucially, the tensors are based on some vector space \mathcal{V} with an inner product \cdot . We consider symmetric tensors to be built from the symmetrized tensor product \odot [7, 31], so a rank- r , degree- n (or order- n) tensor \mathbf{A} is any tensor that can be written as a weighted sum of r (but not less than r) tensors of the form [11]

$$\mathbf{a}^{\odot n} = \underbrace{\mathbf{a} \odot \dots \odot \mathbf{a}}_{n \text{ times}}, \text{ with } \mathbf{a} \in \mathcal{V}.$$

The terms in such a weighted sum form a so-called “rank-one decomposition” of a tensor, analogous to the eigendecomposition of a symmetric matrix. Due to its linearity, the inner product on tensors is fully determined by

$$\mathbf{a}^{\odot n} \cdot \mathbf{b}^{\odot n} = (\mathbf{a} \cdot \mathbf{b})^n.$$

The *identity tensor* \mathbf{I}_n is the unique degree- n symmetric tensor that satisfies $\mathbf{I}_n \cdot \mathbf{a}^{\odot n} = \|\mathbf{a}\|^n$, with n even. See our previous work [21, 50] for details.

2.3 Graphs

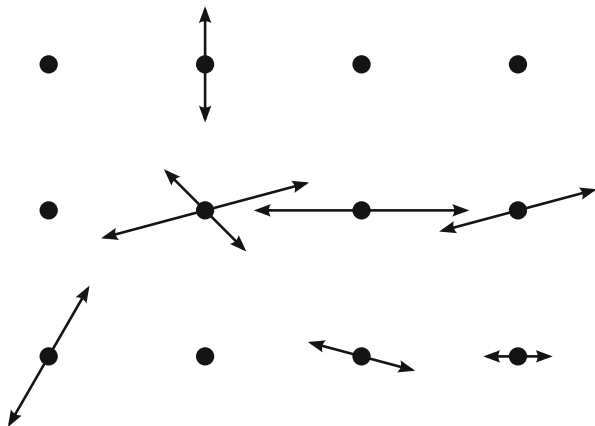
A directed graph G can be identified with a pair (V, E) containing a set of vertices and a set of edges, with all of the edges being pairs of vertices. A graph is a subgraph of another graph if both its vertex set and its edge set are subsets of the vertex set and the edge set, respectively, of the other. A cycle is a sequence of edges of the form $(v_1, v_2), (v_2, v_3), \dots, (v_n, v_1)$ (that is, if an edge ends at vertex v then the next edge starts at vertex v , and the first and last vertex are also the same). If a directed graph does not contain any cycles it is called a directed acyclic graph, or DAG. The vertices in a directed acyclic graph can always be ordered in such a way that if there is an edge from a vertex to another vertex, then this other vertex comes after the first in the ordering. Such an ordering is called a topological ordering of the graph.

3 From Tensor Fields to Paths in Graphs

Previously [21, 50, 51], we looked at generalizing mathematical morphology to vector-valued images and tensor fields. For this, it was found that it is important to construct rotation-invariant operators. This is in line with some older research [2, 8, 42], as well as the more recent work by Angulo [1] for example. We accomplished this by “lifting” to rotation-invariant representations. For tensor fields, this boiled down to taking a function that maps a position to a tensor (describing a tensor field), and turning it into a function that maps a tangent vector¹ to a scalar (for example: if f maps positions to degree- n tensors, then one could construct a function f' on tangent vectors defined by $f'(x, \mathbf{v}) = f(x) \cdot \mathbf{v}^{\odot n}$, where the pair (x, \mathbf{v}) represents a tangent vector). This makes it much easier to apply traditional morphological concepts, as these are typically already suited to scalar images.

¹A tangent vector can be considered an element in the tangent bundle, and is a *combination* of a position and a vector describing an orientation/direction. Physically, a tangent vector can be considered to describe the position and velocity of a particle, for example.

Fig. 1 Illustration of what (spherically deconvolved) diffusion MRI data looks like. There is a regular grid of points, and for each of those points there are zero or more antipodal pairs of directions with weights (the arrow length corresponds to the weight). Any two antipodal directions have the same weight. The idea is that vectors that roughly line up could be indicative of a fibre bundle running through those vectors



By adapting the theorem that showed the usefulness of lifting to a rotation-invariant representation [51, Thm. 1], we can also consider operations like tensor decomposition to be admissible lifting operators. That is, we can consider the tensor decomposition as a rotation invariant map from the tensor space to a rotation invariant lattice. Although we will not prove this here, this *is* part of our (intuitive) justification for using tensor decompositions for morphological filters on tensor fields.

Once we view a tensor field as a function on tangent vectors—or as a sparse set of (weighted) tangent vectors (see Fig. 1)—it becomes natural to look for paths in the data. This is like looking for streamlines [35] in a flow field, or performing tractography on diffusion tensor fields [3, 14, 36, 39]. More fundamentally, orientations really only make sense if they line up somehow, naturally giving rise to the concept of a path.

In some cases, like gradients of functions, it might be more natural to consider something like a “hyperplane/sheet” opening (extracting surfaces or other structures that are more than one dimensional), but this is somewhat outside the scope of the current work. We do show an example (using orientation scores) in 2D though, where a hyperplane is also a line, so that it still makes sense to use a path opening.

In our examples we build graphs in which the vertices correspond to tangent vectors. There is an edge from a tangent vector to another tangent vector with the same orientation at a neighbouring position if and only if the neighbouring position is within 0.65 times the cell spacing of the ray spanned by the first tangent vector. More formally, if we denote the set of all possible edges on the current grid by \mathcal{N} (so regardless of which tangent vectors are in any particular data set), then (with unit cell spacing, x_1, x_2 in some Euclidean space, \mathbf{v} a unit vector in the tangent space, and ignoring the weights associated with the tangent vectors):

$$((x_1, \mathbf{v}), (x_2, \mathbf{v})) \in \mathcal{N} \iff (x_2 - x_1) \cdot \mathbf{v} \geq 0$$

$$\text{and } \|(x_2 - x_1) - ((x_2 - x_1) \cdot \mathbf{v})\mathbf{v}\| \leq 0.65.$$

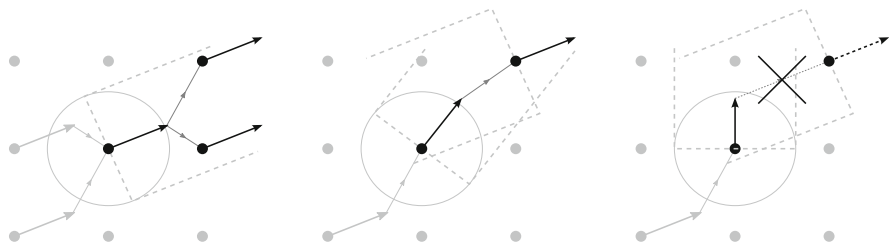


Fig. 2 A tangent vector is connected to another tangent vector with the same direction at a neighbouring position if the neighbouring position is within a set distance from the ray spanned by the first tangent vector. For connecting to different orientations, positions are chosen only if both orientations “agree” on being able to reach/come from each other’s position

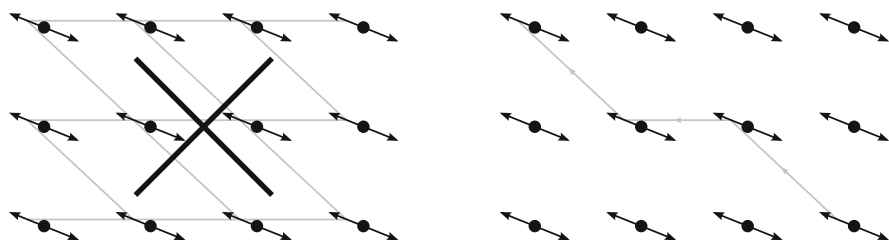


Fig. 3 If we connect vectors to other vectors in a bidirectional way (*left*), we quickly link everything together and lose the idea behind finding paths (the *grey line* could be a single path!). That is why we only connect vectors to other vectors that lie roughly in the direction the vector is pointing to (*right figure* shows an example path). Note that the *grey line* segments connect the midpoints of the *arrows* representing tangent vectors

We also connect tangent vectors to tangent vectors with a different orientation if they “agree” on being able to reach each other’s position (see Fig. 2):

$$((x_1, \mathbf{v}_1), (x_2, \mathbf{v}_2)) \in \mathcal{N} \iff ((x_1, \mathbf{v}_1), (x_2, \mathbf{v}_1)) \in \mathcal{N}$$

$$\text{and } ((x_1, \mathbf{v}_2), (x_2, \mathbf{v}_2)) \in \mathcal{N}.$$

If $G = (V, E)$ is the graph we build for a particular data set, with V containing all the tangent vectors, then $E = (V \times V) \cap \mathcal{N}$. See Fig. 3 for why it is important that the above relation is not symmetric.

The above connectivity scheme strikes a balance that seems to work reasonably well in practice. However, it would definitely be interesting to examine more closely what kind of connectivity makes sense for this kind of data. In particular, though it might be tempting to try to fit “trajectories” through tangent vectors to determine the connectivity between neighbouring positions and orientations, in practice the discrete grid (of positions) works against this. Also, a good balance should be struck between being able to track sharp turns and preventing spurious connections.

Note that since in our applications we always have antipodal pairs of tangent vectors, it is useful to have the concept of an “opposite”. For each graph $G = (V, E)$ we have a map of opposites $o : V \rightarrow V$ that gives the antipodal partner of each vertex. We have: $(v_1, v_2) \in E \iff (o(v_2), o(v_1)) \in E$, for all $v_1, v_2 \in V$.

4 Path Openings on Graphs with Cycles

Traditionally [23, 24], binary path openings are computed by constructing several directed acyclic graphs (DAGs), one for each direction, like in Fig. 4. These graphs are then used to determine the lengths of the longest paths running through each point using a dynamic programming scheme. Said paths are constrained to form a (connected) path in one of the graphs. So one can have an approximately horizontal path, an approximately vertical path, etc. (depending on the number of directions used). Some enhancements have been developed [34], but the basic idea remains the same. This means that path openings are not well suited to extract curved features.

In most work on path openings, weighted graphs are considered, with the weights corresponding to—binary—pixel values. In our setting it is more convenient to simply consider the graph induced by those pixels that are “on”. We define a path a in a graph $G = (V, E)$ to be a sequence of vertices such that any pair of consecutive vertices is an edge in E . The set of vertices in a is denoted by $\sigma(a)$, its length by $|a|$, and the set of all paths in G by $\Pi(G)$. A path opening on a directed *acyclic* graph G , with path length threshold L , can then be defined as:

$$\alpha_L(G) = \bigcup \{ \sigma(a) \mid a \in \Pi(G) \text{ and } |a| \geq L \}.$$

Introducing cycles in the graph poses a problem: there is no such thing as a *longest* path in the presence of a cycle. One solution is to constrain paths to never visit the same *vertex* twice. However, finding the length of the longest path through each vertex would then be NP-hard (even approximations are hard [5, 27]). Constraining paths to never traverse the same *edge* twice might be simpler in some cases, but it is not immediately clear in what cases, nor how we could (easily) find such paths. Also, if the idea is to find the *longest* (or, rather, largest) path, why should we avoid cycles in the path?

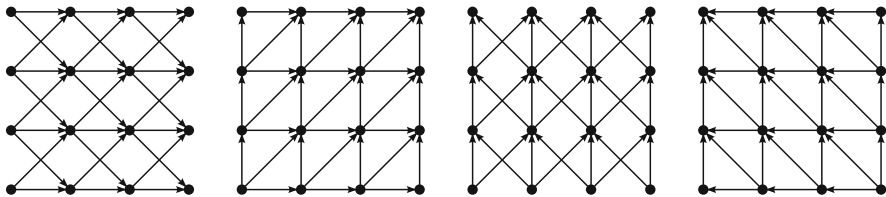


Fig. 4 A set of DAGs that has been used for (traditional) 2D path openings

The key is to focus on path *size* ($|\sigma(a)|$) instead of path *length* ($|a|$). For a DAG, there is no difference, as a vertex can only occur once in a path in a DAG anyway, but in the presence of cycles it can make a big difference. In particular, traversing a cycle multiple times does not make the path larger (in the sense of the size of the set of vertices covered by the path). In the interest of brevity, we will, from now on, say “path size” instead of “the size of the set of vertices covered by a path”. So rather than finding longest paths through vertices, we find, for each vertex, the maximum *size* of all paths through that vertex. We thus (re)define the path opening as follows:

$$\alpha_S(G) = \bigcup \{ \sigma(a) \mid a \in \Pi(G) \text{ and } |\sigma(a)| \geq S \}. \quad (1)$$

The subscript S will be dropped whenever it is immaterial to the matter at hand. Also, when convenient we equate $\alpha_S(G)$ with the subgraph of $G = (V, E)$ it induces: $(\alpha_S(G), \{(a_1, a_2) \in E \mid a_1, a_2 \in \alpha_S(G)\})$.

Note that the above allows us to apply path openings to oriented data and find curved paths, as we demonstrate in Sect. 5. Also, we give an efficient algorithm to compute α_S in Sect. 4.2. But first we show that α_S is indeed worthy of the title *path opening*.

4.1 α_S is an Algebraic Opening

The above develops a generalization of the traditional path opening, but it is not immediately clear that the result is in fact still an opening in the algebraic sense of the word. That is, it should be increasing, anti-extensive and idempotent. Here we first show that the operator does in fact still have these properties in an abstract graph-based setting, and then go on to discuss how this applies to some more concrete settings.

Theorem 1 *A path opening α on directed graphs is an opening: increasing, anti-extensive and idempotent. In particular, for all directed graphs G and H , if G is a subgraph of H , then $\alpha(G)$ is a subset of $\alpha(H)$ (increasing). Also, $\alpha(G)$ induces a subgraph of G (anti-extensive), and $\alpha(\alpha(G)) = \alpha(G)$ (idempotent), where we equate $\alpha(G)$ with the subgraph of G that it induces.*

Proof We start by observing that α is almost trivially anti-extensive: by definition it returns a subset of the vertices of the graph (inducing a subgraph).

To see that α is increasing, observe that if G is a subgraph of H , any path possible in G is also possible in H . The maximum path size through every vertex in H that is also in G should thus be greater than or equal to the maximum path size of the same vertex in G . It follows that $\alpha(G) \subseteq \alpha(H)$.

That α is idempotent follows from the fact that if it returns a vertex, it also returns all vertices in the path that made it return that vertex. So all of the vertices returned by the path opening will have large enough paths in the induced subgraph to be

preserved by a second application of the path opening: $\alpha(\alpha(G)) \supseteq \alpha(G)$. And since we already concluded that α is anti-extensive, implying that $\alpha(\alpha(G)) \subseteq \alpha(G)$, we have that α must be idempotent.

In the above, filtering is done by removing vertices from the graph rather than “turning them off” (as in traditional path openings). However, if we have a graph with boolean weights, then we can simply consider the subgraph that contains only those vertices that are turned on. Since paths do not propagate through vertices that are turned off, this has no effect on the result. Using this trick the above can be interpreted as a(nother) proof that traditional path openings are indeed openings.

When applying the above to orientation scores (see Sect. 5.1), where we use tensor decompositions to compute sparse sets of tangent vectors, it is not clear yet to what extent the whole procedure (decomposition, filtering, reconstruction) can be considered an opening. The main issues are picking a suitable order, and the possible ambiguity of tensor decompositions. Since the path opening itself is anti-extensive, we suspect the Loewner order [8, 50] might be sufficient to prove anti-extensivity, but it is unlikely to be of use for proving increasingness. As for tensors with ambiguous decompositions (analogous to having eigenvalues with multiplicity greater than one), a more detailed study of how often and where they occur would be needed. As it is, we consider this firmly outside the scope of the current work, but look forward to future developments.

4.2 Implementation

The first step in realizing an efficient algorithm to find the maximum path sizes through all vertices, is to observe that all vertices within a cycle share the same longest path size, and that this size is greater than or equal to the cycle size. Also, since for counting the path size it does not matter *where* we enter the loop, we can contract any loop of n vertices into a single vertex that we count as n vertices without affecting the results (see Fig. 5). If we do this with all cycles, then we clearly end up

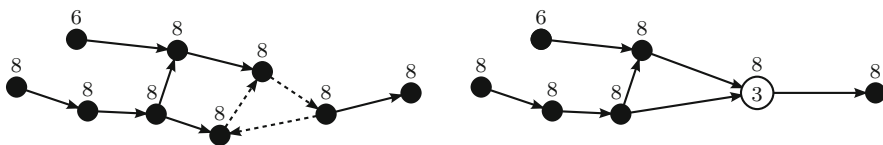


Fig. 5 *Left*: original directed graph containing a cycle (*dashed*), along with the maximum path “lengths” through each vertex. *Right*: same graph after contracting the cycle into a single vertex with weight 3 (rather than 1)

Algorithm 1: Path opening on graphs, allowing for cycles

Input : A directed vertex-weighted graph $G = (V, E, w)$, with $w : V \rightarrow \mathbb{R}$, a map of opposites $o : V \rightarrow V$, and a threshold S .

Output: A set $A \subseteq V$.

$H = (V', E', w')$ \leftarrow graph of strongly connected components in G

construct $o' : V' \rightarrow V'$ such that $v \in v'$ and $u \in u'$ and $o(v) = u \implies o'(v') = u'$

find $\lambda_+ : V' \rightarrow \mathbb{R}$ using Algorithm 2

$A \leftarrow \emptyset$

for $v' \in V'$ **do**

$\lambda \leftarrow \lambda_+(v') + w'(v') + \lambda_+(o'(v'))$

if $\lambda \geq S$ **then**

$A \leftarrow A \cup v'$

with a directed acyclic graph, allowing the application of a traditional path opening algorithm.

Can we easily find the graph resulting from contracting all cycles? (Known as the condensation of a graph.) Yes: by finding strongly connected components. A strongly connected component in a directed graph is a (maximal) set of vertices such that there is a path in the graph from every vertex to every other vertex in the set. The set of all vertices in a directed graph can be partitioned into strongly connected components. As all vertices in a cycle clearly belong to the same strongly connected component, contracting the cycle does not change the partition. So after we have contracted all cycles, each vertex of the contracted graph corresponds to a strongly connected component in the original graph. We use Tarjan's algorithm [46] to find all the strongly connected components in a graph in linear time.

Instead of just counting the number of vertices in a path, Algorithm 1 uses the sum of all vertex (and edge) weights associated with the set of vertices (and edges) covered by the path. The weight of a strongly connected component is the sum of the weights of the vertices (and edges) in the component. Also, we only find the sizes of paths that *end* in each vertex (up to, but not including, the weight of the vertex itself). Since in our context the vertices (tangent vectors) come in antipodal pairs, the maximum size of a path through a vertex is then determined by combining its associated path size with the path size found for its opposite (and the vertex's own weight).

Our implementation of a (one-sided) path opening on a directed (acyclic) graph in Algorithm 2 uses an algorithm by Kahn [26] to compute the topological ordering, but avoids outputting this topological order by integrating the body of the (outer) loop in Algorithm 2 with the topological sort.

Algorithm 2: One-sided path opening transform

Input : A directed acyclic vertex-weighted graph $G = (V, E, w)$, with $w : V \rightarrow \mathbb{R}$.
Output: A map $\lambda : V \rightarrow \mathbb{R}$ giving the maximum lengths of paths ending in each vertex (excluding the weight of the vertex itself).

initialize $\lambda : V \rightarrow \mathbb{R}$ to zero
 $O \leftarrow$ topological sort of V
for o in O **do**
 $l \leftarrow \lambda(o) + w(o)$
 for $s \in V \mid (o, s) \in E$ **do**
 $\lambda(s) \leftarrow \max\{\lambda(s), l\}$

5 Examples

We now apply the above to two types of oriented data: tensorial orientation scores and diffusion MRI data. These proof-of-concepts illustrate how our methods can be applied to oriented data. We use tensor decomposition and spherical deconvolution to obtain sparse sets of tangent vectors and associated weights that together describe the data. We then build graphs on these tangent vectors using simple rules for deciding what vectors can be connected. Finally, we apply the above to filter the data, and show the results.

5.1 Orientation Scores

Our first example² takes a 2D image and computes an orientation score inspired on work by Duits and Franken [15–17, 19, 20], except that we use a fourth order tensor to represent the orientation score rather than a discrete number of directions based on a discrete approximation of a tensor-valued filter defined in the Fourier domain by $F(\xi) = \frac{\xi^{\odot n}}{\|\xi\|^n}$. We then use the (symmetric) higher order power method [30] in combination with multiplicative update rules for non-negative least squares [43] to find a decomposition of the tensors into positively weighted rank-one tensors and the identity tensor (in as far as this is possible). Since it proved slightly simpler to extract positive components from a tensor than negative components, we ignore negative components; our example has a dark background and light foreground to ensure that the main features are represented by positive components. We also ignore the weight of the identity tensor, as we are only interested in oriented features. The compound effect of these measures is shown in Fig. 6.

Having found directions and weights, we rotate the directions by 90° so they point along the edge/line, and turn them into antipodal pairs. For building a graph

²Code available at <http://bit.ly/1zpfIXf>.

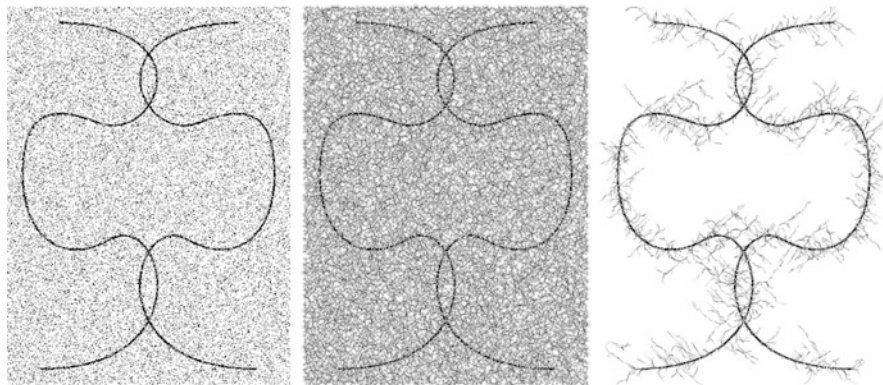


Fig. 6 From *left to right*: the original image; the reconstruction from the vectors and values extracted from the orientation tensors (ignoring any negatively weighted components, as well as the identity component); the reconstruction after path opening with a threshold of 400 pixels. Since the image dimensions are 387×517 , this threshold is way too large for a traditional path opening to have picked up any part of the curves. Note that for the path opening the weights are ignored (and kept unchanged). Grey values have been inverted, so the filter was run on an image with a dark background and light curve

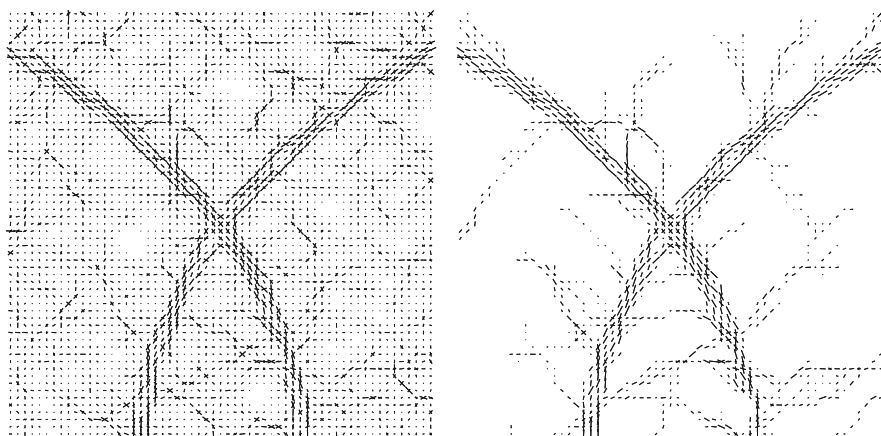


Fig. 7 *Left*: the original orientations found (near the top-most crossing in Fig. 6). *Right*: the output of the path opening. The orientations are shown as small line segments extending symmetrically from the center of each pixel (the length corresponding to the weight)

on the orientation score data, we use the rules already discussed in Sect. 3. For this example we use one additional constraint though: we only connect tangent vectors to other tangent vectors if they differ in angle by less than 30° . We compute a path opening on the graph, and then reconstruct the data from the output graph. Figure 7 shows what the intermediate sets of tangent vectors look like.

Figure 6 shows that we can extract a long curved path from a noisy background, even though we use a threshold that is in fact higher than the width of the image, and definitely higher than any of the approximately straight paths that could have been picked up by a traditional path opening. Also, our method is not limited to fixed set of orientations, it works with whatever orientations are found by the tensor decompositions. This is in stark contrast to traditional path openings, which have to process the entire image once per orientation, and require relatively complicated schemes to be able to distinguish between more than eight orientations (in 2D) [52].

5.2 Diffusion MRI

Our spherically deconvolved diffusion MRI dataset has a (short) list of directions and associated weights for each point in a regular (although slightly anisotropic) 3D grid. Because of the physics of diffusion, these directions again come in antipodal pairs, with the same weight for both directions. To build a graph on the diffusion MRI data, we use the same rules as explained in Sect. 3.

ExploreDTI [33] was used for motion and eddy currents correction, while StarTrack [12] was used for spherical deconvolution. In principle tensor decomposition could be considered a kind of spherical deconvolution, but for diffusion MRI it is often appropriate to use a different kernel (fibre response function) to deconvolve with. Also, we typically do not start with a tensor, but rather with measurements in many discrete directions. The resulting path opening is demonstrated in Fig. 8. Figure 9 shows the largest strongly connected component.

In this example we remove tangent vectors whose weight is below a threshold. However, at crossings, the values tend to be smaller than in regions with a single

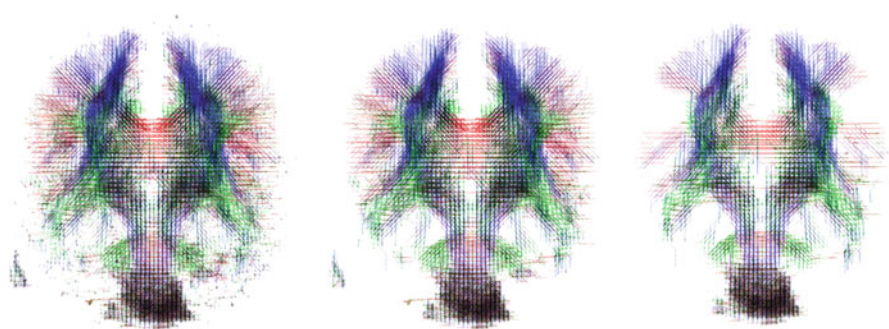
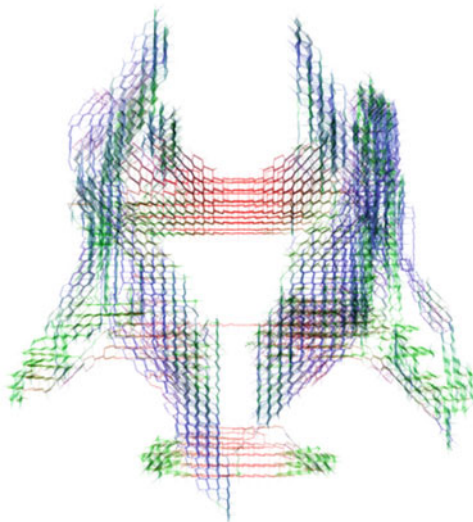


Fig. 8 The result of performing a path opening on spherically deconvolved (and thresholded) diffusion MRI data. From *left to right*: the original graph; a path opening with length threshold five; and a path opening with length threshold 100. The most striking differences are visible along the periphery. Path length was measured in voxels; the complete volume contains $128 \times 128 \times 69$ voxels. Interactive version available at <http://bit.ly/1zpfIXf>

Fig. 9 Largest strongly connected component present in the (thresholded) diffusion MRI data (containing 16,540 elements). Apart from this strongly connected component there was one other of a size of several thousands (corresponding to the noisy blob at the bottom of the data visible in Fig. 8), and two with a size of only four elements. All 668,966 remaining strongly connected components had size one (so were not part of a cycle)



fibre orientation.³ Our attempt at compensating for this assumes that all directions suffer the same attenuation, and that this factor is equal to the maximum weight for a voxel divided by the sum of all the weights in a voxel (compensating for the fact that vectors come in pairs).

A more traditional way of finding paths/fibres in diffusion MRI is using tractography. In a common variant of tractography the diffusion MRI data is used to extract a finite set of trajectories/fibres representing the layout of the actual nerve fibre bundles. Here, tractography was performed using StarTrack [13]. Figure 10 shows some examples of how the constructed graph compares to more traditional tractography. For comparison purposes we took the markers used to select trajectories in the tractography data and used those for what amounts to a flood fill procedure on the graph we constructed, limited to staying within a certain number of steps from the marker. The main differences (in what structures are found) appear to be caused by our method's tendency to be more sensitive to small branches (it picks up on what, when taking the data at face value, could be a connection from the fornices to the corpus callosum for example), and the traditional method's ability to follow trajectories at a low(er) threshold (at least in order to cross small gaps).

³This makes some intuitive sense, as every voxel contains the same amount of space, but at a crossing it is shared by fibres of multiple orientations.

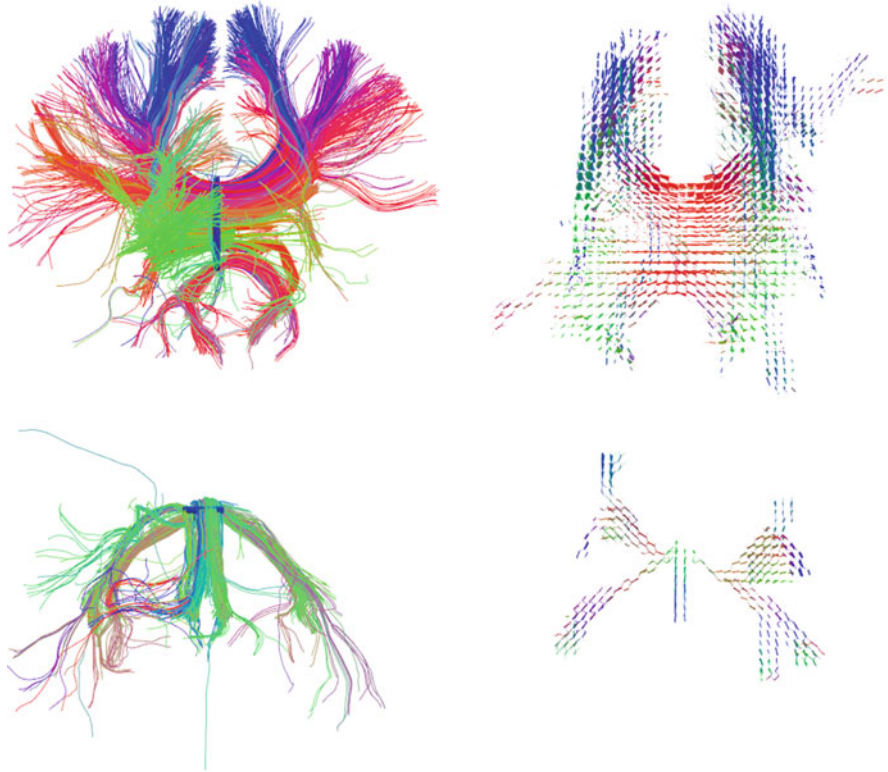


Fig. 10 Tractography of the corpus callosum (*top*) and the fornices (*bottom*), and propagation of the same markers (seedpoints) on the graph we constructed (on the *right*, showing all tangent vectors within 20 steps of the markers). Note that the views are all at roughly the same scale and look onto the anterior of the brain, but that the graph-based images are much sparser due to only plotting tangent vectors at voxel positions (as well lines between connected tangent vectors)

Although our method is perhaps not really suitable as a drop-in replacement for more traditional tractography methods, it does have some interesting characteristics:

- Symmetry of the connections. By construction, if our method finds a path from position and orientation A to position and orientation B, it will also find a reversed path. Tractography methods based on numerically integrating PDEs will tend to miss small branches of larger tracts.
- Output data has the same format as input data (it is just sparser). In fact, as the result of an algebraic opening, the output is a subset of the input.

Also, in principle, we could imagine first thinning the data (within the same kind of framework), and *then* filtering it. If we record the thickness of the bundles, this could then be taken into account during filtering. Essentially this would mean working with bundles rather than paths.

6 Conclusion and Future Work

We have shown how we can deal with cycles in graphs so as to still have an easy and efficient algorithm for path openings. The idea is to contract all cycles (or rather, strongly connected components) to single vertices weighted by the sum of the weights of the vertices in the contracted cycles. The remaining *acyclic* condensation graph can be filtered using more traditional path opening algorithms.

In contrast to traditional path openings, our approach allows extracting curved paths without any problems. We can also easily support tensor fields and similar data. In principle we could even construct path openings for data whose “image domain” is a sphere or other curved space/manifold. We do note, however, that picking any arbitrary graph structure is unlikely to be successful, some amount of sparsity and a tendency to avoid (unnecessary) cycles is crucial in getting sensible results. Our examples show that it is entirely feasible to apply our method to certain classes of real-world data though.

In the future we hope to look at other attributes than length or size for paths and proper greyscale path openings. We suspect that for certain attributes the strongly connected component decomposition can still help, as long as we want to compute those attributes over maximal paths (maximal in the sense that there are no other paths of which they are proper subsets). The condensation of a graph makes it easier to deal with cycles when analyzing paths in directed graphs.

Regarding tensors and tensor decompositions, there are still some unresolved issues. For example, how likely is it to have an ambiguous (minimum symmetric-rank) tensor decomposition? How does this change if we add identity tensors to the possible “basis” tensors? What if we put certain constraints on the tensors? Ultimately, we hope to be able to consider the path openings we have created here as algebraic openings on tensor fields (and not just graphs built on top of those tensor fields), but so far the possibility or impossibility of this is still an open question. In addition, it would be interesting to explore tensor decompositions that can also deal with negative components, for use with orientation scores.

We are also highly interested in looking at *bundles* (paths with a width) rather than paths (without width). In 3D it might also be worthwhile to attempt finding sheets rather than paths. Finally, we would like to develop versions of our path openings that are more robust to noise [10, 45], and experiment with “greyscale” path openings on (decomposed) tensor fields.

Acknowledgements The first and third author were funded by the Netherlands Organisation for Scientific Research (NWO), project no. 612.001.001. Also, we thank Remco Renken and Jelmer Kok from the NeuroImaging Center Groningen for supplying us with the diffusion MRI data (including regions of interest and tractography results), as well as for some inspiring discussions.

References

1. Angulo, J.: Supremum/infimum and nonlinear averaging of positive definite symmetric matrices. In: Nielsen, F., Bhatia, R. (eds.) *Matrix Information Geometry*, pp. 3–33. Springer, Berlin/Heidelberg (2013). doi:10.1007/978-3-642-30232-9_1
2. Astola, J., Haaavisto, P., Neuvo, Y.: Vector median filters. *Proc. IEEE* **78**(4), 678–689 (1990). doi:10.1109/5.54807
3. Bastiani, M., Shah, N.J., Goebel, R., Roebroek, A.: Human cortical connectome reconstruction from diffusion weighted MRI: the effect of tractography algorithm. *NeuroImage* **62**(3), 1732–1749 (2012). doi:10.1016/j.neuroimage.2012.06.002
4. Bismuth, V., Vaillant, R., Talbot, H., Najman, L.: Curvilinear structure enhancement with the polygonal path image - application to guide-wire segmentation in X-ray fluoroscopy. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) *Medical Image Computing and Computer-Assisted Intervention. Lecture Notes in Computer Science*, vol. 7511, pp. 9–16. Springer, Berlin/Heidelberg (2012). doi:10.1007/978-3-642-33418-4_2
5. Björklund, A., Husfeldt, T., Khanna, S.: Approximating longest directed paths and cycles. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) *Automata, Languages and Programming. Lecture Notes in Computer Science*, vol. 3142, pp. 222–233. Springer, Berlin/Heidelberg (2004). doi:10.1007/978-3-540-27836-8_21
6. Booth, B.G., Hamarneh, G.: Multi-region competitive tractography via graph-based random walks. In: *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 73–78 (2012). doi:10.1109/mmbia.2012.6164747
7. Bourbaki, N.: *Algebra I. Elements of Mathematics*. Springer, Berlin (1989)
8. Burgeth, B., Bruhn, A., Didas, S., Weickert, J., Welk, M.: Morphology for matrix data: ordering versus PDE-based approach. *Image Vis. Comput.* **25**(4), 496–511 (2007). doi:10.1016/j.imavis.2006.06.002
9. Citti, G., Sarti, A.: A cortical based model of perceptual completion in the Roto-translation space. *J. Math. Imaging Vision* **24**(3), 307–326 (2006). doi:10.1007/s10851-005-3630-2
10. Cokelaer, F., Talbot, H., Chanussot, J.: Efficient robust d-dimensional path operators. *IEEE J. Sel. Top. Signal Process.* **6**(7), 830–839 (2012). doi:10.1109/jstsp.2012.2213578
11. Comon, P., Golub, G., Lim, L.H., Mourrain, B.: Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.* **30**(3), 1254–1279 (2008). doi:10.1137/060661569
12. Dell’Acqua, F., Scifo, P., Rizzo, G., Catani, M., Simmons, A., Scotti, G., Fazio, F.: A modified damped Richardson–Lucy algorithm to reduce isotropic background effects in spherical deconvolution. *NeuroImage* **49**(2), 1446–1458 (2010). doi:10.1016/j.neuroimage.2009.09.033
13. Dell’Acqua, F., Simmons, A., Williams, S.C.R., Catani, M.: Can spherical deconvolution provide more information than fiber orientations? Hindrance modulated orientational anisotropy, a true-tract specific index to characterize white matter diffusion. *Hum. Brain Mapp.* **34**(10), 2464–2483 (2013). doi:10.1002/hbm.22080
14. Descoteaux, M., Deriche, R., Knosche, T.R., Anwander, A.: Deterministic and probabilistic tractography based on complex fibre orientation distributions. *IEEE Trans. Med. Imaging* **28**(2), 269–286 (2009). doi:10.1109/tmi.2008.2004424
15. Duits, R.: *Perceptual organization in image analysis: a mathematical approach based on scale, orientation and curvature*. PhD thesis, Eindhoven University of Technology (2005)
16. Duits, R., Franken, E.: Left-invariant parabolic evolutions on $SE(2)$ and contour enhancement via invertible orientation scores. Part I: Linear left-invariant diffusion equations on $SE(2)$. *Q. Appl. Math.* **68**(2), 255–292 (2010). doi:10.1090/s0033-569x-10-01172-0
17. Duits, R., Franken, E.: Left-invariant parabolic evolutions on $SE(2)$ and contour enhancement via invertible orientation scores. Part II: Nonlinear left-invariant diffusions on invertible orientation scores. *Q. Appl. Math.* **68**(2), 293–331 (2010). doi:10.1090/s0033-569x-10-01173-3
18. Duits, R., Dela Haije, T.C.J., Creusen, E., Ghosh, A.: Morphological and linear scale spaces for fiber enhancement in DW-MRI. *J. Math. Imaging Vision* **46**(3), 326–368 (2013). doi:10.1007/s10851-012-0387-2

19. Franken, E.M.: Enhancement of crossing elongated structures in images. PhD thesis, Eindhoven University of Technology (2008)
20. Franken, E., Duits, R.: Crossing-preserving coherence-enhancing diffusion on invertible orientation scores. *Int. J. Comput. Vis.* **85**(3), 253–278 (2009). doi:10.1007/s11263-009-0213-5
21. van de Gronde, J.J., Roerdink, J.B.T.M.: Frames for tensor field morphology. In: Nielsen, F., Barbaresco, F. (eds.) *Geometric Science of Information. Lecture Notes in Computer Science*, vol. 8085, pp. 527–534. Springer, Berlin/Heidelberg (2013). doi:10.1007/978-3-642-40020-9_58
22. Heijmans, H.J.A.M.: *Morphological Image Operators*. Academic, Boston (1994)
23. Heijmans, H., Buckley, M., Talbot, H.: Path-based morphological openings. In: *IEEE International Conference on Image Processing*, vol. 5, pp. 3085–3088 (2004). doi:10.1109/icip.2004.1421765
24. Heijmans, H., Buckley, M., Talbot, H.: Path openings and closings. *J. Math. Imaging Vision* **22**(2), 107–119 (2005). doi:10.1007/s10851-005-4885-3
25. Iturria-Medina, Y., Canales-Rodríguez, E.J., Melie-García, L., Valdés-Hernández, P.A., Martínez-Montes, E., Alemán-Gómez, Y., Sánchez-Bornot, J.M.: Characterizing brain anatomical connections using diffusion weighted MRI and graph theory. *NeuroImage* **36**(3), 645–660 (2007). doi:10.1016/j.neuroimage.2007.02.012
26. Kahn, A.B.: Topological sorting of large networks. *Commun. ACM* **5**(11), 558–562 (1962). doi:10.1145/368996.369025
27. Karger, D., Motwani, R., Ramkumar, G.D.S.: On approximating the longest path in a graph. *Algorithmica* **18**(1), 82–98 (1997). doi:10.1007/bf02523689
28. Kingsley, P.B.: Introduction to diffusion tensor imaging mathematics: Part I. Tensors, rotations, and eigenvectors. *Concepts Magn. Reson.* **28A**(2), 101–122 (2006). doi:10.1002/cmr.a.20048
29. Knutsson, H., Westin, C.F., Andersson, M.: Structure tensor estimation: introducing monomial quadrature filter sets. In: Laidlaw, D.H., Vilanova, A. (eds.) *New Developments in the Visualization and Processing of Tensor Fields. Mathematics and Visualization*, pp. 3–28. Springer, Berlin/Heidelberg (2012). doi:10.1007/978-3-642-27343-8_1
30. Kofidis, E., Regalia, P.A.: On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM J. Matrix Anal. Appl.* **23**(3), 863–884 (2002). doi:10.1137/s0895479801387413
31. Kostrikin, A.I., Manin, I.I.: *Linear Algebra and Geometry. Algebra, Logic and Applications*, vol. 1. Gordon and Breach, Amsterdam (1997)
32. Köthe, U.: Edge and junction detection with an improved structure tensor. In: Michaelis, B., Krell, G. (eds.) *Pattern Recognition. Lecture Notes in Computer Science*, vol. 2781, pp. 25–32. Springer, Berlin/Heidelberg (2003). doi:10.1007/978-3-540-45243-0_4
33. Leemans, A., Jeurissen, B., Sijbers, J., Jones, D.K.: ExploreDTI: a graphical toolbox for processing, analyzing, and visualizing diffusion MR data. In: *ISMRM 17th Scientific Meeting & Exhibition*, p. 3537 (2009)
34. Luengo Hendriks, C.L.: Constrained and dimensionality-independent path openings. *IEEE Trans. Image Process.* **19**(6), 1587–1595 (2010). doi:10.1109/tip.2010.2044959
35. McLoughlin, T., Laramée, R.S., Peikert, R., Post, F.H., Chen, M.: Over two decades of integration-based, geometric flow visualization. *Comput. Graph. Forum* **29**(6), 1807–1829 (2010). doi:10.1111/j.1467-8659.2010.01650.x
36. Melhem, E.R., Mori, S., Mukundan, G., Kraut, M.A., Pomper, M.G., van Zijl, P.C.M.: Diffusion tensor MR imaging of the brain and white matter tractography. *Am. J. Roentgenol.* **178**(1), 3–16 (2002). doi:10.2214/ajr.178.1.1780003
37. Morard, V., Dokladal, P., Decencière, E.: One-dimensional openings, granulometries and component trees in $O(1)$ per pixel. *IEEE J. Sel. Top. Signal Process.* **6**(7), 840–848 (2012). doi:10.1109/jstsp.2012.2201694
38. Morard, V., Dokladal, P., Decencière, E.: Parsimonious path openings and closings. *IEEE Trans. Image Process.* **23**(4), 1543–1555 (2014). doi:10.1109/tip.2014.2303647
39. Mori, S., van Zijl, P.C.M.: Fiber tracking: principles and strategies – a technical review. *NMR Biomed.* **15**(7–8), 468–480 (2002). doi:10.1002/nbm.781
40. Nolet, G.: *A Breviary of Seismic Tomography*. Cambridge University Press, New York (2008)

41. Serra, J. (ed.): *Theoretical Advances, Image Analysis and Mathematical Morphology*, vol. 2. Academic, London (1988)
42. Serra, J.: Anamorphoses and function lattices. In: Dougherty, E.R., Gader, P.D., Serra, J.C. (eds.) *Image Algebra and Morphological Image Processing IV*, SPIE Proceedings, vol. 2030, pp. 2–11 (1993). doi:10.1117/12.146650
43. Sha, F., Lin, Y., Saul, L.K., Lee, D.D.: Multiplicative updates for nonnegative quadratic programming. *Neural Comput.* **19**(8), 2004–2031 (2007). doi:10.1162/neco.2007.19.8.2004
44. Sotiropoulos, S.N., Bai, L., Morgan, P.S., Constantinescu, C.S., Tench, C.R.: Brain tractography using Q-ball imaging and graph theory: improved connectivities through fibre crossings via a model-based approach. *NeuroImage* **49**(3), 2444–2456 (2010). doi:10.1016/j.neuroimage.2009.10.001
45. Talbot, H., Appleton, B.: Efficient complete and incomplete path openings and closings. *Image Vis. Comput.* **25**(4), 416–425 (2007). doi:10.1016/j.imavis.2006.07.021
46. Tarjan, R.: Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1**(2), 146–160 (1972). doi:10.1137/0201010
47. Terajima, K., Nakada, T.: EZ-tracing: a new ready-to-use algorithm for magnetic resonance tractography. *J. Neurosci. Methods* **116**(2), 147–155 (2002). doi:10.1016/s0165-0270(02)00039-0
48. Tournier, J.D., Mori, S., Leemans, A.: Diffusion tensor imaging and beyond. *Magn. Reson. Med.* **65**(6), 1532–1556 (2011). doi:10.1002/mrm.22924
49. Valero, S., Chanussot, J., Benediktsson, J.A., Talbot, H., Waske, B.: Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognit. Lett.* **31**(10), 1120–1127 (2010). doi:10.1016/j.patrec.2009.12.018
50. van de Gronde, J.J., Roerdink, J.B.T.M.: Frames, the Loewner order and eigendecomposition for morphological operators on tensor fields. *Pattern Recognit. Lett.* (2014). doi:10.1016/j.patrec.2014.03.013
51. van de Gronde, J.J., Roerdink, J.B.T.M.: Group-invariant colour morphology based on frames. *IEEE Trans. Image Process.* **23**(3), 1276–1288 (2014). doi:10.1109/tip.2014.2300816
52. Vincent, L.: Minimal path algorithms for the robust detection of linear features in gray images. In: Heijmans, H.J.A.M., Roerdink, J.B.T.M. (eds.) *Mathematical Morphology and Its Applications to Image and Signal Processing*, ISMM '98, pp. 331–338. Kluwer Academic Publishers, Norwell, MA (1998)
53. Wilkinson, M.H.F.: Hyperconnectivity, attribute-space connectivity and path openings: theoretical relationships. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) *Mathematical Morphology and Its Application to Signal and Image Processing*. Lecture Notes in Computer Science, vol. 5720, Chap. 5, pp. 47–58. Springer, Berlin/Heidelberg (2009). doi:10.1007/978-3-642-03613-2_5