Active Surface Models for Brain Imaging

Mattias Ohlsson, Peter Toft, Lars Kai Hansen, and Finn Årup Nielsen
Department of Mathematical Modelling,
Technical University of Denmark
mo,pto,lkh,fn@imm.dtu.dk

ABSTRACT

This paper presents a generic approach for surface modelling of 3D objects from volume data. Our strategy is to shape a closed surface of connected triangles to match the edge of the object. The adaptive strategy we suggest invokes minimisation of an energy function containing a set of individually weighted terms reflecting a set of desired goals.

From the closed surface of triangles we are able to do efficient 3D visualisation as well as perform quantitative measurements within the object. The framework is very general and applicable in many research areas, and we show the viability of the approach in the context of medical imaging.

1. Introduction

Identification and segmentation of a 3D object measured indirectly by volumetric data, i.e., without a physical signal, like reflections etc. from the surface itself, is a recurrent inversion problem in many areas of science. While it is relatively easy for humans to spot and identify an object from, e.g., images of the individual slices or directly using volume rendering techniques, modern acquisition techniques routinely produce data sets with hundreds of slices, hence, automatic procedures are urgently needed. Furthermore, automatic procedures can potentially lead to more consistent quantification. The success that humans have in performing segmentation tasks is due much to the fact that we use a priori *forward* information, such as e.g. closed boundaries or a specific known form when locating the structure. Automatic procedures should also incorporate a priori information to obtain good performance.

A very popular group of methods that supports inclusion of a priori information is the so called *active contours* or *active surface* methods that can be described as physically-based deformable models [Terzopoulos and Witkin, 1988, Terzopoulos et al., 1988]. In active surface models the surfaces deform themselves under the action of internal and external *forces.* The a priori information is included by means of an initial shape and position and topological constraints such as e.g. closed surfaces.

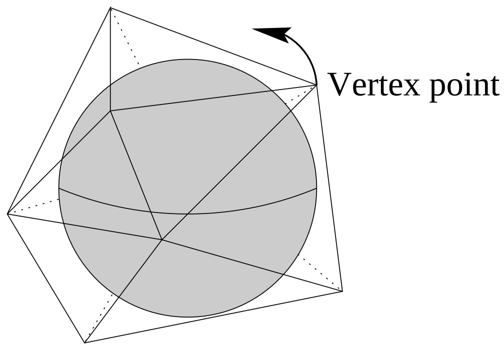

Fig. 1: The closed set of triangles, where the positions of the vertex points are iterated to match the edge of the object, in this case an ellipsoid.

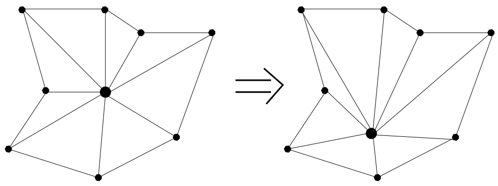

Fig. 2: The displacement of a single vertex will only influence the connections to its surrounding triangles.

Examples of such active surface algorithms are given in [Cohen and Cohen, 1993] focusing on *open* surfaces used to segment Magnetic Resonance (MR) images. Their method is significantly enhanced by the use of a Finite Element Method in the minimisation process. In [Schlesinger et al., 1996] the boundaries of the objects are constrained further by using topological closed geodesic surfaces, that deform under an the potential from an image. They report good results for segmentation of major structures in the human brain.

Another direction is taken in [Collins et al., 1995], where the task again is to locate objects of interest in brain images. Their method requires a completely segmented brain (**a**), where each volume-element (voxel) has a neuroanatomical label. In order to segment another brain (**b**) one can recover the non-linear spatial transformation field that optimally maps the intensity field from **a** to **b**. Each of the labels from **a** is now mapped onto **b**, thus segmenting **b**. The segmentation problem has become a registration problem.

The approach taken in this paper follows the active surfaces idea of having an initial closed surface estimate that matches an object of interest using an energy minimisation process. Key features of our method are:

- A flexible surface description that puts almost no restriction on the closed form on the surface.
- A very general energy formulation that can be custom-made for a wide range of objects/datasets.
- The resulting surface can easily be used for powerful 3-D visualisation or subject to further analysis such as quantitative measurements within the object.

As illustrated in Fig. 1 the boundary of the anatomical object of interest will be represented by a mesh of triangles. While we can consider both open and closed surfaces, we here confine ourselves to closed surfaces. Each vertex in the mesh will have a set of neighbouring vertices that induce a set of triangles around the vertex. All movements and adjustments of the mesh are performed at the vertex level. Note that the displacement of a single vertex will only influence its surrounding triangles (see Fig. 2). Iterating on all vertex positions will move the mesh to the object of interest, as shown schematically in Fig. 3.

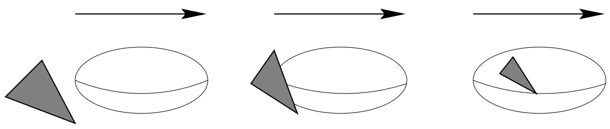

Fig. 3: Iterations on the vertex positions will move the whole mesh, so that the triangles will track the object surface.

2. Solution Strategy

The algorithm presented below will require an initial surface (boundary estimate). It is assumed that this estimate is not to far from (in a Eucledian distance sense) the true underlying boundary. The initial estimate could be the result of a rough manual segmentation, or of a *registration* based segmentation: having a segmented reference brain where the boundary of the object of interest in known. If, in addition, the non-linear spatial transformation field that register the two volumes is known one can map the reference object-surface onto the current volume, thereby providing the surface estimate [Ohlsson et al., 1997].

The adaptation of the initial surface will be performed as an energy minimisation task. This energy function consists of several terms, each representing different properties of the surface.

$$E_{\text{total}} = \sum_k \alpha_k E_k \quad , \tag{1}$$

where k sums over the possible energy terms - listed below - and α_k is a weight parameter, that determines the individual weighting[1]. By this construction it is possible to customise energy functions for various anatomical regions or objects.

The following table lists a set of symbols will be used throughout this paper:

i	The vertex number, i.e., number of the triangle corner.
j	The triangle number.
N_i	Number of triangles.
N_j	Number of vertices.
$V(\boldsymbol{x})$	The value of the volume at $\boldsymbol{x}$.
t	iteration number.

Here follows a list of possible energy terms that can be used to construct the energy function. It should be emphasised that this is not an exhaustive list. Some volumetric data may require special attractive forces or smoothing regulators not listed here.

[1]In practice we are only using $K-1$ weights since a rescaling will put one weight to unity. K is the number of energy terms.

Smoothing As shown in Fig. 4 a smoothing term minimises the squared distance to the centre of mass of the vertex neighbours

$$E_{\text{Smooth}} = \sum_{i}^{N_i} \left(\boldsymbol{x}_i - \boldsymbol{x}_{N(i)}\right)^2 \tag{2}$$

where $\boldsymbol{x}_{N(i)}$ is the mean neighbour vertex position. This energy term will try to keep smoothness. Note that this energy term implies a shrinking effect, because the global minimum of this energy term is found where all vertex positions are equal.

Reference Mesh Inertia One can add an term that prevent large movements away from the initial vertex positions. This is illustrated in Fig. 5.

$$E_{\text{Inertia}} = \sum_{i}^{N_i} (\boldsymbol{x}_i - \boldsymbol{x}_{i,org})^2 \tag{3}$$

where $\boldsymbol{x}_i$ is the current position of vertex point i and $\boldsymbol{x}_{i,org}$ is the original position of the same vertex point. And if the particular energy weight α_{Inertia} is set sufficiently high this term can prevent severe divergence of the vertex positions.

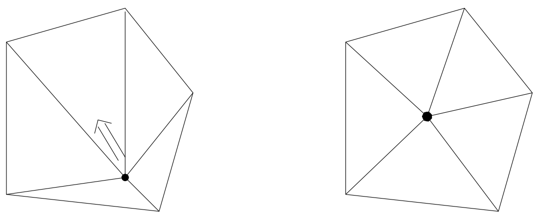

Fig. 4: The smoothing energy term will pull the vertex point to the centre of mass of the neighbours.

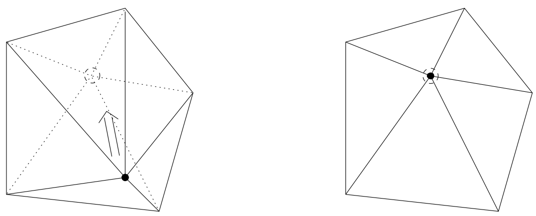

Fig. 5: The reference energy term will try to pull the vertex positions towards the initial positions.

Potential The surface must be adapted to the volume V, and the simplest way is to use the square magnitude of the gradient. In contrast to the other forces presented this potential energy makes the link to the data volume, and this energy term will track edges in the volume. As indicated in Fig. 6 the simplest form of the energy function samples the potential at the vertex positions.

$$E_{\text{Gradient}} = \sum_{i}^{N_i} |S_w \nabla V(\boldsymbol{x}_i)|^2 \tag{4}$$

where $V(\boldsymbol{x})$ is the volume that contains the object to be fitted and S_w is a spatial smoothing operator with a characteristic distance of w. The smoothing has to be used in order to extend a very sharp edge, i.e. gradient, so that it can be sensed in the areas of the current vertex positions. If the initial estimates of the vertex positions are very bad then S should do much smoothing. Similar to simulated annealing and mean field annealing the smoothing parameter w can be lowered (cooled) as a function of the iteration number t in order to get better estimates at the end.

Note that a very coarsely sampled mesh compared to the sampling distance of the volume will suggest that Eq. 4 can be replaced by

$$E_{\text{Potential}} = \sum_{j}^{N_j} \frac{1}{A_j} \int\int d\Omega_j |S_w \nabla V(\boldsymbol{x})|^2 \tag{5}$$

where the integration over Ω_j normalised with the triangle area A_j is the average squared gradient over triangle j. In practice the triangle must be sampled in several positions on the triangle and then averaged. This way of implementing the gradient term will also make it possible to compute the variance of the squared gradient, where a low variance indicates that the mesh is sufficiently fine, and a high value will signal that the triangle should be split in minor triangles (this is discussed in Subsection 3.1.

Template Matching A very useful and general energy term is the *template matching* term, where availability of a reference volume $V^{(ref)}(\boldsymbol{x})$ is assumed with an associated matched surface with vertices $\boldsymbol{x}_i^{(ref)}$. The neighbourhood around $\boldsymbol{x}_i^{(ref)}$ is then matched to the neighbourhood around the same vertex number i in the current volume:

$$E_{\text{Template}} = \sum_{i}^{N_i} \sum_{\boldsymbol{\Delta x}} F(V^{(ref)}, V) \tag{6}$$

where the function F might be chosen as

$$F(V^{(ref)}, V) = (V^{(ref)}(\boldsymbol{x}_i^{(ref)} + \boldsymbol{\Delta x}) - V(\boldsymbol{x}_i + \boldsymbol{\Delta x}))^2 \tag{7}$$

and the sum over $\boldsymbol{\Delta x}$ sums over all voxels within a predefined distance, either radially or in a rectangular region, and $\boldsymbol{x}_i^{(ref)}$ is vertex position i on the reference mesh. The function F may implement the average distance within the template or alike.

This energy term will force the mesh to search for areas in the volume that resembles the reference volume at the corresponding vertex positions, and we expect this energy term to be very relevant for person to person studies.

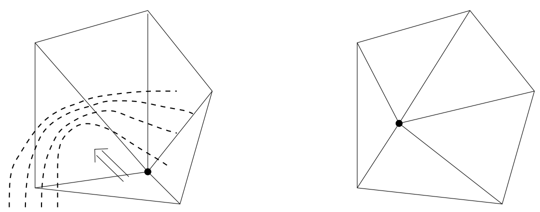

Fig. 6: The gradient attraction term will try to pull the vertex to areas of high gradients.

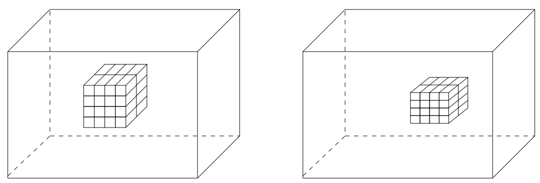

Fig. 7: The template matching term will try to search for a similar neighbourhood to a reference volume with an associated mesh.

3. Minimisation of the Energy Function

We have chosen to minimise the energy in an iterative fashion by optimising over vertex points only. It is natural to move vertex points and not triangles since, usually, all energy terms can be written as a sum over vertex points.

In principle it is possible to compute analytical derivatives of the energy function with respect to the vertex positions. However, we choose to use a simpler approach, namely to compute approximations to the gradient of the energy function by computing the difference in the energy when moving one vertex position a small step. It should be pointed out that moving a vertex point will only influence the triangles around the vertex point. This will significantly lower the computational complexity when computing the gradient for the energy function using the approximation scheme below.

$$\nabla E_i \approx \begin{pmatrix} \frac{1}{\delta\Delta x}\left[E(\boldsymbol{x}_i + \delta\Delta x) - E(\boldsymbol{x}_i)\right] \\ \frac{1}{\delta\Delta y}\left[E(\boldsymbol{x}_i + \delta\Delta y) - E(\boldsymbol{x}_i)\right] \\ \frac{1}{\delta\Delta z}\left[E(\boldsymbol{x}_i + \delta\Delta z) - E(\boldsymbol{x}_i)\right] \end{pmatrix} \tag{8}$$

where Δx is the sampling distance in the volume in the x direction (likewise with Δy and Δz) and δ is a weight between 0 and 1.

With this approximation to the gradient we can, e.g., use a simple steepest descent algorithm to minimise the energy function, by updating the vertex position according to

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} - \eta_t \nabla E_i \tag{9}$$

An alternative to the steepest descent algorithm is simulated annealing [Kirkpatrick et al., 1983], where an alternative to the current vertex position is accepted if it lowers the energy, and accepted with a certain probability if the new position implies an increase in the energy.

Regardless of the way that we actually choose the new vertex position in a certain iteration we will have to choose which vertex number to change. Two strategies are directly applicable, namely linear addressing the vertex points $i(t) = t \text{ MOD } N_i$, where MOD is the modulus operator. The other strategy is to permute the set of indices randomly, so in contrast to choosing the vertex point at random in each iteration it is guaranteed that each vertex point has been updated once every N_i iterations. Another more demanding strategy would be to rank the vertex position after the size of the gradient $|\nabla E|$ and update the vertex with the maximal gradient in each iteration.

3.1. Additional Surface Manipulation

In order to accurately represent the surface of an object the number of vertices and triangles cannot be too small. Due to non-convex and high curvature regions of the object some parts of the mesh may need further refinements. As shown in Fig. 8 two operations may be used to refine the mesh, one local and one global:

(A) The global operator will split each triangle into 4 new ones by connecting the middle point of its three edges.

(B) For the local operation a new vertex is placed on a given edge thereby splitting the two triangles that shares the edge. The location of the new vertex is chosen so that the two child triangles will have equal area.

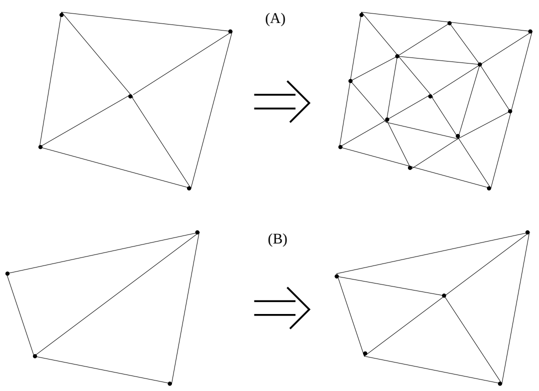

Fig. 8: Examples of mesh refinement.

A problem that can occur is that triangles might collapse as shown in Fig. 9. If the distance between two neighbour vertices is much less than the distance between voxels then the vertices should be connected and two triangles should be removed.

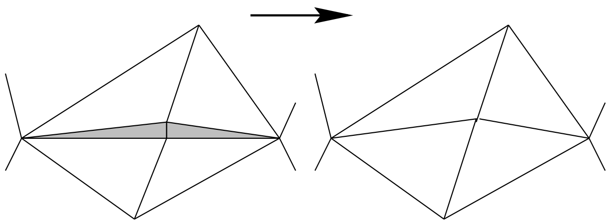

Fig. 9: The distance between two vertices becomes very small and they are replaces by one vertex point.

4. A Medical Imaging Example

In Fig. 11 is shown an initial estimate of a mesh that should match the outer edge of an MRI volume. It has been based on a (bad) hand segmentation of the individual slices. The surface consists of 3201 vertices that builds 6398 triangles. The energy used for this example is a sum of two terms,

$$E_{\text{Total}} = \alpha_1 \sum_i S_w |\nabla V(\boldsymbol{x}_i)|^2 + \alpha_2 \sum_i \left(\boldsymbol{x}_i - \boldsymbol{x}_{N(i)}\right)^2 \quad . \tag{10}$$

Based on manually tuned parameters for the weights α_1 and α_2 and using 300 iterations, requiring approximately 170 seconds on a Pentium 133 MHz Linux machine, the final mesh shown in Fig. 12 was obtained. Fig. 10 shows the total energy as a function of the number of full iterations, i.e., one iteration equals one update of all the vertex points.

The mesh during the optimisation can be seen in Fig. 13, and additional examples and a VRML2 object containing the mesh for all iterations of the same dataset can be found at `http://hendrix.imm.dtu.dk/geo`

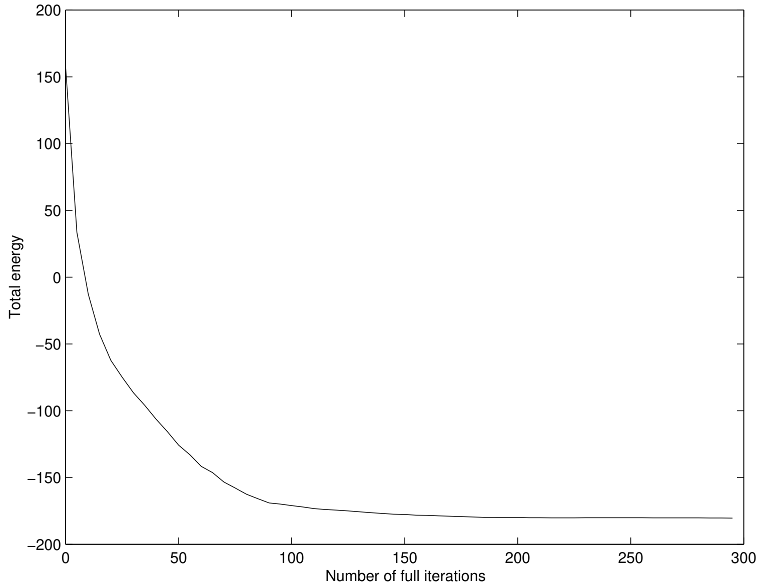

Fig. 10: The total energy as a function of the number of full iterations.

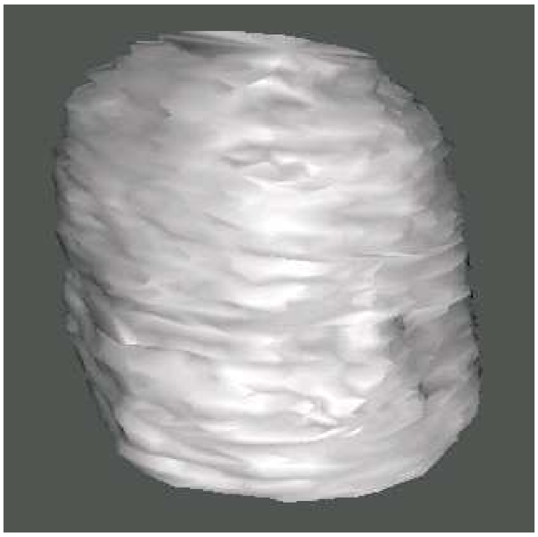

Fig. 11: Initial surface mesh.

Fig. 12: Final surface mesh.

Fig. 13: Adaptation of the surface mesh to the outer skull of an MR volume of the brain.

5. Conclusion and Acknowledgements

We have presented a generic approach for adaptation of 3D active surfaces to volumes. The approach has been implemented and demonstrated for locating of the head in an MRI data set.

Currently the energy weight parameters and the parameter in the steepest descend algorithm are tuned using trail and error, but they can be trained by a learning algorithm. Furthermore, questions such as measuring performance are still under development.

The project is funded by The Danish Research Councils, The Human Brain Project P20 MH57180 "Spatial and Temporal Patterns in Functional Neuroimaging", and by EU within the Training and Mobility of Researchers programme, contract no. ERBFM-BICT961031 "Advanced 3D Visualization of Human Brain Structure and Activation".

REFERENCES

[Cohen and Cohen, 1993] Cohen, L. and Cohen, I. (1993). Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, PAMI-15:1131–1147.

[Collins et al., 1995] Collins, D., Holmes, C., Peters, T., and Evans, A. (1995). Automatic 3-D Model-Based Neuroanatomical Segmentation. *Human Brain Mapping*, 3:190–208.

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220.

[Ohlsson et al., 1997] Ohlsson, M., Toft, P., Hansen, L. K., and Nielsen, F. Å. (1997). Coregistration and active surfaces for medical imaging. Manuscript in Preparation.

[Schlesinger et al., 1996] Schlesinger, D., Snell, J., Mansfield, L., Brookeman, J., Ortega, J., and Kassell, N. (1996). Segmentation of volumetric medical imagery using multiple geodesic-based active surfaces. In *SPIE Medical Imaging*.

[Terzopoulos and Witkin, 1988] Terzopoulos, D. and Witkin, A. (1988). Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications*, 8:41–51.

[Terzopoulos et al., 1988] Terzopoulos, D., Witkin, A., and Kass, M. (1988). Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *AI Journal*, 36:91–123.