

Sparse Finite Element Level-Sets for Anisotropic Boundary Detection in 3D Images^{*}

Martin Weber¹, Andrew Blake², and Roberto Cipolla¹

¹ Department of Engineering, University of Cambridge, UK,
{mw232,cipolla}@eng.cam.ac.uk
<http://mi.eng.cam.ac.uk/research/vision/>

² Microsoft Research, Cambridge, UK

Abstract. Level-Set methods have been successfully applied to 2D and 3D boundary detection problems. The geodesic active contour model has been particularly successful. Several algorithms for the discretisation have been proposed and the banded approach has been used to improve efficiency, which is crucial in 3D boundary detection. In this paper we propose a new scheme to numerically represent and evolve surfaces in 3D. With the new scheme, efficiency and accuracy are further improved. For the representation, space is partitioned into tetrahedra and finite elements are used to define the level-set function. Extreme sparsity is obtained by maintaining data only for tetrahedra that contain the zero level-set. We formulate the evolution PDE in weak form and incorporate a normalisation term. We obtain a stable scheme with consistent sub-grid accuracy without having to rely on any re-initialisation procedure. Boundary detection is performed using an anisotropic extension of the isotropic geodesic model. With the sparse representation, the anisotropic model is computationally feasible. We present experimental results on volumetric data sets including images with a significant amount of noise.

1 Introduction

Boundary surface detection in noisy 3D images is a vital ingredient for the analysis of medical scans, such as 3D ultrasound. In principle, one can attempt to form a surface by combining the contours of slices through the volumetric image. However, noisy images require a 3D method to facilitate the exploitation of the full spatial context of the problem. Geodesic [1–4] surface detection has been proposed and Boykov and Kolmogorov [5] have recently presented an efficient discrete algorithm using *graph cuts*, including the anisotropic case. However, greater accuracy can be achieved in the continuous setting when a consistent sub-grid definition is being employed. *Deformable models* [6] have been used to estimate curve and surface models from image data. *Level-set methods* [7–10, 1, 2] represent the surface implicitly, are topologically flexible and overcome other problems of parametric models, such as self-intersection. However, level-set

^{*} This work was supported by the EPSRC, the Cambridge European Trust and DAAD.

implementations in 3D are computationally involved. Preusser and Rumpf [11] have proposed a level-set framework with cubical finite elements. However, their numerical representation is not sparse (all levels are evolved) which requires a computational power exceeding that of current PCs.

Level set-methods [7–10] introduce a level-set function ϕ , to represent the interface Γ implicitly as the zero level-set: $\Gamma := \phi^{-1}(0)$ where $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ is a *Lipschitz-continuous*, real valued function. The implicit representation links ϕ (as the introduced *analytic entity*) with the *geometric entity* Γ : $\phi \mapsto \Gamma(\phi)$ and allows for changes in the topology during the evolution (Figure 1). Furthermore, it was pointed out [12] that this relationship can be made one-to-one by imposing the signed distance constraint. That is, the constraint fixes the gauge freedom³. The conceptual advantage is then that ϕ is (up to a sign) uniquely determined by Γ and that one can also write $\Gamma \mapsto \phi(\Gamma)$. In this way ϕ gets the intrinsic geometric meaning as the distance function for Γ .

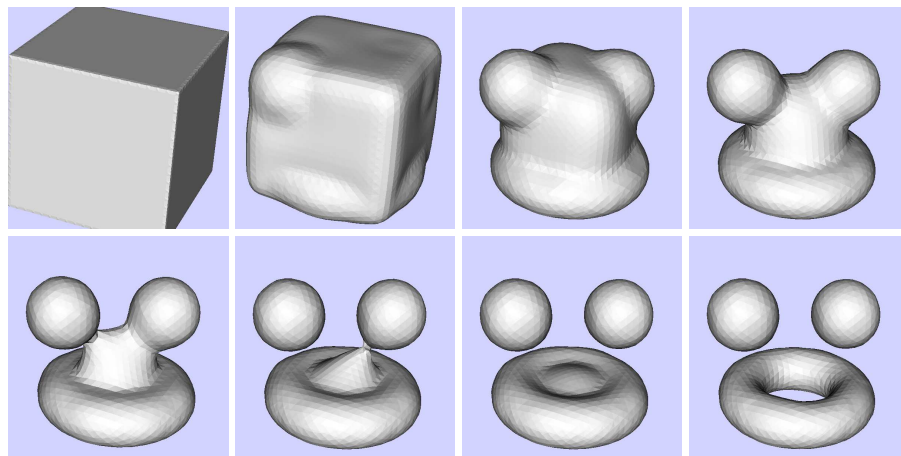


Fig. 1. Topological changes: stages of a geodesic evolution in the novel finite element scheme (starting from the top left). Three synthetic objects are to be detected. The evolution is initialised as a cube and splits automatically into the components. The generation of the hole inside the torus (bottom right) is an example of a topological change where the number of components remains unchanged (resolution $40 \times 40 \times 40$).

1.1 Differential minimisation and level-set evolution

For the evolution, one introduces an evolution parameter $t \in \mathbb{R}$ and ϕ becomes *time-dependent*⁴. One starts with an initial function $\phi(0, \cdot)$ and prescribes an

³ the freedom of using any multiple $\psi \phi$ in the place of ϕ with *any* positive valued function $\psi > 0$.

⁴ One refers to the parameter t as *time* although it is not related to physical time.

evolution $\phi(t, \cdot)$. In Section 2, we introduce a cost-functional C and define the evolution via gradient descent. The resulting evolution equation is a PDE:

$$\frac{d\phi}{dt} = \beta \quad (1)$$

where at the interface Γ , β is the differential of the cost: $\beta|_{\Gamma} := -\frac{\delta C}{\delta \phi}$ and is defined elsewhere in such a way as to maintain the signed distance constraint [12]. The signed distance constraint is well known for its desirable conceptual and numerical properties. Where ϕ is differentiable, we have $|\nabla\phi(x)| = 1$ and, for $x \in \Gamma$ the particularly simple expressions for the *normal* $N(x) = \nabla\phi(x) \in S^2$ and *mean curvature* $\kappa(x) = \Delta\phi(x) \in \mathbb{R}$ along Γ .

1.2 Previous level-set methods and numerical problems

In the following sections, u denotes the numerical representation of the level-set function ϕ . The major issues for the numerical implementation of (1) concern efficiency, stability and accuracy. Improved efficiency and stability was obtained using sparse finite element level-sets, as we reported in [13]. Here we extend the 2D scheme to 3D and include anisotropic cost-functionals. One advantage of level-set methods over discrete methods is the potential for sub-grid accuracy.

Consistent sub-grid accuracy: sub-grid accuracy requires the definition of the level-set function inside each cell of the grid. Figure 2 details the standard way in which this can be achieved [12]. Unfortunately, this sub-grid definition has several problems:

- **mixed polynomial degree:** the standard representation (see Figure 2) singles out the directions of the coordinate axes (along which u is piece-wise affine). In the interior of the cell it is generally a cubic polynomial.
- **tiling ambiguities:** the graphical output of the implicit interface is not straightforward. To obtain a facet approximation, one employs so called *iso-contour algorithms* ([10] p.425) such as the *marching cube algorithm*, in which case ambiguities in the *tiling* have to be resolved.
- **sub-grid definition cannot be used for the evolution:** the definition of *curvature motion* [8] inside a cell is problematic. To see this, let us assume that we have an initial signed distance function ϕ . Then, curvature motion is simply given by the Laplacian: $\beta = \Delta\phi$. However, the Laplacian of the numerical representation u vanishes identically: $\Delta u(x) = 0$ for any x inside the cell, independent of any node-values on the grid. This is related to the fact that u is affine along coordinate directions (alternatively, one can verify in Figure 2 that $\Delta e_0 = 0$, like any other nodal basis function). Therefore, the sub-grid definition cannot be used when the interface is evolved. Instead, one [8] computes differential operators (like the Laplacian) with finite difference operators on *stencils* that contain several cells. However, it is not entirely clear how the evolution in terms of finite difference operators is related to the sub-grid localisation of interfaces as defined in Figure 2.

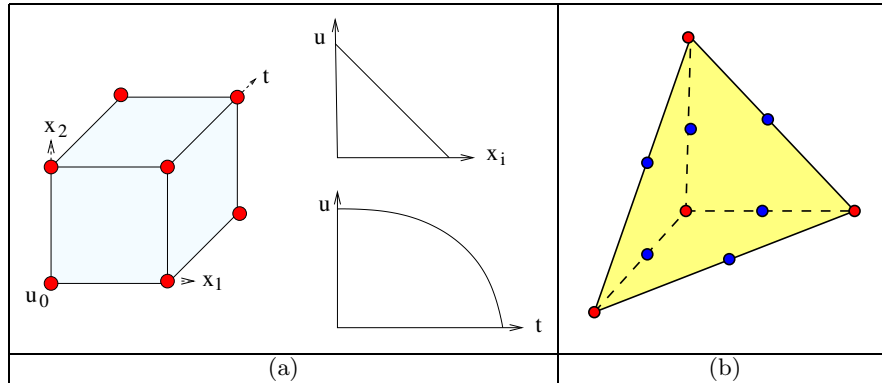


Fig. 2. Sub-grid definition: The figure illustrates the sub-grid definition of a level-set function u in terms of (a) cubical cells and (b) tetrahedral elements. (a) in the case of cubical cells, a product of functions that are affine in the direction of the coordinate-axes is used. One can express u as linear combination $u = \sum_{i=0}^7 u_i e_i$ of nodal basis functions e_i . For instance, the basis function for node 0 is: $e_0(x) = (1 - x_1)(1 - x_2)(1 - x_3)$ which evaluates to 1 at node 0 and vanishes on all other nodes. Unfortunately, the e_i (and hence u) are of mixed polynomial degree: along all coordinate axes they are affine but on the diagonal they are cubic polynomials. (b) tetrahedral elements are defined by the 4 nodes located at the vertices (1st degree case) and, 6 additional nodes at the edge-midpoints in the 2nd degree case. A general 1st or 2nd degree polynomial is prescribed by the values u_i at the nodes. Coordinate axes are not treated as special directions here.

These problems are resolved when simplices are used instead of cubical cells:

- **fixed degree:** inside each element, u is a polynomial of fixed degree.
- **simple graphical output:** with first degree simplex-elements, the interface inside each element is always a planar facet (Figure 3). Hence, no tiling ambiguity occurs and the output of the interface is straightforward.
- **consistent use of the sub-grid definition in the evolution:** the evolution equations are treated in weak formulation, which allows us to define the evolution equation consistently with the defined sub-grid accuracy.

2 Anisotropic interface optimisation

By adopting an interface optimisation formulation, we incorporate the prior knowledge/demand that the interface is a continuous boundary Γ . As is well known, this implies robustness to noise. The task is to find local minima of the cost

$$C = \int_{\Gamma} \gamma \quad (2)$$

which is a two-dimensional *Lebesgue integral* of a density $\gamma > 0$ (specified below) over the interface Γ . In the anisotropic case, γ depends not just on location but

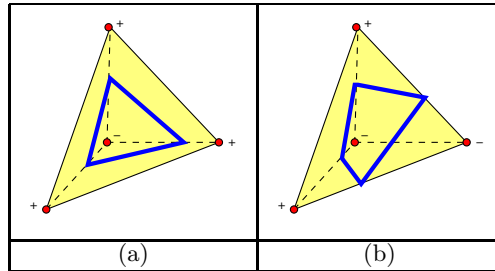


Fig. 3. Implicit facet in 3D: depending on the signs of the node-values, the implicit facet is either (a) a triangle (if one sign differs from the other three) or (b) a planar quadrangle (if there are two pairs of different signs).

also orientation. Global minima of (2) are trivial (sets of measure zero) unless additional constraints are imposed [5]. We follow here the strategy of interface evolution [7, 8] by deforming an initial interface to obtain local minima.

2.1 Edge indication function

We denote the real-valued volumetric image by $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. Edges are related to the gradient of f . In order to *quantify* the strength of edges, a function g is introduced. In the literature [14], g is referred to as *edge indication function*. The particular functional form of g is not crucial. It is however essential that $g \geq 0$ and usually g is monotonous with respect to feature gradient magnitude. We also adopt the convention to normalise g to have values in the unit interval $[0; 1]$. We let $g(x)$ depend on the gradient magnitude $|\nabla f(x)|$ and choose [15]:

$$g := 1 - \exp\left(-\frac{a}{|\nabla f_\sigma|^q}\right) \quad (3)$$

where f_σ denotes the smoothed version of the image f obtained by convolution of f with a Gaussian of width σ . Furthermore, a and q are real, positive constants that determine the response of the edge/feature detection.

In order to complete the definition of the edge detector function g , one can adjust the parameters a and q in (3) automatically to the image-data. For instance we determine a, q in such a way that the average gradient magnitude $\langle |\nabla f_\sigma| \rangle$ over the image results in $g = \frac{1}{2}$ and that the slope of g with respect to the gradient magnitude equals $-1/\langle |\nabla f_\sigma| \rangle$ at this point. These are sensible choices which are invariant to the affine value-transformations of the image [16].

2.2 An anisotropic density

In this section we define a cost $C(\Gamma)$ for the boundary detection by specifying the density γ in (2). One viable density γ is obtained by simply choosing the edge indication function g itself: $\gamma(x, N) = g(x)$. In this case, one refers to the

density as (*isotropic*) *Riemannian metric* [17, 9] since the cost $C(\Gamma) = \int_{\Gamma} g$ is the *Riemannian area*. The attraction to the edges/features in the image is geometrically obtained by driving the evolution towards g -minimal surfaces.

This is the geodesic model, which is capable of detecting boundaries even in the presence of noise. However, the isotropic model is known to be sub-optimal. While the edge-strength is taken into account, the orientation information provided by ∇f_{σ} is not utilised. We can further improve the performance of the geodesic boundary detection by including an orientation-dependent term. Formally, this means that we consider *anisotropic* densities γ which depend non-trivially on the orientation of the interface N .

Orientation-dependent terms have been used previously: Kimmel and Bruckstein [18] detect boundary-curves in 2D images using *Laplacian zero crossings*. However, their formulation requires additional regularising functionals and parameters which are needed to determine their relative strength. Instead, the formulation presented here is based on a single cost-functional that is to be minimised. Boykov and Kolmogorov [5] consider an anisotropic extension of the geodesic model in their discrete setting. In our continuous formulation, the anisotropic extension is conceptually straightforward with consistent sub-grid accuracy and computationally feasible due to the extreme sparsity of the representation.

We consider the general quadratic expression

$$\gamma = \frac{1}{2} \langle N, N \rangle_G \quad (4)$$

where G is a matrix-valued function and $\langle N, N \rangle_G := N^{\top} G N$. We use the edge detector g to define G :

$$G := \mathbf{1} + \frac{g-1}{|\nabla f_{\sigma}|^2} \nabla f_{\sigma} \otimes \nabla f_{\sigma} \quad (5)$$

where $v \otimes v := v v^{\top}$. This choice is motivated by the following properties:

- **alignment case** ($\nabla f_{\sigma} \propto N$): we obtain $G N = g N$, which is equivalent to geodesic motion.
- **non-alignment case** ($df_{\sigma} N = 0$): we obtain $G N = N$, equivalent to *curvature motion*.
- **weak gradient case** ($\nabla f_{\sigma} \approx 0$): since $g \rightarrow 1$, we obtain $G \rightarrow \mathbf{1}$ which results in *curvature motion*.

When compared to the isotropic geodesic motion, there is an additional aligning force. Figure 4 demonstrates the orienting force for the special case where $g = 0$. In this case (5) becomes $G = \mathbf{1} - \frac{1}{|\nabla f|^2} \nabla f \otimes \nabla f$. The following proposition (proved in [16]) quantifies the aligning property :

Proposition 1 *Difference between the anisotropic and geodesic cost-functionals:*

The difference between the cost functionals is given by $\Delta C := 2 \int_{\Gamma} \gamma - \int_{\Gamma} g = \int_{\Gamma} \Delta \gamma$ (here Δ denotes differences, not the Laplacian) with the difference-density

$$\begin{aligned} \Delta \gamma &= \langle N, N \rangle_G - g \\ &= (1 - g)(1 - a) \end{aligned} \quad (6)$$

with the aligning term $a := \left\langle N, \frac{1}{|\nabla f_\sigma|} \nabla f_\sigma \right\rangle^2 \in [0; 1]$. Consequently,

- $\Delta\gamma \geq 0$: hence an additional cost is present in the anisotropic density.
- $\Delta\gamma \propto (1 - g)$: hence the orienting power is most pronounced at edges and vanishes in feature-free regions.
- $\Delta\gamma \propto (1 - a)$: if the interface is aligned with the feature ($a \approx 1$) there is no additional cost; when the alignment is worst (i.e. orthogonal, $a \approx 0$) the orienting force is maximal.

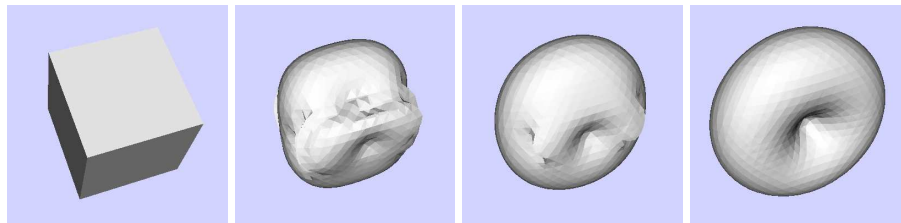


Fig. 4. Orienting motion: the figure illustrates the orienting property of quadratic motion $\gamma = \frac{1}{2}\langle N, N \rangle_G$. Orientation is prescribed here by selecting a distance map f and setting $G := \mathbf{1} - \nabla f \otimes \nabla f$. In this example, f is the distance from a circle. The interface is oriented by the evolution without being contracted or attracted to any features such as edges ($|\nabla f| \equiv 1$). Rather, any initial interface is progressively aligned to the orientation prescribed by f . A temporary local ‘folding’ of the interface can be observed. This is related to the fact that the cost functional does not discriminate between N and $-N$; however, the cost of transitions between the two possible alignments means that eventually there is one consistent alignment.

2.3 Gradient descent for the anisotropic scheme

Surface evolution is defined as the gradient descent of the cost (2) with density (4). Using *variational calculus*, one can derive [16] the following result:

Proposition 2 *Gradient descent of quadratic densities*

The normal motion $\beta|_\Gamma = -\frac{\delta C}{\delta \phi}$ for the gradient descent is given by

$$\beta|_\Gamma = \operatorname{div} V \tag{7}$$

$$V = (G - \gamma \mathbf{1}) N \tag{8}$$

where N is the surface normal. We call V the *descent-generating vector field*. The geodesic model is a special case in which $G = 2g\mathbf{1}$ and hence $V = gN$. Equation (7) has also been used by Vasilevskiy and Siddiqi in their flux maximising finite difference scheme [19]. Here, the vector field V derives from (8) and we exploit equation (7) in our weak scheme. The fact that such a vector field exists is of prime importance to our scheme since it enables a straightforward application of Gauss’s integral theorem for the weak formulation of Section 4.

3 Sparse finite element representation

The new numerical representation u of the level-set function employs a sparse finite element complex [13]. For its definition in 3D, we have to partition space into tetrahedra.

3.1 Partitioning of three-space into tetrahedra

Unfortunately, unlike in the 2D case, it is not possible to partition \mathbb{R}^3 into standard 3-simplices. However, the methodology of [13] is not confined to standard simplices. Here, we obtain a convenient partitioning \mathcal{M} of space as *Delaunay tetrahedrisation* [20] of the vertex-set $(2\mathbb{Z}^3) \cup ((1, 1, 1)^\top + (2\mathbb{Z}^3))$. A particularly convenient feature of this mesh is that all tetrahedra of the mesh have the same shape. Each element has two even $(2\mathbb{Z}^3)$ and two odd $(2\mathbb{Z}^3 + (1, 1, 1)^\top)$ vertices. For instance, the four vertices $\{(0\ 0\ 0)^\top, (2\ 0\ 0)^\top, (1\ -1\ 1)^\top, (1\ 1\ 1)^\top\}$ span a tetrahedron of the mesh \mathcal{M} .

As in the 2D case [13], we restrict the actual numerical representation to the sparse simplicial sub-complex which consists only of those elements which contain the zero level-set. We call this minimal set the *active complex* $\mathcal{A} \subset \mathcal{M}$.

3.2 Finite elements for functional representations

The evolution equations of Section 4 involve the continuous functionals u and G_{ij} which have to be represented numerically. This is achieved with consistent sub-grid definition by using standard finite element methods [21]. We briefly describe the technique for u (every component of G is represented similarly).

u is a linear combination of nodal-basis functions: $u(x) = \sum_j u_j e_j(x)$. The nodes are located at the vertices that are contained in the active complex \mathcal{A} . In the second degree case ($p = 2$), additional nodes are inserted as in Figure 2b. Each node defines a nodal basis function e_j . Inside a tetrahedral element $T \in \mathcal{A}$, e_j is defined as the unique [21] polynomial of degree p which evaluates to 1 at the node and vanishes on all other nodes: $e_j(x_i) = \delta_{ij}$. Note that e_j is a globally defined continuous function over the area Ω covered by the active complex \mathcal{A} .

The fact that the integral over u is a linear map allows us to integrate efficiently by using linear combinations of pre-computed integrals over (products of) basis functions. In fact, this type of integration can be performed analytically [16] and we only need to store one real value u_j at each node. For first degree elements, each active element $T \in \mathcal{A}$ contains one planar facet (Figure 3).

4 Surface evolution with sparse finite elements

We now show how a stable evolution can be defined to realise the gradient descent expression (7) in terms of the sparse representation. The sparse representation is updated in two stages:

- first, a differential update corresponding to a time-step Δt is performed. This alters the values of nodes u_j .
- secondly, the active complex \mathcal{A} is updated to restore the minimal containing property. Elements that no longer contain the zero level-set are deleted while neighbouring elements get activated if the zero-level set has moved into their domain. As in the 2D case, the criteria that control the activation and removal of elements are simply obtained⁵ and functional extrapolation is used to initialise newly activated nodes.

For the algorithmic details of this process, we refer to [13, 16]. The differential update with *time-step* Δt : $u(t + \Delta t, \cdot) = u(t, \cdot) + v(t, \cdot)$, $v = \Delta t \beta$ combines two equations, which we formulate in *weak form (Petrov Galerkin)*:

- the normalisation of the level-set function is maintained by demanding $\langle \nabla u, \nabla u \rangle - 1 = 0$. This is formulated in the weak sense by

$$z_1^i = \int_{\Omega} e_i (\langle \nabla(u + v), \nabla(u + v) \rangle - 1) \quad (9)$$

- the interface motion $v - \Delta t \beta|_{\Gamma} = 0$ is formulated as

$$z_2^i = \int_{\Omega} e_i (v - \Delta t \beta) \quad (10)$$

Note that $\langle \nabla u, \nabla u \rangle - 1 = 0$ (and likewise (9)) has a *flat direction* since it is invariant to any level-shift $u \rightarrow u + c$, $c \in \mathbb{R}$. While this would complicate the use of the equation in isolation⁶, it is not a problem when the equation is combined with the interface motion equation (10).

If there are n active nodes, we have $2n$ equations and determine the update v as the least-square solution to $|z_1|^2 + |z_2|^2$. This is a sparse, banded problem which we solve [13] by the *conjugate gradient method* [22].

The interface speed $\beta_{\Gamma} = \text{div}V$ depends on second order derivatives, since V depends on $N = \nabla u$. In *strong form*, this causes the same problems as the finite-difference approach (Section 1.2) since the desired second order derivatives cannot be obtained from the sub-grid definition inside the elements and are ill defined at element-transitions. This problem is resolved in the *weak* formulation. Instead of having second order derivatives which are ill-defined at element transitions, we encounter boundary integrals (which we compute efficiently [16]). By Proposition 2, the gradient descent has a descent-generating vector field $\beta = \text{div}V$, hence we can apply Gauss's theorem and perform partial integration. This allows us to eliminate second order derivative operators in (10):

$$\int_{\Omega} e_i \beta = \int_{\partial\Omega} e_i \langle V, \mathcal{V} \rangle - \int_{\Omega} de_i V \quad (11)$$

where \mathcal{V} denotes the normal along the boundary $\partial\Omega$. The right hand side does not contain any second order derivatives of the level-set function.

⁵ This reduces to the task of determining if a boundary-face of \mathcal{A} (a triangle) contains any part of the zero level-set. For $p = 1$ this simply depends on whether all signs of u_j at the nodes of the triangle agree.

⁶ This equation is used in some re-initialisation schemes [8].

5 Experimental results

In this section, we apply the new method to real medical image data. Figures 5 and 6 illustrate results of the boundary detection (white-grey matter boundary) based on MRI scans⁷. Figure 7 shows the detection of the vessel-system in a 3D ultrasound scan of a liver⁸. The ultrasound data shows a significant amount of noise and the detection of the vessel-contours in the slice requires a 3D method which takes the full spatial context into account.

For all examples, we used 1st degree elements (i.e. $p = 1$) and determined the constants a, q of (3) automatically, as described in Section 2.1.

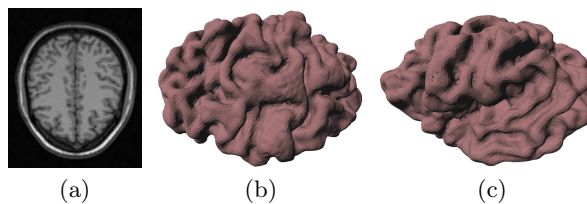


Fig. 5. Cortical brain structure: (a) displays a slice through the volumetric MRI data. (b) and (c) show views of the VRML model created by the described method.

6 Conclusion and future work

We presented a novel sparse finite element scheme and applied it to boundary detection problems in 3D images. The geodesic model was extended to include *quadratic densities*. The performance of the geodesic model was improved by this anisotropic extension and the aligning force was analysed.

Boundary detection was formulated as interface optimisation problem and gradient-descent was used for the differential minimisation. A key observation was the existence of a *descent-generating vector field* which simplified the weak formulation of the evolution. This had several numerical advantages: first degree elements were sufficient for the evolution, a consistent sub-grid definition was established and extreme sparsity was obtained since no additional elements were required to define the evolution. Like in the 2D case, numerical stability was obtained by incorporating a normalising term into the evolution equation with no need for re-initialisation. Applications of the method to synthetic and real data were presented and the robust performance in the case of noisy images was confirmed experimentally. With the efficient representation, high resolution boundary detections were feasible on a standard laptop computer.

Future work will include the implementation of the second degree scheme and the inclusion of further application-specific types of optimisation problems.

⁷ sources: <http://www.bic.mni.mcgill.ca/brainweb/>, <http://www.wbic.cam.ac.uk>

⁸ source: http://svr-www.eng.cam.ac.uk/~rwp/stradx/sample_data.html

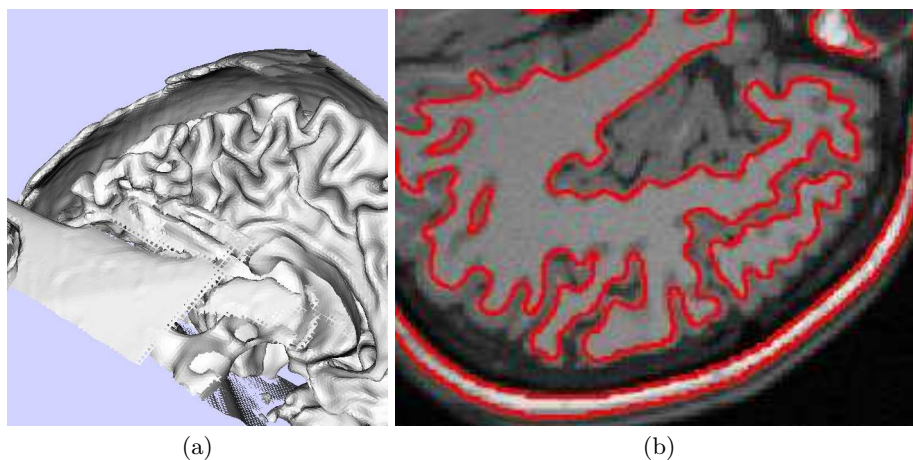


Fig. 6. MRI-brain scan: the cortical brain structure is detected from an MRI-image, using the quadratic cost. The evolution was initialised by a sphere located in the area of white matter and inflated with the aid of a balloon force ($c = -0.1$). (a) shows a projective view of the obtained surface and in (b) a section of the 3D interface is superimposed on the corresponding slice of the input-image.

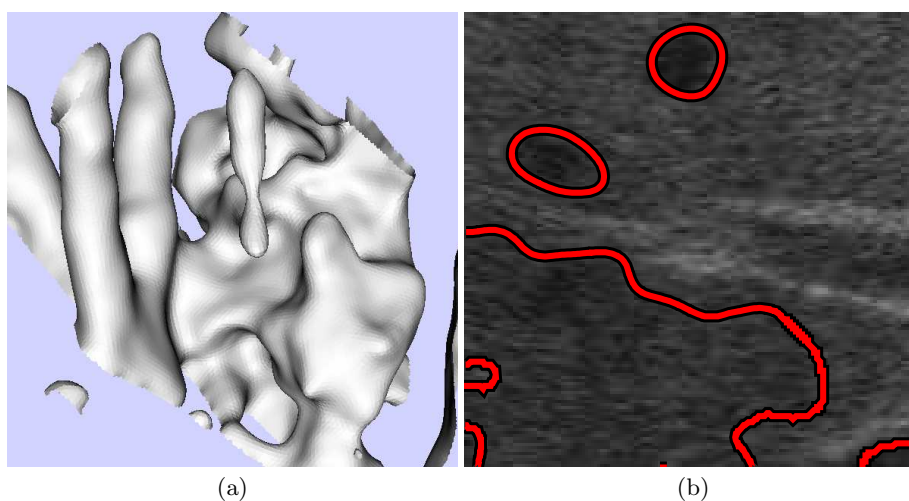


Fig. 7. Ultrasound scan of a liver (vessel system): boundaries of the vessel system are detected in an ultrasound scan. The evolution was initialised by a sphere located inside one of the vessels. (a) shows a projective view of the obtained surface and in (b) a section of the 3D interface is superimposed on the corresponding slice of the input-image. The boundary was detected successfully, despite the considerable amount of noise. $\sigma = 5$ was used to smoothen the intensity and quadratic evolution with additional balloon force $c = -0.1$ was employed. A similar result could not have been obtained by considering slices in isolation.

References

1. Malladi, R., Sethian, J., Vemuri, B.: Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Analysis and Machine Intelligence* **17** (1995) 158–175
2. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Gradient flows and geometric active contour models. In: *Proc. IEEE Int. Conf. on Computer Vision*. (1995) 810–815
3. Yezzi, A., Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A.: A geometric snake model for segmentation of medical imagery. *IEEE Trans. Med. Imaging* **16** (1997) 199–209
4. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces: A geometric three dimensional segmentation approach. *Numerische Mathematik* **77** (1997) 423–425
5. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: *Proc. IEEE Int. Conf. on Computer Vision*. (2003) 26–33
6. Terzopoulos, D., Metaxas, D.: Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Trans. Pattern Analysis and Machine Intelligence* **13** (1991) 703–714
7. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. of Comp. Phys.* **79** (1988) 12–49
8. Sethian, J.: *Level Set Methods*. Cambridge University Press, Cambridge (1999)
9. Sapiro, G.: *Geometric Partial Differential Equations and Image Processing*. Cambridge University Press, Cambridge (2001)
10. Osher, S., Paragios, N.: *Geometric Level Set Methods in Imaging Vision and Graphics*. Springer, New York (2003)
11. Preußner, T., Rumpf, M.: A level set method for anisotropic geometric diffusion in 3D image processing. *SIAM J. on Applied Math.* **62(5)** (2002) 1772–1793
12. Gomes, J., Faugeras, O.: Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation* **11** (2000) 209–223
13. Weber, M., Blake, A., Cipolla, R.: Sparse finite elements for geodesic contours with level-sets. In: *Proc. European Conf. on Computer Vision, Prague, Czech Republic*, Springer, Berlin Heidelberg (2004) 391–404
14. Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: Fast geodesic active contours. *IEEE Trans. Image Processing* **10** (2001) 1467–1475
15. Weickert, J., ter Haar Romeny, B., Viergever, M.: Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Processing* **7** (1998) 398–410
16. Weber, M.: *Curve and Surface Reconstruction from Images and Sparse Finite Element Level-Sets*. PhD thesis, Department of Engineering, University of Cambridge (2004)
17. Do Carmo, M.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall (1976)
18. Kimmel, R., Bruckstein, A.: Regularized Laplacian zero crossings as optimal edge integrators. *International Journal of Computer Vision* **53** (2003) 225–243
19. Vasilievskiy, A., Siddiqi, K.: Flux maximizing geometric flows. In: *Proc. IEEE Int. Conf. on Computer Vision, Vancouver, Canada* (2001) 149–154
20. Edelsbrunner, H.: *Geometry and Topology for Mesh Generation*. Cambridge University Press, Cambridge (2001)
21. Zienkiewicz, O., Morgan, K.: *Finite Elements & Approximation*. John Wiley & Sons, New York (1983)
22. Schwarz, H.: *Numerische Mathematik*. Teubner, Stuttgart (1993)