International Journal of Image and Graphics

DEFORMABLE MESH FOR AUTOMATED SURFACE EXTRACTION FROM NOISY IMAGES

JUSSI TOHKA
Digital Media Institute/Signal Processing, Tampere University of Technology, P.O. Box 553
Tampere, FIN-33101, Finland
jussi.tohka@tut.fi

JOUNI M. MYKKÄNEN
Department of Computer and Information Sciences, University of Tampere,
Tampere, FIN-33014, Finland
jouni.mykkanen@cs.uta.fi

Received 13 September 2002
Revised 27 February 2003
Accepted 31 March 2003

Surface extraction from noisy volumetric images is a problem commonly encountered in medical image analysis. Deformable surface models can, in principle, solve the problem in an automatic manner. However, it is often required that a reasonably close initialization and good parameter values for deformable models are provided. In this paper, novel algorithms for global minimization of the energy of deformable meshes are presented. We demonstrate that global optimization by these algorithms reduces sensitivity of the deformable mesh to its initialization and to its parameter values. Consequently, it becomes easier to automate the initialization process and selection of parameter values. As the second contribution, the internal energy function is derived in a novel way in the framework of deformable surface models. The construction of the internal energy in this way features a simple way to derive the variants of our global optimization algorithm. The experiments with synthetic images are performed to compare variants of the proposed optimization algorithm. Also, we present a practical application of our deformable model to automatic segmentation of positron emission tomography images.

Keywords: Deformable model; segmentation; surface mesh; medical image analysis; energy minimization

1. Introduction

The emerging interest in quantitative analysis of volumetric medical images has set new requirements for extracting biologically meaningful structures automatically from three-dimensional (3-D) image data. Imperfections typical of medical images, such as noise and non-uniformity of intensity values within structures of interest, cause the boundaries of the structures appear disconnected and inhomogeneous in the image. This fact combined with the variability of biological shapes necessitates the use of advanced methods for surface extraction and segmentation within medical imaging.

Deformable models [1,2], based on a Bayesian insight, are widely applied for medical image analysis [3]. With deformable models, surface extraction is formulated as an energy minimization problem. The energy of deformable surface models is divided into two parts: the external energy and the internal energy. The external energy depends on an underlying image and draws the surface towards salient image features while the internal energy regularizes the shape of the surface. A traditional approach for solving locally the minimization problem is to consider a time varying surface under the influence of external and internal forces. External forces are then derived from image data and internal forces control properties, such as smoothness, of the surface. A wide interest in deformable surface models has led to various formulations of them and we do not review these here in detail. Instead, we refer to a recent survey article [4].

Our aim, motivated by the segmentation problem of functional emission tomography images, is to design a novel deformable model for automatic surface extraction from noisy images. The model is based on the simplex mesh representation of surfaces [5]. We expect that surfaces to be extracted are homeomorphic to the sphere and their approximate size is known. Hence, we do not have to deal with mesh decimation or mesh subdivision techniques. The constrained topology of surfaces is also the reason behind applying meshes for surface representation instead of e.g. level sets [6], which allow implicit changes of surface topology. Due to noise, we cannot expect automatic initialization routines based on binarization such as in [7] to perform well. Also, the energy function to be minimized will have numerous local minima. Therefore, surface extraction is treated as a global optimization problem. We propose effective algorithms for the global optimization task, which are inspired by ideas from [8]. As our second contribution we derive and analyze the internal energy function in a novel way within the framework of deformable meshes. For this, we extend the shape modeling scheme for active contours from [9] to cover deformable meshes and to suite our algorithms for global optimization.

The organization of the paper is as follows. In Sect. 2, deformable surface meshes are introduced in a more detailed manner concentrating on the energy minimization methods that have been applied with them. In Sect. 3, we review simplex meshes, which we use for representing surfaces. The original definition of the simplex mesh [5] is modified to contain a global position parameter. The energy function to be minimized is defined and analyzed in Sect. 4. In Sect. 5, we propose algorithms for global energy minimization. Experiments with synthetic images are described and quantitative results are reported in Sect. 6. In Sect. 6, we also summarize the results of our comparative study [10] between a few recent deformable surface mesh models. The deformable model introduced in this paper has been used to solve segmentation problems within positron emission tomography (PET) and we consider this application in Sect. 7. The segmentation procedure is detailed in a conference article [11]. Section 8 discusses our deformable model in comparison to other relevant models and Sect. 9 concludes the paper.

2. Background: Deformable Surface Meshes

Deformable surface meshes are polygon meshes whose geometry evolves driven by image data and shape regularity constraints starting from given initial meshes. These are used for extracting surfaces whose topology is known but whose shape varies individually. There are two basic approaches for controlling the mesh evolution. In the force-based approach one assigns an equation of motion to each node of the mesh (or mexel) and lets it move in a force field formed by internal and external forces until equations of motion stabilize. In the energy-based approach, an energy is assigned to each mesh belonging to the set of admissible meshes. The energy is minimized and the mesh of the minimal energy is selected as the result. Remark that some force-based methods do not have an energy minimization interpretation and correspondingly some energy minimization algorithms cannot be implemented as force-based methods. Hence, the two approaches can be considered to be truly different.

If initializations for processes are not in the close vicinity of the target surface, advanced solutions for both approaches are required. Especially, when images are noisy, the energy function is likely to have multiple local minima and as well the equations of motion may stabilize before the correct surface has been found. Addition of artificial inflation or deflation forces, such as with snakes [2], is not a durable solution to the initialization problem. This can lead to high sensitivity to values of parameters controlling the relative strength of the added artificial forces, which makes it difficult to select proper values of them. Above mentioned problems of initialization and parameter sensitivity limit automatic application of deformable models.

As already mentioned, we focus on the energy-based approach in this paper. A few methods for globalized energy minimization within the framework of deformable meshes have been reported. For example, simple coarse-to-fine schemes have been applied, see e.g. [12]. The approach is to optimize the geometry of a coarse mesh with a local minimization algorithm and then increment the resolution of the resulting mesh. Then, another local optimization is initiated starting from the mesh of the incremented resolution. This is continued iteratively until the desired mesh resolution has been attained. Although the approach has been effective for certain applications, problems could be expected when dealing with a noisy input. Even with the coarsest resolution of the mesh, the optimization problem is hard enough for a simple local optimization method to fail. This will easily lead successive optimizations to wrong tracks.

A divergent method for the minimization of the energy that can be also characterized as coarse-to-fine one was presented in [13] (cf. also [14]). There segments of the surface mesh underwent a deformation which locally minimized the energy of the segment. This was carried out in a hierarchical manner where the surface segments considered became smaller as the process advanced. The deformable mesh was applied particularly for extracting brain structures from magnetic resonance images in [13]. Although their optimization method is interesting, they used it in a combination of the complicated shape model for their specific application. Hence, it is unclear how good the optimization method would be when a broader class of structure extraction problems is considered.

In [15], they globalized energy minimization by modifying the parameters for the internal energy function of the deformable mesh. However, they did not explain how to modify these parameters automatically and neither did they define a stopping condition for the optimization process.

Finally, we studied the applicability of real coded genetic algorithms (RCGA)[16] for deformable mesh optimization in [17]. The problem with RCGAs are their high computational costs which complicate their use for practical applications.

3. Simplex Meshes with a Global Position Parameter

A closed surface is represented by a simplex mesh [5,18]. It consists of discrete points $\mathbf{w}_i \in \mathbb{R}^3$, called *mexels*, and connectivity relations between them. The set of mexels is $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$. The connectivity relations are modeled by a 3-regular simple graph $\mathcal{M}$, the graph of the mesh. Each mexel is connected to three other mexels, which are called its *neighbours.* The three neighbours of $\mathbf{w}_i$ are denoted by $\mathbf{w}_{i_j} (j = 1, 2, 3)$. The graph of the mesh is assumed to be known and constant and hence the set of mexels defines the mesh completely.

Simplex meshes are topological duals of triangular meshes [18]. In other words, for each simplex mesh there exists a triangular mesh whose graph is the dual of the graph of the simplex mesh. This can be interpreted in the sense that each mexel of a simplex mesh represents a triangle face in a triangular mesh. See Fig. 1 for an example of triangular mesh and its dual simplex mesh.

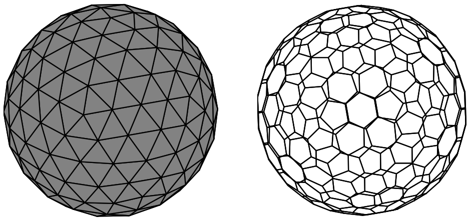


Fig. 1. On the left is an example of triangular mesh and on the right is a dual simplex mesh of the triangular mesh.

While the topological structure of the mesh is exactly as in [5], we introduce a different geometric structure for the mesh for the use of the internal energy function generalized from [9]. This subsequently allows us to develop differing versions of our optimization algorithm. Mexels $\mathbf{w}_i$ are set relative to *a reference point* $\mathbf{g}_W \in \mathbb{R}^3$. Actual positions of mexels in an image are $\hat{\mathbf{w}}_i = \mathbf{w}_i + \mathbf{g}_W$. A surface mesh is then represented as a pair $(\mathbf{W}, \mathbf{g}_W)$, where $\mathbf{W}$ is referred as the *surface centered mesh.* The set $\hat{\mathbf{W}} = \{\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_N\}$ is the unique (image centered) *actual mesh* induced by $(\mathbf{W}, \mathbf{g}_W)$. Note that different pairs consisting of a surface centered mesh and an associated reference point may induce exactly the same actual mesh.

Given an actual mesh $\hat{\mathbf{W}}$, the reference point $\mathbf{g}_W$ can be independent of $\hat{\mathbf{W}}$ as well as a function of $\hat{\mathbf{W}}$. In the latter case, we consider only reference points defined as

$$\mathbf{g}_W = \frac{1}{N} \sum_{i=1}^{N} \hat{\mathbf{w}}_i, \tag{1}$$

where N is the number of mexels and we say that reference points are *constrained.*

4. Energy Model

The energy of the deformable mesh quantifies how well the mesh approximates the surface of interest, when the prior shape for the target surface and image data are given. The energy of the surface mesh $(\mathbf{W}, \mathbf{g}_W)$ is

$$\begin{aligned} E(\mathbf{W}, \mathbf{g}_W) &= \lambda E_{int}(\mathbf{W}) + (1-\lambda) E_{ext}(\hat{\mathbf{W}}) \\ &= \frac{1}{N} \sum_{i=1}^{N} [\lambda E_{int}^{i}(\mathbf{w}_i | \mathbf{w}_{i_1}, \mathbf{w}_{i_2}, \mathbf{w}_{i_3}) + (1-\lambda) E_{ext}^{i}(\mathbf{w}_i + \mathbf{g}_W)], \end{aligned} \tag{2}$$

where $E_{int}(\cdot)$ is the internal energy, $E_{ext}(\cdot)$ is the external energy, and $\lambda \in [0,1]$ is the regularization parameter. The internal energy controls the shape of $\mathbf{W}$ when the prior shape is given. The external energy couples the actual surface mesh $\hat{\mathbf{W}}$ with salient image features. The parameter λ controls the trade-off between the model term ($E_{int}(\cdot)$) and the data term ($E_{ext}(\cdot)$).

For extracting the surface mesh from an image, we seek the minimizer of the energy (2) from the set of admissible meshes $\mathcal{W}$. All the meshes in the set $\mathcal{W}$ have the same graph and there may be also other constraints. The set $\mathcal{W}$ consists of pairs $(\mathbf{W}, \mathbf{g}_W)$. This allows us to interpret reference points in three distinct ways, all leading to different optimization algorithms. 1) Reference points can be *fixed*, i.e. all the pairs in $\mathcal{W}$ have the same reference point $\mathbf{g}$. 2) Reference points can be *constrained.* Then pairs $(\mathbf{W}, \mathbf{g}_W)$ are computed from the associated actual meshes $\hat{\mathbf{W}}$ applying Eq. (1) and the definition of the actual mesh from Sect. 3. In this case, as with fixed reference points, there obviously exists a one-to-one correspondence between actual and surface centered meshes. 3) Finally, we may consider reference points as variables independent on the associated actual and surface centered meshes. In this case we say that reference points are *floating.*

4.1. *Representing the prior shape*

Assume that we have the prior shape for the target surface. This is given in the form of the *reference mesh* $\mathbf{V}$. The reference point $\mathbf{g}_V$ associated with $\mathbf{V}$ is constrained to be the center of the mass of $\hat{\mathbf{V}}$. Hence, we can deal only with the surface centered mesh $\mathbf{V}$. From now on, the symbol $\mathbf{V}$ denotes a reference mesh and symbols $\mathbf{W}, \mathbf{g}_W$, and $\hat{\mathbf{W}}$ refer to a deformable mesh.

Each mexel $\mathbf{v}_i$ can be represented as a linear combination of its neighbours, i.e. for all i

$$\mathbf{v}_i = \sum_{j=1}^{3} \alpha_{i_j} \mathbf{v}_{i_j}. \tag{3}$$

Scalars α_{i_j} are called *shape parameters* of $\mathbf{V}$. In a simplex mesh each mexel has exactly three neighbours and the neighbours of any mexel are assumed to be linearly independent. Therefore shape parameters for a given reference mesh are unique and they can be solved from Eqs. (3).

With certain assumptions, the shape parameters of $\mathbf{V}$ offer an affine invariant description of the shape of $\mathbf{V}$ [9]. The shape parameters are therefore used for the definition of the prior shape for the internal energy.

In principle, it is possible to have any shape as the prior. However, a reference mesh $\mathbf{V}$ should optimally represent the prior shape: There has to be a predefined criterion how mexels should be situated on the approximated surface. Meshes satisfying precisely, or even approximately, the optimality criterion are in most cases impossible to construct, unlike when dealing with two-dimensional (2-D) contours.

If the prior shape is simple, it is possible to analytically approximate the shape parameters without constructing a reference mesh. In this case, a reference mesh satisfying exactly Eqs. (3) does not even necessarily exist, but Eqs. (3) provide the means to derive the shape parameters and to analyze the resulting internal energy function.

The simplest form of the derived shape parameters is obtained by requiring that each mexel should be as close as possible to the center of the mass of its neighbours. We call these the thin-plate shape parameters and they are

$$\alpha_{i_j} = \frac{1}{3}, \tag{4}$$

for all i, j. These parameters are comparable with the notion of the surface orientation continuity in [5].

We have also derived the shape parameters for the sphere prior, where we assume that the reference mesh has the shape of the sphere. The shape parameters are

$$\alpha_{i_j} = \frac{1}{3\cos(2\arctan\frac{2\sqrt{\pi\sqrt{3}}}{3\sqrt{N}})}, \tag{5}$$

for all i, j, where N is the number of mexels. The derivation of Eq. (5) is presented in Appendix. These parameters are optimal in the sense that each mexel represents an equilateral triangle of the same size.

4.2. *Internal energy*

The internal energy measures the distance of the shape of the deformable mesh $\mathbf{W}$ to the given reference shape. The mexel-wise internal energy is

$$E^i_{int}(\mathbf{w}_i|\mathbf{w}_{i_1}, \mathbf{w}_{i_2}, \mathbf{w}_{i_3}) = \frac{||\sum_{j=1}^3 \alpha_{i_j}\mathbf{w}_{i_j} - \mathbf{w}_i||^2}{A(\mathbf{W})}, \tag{6}$$

where scalars α_{i_j} are the given shape parameters and $A(\mathbf{W})$ is the area of $\mathbf{W}$. The shape parameters can be inferred from a reference mesh or derived analytically without constructing a reference mesh. The area $A(\mathbf{W})$ is computed as the average area of faces of the mesh $\mathbf{W}$. Areas of the faces of a simplex mesh are computed by triangulating them. The purpose of the normalization by $A(\mathbf{W})$ is to render the internal energy scale invariant.

The internal energy is invariant to rotations and scalings of the mesh. The derivation of these properties is straight-forward and we do not present it here. The translation invariance requires more consideration. The internal energy is translation invariant for surface centered meshes if $\alpha_{i_1} + \alpha_{i_2} + \alpha_{i_3} = 1$ for all i. If this is not the case, reference points must be floating or constrained in order to the internal energy to be translation invariant when interpreted in terms of actual meshes. To establish this fact consider two actual meshes $\hat{\mathbf{W}}^1$ and $\hat{\mathbf{W}}^2$, where $\hat{\mathbf{W}}^2$ is obtained by translating $\hat{\mathbf{W}}^1$ by a vector $\mathbf{d}$. Then, by the definition of the constrained reference point, Eq. (1), the surface centered meshes obtained from $\hat{\mathbf{W}}^1$ and $\hat{\mathbf{W}}^2$ are equal. Hence, they have the same internal energy. On the other hand, with floating reference points, we are interested only in the pair $(\mathbf{W}, \mathbf{g})$ inducing $\hat{\mathbf{W}}^1$ that yields the minimum energy. And since the pair $(\mathbf{W}, \mathbf{g} - \mathbf{d})$ has the same (minimal) energy and induces $\hat{\mathbf{W}}^2$, the internal energy is translation invariant in this case too.

Defining constrained reference points as in Eq. (1) deserves more elaboration. Assume that shape parameters α_{i_j} have the same value, say α, for all i, j. Let $\hat{\mathbf{W}}$ be a fixed actual mesh and consider pairs $(\mathbf{W}, \mathbf{g})$ inducing $\hat{\mathbf{W}}$. Define a function $f(\mathbf{g}) = E_{int}(\mathbf{W}) = E_{int}(\hat{\mathbf{W}} - \mathbf{g})$ of the reference point $\mathbf{g}$, where $\hat{\mathbf{W}} - \mathbf{g} = \{\hat{\mathbf{w}}_i - \mathbf{g} : i = 1, \ldots, N\}$. Note that $A(\mathbf{W})$ does not depend on the choice of $\mathbf{g}$. Then, by differentiating f with respect to $\mathbf{g}$, one obtains that

$$\arg \min_{\mathbf{g} \in \mathbb{R}^3} f(\mathbf{g}) = \frac{1}{N}\sum_{i=1}^N \hat{\mathbf{w}}_i. \tag{7}$$

Furthermore, the minimum is unique if the value of the shape parameters $\alpha \neq \frac{1}{3}$. If $\alpha = \frac{1}{3}$, $f(\mathbf{g})$ is constant.

4.3. *External energy*

The input for the deformable mesh, an image I, is a preprocessed version of the image to be analyzed that is denoted by I^*. In I voxels are given an intensity value based on their saliency inferred from local characteristics of I^*. We also normalize I to have intensity values from 0 to 1 with the voxel of the greatest saliency having the intensity value of 1. Based on the input image I, the mexel-wise external energy is defined simply as

$$E^i_{ext}(\mathbf{w}) = 1 - I(\mathbf{w}). \tag{8}$$

Preprocessing is always an application specific task. For example in our PET application, where we are interested in locating surfaces defined by edges, we take

$$I = ||\nabla MF(I^*)||, \tag{9}$$

where $MF(\cdot)$ denotes a median filter. The gradient is computed by a 3-D Sobel operator [19].

5. Energy Minimization

5.1. *Overview*

For the problem of minimizing the energy (2) globally, we propose an algorithm inspired by the *dual contour method* [8]. The resulting algorithm is called *dual surface minimization* and abbreviated as DSM. We first give an overview of the iterative algorithm and then specialize it for the three different interpretations of reference points.

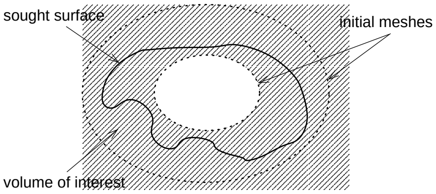


Fig. 2. A cross-section of an initialization for the dual surface minimization algorithm. The term volume of interest refers to the image subvolume where we expect the surface to be extracted to lie.

The initialization for the process consists of two initial surface meshes: the inner mesh $(\mathbf{W}^0_{in}, \mathbf{g}^0_{in})$ and the outer mesh $(\mathbf{W}^0_{out}, \mathbf{g}^0_{out})$. The outer initial surface mesh is located outside and the inner initial surface mesh is located inside of the sought surface, see Fig. 2. The two meshes are generated from a single mesh by scaling. Thus they share the shape, the orientation, and the location in the image.

In each iteration step t, the energies of the inner mesh $(\mathbf{W}^t_{in}, \mathbf{g}^t_{in})$ and the outer mesh $(\mathbf{W}^t_{out}, \mathbf{g}^t_{out})$ are compared and a local energy minimization routine is initiated from the mesh of the higher energy. Two meshes in the iteration $t + 1$ are then the result of the local minimization routine and the mesh of the lower energy at the instant t. In order for the algorithm to converge, it is required that the local minimization process guarantees that $V(\mathbf{W}_{in}^t, \mathbf{g}_{in}^t) \leq V(\mathbf{W}_{in}^{t+1}, \mathbf{g}_{in}^{t+1})$ and $V(\mathbf{W}_{out}^t, \mathbf{g}_{out}^t) \geq V(\mathbf{W}_{out}^{t+1}, \mathbf{g}_{out}^{t+1})$, where $V(\mathbf{W}, \mathbf{g}_W)$ denotes the volume inside the mesh $(\mathbf{W}, \mathbf{g}_W)$. The scale invariance of the internal energy is important for the energy comparison. The alternation of energy comparisons and local optimizations is repeated until $V(\mathbf{W}_{in}^t, \mathbf{g}_{in}^t) > V(\mathbf{W}_{out}^t, \mathbf{g}_{out}^t)$. At this point the mesh of the lower energy is returned as the result.

The local optimization routine can produce its initial mesh ($(\mathbf{W}_{in}^t, \mathbf{g}_{in}^t)$ or $(\mathbf{W}_{out}^t, \mathbf{g}_{out}^t)$) also as its result. If this is the case, the DSM algorithm forces the mesh out of this local energy minimum. It does this either by modifying the energy function or the initial mesh for the local minimization routine. The choice depends on the interpretation of reference points and two alternatives are described in the following two subsections, where also the local minimization routine is detailed.

The algorithm for fixed reference points is described in detail in Sect 5.2. This is later on referred as the standard DSM algorithm, which is meant for processing images that are moderately noisy. The constrained reference point algorithm is a minor modification of the standard DSM and it is also explained in Sect. 5.2. For more drastic conditions, we introduce the floating reference point version of the algorithm. It is slower than the standard DSM algorithm but enables more robust surface recovery when the level of noise is high. This DSM - Floating Reference Point (DSM-FRP) algorithm is described in Sect. 5.3.

5.2. *Algorithm for fixed and constrained reference points*

We first introduce the fixed reference point algorithm, and thereafter explain how it should be modified when the reference points are constrained. The reference point is $\mathbf{g}$ for the inner and outer surface meshes. In each iteration the energy is minimized by a greedy algorithm adapted from [20]. Let the initial surface centered mesh for the greedy algorithm be $\mathbf{W}^t = \{\mathbf{w}_i^t\}$ and denote the resulting surface centered mesh as $\mathbf{W}^{t+1} = \{\mathbf{w}_i^{t+1}\}$. Then, we sequentially update all mexels $\mathbf{w}_1^t, \ldots, \mathbf{w}_N^t$ according to

$$\mathbf{w}_i^{t+1} = \arg \min_{\mathbf{w} \in S(\mathbf{w}_i^t)} E^i(\mathbf{w}, \mathbf{g} | \mathbf{w}_{i_1}^{t*}, \mathbf{w}_{i_2}^{t*}, \mathbf{w}_{i_3}^{t*}). \tag{10}$$

In Eq. (10), $E^i(\cdot|\cdot) = \lambda E_{int}^i(\cdot|\cdot) + (1-\lambda)E_{ext}^i(\cdot)$ is the total energy of the mexel $\mathbf{w}_i$ given its immediate neighbours and t^* is t if $i_j > i$ and $t+1$ otherwise. (This is to say that we use updated mexels as soon as they are available).

For the inner mesh the search space $S(\cdot)$ is defined as

$$\begin{aligned} S(\mathbf{w}_i) &= S_1(\mathbf{w}_i) \cup S_2(\mathbf{w}_i) \\ &= \{(1 + \frac{jL}{||\mathbf{w}_i||})\mathbf{w}_i : j = 0, \ldots, J\} \\ &\cup \{(1 + \frac{jL}{||\mathbf{w}_i||})[(1-2kD)\mathbf{w}_i + kD(\mathbf{w}_{i_a} + \mathbf{w}_{i_b})] \\ &\quad : (j = 1, \ldots, J), (k = 1, \ldots, K), (a, b = 1, 2, 3), (a \neq b)\}, \end{aligned} \tag{11}$$

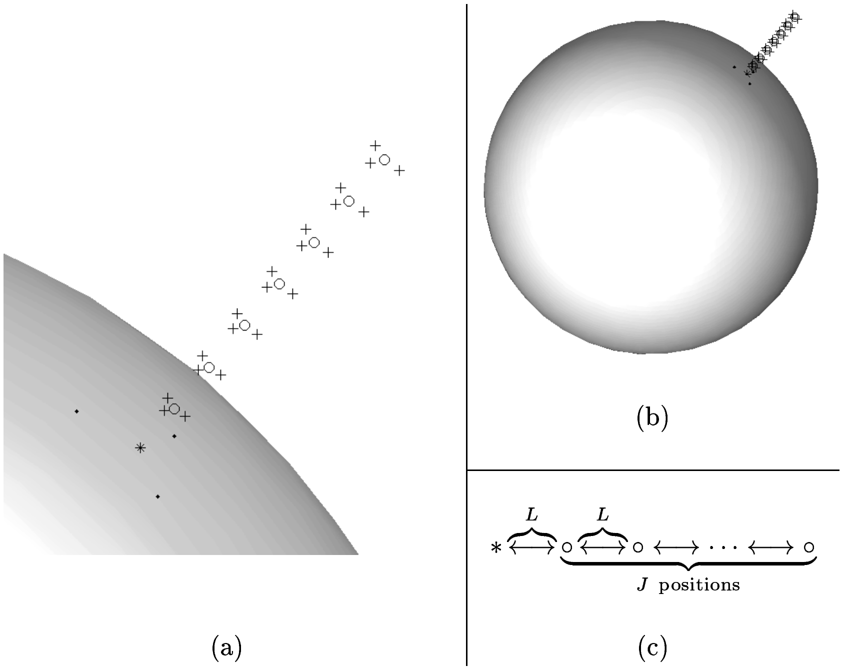


Fig. 3. Search space of the inner mesh for the mexel marked by a star ($*$). Positions marked by a circle ($\circ$) and the mexel position itself belong to $S_1(\cdot)$. Positions marked by a plus sign ($+$) belong to $S_2(\cdot)$. Neighboring mexels are marked by solid dots. Required parameters for the construction of the search space were $L = 0.5, J = 8, D = 0.15, K = 1$. In the panel (a) a close up from a global view of the panel (b) is shown. In the panel (c), a schematic representation of $S_1(\cdot)$ is shown demonstrating the meaning of constants J and L in Eq. (11).

where L, J, K, D are user definable constants. The search space is divided into two parts, $S_1(\cdot)$ and $S_2(\cdot)$, because it is occasionally useful to modify the search space (automatically) during DSM. As can be seen from Eq. (11) and the graphical depiction of $S(\cdot)$ in Fig. 3, the search is directed away from the reference point. The search space for the outer mesh is obtained by setting $j = 0, -1, \ldots, -J$ in Eq. (11). In this case the search is directed towards the reference point $\mathbf{g}$.

We have used experimentally found values $L = 0.5, J = 8, D = 0.15, K = 1$ for search spaces for meshes with 1280 mexels. If the resolution of the mesh is increased, the search space should also be more dense in order to capitalize the increased resolution. This is achieved by adapting L to have a lower value than 0.5. The value of the parameter K should be at least 1, cf. Sect 5.4. Otherwise, slight changes in values of these parameters do not seem to affect the performance of the DSM algorithm.

If the DSM algorithm is stuck in a local minimum, the energy function is modified by adding a penalty term for the current $\mathbf{W}^t$. Instead of minimizing the energy

$E^i(\cdot|\cdot)$ as in Eq. (10), we minimize

$$E^i_{modified}(\mathbf{w}, \mathbf{g}|\cdot) = E^i(\mathbf{w}, \mathbf{g}|\cdot) + r\gamma\delta(\mathbf{w} - \mathbf{w}^t), \tag{12}$$

where $\delta(\cdot)$ is the Discrete Delta Function[a]. The integer $r = 0, 1, 2\ldots$ is gradually incremented until the result of the greedy algorithm differs from its starting mesh. The parameter γ determines the amount of the penalty added at a time. When driving the meshes out of the local minima, only the search space $S_1(\cdot)$ is considered, see Eq. (11), in order to prevent oscillations. The term oscillation refers to undesired phenomenon where mexels vary only between few repeating positions. This can prevent the algorithm from converging.

For the constrained reference point version, necessary for the translation invariance of the internal energy with the sphere prior, the reference point and the surface centered mesh have to be constantly modified according to the current actual mesh. After each update according to Eq. (10), a new reference point and a new surface centered mesh are computed. This can be done efficiently since at any time we only have to know surface centered positions of one mexel and its immediate neighbours. Hence, we can update only the necessary mexels in the surface centered mesh according to the changed reference point.

5.3. *Floating reference point algorithm DSM-FRP*

We modify the standard DSM algorithm by enabling the reference point to float, i.e. considering more than one alternative for the reference point at a time. In each iteration t of the DSM algorithm several greedy optimization algorithms are initiated, all with the same initial surface centered mesh but different reference points. The surface centered mesh and the reference point at the instant $t + 1$ are then the result of the greedy algorithm yielding the lowest energy. If the current reference point is $\mathbf{g}^t_W$, then at each iteration we consider reference points from the set

$$S_g(\mathbf{g}^t_W) = \{\mathbf{x} : ||\mathbf{x} - \mathbf{g}^t_W||_\infty \in \{0, 1\}\}. \tag{13}$$

If the DSM-FRP algorithm is locked in a local energy minimum, the surface mesh of the higher energy is scaled in order to push the procedure out of the local energy minimum. The inner mesh is expanded and the outer mesh is contracted. This is implemented by multiplying the mexels of the surface centered mesh by an appropriate constant. This forced movement method is applicable here, because with the floating reference point the meshes are able to backtrack if the scaling has forced them over the correct target surface. Moreover, the forced movement leads to better results with the floating reference point than the penalty term method that is favored with the standard and constrained reference point algorithms.

[a] If $\mathbf{x} = \mathbf{0}$, $\delta(\mathbf{x}) = 1$ and if $\mathbf{x} \neq 0$, $\delta(\mathbf{x}) = 0$.

5.4. *Modifications, extensions, and implementation*

The standard DSM algorithm can be modified in a few ways to make it more appropriate for a particular application. For example, we may consider only one surface mesh at the time instead of considering two of them as in the standard algorithm. When extracting the brain surface from positron emission tomography (PET) images (see Sect. 7), it is preferable to approach the surface of interest from outside. This is because outside the brain volume there is no radioactivity and therefore the noise level outside the brain volume is lower than inside the brain volume. Consequently, it becomes appropriate to use only one surface mesh approaching the target surface from outside. This modification is straight-forward to implement, since the process only has to remember the mesh of the lowest energy that has been already found and compare results of the greedy algorithm against it instead of the current inner mesh. The algorithm is terminated when the volume of the (outer) mesh becomes smaller than the given threshold. This modification, abbreviated DSM-OS (DSM-outer surface), improved extraction results in cases where the standard DSM algorithm had difficulties, see Fig. 4. These difficulties were due to the existence of good but local minima near the surface of interest, which the mesh approaching from outside found before the correct surface of interest. On the other hand, with the inner mesh it was hard to find as good minima as with the outer mesh and the iterative process remained optimizing the inner mesh.

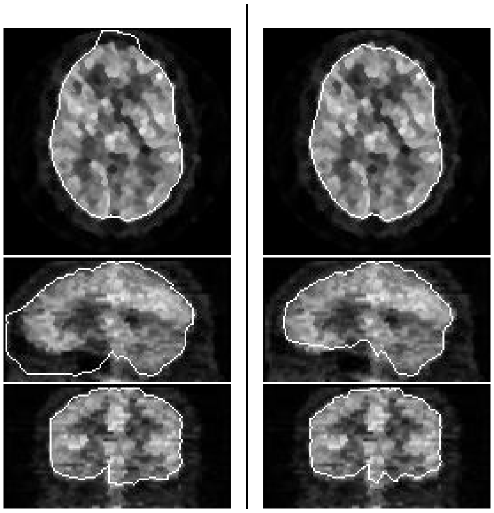


Fig. 4. Failed extraction of the brain surface from a PET image using the standard DSM algorithm (left) and a successful result with the single surface strategy (right). Both extraction results were achieved by using the same parameters values. cross-sections of the extracted surfaces are shown on the PET image. From the top, transaxial, sagittal, and coronal views.

However, the DSM-OS modification should only be used for special problems such as the one described above. The standard algorithm is a better choice if there is no clear indication that it is preferable to approach the surface of interest from outside. This seems reasonable and we noticed this during the experimentation.

The search space as defined in Eq. (11) does not absolutely guarantee the outer meshes to contract and the inner ones to expand. Hence, it may happen that meshes are caught in oscillating between few positions. This has never happened with the floating reference point version of the algorithm. However, undesired oscillations are easily detected by examining the volumes of the meshes. Meshes may then be forced out of the oscillating state by setting the search space for the greedy algorithm to $S_1(\cdot)$ as in (11) until the volume of the mesh changes. Applying the search space $S_1(\cdot)$, i.e. setting $K = 0$ in Eq. (11), all the time during the DSM algorithm could result in difficulties, for example when the (initial) reference point is close to the surface of interest.

6. Experiments

We have performed experiments with synthetic images in order to obtain quantitative results concerning relative advantages of different versions of our deformable model. The set of synthetic images was also used for comparing the standard DSM algorithm to a few other methods to control the evolution of deformable meshes.

6.1. *Synthetic images*

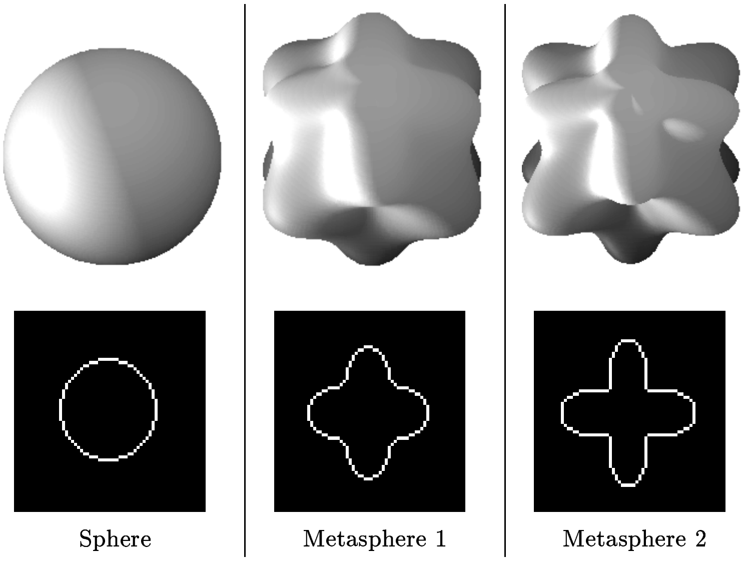
The set of the applied synthetic images can be divided into two subsets: The images in the first set contained a closed surface of intensity values of one. Surfaces were metaspheres [21], see Fig. 5. These were drawn to $64 \times 64 \times 64$ grid and then white Gaussian noise was added to the images. Variances of noise were 0.2, 0.6 and 1.0. Thereafter, images were filtered with a Gaussian filter. The filter had $3 \times 3 \times 3$ kernel with the standard deviation equal to 1.0. The input images for the deformable mesh were obtained by scaling the intensities of each image to the range from zero to one.

The images in the second set consisted a true surface of intensity of one and few false surfaces of the intensity of 0.5. Examples of cross-sections of images are shown in Fig. 6. Dimensions for these images were also $64 \times 64 \times 64$.

6.2. *Implementation of the experiments*

Test results were evaluated with the following measure. Let the set of voxels inside the true (digital) surface be TV and the set of voxels inside the extracted surface be EV. Then the measure is

$$\epsilon = 1 - \frac{|TV \cap EV|}{|TV \cup EV|}. \tag{14}$$



Sphere Metasphere 1 Metasphere 2

Fig. 5. Surfaces in the first image set are shown on top and the central cross-sections of these surfaces in xy-plane are shown below.

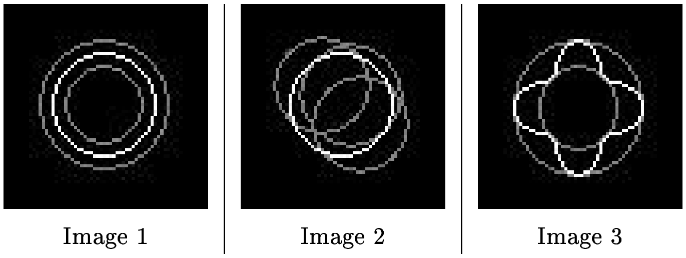


Image 1 Image 2 Image 3

Fig. 6. The central cross-sections in xy-plane of the images of the second set. The cross-sections of the true surfaces are depicted in white (intensity 1.0) and the cross-sections of the false ones are depicted in gray (intensity 0.5).

To depict the measure assume that the true surface is a sphere of the radius 16. The extracted surface is identical expect that it is displaced one unit in each direction from the true surface. Then $\epsilon = 0.16$. (The quantity $1 - \epsilon$ is known as the Tanimoto coefficient.)

Meshes in the experiments consisted of 1280 mexels unless otherwise mentioned. The standard DSM algorithm was tested also with meshes of 5120 mexels. Five different dual initializations were used. The listed error value (in Tables 1 and 2) for each experiment is the median of the error values resulting from these five initializations. The initial surface meshes had the shape of the sphere. Examples of the initializations are shown in Fig. 7.

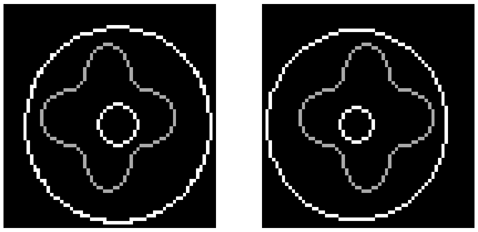


Fig. 7. Two of the five dual initializations used in the experiments. The central cross-sections of the initial surface meshes are shown in white. For the reference, also the central cross-section of the Metasphere 1 target surface is shown in gray.

We used the thin-plate shape parameters Eq. (4) for the internal energy unless otherwise mentioned. The value of the regularization parameter λ was 0.3, which seems to be the best choice on the average. The parameters for search spaces were as suggested in Sect 5.2 when meshes with 1280 mexels were considered. For finer meshes with 5120 mexels, the value of the parameter L was lower, 0.25, to obtain the gain from the increased mesh resolution as outlined in Sect. 5.2. To drive the standard and the constrained reference point algorithms out of local minima, the parameter γ as in Eq. (12) was 0.05 with 1280 mexels and 0.03 with 5120 mexels. These selections provided fast convergence combined with good extraction results. The lower value for finer meshes was again to obtain the advantage from the finer mesh resolution. The quality of extracted surfaces may be improved by diminishing the value of γ, at the expense of increased computation time. This is demonstrated in Fig. 8, where error values Eq. (14) and required number of iterations are plotted against the value of γ for two of the test images.

6.3. *Results with the variants of DSM and the two prior shapes*

Quantitative test results of the variants of the DSM algorithm with the two prior shapes are provided in Table 1. The standard algorithm with the sphere prior shape Eq. (5) was not included in the experiments because in this case the internal energy would not be translation invariant. As can be seen from the test results in Table 1, the DSM-FRP algorithm was more robust with higher noise levels. This is further demonstrated in Fig. 9, where spheres extracted from the images with different noise levels using the standard DSM and DSM-FRP are shown. The shown surfaces are those which yielded the median value of the error measure. When the surface to be extracted had a complex shape (Metasphere 2) DSM-FRP was found to be less accurate than the standard algorithm. With the image set 2 (Fig. 6), good results

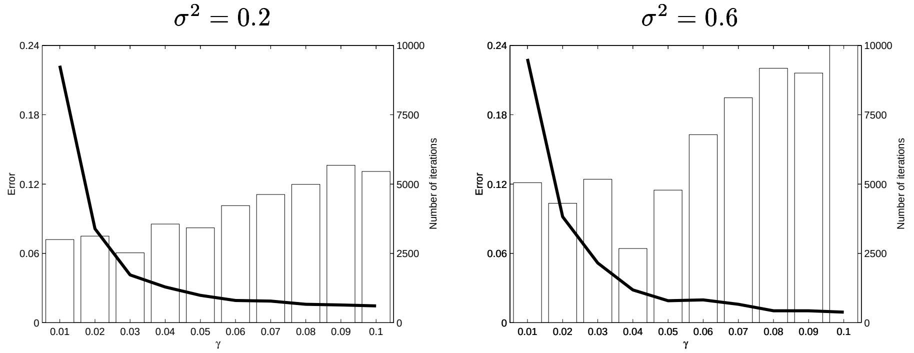


Fig. 8. Influence of value of the parameter γ, cf. Eq. (12), to the required number of iterations (black lines) and to the error value (bars). The surface to be extracted was Metasphere 1 and noise variances σ^2 were 0.2 and 0.6. Meshes with 1280 mexels were applied. All values are medians resulting from the five initializations.

were obtained with every variant of the DSM algorithm. There were only minor differences in the results of the standard algorithm and the constrained reference point variant.

The usage of the sphere shape parameters improved the results with the constrained and the floating reference point algorithms. This is probably because the sphere prior shape offers locally better expected position for a particular mexel. In a more global sense the sphere prior shape is not necessarily better prior model for the shape of the surface of interest than the thin-plate prior shape. However, usually evolution procedures for deformable meshes consider only one mexel at the time in a greedy manner. Our method makes no exception and in this respect better expected positioning of a single mexel may improve the quality of the results.

Using 5120 mexels as opposed to 1280 mexels resulted in bit lower error values when the variance of noise was 0 or 0.2 in the images of the first set. However, with higher noise levels no such gain was achieved by using finer meshes. With the second image set, increment in the applied number of mexels resulted in slightly higher error values. These results are quite natural, since 1280 mexels are sufficient for representing the test surfaces properly. On the average, the performance of DSM was similar with both numbers of mexels. This suggests that DSM can be used with meshes of different resolutions. Only a slight and justifiable change of some parameter values could be required.

When considering meshes with a certain number of mexels, we performed all the experiments with the same parameter values. Also, the images in the first and the second image sets were quite divergent. This implies that it is possible to process dissimilar images without adapting the parameter values. This reduced sensitivity to parameter values is obviously an important property for automatic surface extraction. Moreover, the DSM algorithm was rather insensitive to its initialization,

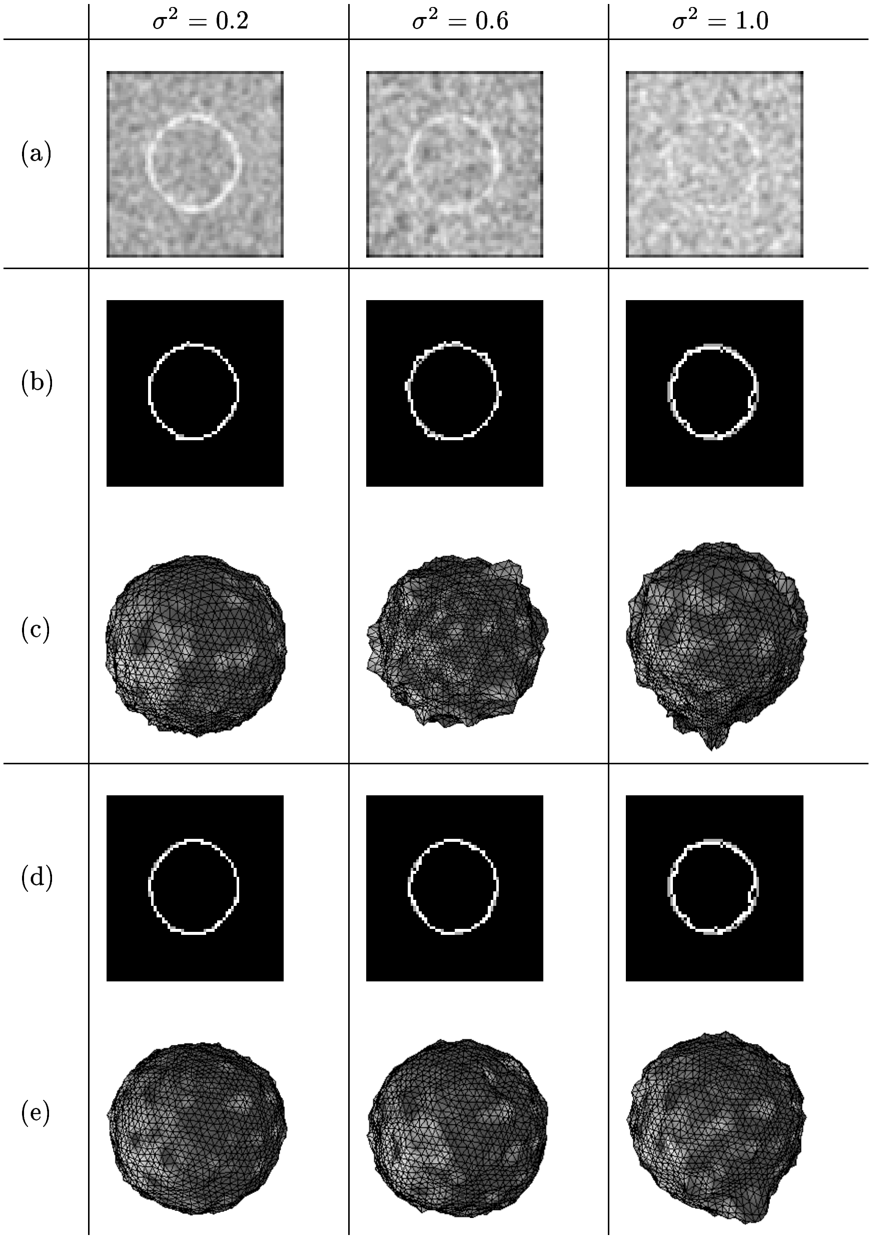


Fig. 9. The effect of the increased noise level on the standard DSM and DSM-FRP algorithms. The noise variance is denoted by σ^2. Row (a): The central cross sections of the input images are shown. Row (b): The cross-sections (in white) of the extracted surfaces with the standard DSM against the cross-sections of the true surfaces (in gray). Row (c): The surface meshes extracted using the standard DSM. Rows (d) and (e): The extraction results using DSM-FRP. The thin-plate shape parameters were applied.

Table 1. Values of the error measure of the variants of the DSM algorithm and the two prior models from the experiments with the synthetic images. Error values listed are medians of the error values resulting from the five initializations. Standard refers to the standard DSM algorithm with the thin-plate prior shape. CRP-TP and CRP-S, respectively, refer to the constrained reference point algorithm with thin-plate and sphere prior shapes. FRP-TP and FRP-S, respectively, refer to the floating reference point algorithm with the thin-plate and the sphere prior shapes. The column labeled 5120 shows results of the standard DSM algorithm with meshes of 5120 mexels. All the other meshes consisted of 1280 mexels.

The first image set

surface	σ^2	DSM version					
		Standard	CRP-TP	CRP-S	FRP-TP	FRP-S	5120
sphere	0	0.03	0.03	0.04	0.04	0.04	0.02
	0.2	0.07	0.09	0.06	0.06	0.05	0.06
	0.6	0.11	0.10	0.11	0.06	0.05	0.12
	1.0	0.18	0.19	0.17	0.08	0.08	0.17
Metasphere 1	0	0.03	0.03	0.02	0.03	0.02	0.01
	0.2	0.08	0.10	0.02	0.09	0.04	0.02
	0.6	0.12	0.11	0.06	0.12	0.07	0.06
	1.0	0.15	0.13	0.12	0.13	0.11	0.15
Metasphere 2	0	0.07	0.08	0.08	0.07	0.08	0.06
	0.2	0.13	0.14	0.11	0.16	0.16	0.12
	0.6	0.18	0.21	0.18	0.25	0.22	0.20
	1.0	0.27	0.30	0.26	0.28	0.25	0.31

The second image set

Image	DSM version					
	Standard	CRP-TP	CRP-S	FRP-TP	FRP-S	5120
Image 1	0.03	0.03	0.03	0.03	0.03	0.09
Image 2	0.05	0.05	0.05	0.09	0.06	0.07
Image 3	0.02	0.07	0.06	0.12	0.12	0.10

although this does not mean that all the initializations led to exactly the same result. However, in most cases differences in extracted surfaces were small and are better seen from the quantitative results than by visually inspecting the extracted surface meshes. In Fig. 10 the best and the worst extracted Metasphere 1 resulting from the different initializations are shown. The extracted surfaces were ranked based on their error value ϵ as in Eq. (14).

We could rather reliably rank the quality of the extracted surfaces resulting from different initializations based on their energies. This observation serves as a partial justification for the global minimization framework. However, it is worth to note that we could not rank so easily surfaces extracted with the constrained reference point algorithm and the sphere prior shape. This was probably since the internal energy with the sphere shape parameters is not translation invariant for the surface centered meshes.

In the experiments the standard DSM-algorithm typically took approximatively

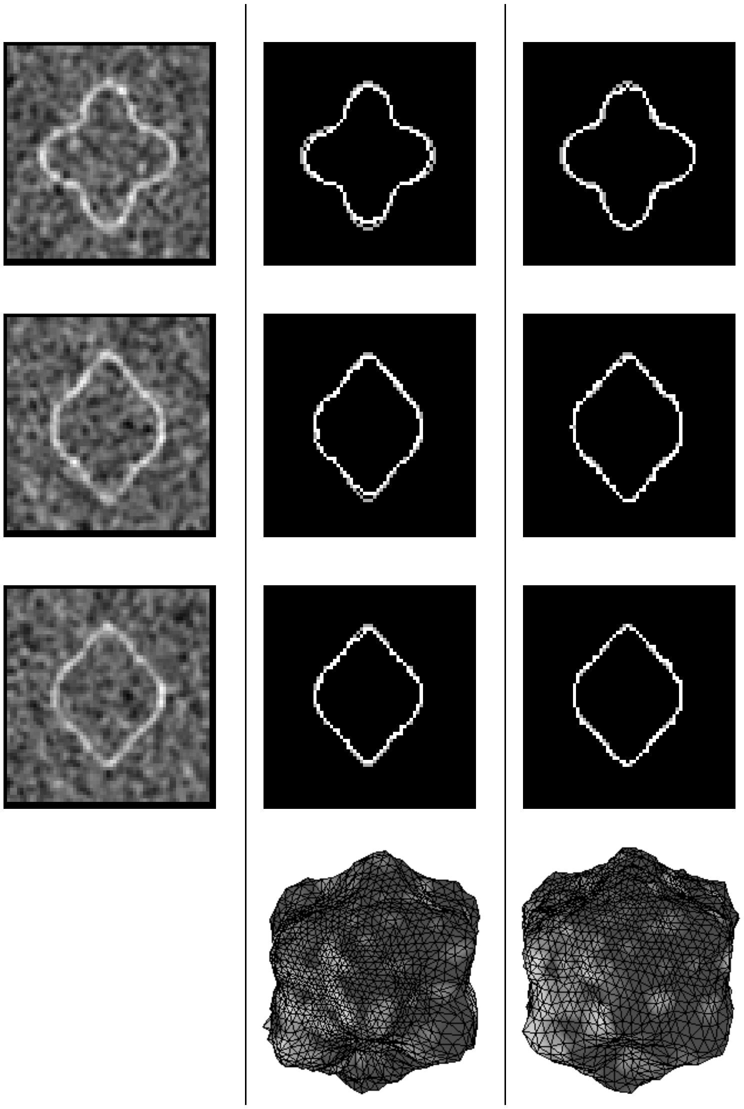


Fig. 10. The best ($\epsilon = 0.01$) and the worst ($\epsilon = 0.10$) extracted Metasphere 1 from the image with the noise variance of 0.2 resulting from different initializations. Surface extraction was performed using the standard DSM algorithm and the thin-plate prior shape. Left: The input image for the deformable model. Center: The worst extraction result. Right: The best extraction result. In the three topmost rows are the cross-sections in xy, xz, and yz-planes. The true surface is depicted in gray and the extracted surfaces are in white. The extracted surface meshes are shown in the last row.

1000 iterations of the greedy algorithm to converge when the number of mexels was 1280. With meshes of 5120 mexels, the DSM algorithm took on the average 4000 iterations to converge. Since the computational complexity of the greedy algorithm is linear as compared to the number of mexels, the standard DSM algorithm is quite efficient. The DSM-FRP variant took usually under 500 iterations to converge. However, each iteration of the DSM-FRP is about 27 times slower than an iteration of the standard DSM algorithm as can be deducted from Eq. (13). So the standard algorithm is on the average over ten times faster than the DSM-FRP variant.

6.4. *Comparison of DSM and other methods*

A comparative study between recent methods for controlling the mesh evolution where the standard DSM algorithm was included has been presented in [10]. Here we present only a brief summary of results and conclusions of that study, for more details the reader is referred to the article. The results are not easily interpretable due to the difficulty of selecting the best set of parameters for each method and for each type of experiments. Therefore, the purpose of the comparative study was to find out relative strengths and weaknesses of each method, not to rank them.

Three force-based methods for controlling the mesh evolution were included. These were gradient vector flow (GVF) [22], generalized gradient vector flow (GGVF) [23] and method proposed by Delingette (Delingette) [5]. The methods based on the minimization of the energy included in the study were hybrid of RCGA and the greedy algorithm (GAGR) [17] and the standard version of the DSM algorithm. The energy function to be minimized was as in Eq. (2) with the thin-plate shape parameters for the internal energy. The implementation of the experiments was as in this paper.

We remark that in [10] areas of the meshes were used to compute the termination condition for the DSM algorithm. Hence, those results differ a little from the ones listed in Table 6.4, because here we use the volumes inside the meshes to compute the termination condition. The oscillation control (see Sect. 5.4) is a new feature of the DSM algorithm that was not implemented in the version tested in the comparative study. The oscillation control improved considerably the results of DSM with the second image set.

The quantitative test results are listed in Table 2. As can be seen from these, the results achieved with DSM were among the best when the variance of noise in the images of the first set was 0, 0.2 or 0.6. (An exception was the image containing the sphere degraded with noise of the variance 0.6.) The standard DSM algorithm did not perform as well compared to the others, when the variance of the Gaussian noise was 1.0. However, DSM-FRP produced error measures similar to the best ones when the variance of noise was 1.0. This can be seen by comparing the error values listed in Tables 1 and 2. For example, when the variance of noise was 1.0 and surface to be extracted was sphere, standard DSM, DSM-FRP and GGVF (the best method in this case) yielded, respectively, error values of 0.18, 0.08 and 0.06.

The standard DSM algorithm performed well with all the images from the second set. This was not the case in [10], where we could not extract adequate surfaces from

Table 2. Values of the error measure resulting from different methods to control the mesh evolution. Abbreviations of the methods are given in the text. Remark that the error values of DSM are obtained by using the implementation described in this paper and they may differ from the ones listed in the comparative study article.

The first image set

surface	σ^2	Method				
		GVF	GGVF	Delingette	DSM	GAGR
sphere	0	0.03	0.03	0.08	0.03	0.05
	0.2	0.09	0.05	0.07	0.07	0.06
	0.6	0.09	0.06	0.08	0.11	0.06
	1	0.13	0.06	0.08	0.18	0.07
Metasphere 1	0	0.03	0.02	0.09	0.03	0.06
	0.2	0.11	0.05	0.09	0.08	0.07
	0.6	0.16	0.10	0.11	0.12	0.08
	1.0	0.20	0.14	0.12	0.15	0.11
Metasphere 2	0	0.07	0.06	1	0.07	0.10
	0.2	0.30	0.13	0.19	0.13	0.16
	0.6	0.27	0.19	0.21	0.18	0.20
	1.0	0.27	0.24	0.26	0.27	0.22

The second image set

Image	Method				
	GVF	GGVF	Delingette	DSM	GAGR
Image 1	0.16	0.50	0.07	0.03	0.05
Image 2	0.06	0.51	0.07	0.05	0.05
Image 3	0.07	0.42	0.09	0.02	0.10

Image 2 due to oscillations.

An interesting issue to note is that with the GGVF method extracted surfaces from the images of the second set were of poor quality. On the contrary, GGVF method worked very well with the first set of images. This observation implies that the nature of input data can have a major effect on how well a particular method to control the mesh evolution functions.

7. Application to Analysis of PET Brain Images

Positron emission tomography (PET) is a unique method for the investigation of physiological processes in brain quantitatively cf. e.g. [24]. Automated and repeatable extraction of biologically meaningful structures directly from 3-D PET images could be beneficial for drug development among other tasks [25]. High noise levels in images as compared to their contrast and non-uniformity of intensities within the structures of interest make the segmentation problem hard. For more insights about the level of noise in PET images cf. [26]. We have studied the delineation of the cortical structure from FDG (fluorodeoxyglucose) PET images, i.e. functional cortex, using the deformable model introduced in this paper [11]. Note that difficulties

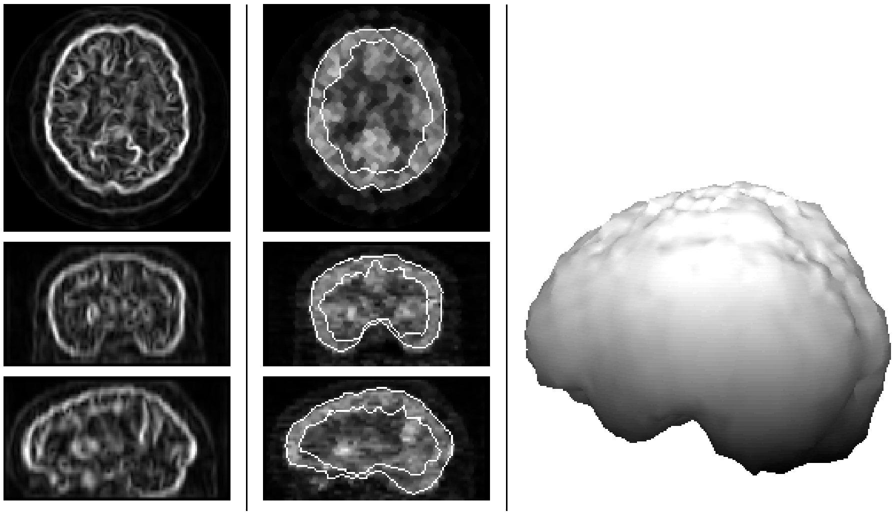


Fig. 11. The extracted cortical structure from a FDG-PET image. The left column: The result from the edge operator (see Eq. (9)) that was used as the input image for the deformable model, from the top transaxial, coronal, and sagittal views. The central column: The extracted cortical structure, from the top transaxial, coronal, and sagittal views. The right column: The extracted brain surface from the image.

in this task are rather different than those in cortex extraction from (anatomical) magnetic resonance (MR) images [12,27], where highly convoluted shape of the cortex is problematic to capture. Fine details of the shape cannot be seen in PET images due to their lower resolution compared to MR images. Also, the contrast to noise ratio within PET is typically lower than it is within MR.

The structure extraction process was divided into two steps. First the brain surface, which is also the outer surface of the functional cortex, was extracted. Here, we used DSM-OS modification for energy minimization for reasons that were outlined in Sect. 5.4. Thereafter, we extracted the surface between white and gray matter, which is treated as the inner surface of the functional cortex. For this, we applied the standard DSM-algorithm. We emphasize that the process is designed to be fully automatic, particularly initializations for the deformable model were obtained without any guidance from the user. For more details about the segmentation procedure, we refer to [11].

The functional cortex was delineated from 17 FDG-PET images and results were in most cases satisfactory in visual inspection. An example of typical result can be seen in Fig. 11 and more results are shown in [11]. Especially, brain surfaces were found accurately and reliably from all 17 images. Also white/gray matter surfaces were found reliably from all the images. However, as one might expect (cf. the edge image in Fig. 11), surface details were not captured as well as with the case of brain surface.

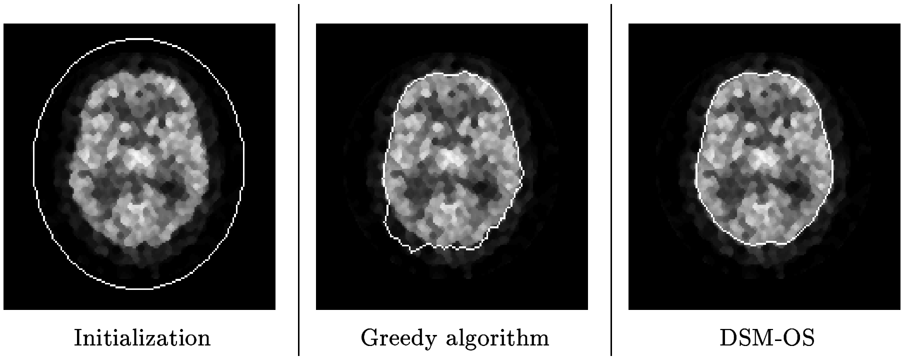


Fig. 12. Transaxial cross-sections of the initial mesh and brain surfaces extracted with the greedy algorithm and the DSM-OS algorithm.

The main merit of our deformable mesh for the task lies in its reduced sensitivity to its parameters and its initializations that enables us to process a large set of images in a completely automated manner. To demonstrate the initialization problem, we tried to extract the brain surface from a PET image by optimizing the deformable mesh with a simple greedy algorithm. To avoid weak local energy minima, a shrinking effect similar as in snakes [2] was generated. This was achieved by considering the area of the mesh $A(\mathbf{W})$ in the internal energy as a constant calculated from the initial mesh. The greedy algorithm was as the one described in Sect. 5.2, but it was iterated until it converged. As shown in Fig. 12, the extraction result using DSM-OS is better than the one using the greedy algorithm. Particularly, the greedy algorithm was not able climb up from a local energy minimum caused by slight tracer uptake outside the brain volume. Initializations for both methods were same as were the parameter values.

The PET acquisitions were made with GE Advance Scanner (GE, Milwaukee, US) and images were reconstructed by the MRP-algorithm [28]. The image dimensions were $128 \times 128 \times 35$ with the voxel dimensions of $1.72mm \times 1.72mm \times 4.25mm$.

8. Discussion

In this section, we discuss similarities and differences between our deformable model and other deformable models. Special attention is given to the explanation of differences and their reasons between our model and those models [8,9] which we used as the sources of ideas.

Our energy definition utilized ideas proposed in [9] for the shape modeling for active contours. While the basic idea of shape parameters as in Eq. (3) is a rather straight-forward generalization from [9], we used it to analytically derive global models for simple 3-D shapes rather than to model the shape of the given surface mesh. In [9], the central idea was to build the reference shape (i.e. derive shape parameters) based on given example contours. In principle, this is also possible with surface meshes and has already been considered for example in [13,5]. Compared to the contour shape modeling, the problem in building a shape model for a deformable surface based on example meshes is that the uniformity of the elements of the example surface meshes is hard to control. This issue was not explicitly addressed in articles [13,5]. The term *uniformity of the elements* is more easily clarified by using a triangular mesh as an example. The elements of the triangular mesh are the triangles that it is composed of and a perfectly uniform triangular mesh refers to a mesh where all triangles are equilateral and of the same size. Also other surface representations besides surface meshes are amenable for building models of 3-D shapes. For instance, *m-reps* [29] are based on a medial description of surfaces.

The sphere prior shape, Eq. (5), derived in this paper is not a specialized shape model for surfaces encountered in medical imaging applications. It is meant to serve as an example on how to analytically derive the shape parameters. Also, as discussed in Sect. 6.3, the application of the sphere prior shape improved experimental results as compared to the thin-plate prior shape. Combination of shape information analytical smoothness models, like the sphere prior, and exemplars as in [30,31] could be interesting topic for further research.

Compared to [9], several possible interpretations of reference points is a new idea. The distinct interpretations of reference points led to the distinct versions of the dual surface minimization algorithm applicable in differing situations. In addition, we proposed the outer surface modification (DSM-OS) of the DSM algorithm useful in the brain surface extraction from PET images. The variants of the algorithm would have been hard to derive with a direct generalization of the dual contour method [8] to the optimization of deformable surface meshes. In the direct generalization, each mexel in the mesh of the greater energy would be pushed towards the corresponding mexel in the other surface mesh instead of pushing it towards (or away from) the reference point. This would necessitate the use of the two surfaces and make the DSM-OS modification impossible and complicate the DSM-FRP modification. In the dual contour method the contour of the greater energy was updated based on the evolution equation similar to ones used with force-based deformable meshes during each iteration. Instead with DSM, we applied greedy local search for updating the meshes, mainly to make the DSM-FRP modification easier. The contours in the dual contour algorithm were not tightly constrained to move only to certain directions. This might result to problems in selecting proper parameter values for the driving force pushing the contour of the higher energy towards the contour of the lower energy.

The floating reference point version of the DSM algorithm, DSM-FRP, utilizes global translations of the mesh combined with local free-form deformations. The application of global deformations in the combined with local ones has been studied also by other researchers. For example, in [32] superquadrics with six degrees of freedom were combined with locally deforming splines and in [33] simplex meshes under the influence of global as well as local forces were considered. Our approach of combining local and global transformations of the mesh differs from the abovementioned approaches in the respect that we consider global and local transformations simultaneously and then select the best combination. In [32,33] they considered global and local transformations in an alternating manner but not simultaneously.

A large body of work has been devoted to reduce the initialization sensitivity of deformable surface models. Combination of local and global deformations and advanced construction of external forces [22,23] are instances of techniques that have been used to address this problem. Model libraries have recently been introduced [34] for improving reliability of initialization techniques based on rigid registration. Also, gradient flows usually associated with level sets [6] have been combined with the mesh representation of the surface [35]. This could result in an attractive framework for automatic extraction of surfaces of constrained topology. However, the motivation in [35] was implementational efficiency, not topological constraints.

Often techniques that are applied to reduce the initialization sensitivity come with an extra set of parameters and it might not be obvious how to set parameter values within a particular application. This easily leads to the problem of high parameter sensitivity. Eventually, it may require a considerable effort to a find suitable parameter values for a specific application especially if a large set of images are to be processed. This problem which often restricts the automation of deformable models is less addressed in the literature than the initialization problem. We have demonstrated that our deformable mesh can process dissimilar images with a same set of parameter values. This property is obviously desirable for automatic surface extraction. When finer meshes of 5120 mexels were considered, a change in parameter values was found to be appropriate and also the reasons behind the change were explained.

9. Conclusion

We have presented a novel deformable surface mesh for extracting surfaces from noisy volumetric images. The main idea was to treat surface extraction as a global optimization problem. For the global optimization task, we introduced several variants of the dual surface minimization algorithm. The variants were found to be applicable for differing surface extraction tasks and therefore they complement each other. As our experiments revealed, the algorithms effectively avoided getting caught in weak local energy minima. For this reason, our deformable model was rather insensitive to its initialization. Also, we were able to extract surfaces from all the test images without changing the parameter values for the deformable model. These properties of the deformable model are highly desirable for automated surface extraction.

The internal energy for the deformable surface model was derived utilizing ideas from [9]. The internal energy is translation, rotation and scale invariant. By definition it included the global position parameter, that allowed us to derive the floating reference point variant of the optimization algorithm. The variant was advantageous when the noise level in images was high. We also derived analytically the sphere prior shape for the internal energy.

We have applied the new deformable model with success for a practical task of extracting brain structures from positron emission tomography brain images [11]. Other applications of the deformable model within PET images are planned. The deformable model could provide new possibilities for automating surface extraction from also other volumetric images.

Acknowledgments

The authors wish to thank Turku PET Centre for providing the PET images and Ph.D. Ulla Ruotsalainen for discussions and comments about this paper. For this study, Mr. J. Tohka received financial support from *Tampere Graduate School in Information Science and Engineering* (TISE) the *Finnish Cultural Foundation*, the *Science Foundation of Instrumentarium* and *the Academy of Finland.* Mr. J.M. Mykkänen received financial support from TISE and the *Oskar Öflund Foundation.*

Appendix

The derivation of Eq. (5) is presented.

Assume that the unit sphere is triangulated by N equilateral triangles of the same size. Then the (reference) simplex mesh $\mathbf{V}$ is generated by choosing the center point of each triangle as a mexel. The aim is to approximate shape parameters α_{i_j} based on local properties of $\mathbf{V}$ such that

$$\mathbf{v}_i = \sum_{j=1}^{3} \alpha_{i_j} \mathbf{v}_{i_j}, \tag{15}$$

for all $i = 1, \ldots, N$.

All parameters α_{i_j} have the same value, say α, due to the symmetry of the problem. Also, it is assumed that all mexels are placed on the unit sphere. That is, $||\mathbf{v}_i|| = 1$ for all i. In this case the reference point $\mathbf{g}_V = \mathbf{0}$. Let us study an arbitrary mexel $\mathbf{v}_k$. Now

$$1 = ||\mathbf{v}_k||^2 = \langle \mathbf{v}_k | \mathbf{v}_k \rangle = \langle \mathbf{v}_k | \sum_{j=1}^{3} \alpha \mathbf{v}_{k_j} \rangle = 3\alpha \langle \mathbf{v}_k | \mathbf{v}_{k_1} \rangle, \tag{16}$$

where the last equality follows from the symmetry of the problem. Denote $\mathbf{v}_{k_1}$ by $\mathbf{v}_l$. Now we need to calculate the inner product $\langle \mathbf{v}_k | \mathbf{v}_l \rangle$ to obtain α. This can be done by calculating the angle θ between vectors $\mathbf{v}_k$ and $\mathbf{v}_l$. To do this we must consider the original triangular mesh. Remember that mexels are the centers of the original triangles.

The sum of the areas of triangles can be approximated by the area of the sphere. The area of a single triangle is $A_{triangle} = \frac{4\pi}{N}$. Since all triangles are equilateral,

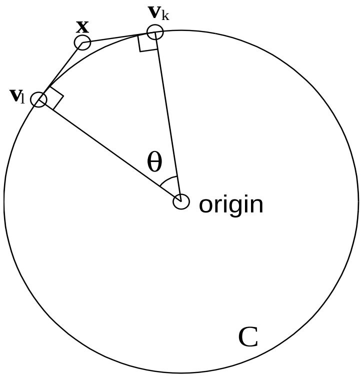


Fig. 13. The circle C.

their height

$$h = \sqrt{A_{triangle}\sqrt{3}} = \sqrt{\frac{4\pi\sqrt{3}}{N}}. \tag{17}$$

We assume that triangles lie on the tangent planes of the unit sphere. Let us form a circle C (in $\mathbb{R}^3$) with its center at the origin and $\mathbf{v}_k$ and $\mathbf{v}_l$ on the circle. Let us denote by $\mathbf{x}$ the point of intersection of the two tangents of C drawn at $\mathbf{v}_k$ and $\mathbf{v}_l$. See Fig. 13. Now it is clear that $\mathbf{x}$ is also on the edge that triangles corresponding $\mathbf{v}_k$ and $\mathbf{v}_l$ share. Since the triangles are equilateral, the closest point on the edge to $\mathbf{v}_k$ and to $\mathbf{v}_l$ is the same point, $\mathbf{x}$. We have

$$||\mathbf{v}_k - \mathbf{x}|| = ||\mathbf{v}_l - \mathbf{x}|| = \frac{1}{3}h, \tag{18}$$

where h is the height of the triangle. Now $\tan\frac{\theta}{2} = \frac{h}{3}$ and

$$\alpha = \frac{1}{3\cos\theta} = \frac{1}{3\cos(2\arctan\frac{2\sqrt{\pi\sqrt{3}}}{3\sqrt{N}})}. \tag{19}$$

References

1. D. Terzopoulos, A. Witkin, and M. Kass. *Artif. Intell.*, **36**, 91 (1988).
2. M. Kass, A. Witkin, and D. Terzopoulos. *Int. J Comput. Vision*, **1**, 321 (1988).
3. T. McInerney and D. Terzopoulos. *Medical Image Analysis*, **2**, 91 (1996).
4. J. Montagnat, H. Delingette, and N. Ayache. *Image and Vision Computing*, **19**, 1023 (2001).
5. H. Delingette. *Int. J. of Comput. Vision*, **32**, 111 (1999).

6. R. Malladi, J. Sethian, and B. C. Vemuri. *IEEE Trans. Patt. Anal. Mach. Intell.*, **17**, 158 (1995).
7. H. Delingette. In *Proc. of the 6th Int. Conf. on Computer Vision, ICCV'98* (IEEE-CS Press 1998), pp. 311 – 316.
8. S.R. Gunn and M.S. Nixon. *IEEE Trans. Patt. Anal. Mach. Intell.*, **19**, 63 (1997).
9. K.F. Lai and R.T. Chin. *IEEE Trans. Patt. Anal. Mach. Intell.*, **17**, 1084 (1995).
10. J. Tohka. In *Proc. of 7th Eur. Conf. on Computer Vision, ECCV02*, LNCS 2352 (Springer-Verlag 2002) pp. 350 – 364 .
11. J. Mykkänen, J. Tohka, and U. Ruotsalainen. In *The new navigators: from professionals to patients, vol. 95, Proc. of Medical Informatics Europe, MIE03* (IOS Press 2003) pp. 33 – 38 .
12. D. MacDonald, N. Kabani, D. Avis, and A.C Evans. *Neuroimage*, **12**, 340 (2000).
13. D. Shen, E.H Herskovits, and C. Davatzikos. *IEEE Trans. Med. Imag.*, **20**, 257 (2001).
14. D. Shen and C. Davatzikos. *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**, 906 (2000).
15. Z. Zhang and M. Braun. In *Proc. 1997 IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'97)* (IEEE-CS Press 1997) pp. 85 – 90.
16. F. Herrera, M. Lonzano, and J. L. Verdegay. *Artif. Intell. Rev.*, **12**, 265 (1998).
17. J. Tohka. In *Proc. of 11th Int. Conf. on Image Analysis and Processing, ICIAP'01* (IEEE-CS Press 2001), pp. 459 – 464.
18. H. Delingette. Simplex meshes: a general representation for 3D shape reconstruction. Technical Report 2214, INRIA, France, March 1994.
19. S. Zucker and R. Hummel. *IEEE Trans. Patt. Anal. Mach. Intell.*, **3**, 324 (1981).
20. D.J. Williams and M. Shah. *CVGIP: Image Understanding*, **55**, 14 (1992).
21. C. Xu. Ph.D. Thesis, the Johns Hopkins University, 1999.
22. C. Xu and J.L. Prince. *IEEE Trans. Image Processing*, **7**, 359 (1998).
23. C. Xu and J.L. Prince. *Signal Processing*, **71**, 131 (1998).
24. M. Phelps and J. Mazziotta. *Science*, **228**, 799 (1985).
25. U. Ruotsalainen, J. Mykkänen, J. Luoma, J. Tohka, and S. Alenius. In *World Congress on Neuroinformatics: Part II Proceedings, ARGESIM Report no. 20* (Agresim-Verlag 2001), pp. 659 – 664.
26. S. S. Furuie, G. T. Herman, T. K. Narayan, P. E. Kinahan, J. S. Karp, R. M. Lewitt, and S. Matej. *Phys. Med. Biol.*, **39**, 341 (1994).
27. X. Zeng, L.H. Staib, R.T. Schultz, and J.S. Duncan. *IEEE Trans. Med. Imag.*, **18**, 927 (1999).
28. S. Alenius, U. Ruotsalainen, and J. Astola. *IEEE Trans. Nuc. Sci.*, **45**, 3097 (1998).
29. S. Joshi, S. Pizer, P.T. Fletcher, A. Thall, and G. Tracton. In *Proc. of Information Processing in Medical Imaging, IPMI01*, LNCS 2082 (Springer-Verlag 2001), pp. 64 – 77.
30. Y. Wang and L. H. Staib. *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**, 738 (2000).
31. J. Weese, M. Kaus, C. Lorenz, S. Lobregt, R. Truyen, and V. Pekar. In *Proc. of Information Processing in Medical Imaging, IPMI01*, LNCS 2082 (Springer-Verlag 2001), pp. 380 – 387.
32. D. Terzopoulos and D. Metaxas. *IEEE Trans. Patt. Anal. Mach. Intell.*, **13**, 703 (1991).
33. J. Montagnat and H. Delingette. *Signal Processing*, **71**, 173 (1998).
34. J. Koikkalainen and J. Lötjönen. In *Proc. of Medical Image Computing and Computer Assisted Intervention, MICCAI02*, LNCS 2488 (Springer-Verlag 2002), pp. 540 – 547.
35. A. Yezzi and A. Tannenbaum. In *Proc. of Medical Image Computing and Computer Assisted Intervention, MICCAI02*, LNCS 2488 (Springer-Verlag 2002), pp. 667 – 673.

Jussi Tohka received his M.Sc. degree in mathematics from University of Turku, Finland in 1999. Since then he has been with the Institute of Signal Processing in Tampere University of Technology, where he is presently pursuing his Ph.D. degree. From October 2001 to May 2002 he visited McConnell Brain Imaging Centre in Montreal Neurological Institute, Canada. His research interests include deformable models, medical image analysis, statistical pattern recognition and optimization algorithms.

Jouni Mykkänen received degrees in computer science, M.Sc. from the Department of Computer Science and Applied Mathematics at University of Kuopio (1996) and PhD. from the Department of Computer Science at University of Tampere (2003). Currently, he is working as assistant professor in the Department of Computer Science at University of Tampere. His research interests are in medical image analysis, deformable models and algorithmics.