Image Anal Stereol 2008;27:39-45
Original Research Paper

REGION MERGING VIA GRAPH-CUTS

JEAN STAWIASKI AND ETIENNE DECENCIÈRE

Ecole des Mines de Paris, Centre de Morphologie Mathématique
e-mail: Jean.Stawiaski@cmm.ensmp.fr, Etienne.Decenciere@cmm.ensmp.fr

(Accepted November 12, 2007)

ABSTRACT

In this paper, we discuss the use of graph-cuts to merge the regions of the watershed transform optimally. Watershed is a simple, intuitive and efficient way of segmenting an image. Unfortunately it presents a few limitations such as over-segmentation and poor detection of low boundaries. Our segmentation process merges regions of the watershed over-segmentation by minimizing a specific criterion using graph-cuts optimization. Two methods will be introduced in this paper. The first is based on regions histogram and dissimilarity measures between adjacent regions. The second method deals with efficient approximation of minimal surfaces and geodesics. Experimental results show that these techniques can efficiently be used for large images segmentation when a pre-computed low level segmentation is available. We will present these methods in the context of interactive medical image segmentation.

Keywords: graph-cuts, region merging, watershed transform.

INTRODUCTION

The watershed transform (Beucher and Lantuejoul, 1979) is one of the most popular segmentation methods. It is a fundamental concept developed in the context of mathematical morphology. Interactive watershed segmentation is widely used for medical image segmentation. It offers a fast and stable way to segment complex structures. Unfortunately, the watershed suffers from a few limitations including sensitivity to noise, poor detection of low boundaries and over-segmentation. We will show in this paper that the watershed segmentation can be used for a pre-segmentation step.

In the last few years, since publication by Greig *et al.* (1989) and Boykov *et al.* (1998), *s-t* graph-cuts have become well established as a leading method for image segmentation and restoration. Graph-cuts is basically an energy minimization technique based on combinatorial optimization. It has many interesting characteristics such as global minimization, solid theoretical background and flexibility in the energy function. Although this approach is feasible for many applications, it suffers from an excessive computational cost and large memory requirements. However a few improvements have been proposed to reduce the computational cost of the max-flow algorithm. An efficient algorithm was proposed by Boykov and Kolmogorov (2004) and a narrow-band like method has been developed by Lombaert *et al.* (2005).

On the other side, authors have combined graph based segmentation with the watershed transform by using spectral approaches (O'Callaghan and Bull, 2005), minimal cuts (Li *et al.*, 2004) and minimum spanning trees (Meyer and Beucher, 1990; Meyer, 1994). As shown by Li *et al.*, combining the watershed transform and graph-cut partitioning can speed-up the classical graph-cut segmentation and reduce memory requirements. In fact graph based techniques as random walker, isoperimetric partitioning, normalized cuts (O'Callaghan and Bull, 2005) and minimal cuts (Li *et al.*, 2004), can efficiently be used with a low level segmentation. We will use the numerous regions obtained by the watershed transform to build a binary segmentation using graph-cuts optimization. In this paper we will primarily consider the binary segmentation of an image with interactive placement of markers.

METHODS

IMAGE SEGMENTATION WITH GRAPH-CUTS

Since the first application by Greig *et al.* (1989) and thanks to the work of Boykov, Kolmogorov, Zabih, Veksler and Jolly (Boykov *et al.*, 1998; Boykov and Jolly, 2001; Boykov *et al.*, 2001; Boykov and Kolmogorov, 2003; Kolmogorov and Zabih, 2004), (*s-t*) graph-cuts have been widely used for image segmentation, restoration and many other applications. Because image segmentation is frequently viewed as an energy minimization problem, (*s-t*) minimal cuts can natively offer a solid theoretical background for image segmentation. Many energy functions can be minimized via graph-cuts as soon as the problem can be formalized as a network flow problem.

Images are seen as non-oriented graphs, the nodes correspond to the pixels and the arcs to the adjacency relations between pixels. The construction of such graphs is fully described in Kolmogorov and Zabih (2004). As shown in Fig. 1, the nodes of the graph are the pixels of the image with two supplementary nodes *s*, the source, and *t*, the sink. The graph arcs are composed of :

- "*s*-links": arcs connecting a node *i* to the source *s*,
- "*t*-links": arcs connecting a node *i* to the sink *t*,
- "*n*-links": arcs connecting neighbor pixels *i* and *j*.

A (*s*-*t*) cut produces a binary partition $\{S,T\}$ of the set of nodes, such that the source *s* belongs to *S* and the sink *t* belongs to T. In the next sections we will name the set *S* the foreground and the set T the background. The value of the cut is:

$$c(S,T) = \sum_{i\in S, j\in T} c(i,j) \ , \qquad (1)$$

where $c(i,j)$ is the capacity of the arc connecting nodes *i* and *j*.

A minimal cut can be computed in polynomial time via algorithms based on network flow such as the one proposed by Ford and Fulkerson (1955) or more recently by Boykov and Kolmogorov (2004).

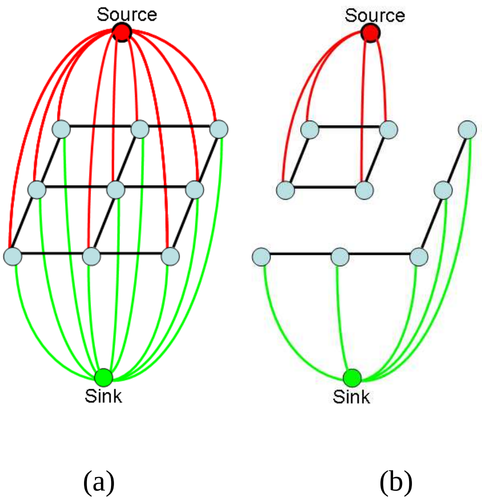


Fig. 1. *A Graph and a (s-t) cut. (a) Graph construction for a 3×3 image. (b) Arcs of the (s-t) cut have been removed from the graph. Note that a cut severs always the "t-link" or the "s-link" of each pixel. The graph-cut partitions the nodes into two different sets S and T, one containing the source and the other containing the sink.*

ARC CAPACITIES

Arc capacities drive the properties of the segmentation. The capacity $c(i,j)$ of arcs connecting adjacent pixels plays the role of an edge indicator. The term may be based on any edge indicator function: gradient, Laplacian, etc. $c(i,j)$ should be large when nodes *i* and *j* have similar gray values.

The terms $c(s,i)$ and $c(i,t)$ are the "*t*-links" and "*s*-links" capacities. Contrary to the term $c(i,j)$, $c(s,i)$ and $c(i,t)$ are related to regional properties of the segmentation. For instance $c(s,i)$ can reflect how a pixel *i* fits into a given intensity profile of the foreground. Usually, prior information has to be known to define a regional term. Some properties of the background or the foreground have to be extracted before defining the arc capacities.

For some applications, it is useful to consider an interactive segmentation, where the user hard-constraints some pixels to be in the background *T* or in the foreground *S* (Boykov and Jolly, 2001). Graph construction and arc capacities are, in this case, slightly different. These hard-constraints result in a modification of arc weights. Nodes marked as foreground and background have respectively "*s*-links" and "*t*-links" of infinite capacity so that these arcs cannot be in the minimal cut.

COMBINING (*S*-*T*) GRAPH-CUTS AND THE WATERSHED TRANSFORM

The Watershed Transform (Beucher and Lantuejoul, 1979) is a well-known segmentation method. The result of the transform is a partition of the image. It uses an intuitive description of boundaries in an image: considering an image as a topographic surface where the height of each point is directly related to its gray level, it simulates the flooding of the surface from a finite set of points (local minima of the image or any set of markers). In order to avoid the merging of water that comes out of different sources, a watershed line is constructed. Computed on the gradient of an image, the watershed transform tends to give the high gradient points which are related to boundaries of the image.

The watershed transform partitions the image into numerous regions depending on the number of local minima of the gradient. When the flooding starts from all gradient minima, the watershed tends to produce an over-segmentation of the image (see Fig. 3c). The use of markers avoids this problem, but the watershed might still be prone to leaks (see the leak on the front of the cameraman in Fig. 3d).

These regions can be represented by an adjacency graph. The watershed adjacency graph has been used in many segmentation applications including hierarchical segmentation (Meyer, 1994), spectral graph partitioning (O'Callaghan and Bull, 2005), graph-cuts segmentation (Li *et al.*, 2004). Following the approach of Li *et al.*, we will study the application of (*s*-*t*) graph-cuts to the graph produced by the watershed transform. The adjacency graph can be efficiently computed during the flooding process of the watershed algorithm. Moreover the flooding process can be used to compute different measures on regions such as histograms, surfaces, and volumes.

The graph we consider in our application is the non-oriented adjacency graph of the watershed regions with two additional nodes: the source and the sink, as illustrated in Fig. 2. The result of the graph-cut is then two subsets (S,T) of regions of the watershed transform. The graph-cuts partitioning can be seen here as a region merging process.

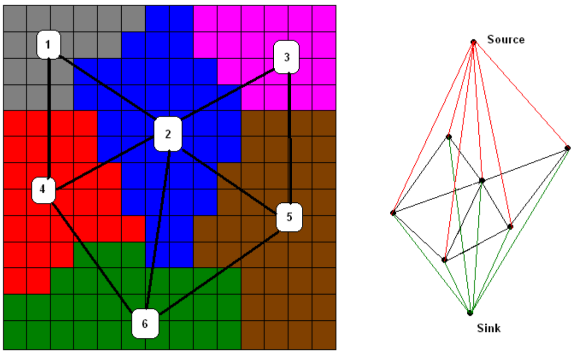
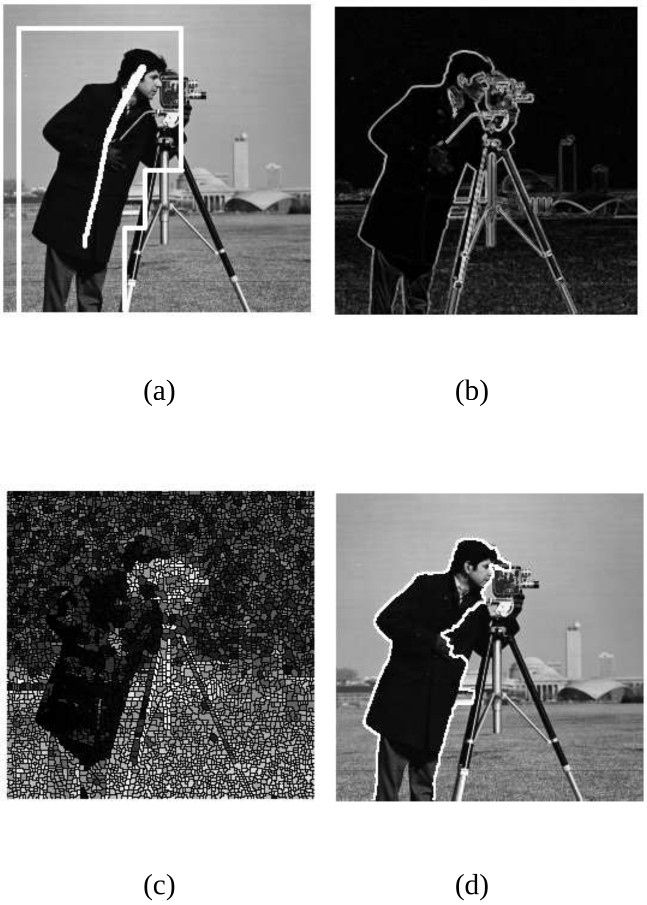


Fig. 2. *Adjacency graph on regions and the corresponding (s-t) graph.*

REGION BASED MERGING

We will now introduce a region merging process, based on graph-cuts, that takes into account dissimilarities between regions. Many energy functions are related to measures on regions, for instance, length of the region boundaries, histograms, areas, volumes, etc. The aim of this section is to define arc capacities of the adjacency graph so that a cut merges regions of similar characteristics. In this section we will consider the gray levels of an image and the watershed transform of its gradient image. We will then consider a region merging process using markers provided by the user as illustrated in Fig. 3.


(a) (b)

(c) (d)

Fig. 3. *Segmentation process: a) Markers Specified by the user superposed with the original Image. b) Gradient of the original image. c) Watershed of the gradient. d) Marker-controlled watershed.*

"*s*-links" and "*t*-links" capacities

"*s*-links" and "*t*-links" capacities reflect how a node of the graph fits into one of the two sets (S,T). For instance we can define the following arcs capacities:

$$\begin{cases} c(s,i) = \frac{d^F}{d^F+d^B} & \forall i \in I\,, \\ c(i,t) = \frac{d^B}{d^F+d^B} & \forall i \in I\,, \end{cases} \tag{2}$$

where $d^F = |m_F - m(i)|$, and $d^B = |m_B - m(i)|$. m_F and m_B are the average gray levels of the regions marked respectively as "foreground" and "background", $m(i)$ is the average gray level of the region i. One should note that m_F and m_B should be different to prevent any numerical problems.

Different energies can be used to measure the dissimilarity between regions. Previously, we have only considered the average gray value of the marked region. It is also possible to use measures based on the histogram of gray values in each region. We will discuss the use of such measures in the next section.

“*n*-links” capacities

The capacities of the graph edges are directly related to the dissimilarity between regions. Researchers dealing with region merging have proposed a large number of ways to evaluate this dissimilarity. For instance we can define a capacity of the general form (Boykov and Jolly, 2001):

$$c(i,j) = \exp\left(-\frac{d(i,j)^2}{2\sigma^2}\right) \quad \{i,j\} \in N \,, \tag{3}$$

where $d(i,j)$ is a dissimilarity measure between adjacent regions i and j, σ is a free parameter. N is the set of neighbor nodes different from s and t.

Different dissimilarity indices between regions can be computed, the most widely used being distances between histograms of gray levels in each regions. Let h_i be the histogram of the region i (in the original image), h_i^k is the k^{th} bin of h_i, then we can define:

$$\begin{aligned} c(i,j) &= \exp\left(-\frac{d(i,j)^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{(\sum_k |L_i^k - L_j^k|)^2}{2\sigma^2}\right) \,, \end{aligned} \tag{4}$$

where $L_i^k = \sum_{j<=k} h_i^j$ is the cumulative histogram of the region i.

This measure has better perceptual properties than simple bin by bin comparisons, since it incorporates a notion of ground distance (O’Callaghan and Bull, 2005). However, finding a satisfying dissimilarity measure between regions is still an open problem and is often application dependant.

This kind of measure is only related to a global measure in each region. The aim of the method is to merge regions that have similar histograms. The method will be illustrated in the last section of this paper dedicated to experimental results. The energy function we minimize with graph-cuts does not take into account image’s edges information such as gradient. In the next section we will introduce a method that takes into account only the boundary values of each region.

BOUNDARY BASED MERGING

Geodesics and minimal surfaces are widely used in medical image segmentation. Two approaches can be distinguished to compute such segmentations. Geodesic active contours (Caselles *et al.*, 1997) use differential geometry to compute an optimal surface minimizing a given Riemannian metric. Riemannian geometry is a non-Euclidean geometry that studies local properties of a smooth manifold. In this kind of spaces the metric varies smoothly from point to point. For instance a continuous and smooth image can be considered as a Riemannian space (a smooth surface) whose metric depends on the local gradient. The aim of Geodesic Active Contours is to compute a curve of minimal length in a Riemannian space.

On the other side Boykov and Kolmogorov (2003) have proposed a method based on integral geometry to compute minimal surfaces and geodesics by using graph-cuts optimization. Authors proposed to use the Cauchy-Crofton formulaes to compute the length of a curve under a Riemannian metric. From these formulaes they derived a mean to compute arc capacities so that a cut in the pixel adjacency graph approximates a given riemannian metric.

In this paper we present a technique to compute approximate geodesics and minimal surfaces using the watershed over-segmentation and graph-cuts optimization. This method includes a simplification of the existing methods and a possible speed-up of the computation. Our approach of combining graph-cuts and watershed segmentation allows to have a more simple way to compute geodesics and minimal surfaces. We use the heuristic that the geodesic or the minimal surface to be computed should be built by a finite union of watershed lines or surfaces. This proposition is motivated by two observations. First the watershed transform produces an over-segmentation of real images. Secondly, the watershed lines or surfaces are composed by all important boundaries of the image. We propose to solve the following combinatorial problem: Finding a curve (or a surface) composed of a finite union of watershed lines (or surfaces) so that the curve minimizes a given Riemannian metric. We solve this problem using graph-cuts optimization on the region adjacency graph.

Following Caselles *et al.* (1997), a geodesic curve C can be computed via the minimization of the energy:

$$E(C) = \int_0^{|C|_\varepsilon} g\left(\nabla I(C(s))\right) ds \,, \tag{5}$$

where $|C|_\varepsilon$ is the Euclidean length of the curve C and s is the curvilinear coordinate on the curve C. g is a positive decreasing function, for instance :

$$g(\nabla I) = \frac{1}{1+|\nabla(G_\rho * I)|^p} \quad p \geq 1 \,, \tag{6}$$

where G_ρ is a gaussian kernel of variance ρ. Finally, The energy $E(C)$ is obtained by weighting the Euclidean element of length ds by an indicator g of image’s edges. This energy can be minimized via a gradient descent approach (Caselles *et al.*, 1997) or via graph-cuts (Boykov and Kolmogorov, 2003).

The energy $E(C)$ can be minimized directly via graph-cuts segmentation based on the watershed regions. Using the watershed adjacency graph allows us to use a simple expression for arc capacities so that a cut on the graph approximates $E(C)$. The capacity of the arc connecting the regions i and j have the following expression :

$$c(i,j) = \int_{F_{i,j}} g(\nabla I(F_{i,j}))ds \qquad (7)$$

$$c(i,j) = \int_{F_{i,j}} \frac{1}{1+|\nabla(G_\rho * I(F_{i,j}))|^p} ds \qquad (8)$$

where $F_{i,j}$ is the common border between regions i and j.

Considering this weighting function, the minimal cut of the adjacency graph of the watershed regions is finally equivalent to an approximation of the geodesic minimizing $E(C)$. The minimal cut of the graph is the solution of the minimization of:

$$E(C2) = \sum_{i\in F, j\in B} c(i,j) = \int_{\cup F_{i,j}} g(\nabla I(F_{i,j}))\, ds\,, \qquad (9)$$

where $\cup F_{i,j}$ is a curve formed by an union of watershed lines. We only consider the capacities of "*n*-links" in our graph. "*s*-links" and "*t*-links" of the graph have a infinite capacity if corresponding nodes are marked respectively as foreground and background. Otherwise arc capacities are equal to zero. This ensures that the segmentation have at most the same number of connected components than the markers.

The minimal cut produces two regions whose boundaries minimize the energy $E(C2)$. The initial combinatorial problem is a choice of a curve among all possible curves in the image, the problem we solved is reduced to a choice of a curve among all curves formed by an union of watershed lines.

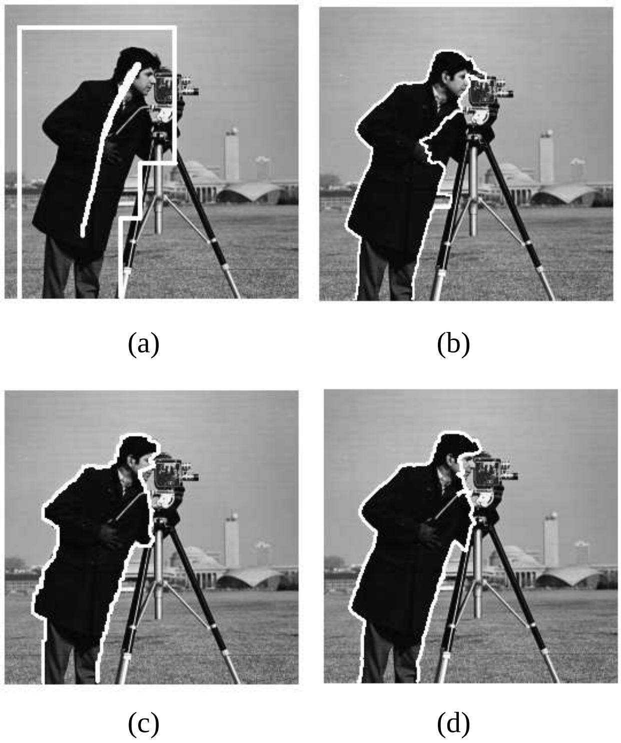
Considering that the watershed transform contains all major boundaries of the image, this approximation can be quite efficient. Moreover the search space of the initial problem increases exponentially with the size of the image. Using the heuristic that the geodesic or the minimal surface should be a union of watershed lines produces always a smaller search space. This method merges two regions if there is no high gradient points between each region. The aim of the method is to find two sets of regions so that the gradient between the two sets is maximal. This kind of energies is very useful to find a region which presents missing boundaries or low contrasted edges. On the other hand geodesics and minimal surfaces does not take into account gray values information in each region.

RESULTS

In this section $\{F,B\}$ denotes the sets of nodes marked respectively as foreground and background and N denotes an adjacency relation between nodes.

Fig. 4 illustrates the comparison between several marker-based segmentation methods: marker-controlled watershed, graph-cuts segmentation using capacities described in (Boykov and Jolly, 2001), and region merging via graph-cuts. The following capacities were used to compute the results:

$$\begin{cases} c(i,s) = 0 \quad \forall i \in I - \{F \cup B\} \\ c(i,t) = 0 \quad \forall i \in I - \{F \cup B\} \\ c(i,s) = \infty \quad \forall i \in \{F\} \\ c(i,t) = \infty \quad \forall i \in \{B\} \\ c(i,j) = exp(-\frac{(\sum_k |L_i^k - L_j^k|)^2}{2\sigma^2}), \quad \{i,j\} \in N \end{cases} \qquad (10)$$


(a) (b)

(c) (d)

Fig. 4. *Comparison between several segmentation methods. (a) Markers superposed with the original image. (b) Marker-controlled watershed segmentation. (c) Graph-cuts segmentation. (d) Region merging via graph-cuts.*

This kind of energy is only based on regional properties in each region. "*s*-links" and "*t*-links" have infinite capacities if the corresponding nodes have been marked as foreground or background, otherwise the corresponding capacities are equal to zero. This experimental result shows that the region merging based on graph-cuts offers a better result than the classical marker-controlled watershed. On the other hand the region merging approach offers quite similar results than the classical graph-cuts segmentation by using a region adjacency graph instead of the pixel adjacency graph. Moreover, using the region adjacency graph reduces the computational cost since the graph have less nodes and arcs than the pixel graph.

Figs. 5 and 6 show 3D examples of marker-based graph-cuts segmentation combined with the watershed transform using the same capacities as defined previously. These examples illustrate multiple binary segmentations of the same image using different sets of markers.

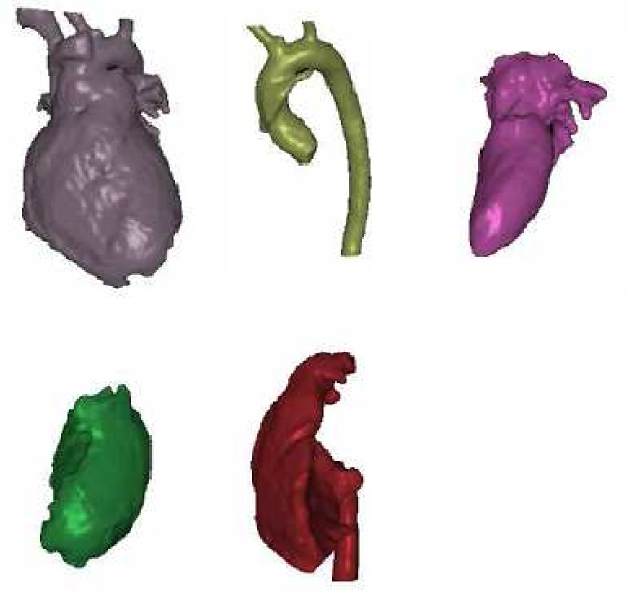


Fig. 5. *3D surface rendering of a segmented heart.*

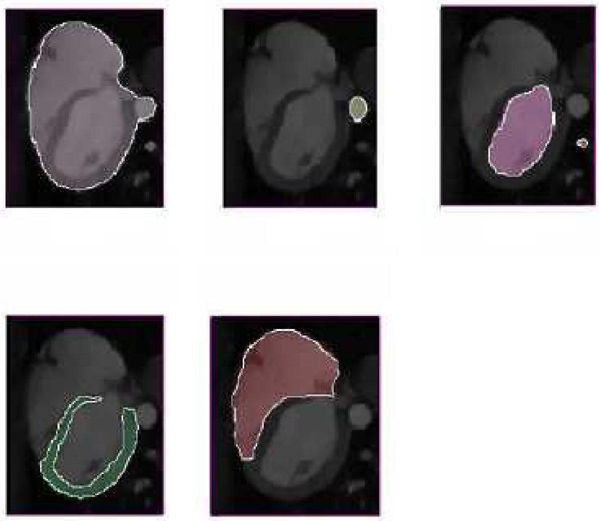


Fig. 6. *Axial cuts of the segmentation superposed with the original data.*

Fig. 7 shows an application of marker-based graph-cuts segmentation combined with the watershed transform using an objective function which takes into account only the regions boundaries.

$$\begin{cases} c(i,s) = 0 \quad \forall i \in I - \{F \cup B\} \\ c(i,t) = 0 \quad \forall i \in I - \{F \cup B\} \\ c(i,s) = \infty \quad \forall i \in \{F\} \\ c(i,t) = \infty \quad \forall i \in \{B\} \\ c(i,j) = \int_{F_{i,j}} g(\nabla I(F_{i,j}(s)))ds \quad \{i,j\} \in N \end{cases} \tag{11}$$

where g is a positive decreasing function (see Eq. 6) and ∇I is the gradient of image I. $F_{i,j}$ is the common boundary between adjacent regions i and j. This kind of energy is related to geodesics and minimal surfaces.

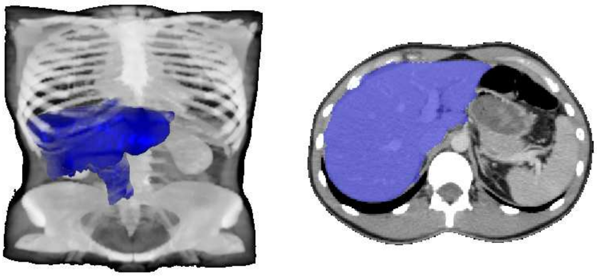


Fig. 7. *3D surface rendering of a segmented liver superposed with the original data.*

Table 1 summarizes the comparisons between the speed of our method with classical graph-cuts (method 2) (Boykov *et al.*, 2001) and the marker controlled watershed segmentation (method 1) (Beucher and Lantuejoul, 1979). For graph based methods, Table 1 contains the time needed to construct the graph, to compute the minimal cut and finally to build the output image. The results were computed on a laptop with Intel Core Duo 2.16 GHz processor and 1 GB of memory. This limited platform does not allow to use the classical graph cut (Boykov and Jolly, 2001) on large datasets such as the thoracic CT illustrated in Fig. 7. This limitation is only due to memory limitation of the platform.

DISCUSSION

Experimental results show that our region merging process is stable. Moreover the approximation done using the region graph instead of the pixel graph is negligible when the watershed transform produces a heavy over-segmentation. When this assumption is not verified, for instance when some important edges have not been detected by the watershed transform, the two methods produce different results. Using the

Table 1. *Comparisons of computation time.*

Image	Size	Method 1	Method 2	Our method
Fig. 4	256×256	0.08 s	0.5 s	0.09 s
Figs. 5-6	$128 \times 85 \times 90$	1.60 s	14.8 s	2.39 s
Fig. 7	$255 \times 255 \times 128$	14.16 s	—	152.3 s

adjacency graph of the watershed regions instead of the pixel adjacency graph can also reduce the time to compute the minimal cut since it offers in experiments an important reduction of the number of nodes and arcs of the graph.

CONCLUSION

The region merging process tries to combine advantages from the watershed transform and the graph-cuts partitioning. The combination of graph cuts and watershed offers an improvement in terms of computational cost and memory requirements of the graph-cut based at the pixel level.

We are now working on a better integration of morphological and graph based segmentation. We would like to combine the two methods presented in this paper. We will also consider different means to construct and compute graphs and arc weights associated with regions. This work has already been applied to 3D medical images segmentation. Our perspectives are now to develop morphological and graph-based methods for 3D temporal image segmentation.

ACKNOWLEDGEMENTS

We would like to thank Région Ile-de-France and the Cancéropôle for funding our research on medical image segmentation. We would also like to thank The Center for Advanced Visualization and Interaction (CAVI), Aarhus University Denmark, especially Thomas Sangild Sorensen and Jesper Mosegaard, for their support and advices about this paper.

REFERENCES

Beucher S, Lantuejoul C (1979). Use of watersheds in contour detection. In: Proc Int Worksh Image Proc, Rennes, France. September 17–21, 1979.

Beucher S (1994). Watershed, hierarchical segmentation and waterfall algorithm. In: Proc ISMM’94, 69–76.

Boykov Y, Jolly MP (2001). Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In: Proc Int Conf Comput Vis, Vancouver, Canada 1:105–12.

Boykov Y, Kolmogorov V (2003). Computing Geodesics and Minimal Surfaces via Graph Cuts. In: Proc Int Conf Comput Vis, Nice, France 1:26–33.

Boykov Y, Kolmogorov V (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans Pattern Anal 26:1124–37.

Boykov Y, Veksler O, Zabih R (2001). Fast Approximate Energy Minimization via Graph Cuts. IEEE Trans Pattern Anal 23:1222-39.

Boykov Y, Veksler O, Zabih R (1998). Markov random fields with efficient approximations. In: Proc IEEE Conf Comput Vis Pattern Recogn 648–55.

Caselles V, Kimmel R, Sapiro G (1997). Geodesic Active Contours. Int J Comput Vision 22(1):61–79.

Ford F, Fulkerson D (1955). A simplex algorithm finding maximal networks flows and an application to the Hitchock problem. Rand Report, Rand Corporation, Santa Monica 1955.

Greig D, Porteous B, Seheult A (1989). Exact maximum a posteriori estimation for binary images. J Roy Stat Soc. 51(2):271–9.

Kolmogorov V, Zabih R (2004).What energy functions can be minimized via graph cuts? IEEE Trans Pattern Anal 26:147–59.

Li Y, Sun J, Tang C, Shum H (2004). Lazy snapping. SIGGRAPH 2004. ACM Trans Graphics 23:303–8.

Lombaert H, Sun Y, Grady L, Xu C (2005). A multilevel banded graph cuts method for fast image segmentation. In: Proc 10th IEEE Int Conf Comput Vis, Vol. 1.

Meyer F, Beucher S (1990). Morphological Segmentation. J Vis Commun Image R 1:21–46.

Meyer F (1994). Minimal spanning forests for morphological segmentation. In: Proc ISMM’94. 13–14.

O’Callaghan RJ, Bull DR (2005). Combined Morphological-Spectral Unsupervised Image Segmentation. IEEE Trans Image Proc 14:49–62.