Deformable Model with Adaptive Mesh and Automated Topology Changes

Jacques-Olivier Lachaud Benjamin Taton
Laboratoire Bordelais de Recherche en Informatique
351, cours de la Libération
33405 TALENCE CEDEX - FRANCE

Abstract

Due to their general and robust formulation deformable models offer a very appealing approach to 3D image segmentation. However there is a trade-off between model genericity, model accuracy and computational efficiency. In general, fully generic models require a uniform sampling of either the space or their mesh. The segmentation accuracy is thus a global parameter. Recovering small image features results in heavy computational costs whereas generally only restricted parts of images require a high segmentation accuracy.

This paper presents a highly deformable model that both handles fully automated topology changes and adapts its resolution locally according to the geometry of image features. The main idea is to replace the Euclidean metric with a Riemannian metric that expands interesting parts of the image. Then, a regular sampling is maintained with this new metric. This allows to automatically handle topology changes while increasing the model resolution locally according to the geometry of image components. By this way high quality segmentation is achieved with reduced computational costs.

1. Introduction

Deformable models are extensively used in the field of image analysis. Their formulation provides a very robust and general framework that remains available in a wide range of applications and many variants have been tailored to solve efficiently given problems. Since they often use *a priori* knowledge on the shape and the topology of recovered objects these models remain very application specific and are not well suited to general object reconstruction.

In contrast, various models achieve extraction of image components with arbitrary complex topology. That kind of models is of great interest in the field of 3D biomedical image analysis where objects have complex and sometimes unexpected shapes, and where accurate initialization of models is difficult. As a counterpart to their genericity these solutions come in general with important increases of the number of shape parameters and hence of time and space complexity. Of course segmentation is expected to recover the finest image details. Time and space complexity is thus fully determined by the image resolution and computational costs become prohibitive due to the steady improvements of acquisition devices.

This emphasizes the need for multi-scale approaches. These methods should be able to recover accurately objects with arbitrary topology while keeping a restricted set of shape parameters. The contribution of this paper is to present an attempt in this direction. The model we propose is an extension of the model first presented in [7]. It is a closed oriented triangular mesh designed to reconstruct objects with an arbitrary topology from large 3D images. Our model changes its topology in a fully automated way and has the ability to adapt its resolution according to the geometry of image features. The computational costs of the segmentation step and of any subsequent operation are thus reduced, while object shapes remain accurately represented. In addition, our model comes with the usual advantages of parametric models, namely intuitive user interaction as well as easy introduction of additional constraints through the definition of new forces [3, 15]. Moreover, coarse to fine approaches remain available.

The first part of the paper recalls significant previous works that involve topology and resolution adaptation in the context of 3D image segmentation. The second part describes our model and explains how topological changes are detected and handled and how adaptive resolution is achieved by changing metrics. The third part shows how these metrics are computed directly from images. Experimental results are presented and discussed in the last part of the paper.

2. Previous works

2.1. Topology adaptation

The various models that achieve adaptive topology are usually classified in two categories.

In the parametric approach, objects are represented explicitly as polygonal meshes. Topological changes are performed by applying local reconfigurations to the model mesh [4, 6, 7] or by computing a new topologically consistent parameterization of the model [9]. The main difficulty consists in detecting self-collisions of the model. McInerney and Terzopoulos [9] investigate the intersections of the polygonal mesh with a simplicial grid that covers the image. Then they detect special configurations that characterize self intersections of the model. Lachaud and Montanvert [7] keep the sampling of their triangulated surface regular. This makes it possible to detect self collisions of the deformable model by checking distances between non neighbor vertices. With an adapted data structure the complexity of this algorithm is $O(n \log n)$ where n denotes the number of vertices used to sample the surface.

In the level-set approach [2, 8] object boundaries are no longer described explicitly: they are represented as the zero level-set of a mapping f defined over the image space. The boundary evolution is translated into evolution equations involving f which are solved iteratively to make the model approach image contours. With these methods topological changes do not require additional procedures since they are embedded in the evolution of f. Theoretically f has to be computed on the whole image space, which is computationally expensive. These costs are significantly reduced by updating only in narrow band around the zero level of f [1]. This band must however be reset periodically and the method remains costly.

2.2. Adaptive resolution

This section recalls different extensions to the deformable models described in Sect. 2.1 that were proposed to achieve an adaptive resolution.

In the framework of level-set the mapping f is usually computed on a regular grid. In the same way, the grid used by McInerney and Terzopoulos is also regular. To recover the finest image features the grid should have the same resolution as the image. Similarly in [7] automated topological changes are made possible through a uniform sampling of the model triangulated surface. The finest image structures can only be detected if all the mesh triangles have the same dimension as image voxels. All these methods have therefore heavy computational costs when working with high resolution images.

One way to reduce computation times is to adopt a "coarse to fine" approach. Deformable models are initialized with a coarse resolution and then deformed toward their energy minimum. Then they are globally refined and their energy is minimized again. This process is repeated until the expected accuracy is reached. With this method the costs of the first deformations are significantly reduced. However algorithms remain expensive when the requested accuracy increases. In addition, it is not always possible to recover the image details that are lost or ignored during the first steps of the algorithm.

In the context of parametric models, an other approach is to adapt the vertex density on the polygonal mesh depending on the geometric properties of objects. Delingette [5] propose to adapt locally the resolution of simplex meshes according to the model curvature. This is achieved in two ways. First, an internal force that attracts vertices towards highly curved parts of the mesh is introduced. Secondly, faces of the model are checked periodically, and those with a curvature higher than a given threshold are refined. The vertex number is optimized without spoiling segmentation quality. However, automated topological changes of simplex meshes are possible only in the context of 2-D images [10]: volume image reconstruction requires the help of a supervisor. In addition, mesh refinement is governed by quantities that are computed from the deformable model itself and are thus clearly *a posteriori* information. As a consequence, vertex number optimization should only be performed once an accurate segmentation has been obtained. This requires a uniform high resolution of the model mesh and implies important computational costs. At last Delingette explains [5] that this method does not perform well on noisy images, since outliers and hence noise tend to be refined.

3. Model description

3.1. Resolution adaptation

As the model presented in [7], our model keeps a regular sampling. This is achieved by maintaining length constraints on the deformable surface edges. Namely, for two neighbor vertices u and v on the mesh we make sure that

$$\delta \leq d_E(u, v) \leq \zeta\delta \tag{1}$$

where $d_E(u, v)$ denotes the Euclidean distance between u and v. The parameter δ determines the global resolution of the model. The parameter ζ determines the ratio between the length of the longest and shortest edges allowed for the model. After each deformation of the model, each edge is checked. If one of the constraints no longer holds then the investigated edge is either split or contracted as shown on Fig 1.

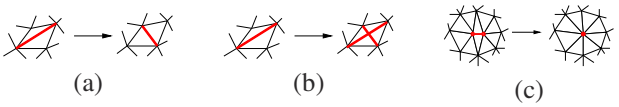


Figure 1. Operators used to restore the constraint on edge lengths. (a) edge inversion, (b) vertex creation, (c) vertex merge. Transformation in (c) must be handled in a special way when the two merged vertices share a common neighbour (see Fig. 3.b).

In [12, 13], the authors consider their mesh as a spring mass system. They achieve adaptive resolution by locally changing the stiffness of the springs according to the features found in the image. Instead, we propose to leave the mechanical properties of the system unchanged and to change the way distances are estimated.

Namely, the Euclidean distance d_E is replaced by a Riemannian distance d_R that geometrically expands area of interest. Informally speaking, d_R is chosen to overestimate distances in the vicinity of interesting image features. As a consequence, the lengths of the model edges tend to overrun the threshold $\zeta\delta$. Therefore edges tend to split (Fig 1.a) and the model mesh gets locally finer. Symmetrically, d_R underestimates distances in areas with no significant image information. Therefore, edge lengths decrease under the threshold δ and are contracted (Fig 1.c). This results in a locally coarser resolution.

More precisely, a Riemannian metric is a way to measure elementary displacements depending on both their origin and their orientation. The length of a displacement $ds = (dx, dy, dz)$ starting form (x, y, z) is given by $\|\mathbf{ds}\|_E^2 = dx^2 + dy^2 + dz^2$ in a Euclidean space. In a Riemannian space, it takes the more general form:

$$\|\mathbf{ds}\|_R^2 = {}^T ds \times G(x, y, z) \times ds \tag{2}$$

where $G(x, y, z)$ denotes a symmetrical positive-definite matrix, namely a dot-product. The mapping G should be $\mathcal{C}^1$ and is called a *Riemannian metric*. The geometrical interpretation of metric changes as well as the way metrics are built from images are discussed in detail in Sect. 4.

Theoretically, computing exact distances between two vertices u and v requires the minimization of the functional

$$L(c) = \int_0^1 \|c'(t)\|_R \, dt$$

over set of the paths that join u and v. Namely, it consists in finding and measuring a shortest path (*a geodesic*) between u and v. However, on the model mesh, neighbor vertices are close from each other so that G may be considered as constant along edges. With this assumption the shortest path between u and v is a straight line the length of which is easily and efficiently computed.

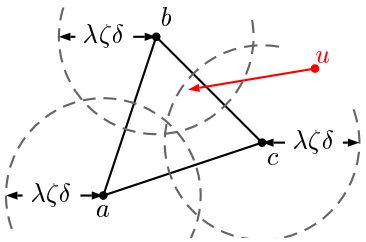


Figure 2. Collision detection method. A vertex u cannot cross over a face (a, b, c) of the deformable model unless it enters one of the sphere centered in a, b or c with radius $\lambda\zeta\delta$.

3.2. Topology adaptation

In the Euclidean context, mesh regularity allows to detect self collisions of the model [7], and hence to determine when topological changes must be performed and how they must be handled. The main idea is that a vertex u cannot cross over a face (a, b, c) without entering one of the spheres with well chosen radius $\lambda_E\delta\zeta$ centered in a, b or c (Fig. 2). After each iteration of the model, the pairs of vertices (u, w) such that

$$d_E(u, w) \leq \lambda_E\zeta\delta. \tag{3}$$

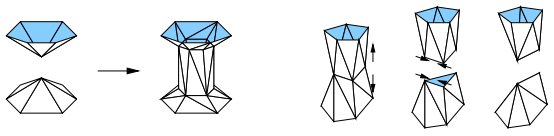
are detected and the transformation described in Fig. 3.a is performed. It may be shown that choosing λ_E such that

$$\sqrt{1 - \frac{1}{\zeta} + \frac{1}{\zeta^2}} + \frac{v_{max}}{2\zeta\delta} \leq \lambda_E \tag{4}$$

is enough to ensure that no vertex may cross over a face without being detected. In (4) the parameter v_{max} denotes the length of the largest vertex displacement allowed during an iteration. If v_{max} is too small, too many iterations are needed for the model to reach its equilibrium position. In contrast, if v_{max} is too large, collisions are detected and tunnels are created between too distant parts of the model. As a consequence, thin objects cannot be recovered properly. A convenient trade-off is $v_{max} = \frac{1}{2}\delta$. By this way the minimum possible thickness corresponds to edge length and computation times remain reasonable.

With a naive algorithm, finding vertex pairs that do not follow (3) would require a complexity of $O(n^2)$ for a mesh containing n vertices. Nevertheless, storing vertices in an octree structure reduces the complexity to $O(n \log n)$.

One may check that this way of detecting the model self collisions also work in non-Euclidean spaces, provided λ_E



(a) (b)

Figure 3. Transforms used to restore the topological consistency of the model. When two parts of the surface collide the local reconfiguration of the mesh shown in (a) is performed. When a tunnel between two parts of the volume becomes too tight, reconfiguration (b) is performed.

is replaced with a new constant λ_R that takes account of the metric. This results in the following choice

$$1 + \frac{v_{max}}{2\zeta\delta} \leq \lambda_R.$$

3.3. Dynamics

As most parametric models our deformable surface is viewed as a set of vertices, each of which evolves under the influence of various forces, namely the usual internal regularizing forces $\mathbf{F_{int}}$, a damping force, along with any additional external force $\mathbf{F_{ext}}$.

On a Riemannian manifold, the trajectory of a vertex submitted to a force $\mathbf{F}$ is theoretically described by the following set of equations:

$$m\ddot{x}_k + \sum_{i,j=0}^{2} \Gamma_{ij}^k \dot{x}_i \dot{x}_j = F_k, \ k = 0, 1, 2 \qquad (5)$$

where m and $x(t) = (x_0(t), x_1(t), x_2(t))$ respectively denote the vertex mass and position. The Γ_{ij}^k coefficients are known as the *Christoffel's symbols* and are given as

$$\Gamma_{ij}^k = \frac{1}{2} \sum_{l=0}^{2} g^{kl} \left(\frac{\partial g_{il}}{\partial x_j} + \frac{\partial g_{jl}}{\partial x_i} - \frac{\partial g_{ij}}{\partial x_l} \right)$$

where g^{kl} denotes the coefficient at position (k, l) in the matrix G^{-1}. Compared with the Newton's laws of motion, equation (5) includes the additional term $\sum_{i,j=0}^{2} \Gamma_{ij}^k \dot{x}_i \dot{x}_j$.

Since G is C^1 and is everywhere positive-definite, the Christoffels symbols and hence the additional term could easily be computed. However this term is second order in $\dot{\mathbf{x}}$. Therefore it has no influence on the deformable model equilibrium position and is negligible compared to the force vector $\mathbf{F}$ which includes the damping force $-\gamma\dot{\mathbf{x}}$. That is why this term is not taken into account so that we get back the usual Newton's laws

$$m\ddot{\mathbf{x}} = -\gamma\dot{\mathbf{x}} + \mathbf{F_{int}} + \mathbf{F_{ext}} \qquad (6)$$

Experimentally, ignoring the additional term does not induce any visible change on the model behavior and significantly reduces computational costs. Equation (6) is classically discretized in time using the finite difference scheme, which allows to iteratively approach the model rest position.

4. Riemannian metrics

This section gives an intuitive geometrical interpretation of metric changes. Then it recalls the definition and properties of structure tensor and explains how this tool is used to build metrics from images.

4.1. Geometrical interpretation

This section shows that changing metrics is equivalent to locally expands or contracts the space along directions and with ratios determined by the local metric eigenstructure. This is seen by oberving that small balls are deformed into ellipsoids when the Eulidean metric is replaced by a Riemannian metric.

Consider the Riemannian ball $B_R(0, \epsilon)$. For small enough values of ϵ , $B_R(0, \epsilon)$ is contained in a neighborhood over which G may be considered as constant. Thus the following approximation holds for any point P in $B_R(0, \epsilon)$:

$$\|\mathbf{OP}\|_R^2 = {}^T\mathbf{OP} \times G \times \mathbf{OP} \simeq d_R^2(O, P) = \epsilon^2. \qquad (7)$$

If (x_0, x_1, x_2) denote the coordinates of P in an orthonormal basis $(\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2})$ of eigenvectors of G, equation (7) is rewritten as

$$\mu_0 x_0^2 + \mu_1 x_1^2 + \mu_2 x_2^2 = \epsilon^2$$

where μ_0, μ_1 and μ_2 denote the eigenvalues of G. This is the equation of an ellipsoid oriented along $\mathbf{v_0}$, $\mathbf{v_1}$ and $\mathbf{v_2}$ and with the length of its semi axes given as $\epsilon/\sqrt{\mu_0}$, $\epsilon/\sqrt{\mu_1}$ and $\epsilon/\sqrt{\mu_2}$.

As a consequence, changing the Euclidean metric with a Riemannian metric is equivalent to locally expand or shrink the space along the local eigendirections of the new metric. Expansion ratio are given as $1/\sqrt{\mu_0}$ along $\mathbf{v_0}$, $1/\sqrt{\mu_1}$ along $\mathbf{v_1}$ and $1/\sqrt{\mu_2}$ along $\mathbf{v_2}$.

4.2. Metric choice

As explained in Sect 3.1 a regular sampling of the model mesh is maintained with the deformed metric. In addition, this metric locally expand or contract the space according

to its eigenstructure. As a result resolution is locally multiplied by $\sqrt{\mu_0}$, $\sqrt{\mu_1}$ or $\sqrt{\mu_2}$ depending on the relative orientation of the mesh and the vectors $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$.

Classically, deformable models detect high norms of the image gradient to track object boundaries. In the following these places are called *contours* and are used to determine how metrics are built.

In a place with weak contours, resolution should remain coarse. This is achieved by choosing

$$\mu_0 \simeq \mu_1 \simeq \mu_2 \simeq \mu_{min}$$

where μ_{min} denotes the minimum allowed metric eigenvalue. We choose $\mu_{min} = 1$. By this way the Euclidean length of the longest edge on the surface is directly given by δ. The mesh orientation has no importance in this context, the only constraint on $\mathbf{v_0}$, $\mathbf{v_1}$ and $\mathbf{v_2}$ is thus orthonormality.

In contrast, in the vicinity of contours the metric should reflect their geometrical properties. Let S and $\mathbf{N}$ denote the contour strength and normal. Let also $\mathbf{v_{K_1}}$ and $\mathbf{v_{K_2}}$ and K_1 and K_2 denote the contour principal directions and curvatures. With these notations the local metric eigendecomposition is chosen as

$$\begin{cases} \mathbf{v_0} = \mathbf{N} \text{ and } \mu_0 = \phi(S) \\ \mathbf{v_1} = \mathbf{v_{K_1}} \text{ and } \mu_1 = \psi(S, K_1) \\ \mathbf{v_2} = \mathbf{v_{K_2}} \text{ and } \mu_2 = \psi(S, K_2) \end{cases}$$

where ϕ and ψ denote increasing functions of their arguments. With these choices, resolution is adapted according to the local geometry of object boundaries, the relative positions of the deformable surface and the image contour, and the confidence we have in this contour (*i.e.* its strength).

When the model crosses over an image contour, the vertex density on its mesh increases. The maximum resolution is determined by μ_0 and is reached when the deformable surface is tangent to $\mathbf{N}$ (*i.e.* when it is orthogonal to image component boundaries). The model is thus given more degrees of freedom to adapt its geometry and fit the contour. In addition, the thickness of thin flat objects is overestimated. This prevents collisions from being detected between parts of the model located on both sides of these objects. As a result, thin surfaces are properly recovered while edges remain significantly longer than the surface thickness.

In contrast, on its rest position the model has the same geometry as the contour so that its principal directions match the metric eigendirections $\mathbf{v_1}$ and $\mathbf{v_2}$. As a result, the mesh resolution is determined by the curvatures K_1 and K_2: resolution remains coarse in both direction along plane contours ($K_1 \simeq K_2 \simeq 0$), increases in the direction of $\mathbf{v_1}$ only along sharp edges or tubular structures ($0 \simeq K_2 \ll K_1$), and increases in both directions in the vicinity of object corners ($0 \ll K_1 \simeq K_2$).

no contour	$\xi_0 \simeq \xi_1 \simeq \xi_2 \simeq 0$
flat contour	$0 \simeq \xi_1 \simeq \xi_2 \ll \xi_0$
sharp edges, tubular structures	$0 \simeq \xi_2 \ll \xi_0 \simeq \xi_1$
corner	$0 \ll \xi_0 \simeq \xi_1 \simeq \xi_2$

Table 1. Eigenvalues of the structure tensor in the vicinity of different image features

4.3. Structure tensor

As pointed out by Weijer *et al.* [14], the contour principal curvatures and directions are difficult to estimate directly from images. The *structure tensor* provides a estimation of local structure orientations, and is often used to build robust curvature estimators [11, 14].

The structure tensor is a positive-definite matrix $J_{\rho,\sigma}$ that is evaluted at each point of the image. It represents the mapping

$$\mathbf{v} \in \mathbb{R}^3 \longrightarrow {}^T\mathbf{v} \cdot J_{\rho,\sigma} \cdot \mathbf{v} = G_\rho * (\nabla_\sigma I \cdot \mathbf{v})^2,$$

where G_ρ and $\nabla_\sigma I$ respectively denote the Gaussian filter with standard deviation ρ and the gradient of the image I smoothed with the Gaussian filter G_σ. This mapping clearly characterizes the gradient orientation as well as its variation in a neighborhood.

Let $\xi_0 \geq \xi_1 \geq \xi_2$ denote the eigenvalues of $J_{\rho,\sigma}$ and let $\mathbf{w_0}$, $\mathbf{w_1}$ and $\mathbf{w_2}$ denote the corresponding eigenvectors. It is easily seen than the average gradient direction in a neighborhood, namely the normal to the contour, is given by $\mathbf{w_0}$. The eigenvectors $\mathbf{w_1}$ and $\mathbf{w_2}$ form a basis of the contour tangent plane and respectively characterize the directions of maximum and minimum curvature on the contour. All the eigenvalues are increasing functions of the contour strength and ξ_1 and ξ_2 also increase with the contour principal curvatures K_1 and K_2. This is summarized in Table 1.

With these properties, the structure tensor is conveniently used to compute metrics well suited to achieve adaptive resolution. Its eigenvectors are used directly to define the metric eigendirections. Its eigenvalues are rescaled and thresholded, so as to vary between $\mu_{min} = 1$ and μ_{max}. The parameter μ_{max} determines the allowed range of variation for the mesh resolution over the whole image.

The parameter σ required to compute image structure tensor determines how much the image is smoothed before estimating its gradient. It is thus chosen according to the noise level in the image. The parameter ρ determines the size of the window over which gradient variations are investigated. It should therefore correspond to the size of the structures to be detected in images.

5. Experimental results

In this section, we investigate and compare segmentation accuracy and efficiency of (i) our model, (ii) the Euclidean model [7] with a fine uniform resolution (iii) the Euclidean model with a coarse uniform resolution. From one model to another only the metric and the edge length (*i.e.* the parameter δ in (1)) are changed. The set of forces applied to the model vertices as well as their balancing are left strictly identical for the three models. The mesh resolutions of the Euclidean models are chosen so as to correspond to the minimal and maximal edge lengths achieved by our Riemannian model.

For both examples, initialization is performed manualy outside objects. The model vertices evolve under the action of the classical regularizing forces, a damping force and a inflation/deflation force that attracts vertices toward a given image isosurface. The model is considered to have reached its rest position when its average kinetic energy stabilizes (see Fig. 5 and Fig. 9).

In the first example (Fig. 4-5), the image is composed of a cube with a spherical cavity. The cavity is connected to the outside through a small hole in one of the cube faces. The object to recover is therefore characterized by large flat surfaces as well as sharp edges and corners. As shown on Fig. 4.a the image is spoiled with a strong additive Gaussian noise. In the second example (Fig. 7-9), the 3D image is a cranial CT scan. The object is characterized by a complex topology, large flat and thin structures (*e.g.* on the upper part of the skull) and sharp edges (*e.g.* arround the orbits).

As expected, the mesh of our model is refined in highly curved parts of object boundaries. In contrast, resolution remains coarse along flat parts of objects. By this way, fewer vertices are used and shape description is enhanced. Furthermore, along flat contours, the resolutions of both the coarse and the Riemannian model are approximately the same. Since the Riemannian metric overestimates distances in the direction normal to these contours, constraint (3) holds and object topology is preserved. In contrast, with the coarse model topology changes are performed and tunnels are spuriously created through thin flat structures (see for example the holes in all cube faces in Fig. 4.d).

Of course, changing the metric requires additional precomputations and makes distances computations more expensive. However, during the model evolution, the mesh is coarser and vertices are thus allowed to travel faster in the image. The number of iterations required to achieve convergence is therefore reduced (see Fig. 5 and Fig. 9). Furthermore, for a given segmentation quality, the required number of vertices is reduced. Therefore, the time required to compute the vertex new positions and to check the model topological consistency at each iteration is reduced. For a given segmentation accuracy, this results in significantly reduced computation times. This is summarized in Table 2 and Table 3.

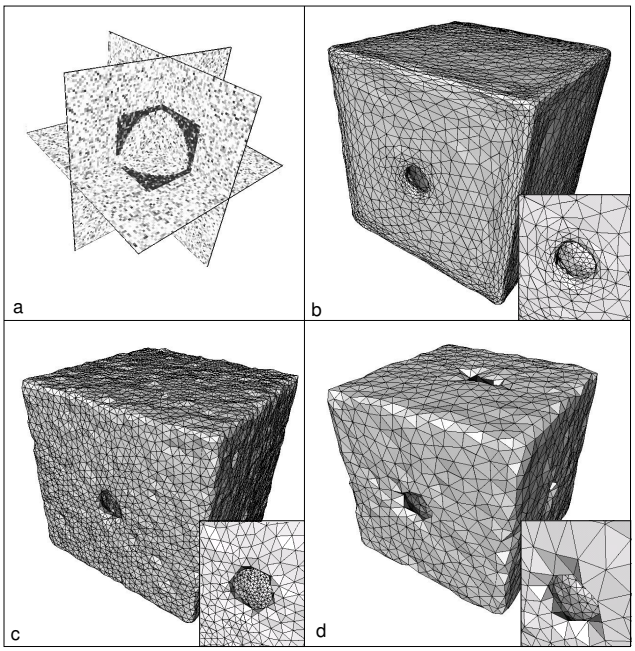


Figure 4. Results on a computer generated image. (a) three orthogonal slices of the 3D image, (b): segmentation with our model, (c) and (d) segmentation result with a euclidean metric and fine and coarse model resolutions.

6. Conclusion

We presented a highly deformable model that is able to recover objects with an arbitrary topology and adapts its resolution according to object geometry. Experimental results show that our model provides both enhanced segmentation quality and significantly reduced computation times.

Further works will focus on the model initialization through marching cubes like techniques. Indeed, these algorithms inherently produce meshes with a regular resolution. This of course results in heavy computational and memory costs. We expect that working with grids that are kept regular in a Riemannian space with an appropriate metric would allow to significantly reduce these costs.

References

[1] D. Adalsteinsson and J.A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

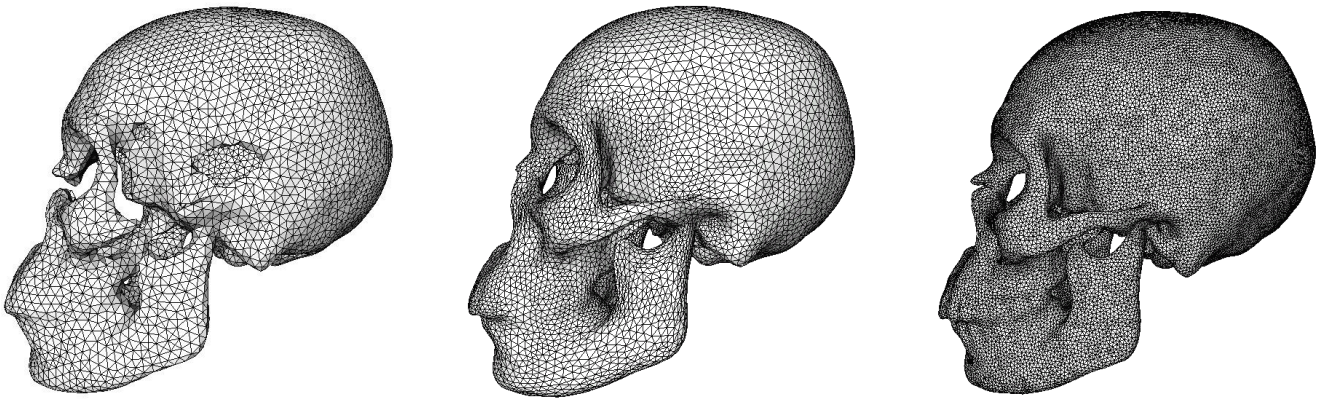


Figure 7. Segmentation of the medical image shown in Fig. 6. From left to right: segmentation results with a coarse Euclidean model, our Riemannian model and a fine Euclidean model. The coarse Euclidean model is unable to recover the correct object topology. The resolution of the Riemannian model keeps coarse along flat parts of the object while getting finer in its highly curved parts. In addition topology is recovered properly. With the fine Euclidean model, the segmentation quality is slightly increased but computational costs are far more important (see Fig. 9).

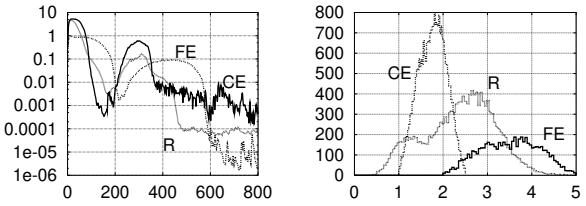


Figure 5. Results on the computer generated image shown in Fig. 4.a. Right: average vertex kinetic energy depending the iteration number. Convergence is obtained after approximately 275, 475 and 600 iterations for the coarse Euclidean model (CE), the Riemannian model (R) and the fine Euclidean model (FE) respectively. Right: histogram showing the repartition of the edges lengths of the models shown in Fig. 4.

	Computation time	Vertex number
Coarse	27”	1,751
Riemannian	3’20”	4,829
Fine	8’04”	8,499

Table 2. Convergence times and vertex numbers for the segmentation of the computer generated image (see Fig. 4). Computing the structure tensor and the metric used by the Riemannian model requires 19 seconds (image size is $100 \times 100 \times 100$).

[2] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66, 1993.

[3] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, mar 1991.

[4] H. Delingette. Simplex meshes: A general representation for 3d shape reconstruction. In *Proc. Computer Vision and Pattern Recognition*, pages 856–859, Los Alamitos, CA, USA, jun 1994. IEEE Computer Society Press.

[5] H. Delingette. General object reconstruction based on simplex meshes. Research Report 3111, INRIA, Sophia Antipolis, France, feb 1997.

[6] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):111–146, sep 1999.

[7] J.-O. Lachaud and A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine 3D surface extraction. *Medical Image Analysis*, 3(2):187–207, 1999.

[8] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modelling with front propagation: A level set ap-

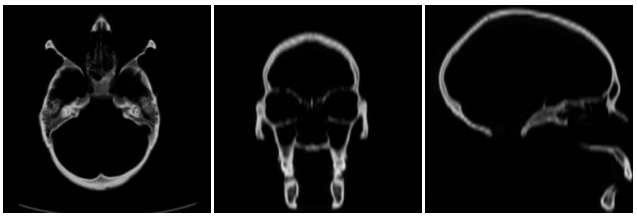


Figure 6. Slices extracted from medical data. Segmentation results are presented in Fig. 7.

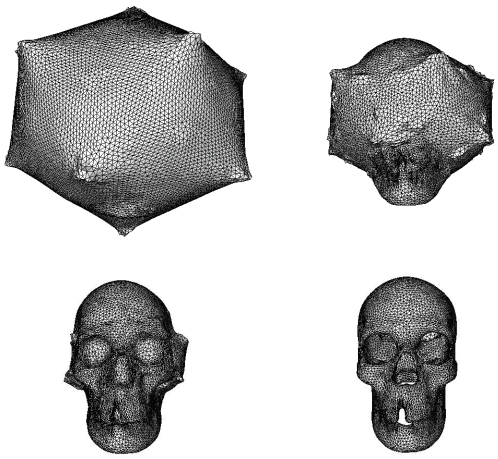


Figure 8. Segmentation of the medical image shown in Fig. 6 with the Riemannian model. Four steps of the model evolution. The model get refined only when it approaches interesting parts of the image.

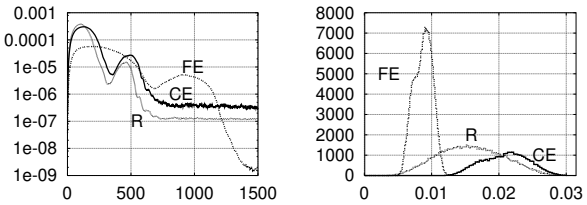


Figure 9. Results on the biomedical image shown in Fig. 6. Left: average kinetic energy of the vertices depending on the interation number. Coarse Euclidean model (CE) and Riemannian model (R) require both approximately 750 iterations to converge, while the fine Euclidean (FE) model only reaches its rest position after approximately 1,600 iterations. Right: Histogram showing the repartition of the edge lengths of the models shown in Fig. 7.

	Computation time	Vertex number
Coarse	13'23"	10,876
Riemannian	32'38"	19,438
Fine	151'26"	68,949

Table 3. Convergence times and vertex numbers for the segmentation of the head CT (Fig. 7). Structure tensor and metric computation over the image requires 59 seconds (image size is 148 $\times$ 148 $\times$ 148).

proach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2):158–174, feb 1995.

[9] T. McInerney and D. Terzopoulos. Medical image segmentation using topologically adaptable surfaces. In J. Troccaz, E. Grimson, and R. Mösges, editors, *Proc. of CVRMed-MRCAS*, volume 1205 of *LNCS*, pages 23–32, Grenoble, France, mar 1997. Springer-Verlag.

[10] J. Montagnat, H. Delingette, N. Scapel, and N. Ayache. Representation, shape, topology and evolution of deformable surfaces. application to 3d medical image segmentation. Technical Report RR-3954, INRIA, 2000.

[11] Bernd Rieger and Lucas J. van Vliet. Curvature of n-dimensionnal space curves in gray-value images. *IEEE Trans. on Image Processing*, 11(7):738–745, July 2002.

[12] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *Proc. CVPR'91*, pages 70–75, Lahaina, jun 1991. IEEE.

[13] M. Vasilescu and D. Terzopoulos. Adaptive meshes and shells. In *Proc. CVPR'92*, pages 829–832, Champain, jun 1992. IEEE.

[14] Joost von Weijer, Lucas J. van Vliet, and Michael van Ginkel. Curvature estimation in oriented patterns using curvilinear models applied to gradient vector fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9):1035–1043, September 2001.

[15] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. on Image Processing*, 7(3):359–369, mar 1998.