

# Computing approximate geodesics and minimal surfaces using watershed and graph-cuts

JEAN STAWIASKI<sup>1</sup>, ETIENNE DECENCIÈRE<sup>1</sup> and FRANÇOIS BIDAULT<sup>2</sup>

<sup>1</sup> *Centre de Morphologie Mathématique (CMM), Ecole des Mines de Paris, Fontainebleau, France*

*{Jean.Stawiaski, Etienne.Decenciere}@ensmp.fr*

<sup>2</sup> *Institut Gustave Roussy (IGR), Villejuif, France*  
*bidault@igr.fr*

**Abstract** Geodesics and minimal surfaces are widely used for medical image segmentation. At least two different approaches are used to compute such segmentations. First, geodesic active contours use differential geometry to compute optimal contours minimizing a given Riemannian metric. Second, Boykov and Kolmogorov have proposed a method based on integral geometry to compute similar contours using a graph representation of the image and combinatorial optimization. In this paper we present a technique to compute approximate geodesics and minimal surfaces using a low-level segmentation and graph-cuts optimization. Our approach speeds-up the computation of minimal surfaces when a low-level segmentation is available.

**Keywords:** watershed, minimal surfaces, geodesics, graph-cuts.

## 1. Introduction

Computation of minimal surfaces and geodesics is a common problem in image processing [1, 2, 4, 6]. Like many other techniques, the segmentation by optimal surfaces is a classical minimization problem. At least two different approaches of the problem have been successfully applied to image segmentation: geodesic active contours [6] and graph-cuts segmentation [4]. The methods will be briefly introduced in Section 2 and Section 3.

In this paper we propose a method to compute approximate geodesics and minimal surfaces by using the watershed low-level segmentation (watershed from all the local minima of the image's gradient). Our approach is motivated by the simplification it offers in the formalization of the problem. We propose to compute a curve (or a surface) that minimizes a given geometric functional in the space of curves (or surfaces) composed by a sub-set of watershed contours. The segmentation is driven by the search of a minimal cut in a region adjacency graph. Experimental results show

that the approximation error is negligible on natural images. Results will be presented on 3D medical images.

## 2. State of the art

### 2.1 Geodesic active contours

The first application of differential geometry in image segmentation has been introduced by Kass et al. in [9]. The method, called “snakes”, as well as many variants of active contours models, has been widely used for image segmentation. “Snakes” use a parametric representation of a curve. An important development has been introduced via a new representation of active contours [11, 13]. Parametric active contours have been replaced by an implicit representation of curves and surfaces via level-sets. This representation allows topological changes of curves and a better handling of numerical schemes to achieve the energy minimization. A further development of active contours has been introduced by Caselles et al. in [6] with “Geodesic active contours”. This method simplifies the energy function to be minimized. The problem is formalized as the minimization of the energy:

$$E(C) = \int_0^{|C|_\varepsilon} g(\|\nabla I(C(s))\|) ds, \quad (1)$$

where  $|C|_\varepsilon$  is the Euclidean length of a contour  $C$ , and  $s$  is the arc length on the contour.  $g$  is a positive and strictly decreasing function and  $\nabla$  is the gradient operator computed on the image  $I$ .

This method is equivalent to the minimization of the length of the curve  $C$  according to a Riemannian metric. The Riemannian metric depends here on the local gradient of the image  $I$ . For general curves, length in a Riemannian space can be written as:

$$|C|_R = \int_0^{|C|_\varepsilon} \sqrt{\tau_s^T D(C(s)) \tau_s} ds, \quad (2)$$

where  $\tau_s$  is a unit tangent vector to the contour  $C$  and  $D$  is a positive definite matrix, called the metric tensor, specifying the local Riemannian metric. In “Geodesic active contours” the local Riemannian metric is given by the following metric tensor:

$$D = \begin{pmatrix} g(\nabla I_x) & 0 \\ 0 & g(\nabla I_y) \end{pmatrix},$$

where  $(\nabla I_x, \nabla I_y)$  are the components of the gradient of  $I$ .

“Geodesic active contours” minimize the Equation 1 via a gradient descent scheme and a level-sets representation of the curve. Unfortunately, the method is sensitive to initialization and the global minimum of Equation 1 is not always found. However the method can also be extended to three dimensions [7].

## 2.2 Cauchy-Crofton formulae

Boykov and Kolmogorov [4] have considered the computation of minimal surfaces and geodesics based on the Cauchy-Crofton formulae of integral geometry. Cauchy established a formula which relates the length of a curve  $C$  to a measure of a set of lines intersecting it. Let  $L(\rho, \theta)$  be a straight line characterized in polar coordinates by the two parameters  $(\rho, \theta)$ . The Cauchy-Crofton formula establishes that the Euclidean length of a curve  $C$  is given by:

$$|C|_\varepsilon = \frac{1}{2} \int_0^\pi \int_{-\infty}^\infty N(\rho, \theta) d\rho d\theta, \quad (3)$$

where  $N(\rho, \theta)$  is the number of intersections of  $L(\rho, \theta)$  with  $C$ , and  $C$  is a regular curve. This formula can be extended to a Riemannian space, then the length of a curve  $C$  according to the metric tensor  $D$  is given by:

$$|C|_R = \frac{1}{2} \int_0^\pi \int_{-\infty}^\infty \frac{\det D}{2(u_L^T D u_L)^{3/2}} N(\rho, \theta) d\rho d\theta, \quad (4)$$

where  $u_L$  is the unit vector in the direction of line  $L$ . This formula is verified by any continuously differentiable and regular curve in  $\mathbb{R}^2$  [12].

Let  $N_g$  be a neighborhood system on a discrete grid.  $N_g$  can be described by a finite set of undirected vectors  $e_k$ ,  $N_g = \{e_k : 1 < k < n_g\}$ . Each vector  $e_k$  generates a family of lines as shown in Figure 1. Each line in a family is separated by a distance  $\Delta\rho_k$  from the closest line of the family. Now let  $\theta_k$  be a discrete angular parameter. For a fixed  $\theta_k$ , we obtain a family of parallel lines separated by a distance  $\Delta\rho_k$  as shown Figure 1.

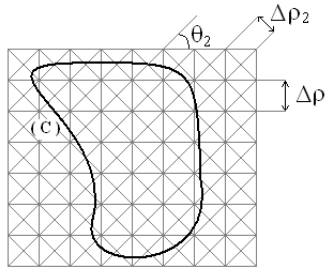


Figure 1. 8-Neighborhood system. Cauchy-Crofton formula established a link between a finite set of lines and the Euclidean length of a curve  $C$ .

The discretization of Equation 3 gives the following approximation of the Euclidean length of the curve  $C$ :

$$|C|_\varepsilon \approx \frac{1}{2} \sum_{k=1}^{n_g} \left( \sum_i n_c(i, k) \Delta\rho_k \right) \Delta\theta_k = \sum_{k=1}^{n_g} n_c(k) \frac{\delta^2 \Delta\theta_k}{2|e_k|}, \quad (5)$$

where  $i$  indexes the  $k^{th}$  family of lines.  $n_c(i, k)$  counts the number of intersections of line  $i$  of the  $k^{th}$  family of lines with the curve  $C$ .  $n_c(k) = \sum_i n_c(i, k)$  is the total number of intersections of the  $k^{th}$  family of lines with  $C$ .

### 3. Graph-cuts

#### 3.1 Basics

Graph-cuts are based on the well-known combinatorial problem of finding a minimal cost cut in a weighted graph. Suppose that each arc  $(i, j)$  of a graph  $G$  has assigned to it a non negative number  $c(i, j)$  called the capacity or the weight of the arc. This capacity is seen as the maximum amount of some commodity that can “flow” through the arc. Let us consider the problem of finding a maximal flow from a node  $s$ , called the source, to a node  $t$ , called the sink. Finding the maximal possible flow between  $s$  and  $t$  is related to finding a minimal cut in the graph. A (s-t) cut is identified by a pair  $(S, T)$  of complementary subsets of nodes, with  $s \in S$  and  $t \in T$ . The cost of the cut is defined by:

$$c(S, T) = \sum_{i \in S} \sum_{j \in T} c(i, j). \quad (6)$$

The minimal cut can be efficiently computed in polynomial time using classical combinatorial algorithm such as the Ford-Fulkerson algorithm or more efficient algorithms as the one proposed by Boykov et al. in [5]. Graph cuts are well suited for image segmentation since a node can represent a pixel and edges represent neighborhood relations between pixels. Graph-cuts have already been used in many imaging applications [3, 4, 10].

#### 3.2 Computing geodesics and minimal surfaces via graph-cuts

Boykov and Kolmogorov have considered the computation of minimal surfaces and geodesics with interactive placement of markers. The user has to specify “background” and “object” seeds and their method finds automatically the optimal curve (or surface) separating the two sets of seeds. The image is represented by a graph. Two additional nodes  $s$  and  $t$  are respectively connected to “object” seeds and “background” seeds. “s-links” and “t-links”, arcs connected to  $s$  or  $t$ , have infinite capacity to ensure that the sets  $S$  and  $T$  respectively contain a “background” seed and a “foreground” seed.

The aim of the method is to relate the cost of a graph-cut to the length of a underlying curve as shown in Figure 1. Let us consider an image embedded on a discrete grid and let  $N_g$  be a neighborhood system on the image. As

described in the previous section, the neighborhood system defines a family of lines. The cost of a (s-t) cut in the constructed graph is then equal to:

$$c(S, T) = \sum_{i \in S} \sum_{j \in T} c(i, j) = \sum_{k=1}^{n_g} n_c(k) w_k, \quad (7)$$

where  $n_c(k)$  is the number of arcs of family  $k$  that connect  $S$  to  $T$ , and  $w_k$  is the weight of the arcs of family  $k$ .

The Cauchy-Crofton formula given by Equation 5 can be directly used to set arcs weights such that the cost of a graph-cut approximates the Euclidean length of the contour separating the two sets  $S$  and  $T$ :

$$c(S, T) = \sum_{k=1}^{n_g} n_c(k) w_k \quad \text{with} \quad w_k = \frac{\delta^2 \Delta \theta_k}{2|e_k|}. \quad (8)$$

The previous relation can also be extended to deal with Riemannian metric by using the following weights:

$$w_k(p) = \frac{\delta^2 \cdot |e_k|^2 \cdot \Delta \theta_k \cdot \det(D(p))}{2(e_k^T D(p) e_k)^{3/2}}, \quad (9)$$

where  $w_k(p)$  is the weight of the arcs leaving the node  $p$ , and  $D(p)$  is the local Riemannian metric at point  $p$ . The previous expressions can also be extended to 3D spaces [4].

These formulae show explicitly that the cost of a graph-cut is related to the geometric length of the contour separating the sets  $S$  and  $T$  defined by the cut. Unfortunately existing methods are computationally costly and cannot always be used interactively on large datasets such as 3D medical images. Inspired by the approaches presented in Section 2 and Section 3, we propose a method to compute fast approximate geodesics and minimal surfaces from an initial low level segmentation of the image.

## 4. Approximate geodesics and minimal surfaces using watershed segmentation

### 4.1 Problem statement

The combination of graph-cuts with a watershed low-level segmentation (watershed from all the local minima of the image's gradient) provides us an explicit way to compute geodesics and minimal surfaces. Our basic assumption is that the geodesic to be computed is embedded in the watershed low-level segmentation. This proposition is motivated by two observations. Firstly, the watershed transform (computed from the local minima of image's gradient), without pre-processing or marker selection, produces an over-segmentation of real images. Secondly, the watershed lines contain all

major boundaries of real images. Thus, we propose to solve the following combinatorial problem: finding a curve composed of a finite union of watershed lines such that the curve minimizes a given geometric functional. We will solve this problem by using graph-cuts optimization on a region adjacency graph, as suggested by Li et al. [10].

## 4.2 Combining graph-cuts and watershed segmentation

Following [6], we will consider a geodesic curve  $C$  than can be computed via the minimization of the energy given by Equation 1. Let us consider the graph  $G = [X, U, W]$  of the watershed regions where  $X = \{x_k\}$  is the set of nodes (i.e the regions of the watershed transform),  $U$  is the set of arcs (i.e the neighborhood relations between regions) and  $W$  is the weights of the arcs as illustrated in Figure 2.

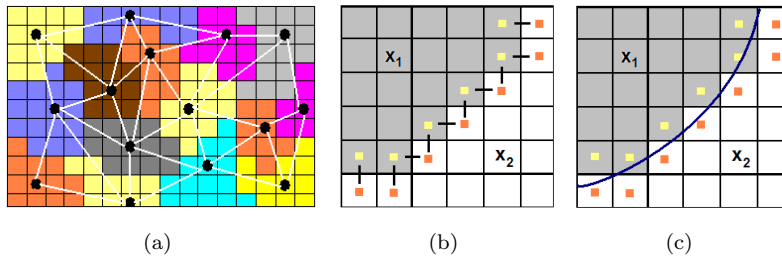


Figure 2. (a) Region adjacency graph of a low-level watershed segmentation. (b) The sets of pixels considered to compute boundary properties between adjacent regions, with a  $V_4$  adjacency relation. (c) An implicit curve defined by the regions  $x_1$  and  $x_2$ .

We present a way of defining arcs weight such that a cut partitions the image by an approximate minimal curve (curve of minimal length in a Riemannian space). Let us define  $F_{(x_i, x_j)}$  as the border between two regions  $x_i$  and  $x_j$  of the low-level watershed segmentation:

$$F_{(x_i, x_j)} = \{(p_m, p_n) \mid p_m \in x_i, p_n \in x_j, (p_m, p_n) \in N\}. \quad (10)$$

One should note that the set  $F_{(x_i, x_j)}$  depends on the adjacency relation  $N$ . This set of nodes implicitly describes a set of curves between the regions  $x_i$  and  $x_j$  as illustrated in Figure 2(c). Let us define  $C_{(x_i, x_j)}$  as the set of curves that can go through the nodes of  $F_{(x_i, x_j)}$ . Thus we can explicitly compute the energy  $E(C_{(x_i, x_j)})$  for all pairs of regions using the Cauchy-Crofton formulae detailed in Section 2. Note also that if regions  $x_i$  and  $x_j$  are not adjacent,  $F_{(x_i, x_j)}$  and  $C_{(x_i, x_j)}$  are empty sets.

Let us define a strictly positive function  $g$  of  $(F_{(x_i, x_j)})$  as:

$$g(F_{(x_i, x_j)}) = \sum_{(p_m, p_n) \in F_{(x_i, x_j)}} \frac{1}{1 + \|I(p_m) - I(p_n)\|^2}, \quad (11)$$

$$g(\emptyset) = 0. \quad (12)$$

The function  $g$  works as an edge indicator of the image  $I$  and takes a small value if the gradient of  $I$  is high between the regions  $x_i$  and  $x_j$ . One should note that the function  $g$  approximates the Riemannian length of the implicit curves  $C_{(x_i, x_j)}$  in case of the 4-neighborhood system. According to the Cauchy-Crofton Formulae, the number of adjacent pixels  $((p_m, p_n) \in F_{(x_i, x_j)})$  indicates the number of intersection of an implicit curve between the regions  $x_i$  and  $x_j$  with the horizontal and vertical lines describing the 4-neighborhood system:

$$|C_{(x_i, x_j)}|_R = E(C_{(x_i, x_j)}) \approx g(F_{(x_i, x_j)}). \quad (13)$$

Alternatively, the Cauchy-Crofton formulae can also be used to compute the approximate Riemannian length of the curves  $C_{(x_i, x_j)}$  between two adjacent regions in case of a larger neighborhood system. However in this section we will only consider the 4-neighborhood system for simplicity.

The cost of a (s-t) cut in the region adjacency graph weighted by the function  $g$  is equal to:

$$c(S, T) = \sum_{x_i \in S} \sum_{x_j \in T} w(x_i, x_j), \quad (14)$$

$$c(S, T) = \sum_{x_i \in S} \sum_{x_j \in T} (g(F_{(x_i, x_j)})). \quad (15)$$

As a consequence a (s-t) cut in the region adjacency graph is equal to the Riemannian length of an curve between the sets  $S$  and  $T$ . Considering the weighting function given by Equation 11, the minimal cut of the weighted adjacency graph of watershed regions is equal to:

$$\min_{(S, T)} c(S, T) = \min_{(C \in (\cup C_{(x_i, x_j)}))} E(C), \quad (16)$$

where  $(\cup C_{(x_i, x_j)})$  is the union of all the implicit curves defined by the watershed regions.

Our minimization problem is reduced to the search of a curve among all curves implicitly described by the watershed regions instead of searching among all curves in the domain of the image  $I$ . This approximation reduces drastically the search space.

### 4.3 Computing minimal surfaces

The method can easily be extended to three dimensions by considering integrals on surfaces instead of curves. The aim of the segmentation task is now to find a surface  $S$  that minimizes the following energy:

$$E(S) = \int \int_S g(\|\nabla I(x, y)\|) dx dy, \quad (17)$$

where  $S$  is a surface and  $g$  a positive and strictly decreasing function.

Thus we can use the same capacities defined in Equation 11, and apply it on the region adjacency graph in 3D:

$$w(x_i, x_j) = g(F_{(x_i, x_j)}). \quad (18)$$

In 3D,  $F_{(x_i, x_j)}$  defines implicitly a set of surfaces  $S_{(x_i, x_j)}$  between the regions  $x_i$  and  $x_j$ . Thus the minimal cut of the region adjacency graph in 3D is equal to:

$$\min_{(S, T)} c(S, T) = \min_{(S \in (\cup S_{(x_i, x_j)}))} E(S). \quad (19)$$

### 4.4 Adding geometric constraints

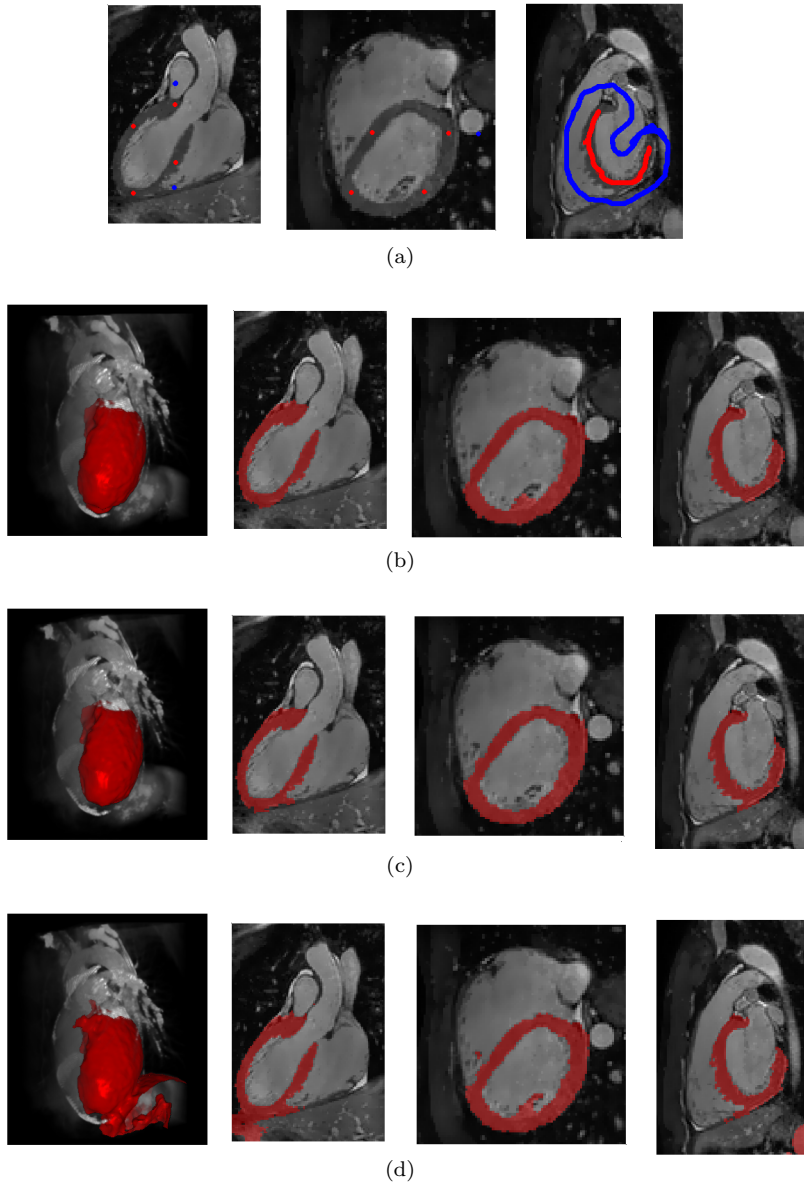
Using a region adjacency graph instead of the pixel adjacency graph can be advantageous in some situations. A wide class of geometric functionals can be computed on each regions of the watershed transform. As a consequence, a large class of geometric functionals can be added to the energy defined by Equation 1. For instance, it remains unclear how to introduce curvature constraints in the graph-cuts method used at the pixel level, but it is clear that a curvature term can easily be added in our methodology. For instance curvature of the border between two adjacent regions can be computed and used to add a shape constraint to the energy to be minimized.

## 5. Results

This section presents some results obtained by our method on 3D medical images. Figure 3 illustrates our segmentation method (Figure 3(b)) and compares it with the classical marker-controlled watershed segmentation (Figure 3(d)) and the minimal surface computed with the technique proposed by Boykov et al. in [4] (Figure 3(c)). Our method outperforms the marker-controlled segmentation and produces approximately the same segmentation as the graph-cuts method proposed by Boykov et al. in [4].

The next example illustrates the method on a 3D CT image. Figure 4 illustrates the segmentation of a liver. The liver presents low-contrasted boundaries and the segmentation of such organs remains a difficult task.





*Figure 3.* (a) Heart MRI superposed with user-provided markers. (b) Approximate minimal surface by our method. (c) Graph cuts minimal surface. (d) Marker-controlled watershed segmentation.

For this application minimal surface remains a leading method. The graph-cuts used on the watershed adjacency graph (Figure 4(b)) improves the results given by the classical marker-controlled watershed segmentation (Figure 4(d)) and speeds up the graph-cuts method proposed by Boykov et al. [4].

Extensive tests have been undertaken with 3D medical images provided by the Institut Gustave Roussy<sup>1</sup> (mainly CT images of the thorax) and the Centre for Advanced Visualization and Interaction<sup>2</sup> (MRI images of the heart). According to specialists, the results are very promising, even for the interactive segmentation of anatomical structures which are difficult to contour, such as the liver.

Table 1 illustrates the computation time needed by the three methods we considered: the marker-controlled watershed, our method, and the minimal surfaces by graph-cuts proposed in [4]. The results shows that our method reduces drastically the computation time needed by graph based techniques. Moreover our method do not seem to affect the results quality.

*Table 1.* Comparisons of computation time. (Laptop Pentium Core Duo 2.16 Ghz, 1Go memory)

<i>Image</i>	<i>Watershed</i>	<i>Our method</i>	<i>Graph cuts</i>
Heart MRI	1,5 sec.	4,2 sec.	15,2 sec.
Liver CT	9,7 sec.	41,6 sec.	1400,2 sec.

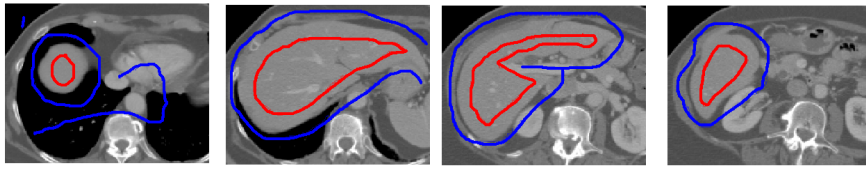
## 5.1 Conclusion

Considering that the watershed transform contains all major boundaries in real images, our approximate segmentation is in practise quite efficient. However the graph-cuts approach works slower than the classical marker-controlled watershed but offers more stable results. In the other hand our method is not as precise as the graph-cuts method proposed by Boykov et al. [4], but it offers a good trade off between speed and precision.

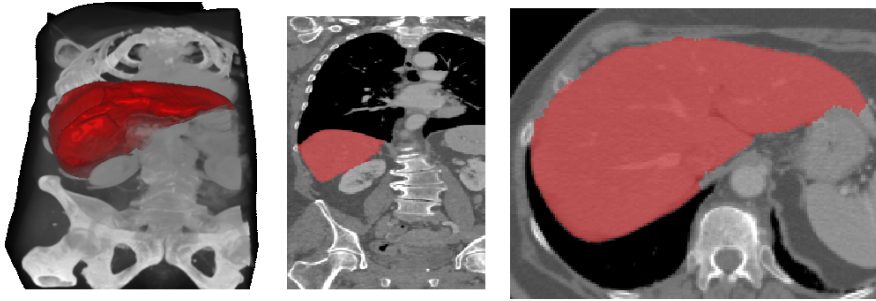
A Graph-cuts approach cannot always be used on large images when the graph considered is the pixel adjacency graph because of the memory requirements and the computational complexity of the method. The developed method can efficiently be used on large images considering the region adjacency graph instead of the pixels graph. Our method do not seem to introduce large biases in the resulting segmentation of natural images. Limitations of our methods are quite clear since it can only be used when an

<sup>1</sup>The Institut Gustave-Roussy is a non-profit private institution, exclusively devoted to oncology, located near Paris, France

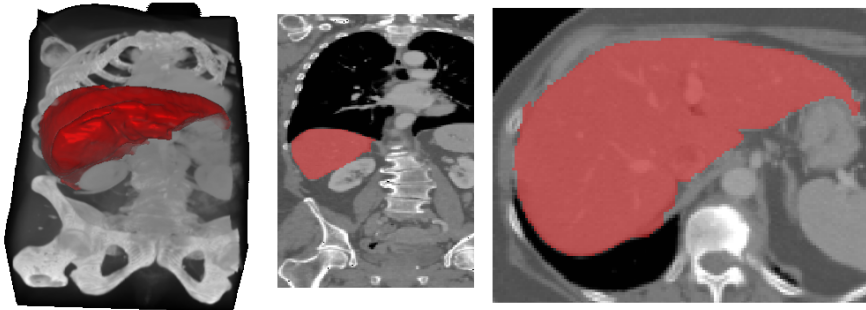
<sup>2</sup>University of Aarhus, Denmark



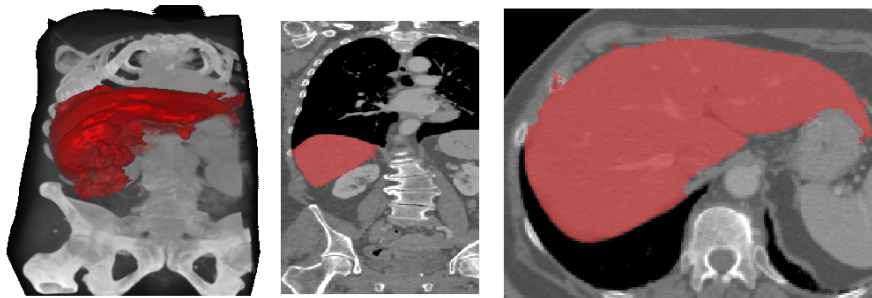
(a)



(b)



(c)



(d)

*Figure 4.* 3D CT image . (a) User-provided markers. (b) Approximate minimal surface by our method. (c) Graph-cuts minimal surface. (d) Marker-controlled watershed segmentation.

over-segmentation can be obtained. Graph-cuts can also be used on other adjacency graphs. For instance  $\lambda$ -flat zones [8] adjacency graph should be a good solution to increase the precision of our method since it can offer a pixel-precision in some situations.

Our methodology can take into account a wide class of geometric functionals since we can compute all kind of measures on the boundaries of the watershed regions. For instance this method can take into account curvature of the boundaries, which remains impracticable for classical graph-cuts methods used at the pixel level.

## 6. Acknowledgements

We would like to thank Région Ile-de-France and the Cancéropôle Ile-de-France for funding our research on medical image segmentation.

## References

- [1] B. Appleton and H. Talbot, *Globally optimal geodesic active contours*, JMIV **23** (2005), 67–86.
- [2] ———, *Globally Minimal Surfaces by Continuous Maximal Flows*, PAMI **28**(1) (2006), 106–118.
- [3] Y. Boykov and M. P. Jolly, *Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images*, ICCV (2001), 105–112.
- [4] Y. Boykov and V. Kolmogorov, *Computing Geodesics and Minimal Surfaces via Graph Cuts*, ICCV (2003), 26–33.
- [5] ———, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), 1124–1137.
- [6] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic Active Contours*, International Journal of Computer Vision (1995), 694–699.
- [7] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, *Minimal surfaces: A three dimensional segmentation approach*, Technion EE Pub. 973, (1995).
- [8] J. Crespo, R. Schafer, J. Serra, C. Gratin, and F. Meyer, *The flat zone approach : a general low-level region merging segmentation method*, Signal Processing **62** (1997), 37–60.
- [9] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, International Journal of Computer Vision **1**(4) (1988), 321–331.
- [10] Y. Li, J. Sun, C. Tang, and H. Shum, *Lazy snapping*, IGGRAPH, ACM Transaction on Graphics **23** (2004), 303–308.
- [11] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, Springer, 2003.
- [12] L. A. Santalo and R. Fedkiw, *Integral Geometry and Geometric Probability*, Addison-Wesley, 1979.
- [13] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press.