Eikonal based region growing for superpixels generation: Application to semi-supervised real time organ segmentation in CT images

Pierre Buyssens, Isabelle Gardin, Su Ruan

Eikonal based region growing for superpixels generation : Application to semi-supervised real time organ segmentation in CT images

Pierre Buyssens[a], Isabelle Gardin[a,b], Su Ruan[a]

[a]*LITIS EA 4108 - QuantIF, Université de Rouen,
22 boulevard Gambetta, 76183 Rouen Cedex, France*
[b]*Département de médecine nucléaire, centre Henri-Becquerel,
1 rue d'Amiens, 76038 Rouen, France*

Abstract

We describe a semi-supervised organ segmentation method for Computed Tomography images. In a first step, a dense oversegmentation of the image is created with an Eikonal-based algorithm. The proposed superpixel algorithm ourperforms state-of-the-art algorithms on classical metrics. In a second step, the semi-supervised segmentation is performed on the underlying Region Adjacency Graph created from the oversegmentation. As the complexity is greatly reduced, the organ segmentation can be performed in real-time.

Keywords: Superpixels, Eikonal equation, semi-supervised segmentation

1. Introduction

Fast and accurate computer aided organ segmentation is more and more required in many medical applications such automatic detection of liver cancer [19], measurements of liver volume [14], helpful both for surgical planning prior to hepatic resection [9], and for developing effective radiation treatments programs [11]. More precisely our work is part of a Selective Internal Radiation Therapy (SIRT) project [7] whose goal is to compute a 3-dimensional dosimetry of the liver for treatment planning.

To this end the volume of the liver has to be known to deliver the right dose to the patient. Lung volume has also to be known to compute the Lung Shunt Fraction (LSF) in order to avoid irradiation of the lungs during the treatment [2]. In this work we propose a semi-supervised organ segmentation scheme that gives the control of the segmentation to the radiologist. A major challenge of such a scheme in medical applications is the volume of data that has to be processed, which is often incompatible with real-time processings.

To tackle this problem, we propose a two-step scheme that, first simplifies Computed Tomography images by computing an oversegmentation of the volumes, and second performs the segmentation on the graph created from this oversegmentation. The oversegmentation proposed in this paper belongs to the *superpixel* algorithms that aim at grouping perceptually and adjacent pixels into meaningful regions, the so-called *superpixels*.

Numerous superpixel algorithms have been proposed in the literature, they can roughly be divided into Graph-based methods [18, 21] and gradient (or color) based ones [12, 20, 1]. More details can be found on these methods in Section 3 and a review of these in [1]. Other works make use of similar oversegmentations such [4, 3] based on an energy that favors partitions of similar pixels, or [15] based on the watershed transformation. These last methods however do not belong to the superpixel algorithm class since they do not control the amount of regions nor their compactness.

Eikonal-based Region Growing Clustering algorithm[1] (ERGC) is a new superpixel algorithm that formulates the clustering of pixels as a solution of an Eikonal equation :

$$\begin{cases} ||\nabla U(x)|| = F(x) & \forall x \in \mathcal{I} \\ U(x) = 0 & \forall x \in \Gamma \end{cases} \tag{1}$$

where $\mathcal{I}$ is the image domain, F a positive function, Γ the set of initial seeds, and $U(x)$ the traveling time or geodesic distance of x from source Γ.

The main difference from other eikonal-based algorithms comes from the computation of $F(x)$ which is based on a distance between pixel and region features.

Efficiently numericaly solved with the *Fast Marching* method [17], ERGC leads to the creation of homogeneous superpixels. An additional spatial constraint can also be added to F to enforce superpixel compactness.

The rest of the paper is organized as follows : Section 2 details ERGC algorithm. Its performance is evaluated and compared to state-of-the-art superpixel algorithms in Section 3. The application of ERGC as a preprocessing step of the graph-based semi-supervised segmentation scheme is detailed in Section 4. Section 5 concludes the paper.

2. Superpixels method

2.1. Notations

In the following several notations are adopted to simplify the reading of the paper. A pixel/voxel is noted $\boldsymbol{p}$ and consists of a coordinate triplet $(x_{\boldsymbol{p}}, y_{\boldsymbol{p}}, z_{\boldsymbol{p}})$ and a color vector $\boldsymbol{C_p}$. For $2D$ images, the z coordinate is equal to zero. A superpixel/supervoxel R_i consists of a seed pixel/voxel $\boldsymbol{s}_i$, a mean color vector $\boldsymbol{C}_i$ and the number N_i of pixels/voxels that belong to R_i.

Note that these formulations are independant of the number of channels of the images. For gray level images, $\boldsymbol{C_p}$ reduces to the gray level of the pixel $\boldsymbol{p}$.

2.2. Algorithm

The initialization of the algorithm consists of sampling the number of desired seeds pixels/voxels K on a regular grid. Without any prior, the sampling step for $2D$ images is set to $S = \sqrt{N/K}$ with N the number of pixels/voxels of the image (similar to [1] and [12]). These initial seeds are then perturbed and moved to the minima of the gradient map in a V_8 neighborhood (V_{26} for $3D$ images). The mean color vector of each superpixel/supervoxel $\boldsymbol{C}_i$ is then initialized with C_{S_i} and its neighbors in V_4 (V_6 for $3D$ images).

To compute the numerical solution of equation 1, we adopt the Fast Marching method introduced by Sethian in [17]. It consists of computing locally the geodesic distances for an evolving front with a priority queue (see [16] for a detailed description of the fast marching method). As a front can only evolve with a positive speed function, it passes through a pixel/voxel only once, and the algorithm terminates when all pixels/voxels have been visited.

Contrary to many front–based methods that rely on a fixed potential map (often the magnitude of the gradient of the image : $F = ||\nabla I||$),

[1] Source code and executable of ERGC can be found at `https://sites.google.com/site/pierrebuyssens/ergc`

the right term of equation 1 evolves during the diffusion process and is not fixed. Given a pixel $\boldsymbol{p}$, the proposed function F (right term of equation 1) is computed according to an adjoining region R_i as :

$$F_c(\boldsymbol{p}, R_i) = \|\boldsymbol{C_p} - \boldsymbol{C}_i\|_2^2 \qquad (2)$$

This formalization favors the creation of superpixels/supervoxels that are more homogeneous than those created by classical gradient–based methods. Figure 1 compares the behaviors of both approaches on a toy problem and on a natural scene.

Within the fast marching algorithm, a superpixel/supervoxel R_i has to be updated every time a pixel $\boldsymbol{p}$ is integrated to R_i. This update is easily computed with :

$$\begin{aligned} \boldsymbol{C}_i &\leftarrow \frac{\boldsymbol{C}_i \times N_i + \boldsymbol{C_p}}{N_i + 1} \\ N_i &\leftarrow N_i + 1 \end{aligned}$$

A similar potential function has been proposed in [13]. Dedicated to the $3D$ brain segmentation, the potential is computed according to a histogram analysis prior to the evolution of active contours. Contrary to our approach, the features (i.e.: mean and standard deviation of the color) of the regions that initially define the active contours do not evolve. Moreover the analysis of the histogram of the brain allows the detection of 3 well defined peaks (corresponding to the gray matter, the white matter and the cerebrospinal fluid), which is not the case for many kind of images, especially with CT where different organs may be composed of pixels with similar or equal gray values. Our framework is far more general since it can deal with any kind of multispectral images.

In this primal form, ERGC superpixels may be very irregular in shape and size. However an additional spatial constraint can be added to the right member of equation 1 to enforce superpixel compactness.

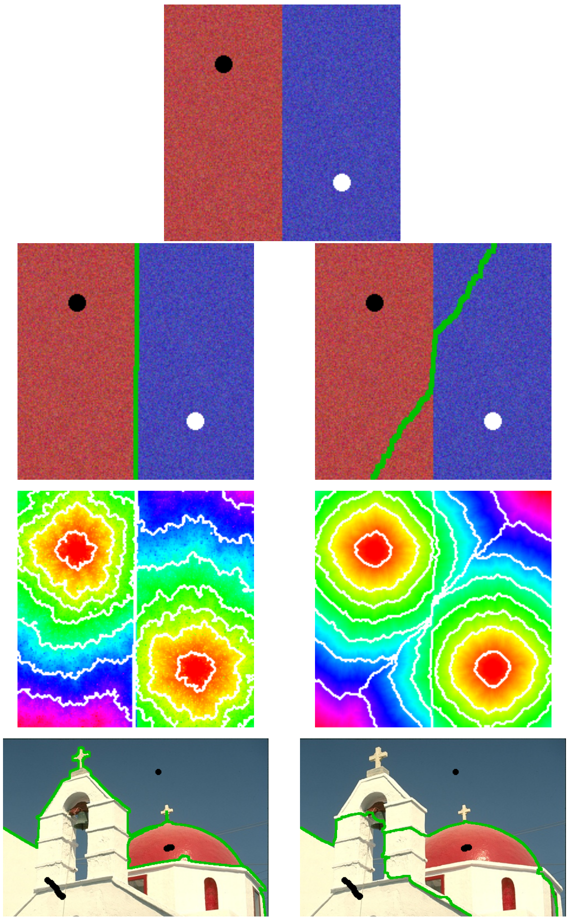


Figure 1: First row : initial image. The back and white circles depict the initial seeds. Second row : the result segmentation with the proposed F (left) and the gradient–based potential function (right). Third row : geodesic distances map U in fake color. Some isocontours are shown in white. Last row : segmentation of a natural scene. Seeds are depicted in black.

2.3. *Superpixels compactness*

According to the application, it may be desirable to have compact superpixels. Following the idea introduced in [1], a spatial constraint term can be added to the computation of F. This spatial constraint penalizes the geodesic distances of a pixel $\boldsymbol{p}$ that is spatially far from a seed $\boldsymbol{s}_i$ via the $\mathcal{L}_2$ norm $||\boldsymbol{p} - \boldsymbol{s}_i||_2$.

Since the spatial and color proximities are not defined on the same range (especially with medical images), a compactness factor m is used to weigh the relative importance between the two metrics.

With this spatial constraint, the second term of equation 1 becomes :

$$F(\boldsymbol{p}, R_i) = \sqrt{F_c(\boldsymbol{p}, R_i)^2 + F_s(\boldsymbol{p}, \boldsymbol{s}_i)^2} \qquad (3)$$

with

$$F_s(\boldsymbol{p}, \boldsymbol{s}_i) = \frac{||\boldsymbol{p} - \boldsymbol{s}_i||_2^2}{S} \times m \qquad (4)$$

where S in the initial sampling step. Setting $m = 0$ leads to $F(\boldsymbol{p}, R_i) = F_c(\boldsymbol{p}, R_i)^2$ which means that no spatial constraint is considered. Figure 2 shows the resulting superpixels for a CT image with varying the number of superpixels and the compactness factor.

2.4. 3*D extension*

ERGC naturally extends to 3D images with minor modifications of the algorithm. Figure 3 shows a supervoxel oversegmentation of a 3D volume from the MICCAI-2007 Grand Challenge dataset. For display purposes only the interior part of the body is shown.

3. Evaluation of ERGC

In this section, we evaluate ERGC algorithm on a natural image dataset and on a medical image dataset, and compare it to the superpixel methods published in the litterature.

The classical metrics to compare oversegmentation results are used in this evaluation :

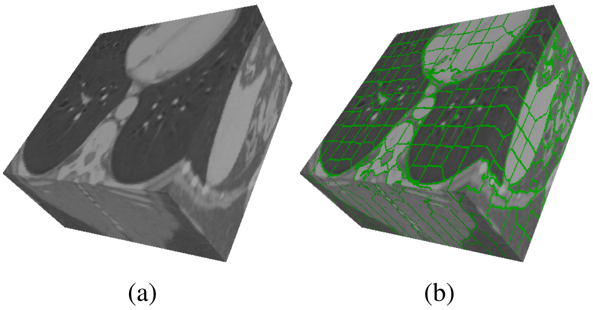


Figure 3: Approximately 1000 ERGC supervoxels for a Computed Tomography volume of a human body. Only the interior part of the body is shown for visualization convenience.

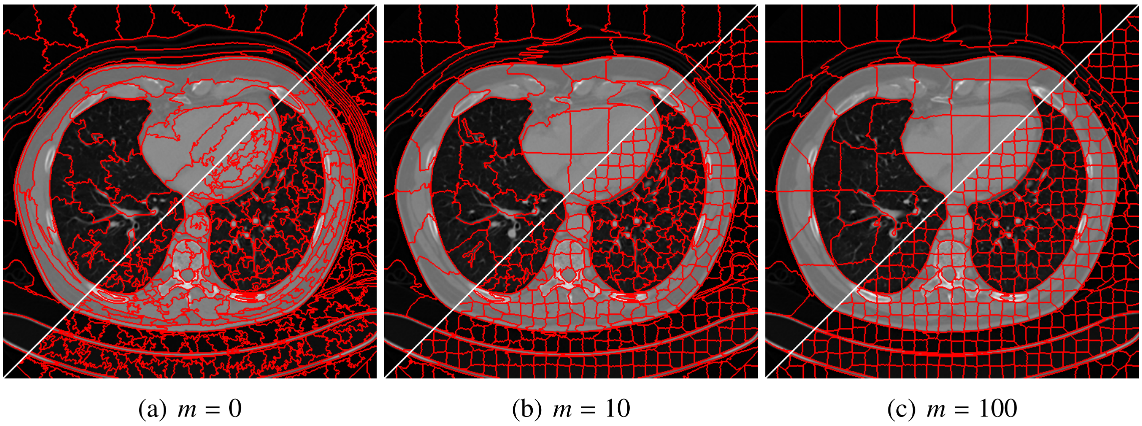
- The *boundary recall* (BR) indicates how well superpixels adhere to true boundaries of the objects in the image. It measures the fraction of ground truth edges that is also present in superpixel segmentation within a distance t. In our experiments, t is fixed to 2 (as in [1, 12] to cite few).

- The *under-segmentation* (UE) error measures the fraction of *bleeding* caused by superpixels that overlap a given ground truth segment. For a given ground truth segmentation into segments $g_1, \ldots, g_k$ and a superpixel segmentation $R_1, \ldots, R_N$, the undersegmentation error for a segment g_i is quantified with

$$\frac{\left[\sum_{\{R_j | R_j \cap g_i > B\}} Area(R_j)\right] - Area(g_i)}{Area(g_i)}$$

where B is set to 5 percent of $Area(R_j)$ (as in [1]) to account for ambiguities of the ground truth segments. Finally this fraction is simply averaged across all ground truth segments and all images.

- The *Achievable Segmentation Accuracy* (ASA) is a performance upperbound



(a) $m = 0$ (b) $m = 10$ (c) $m = 100$

Figure 2: CT image with 100 superpixels in the upper left of each image and 900 in the lower right for different values of the compactness factor.

measure. It gives the highest performance when taking superpixels as units for object segmentation. Each superpixel is labeled with the label of the ground truth segment with the largest overlap. The fraction of correctly labeled pixels is the achievable accuracy :

$$ASA(s) = 100 \times \frac{\sum_k \max_i |s_k \cap g_i|}{\sum_i g_i}$$

- As superpixels are often used as a preprocessing step, the *processing time* is an important factor such that it does not slow down the image processing pipeline too much.

Performances are evaluated and compared to five other algorithms :

- The Normalized Cuts approach (NC05) proposed in [6] partition a graph defined on pixels by minimizing a global cost function with an explicit compactness constraint.
- Superpixels method proposed in [21] formulates the superpixel partitioning problem in an energy minimization framework, and optimize it with graph cuts. Two versions of the algorithm are proposed differing in compact (GCa10) or constant intensity (GCb10) resulting superpixels.
- The turbopixels method (TP09) proposed in [12] dilates a set of initial seeds using a level–set geometric flow to adapt to local image structure.
- Quick shift method (QS08) proposed in [20] exploits kernels to extend mean shift [5] using a medoid shift procedure.
- SLIC method proposed in [1] adapts k–means in the image plane to exhibit superpixels. The addition of a spatial constraint term produces regular regions that adhere well to image boundaries.

3.1. Berkeley dataset

The Berkeley image dataset contains 300 $2D$ images (in the RGB colorspace) of size 481×312 (or 321×481) and approximately 10 ground truth manual segmentations for each image.

Table 1 exhibits results on standard metrics for the considered algorithms. Results have been obtained with approximately 500 superpixels per image and have been computed for the 3000 ground truth images. ERGC outperforms all other superpixels algorithms on the standard metrics, while being faster. Moreover, Student's t-test on these results exhibits high values ($t > 35$), which validates the statistical significance of the results.

Complexity of ERGC relies essentialy on the complexity of the fast marching algorithm, which is roughly in $O(n \log n)$ with an appropriate heap for sorting the pixels according to their geodesic distance. Despite this theorical complexity, ERGC is very fast in practice, and the algorithm is nearly linear in time. Note that these results have been obtained without spatial constraints $m = 0$. Adding this constraint degrades slightly ERGC performances, but produces more visually pleasing superpixels.

3.2. MICCAI-2007 Grand Challenge dataset

The MICCAI-2007 Grand Challenge dataset [10] consists of 20 CT $3D$ images (volumes) used for training and 10 for tests. In this evaluation we focus on the training part (and its ground truth) and especially the slices which contain some liver parts. The sub–dataset consists then of 2750 $2D$ gray images (of size 512×512), for a total amount of more than 1.5 million liver boundary pixels. Table 2 summarizes the results of ERGC and the current state-of-the-art SLIC algorithm. Since CT images are grayscale and for fair comparisons purposes, results of SLIC and ERGC of table 2 have been obtained by simulating RGB images (i.e. 3 channels) for a fair comparison. ERGC$_g$ is more realistic and works directly on monochannel images. ERGC$_g$ then gives the same results as ERGC while being faster.

As shown in table 1 and 2, ERGC and SLIC give the lowest *Undersegmentation Error*. However ERGC gives the better *Boundary Recall* and *Achievable Segmentation Accuracy* results upon the considered algorithms, and especially SLIC.

Table 2: Boundary Recall, Undersegmentation Error, Achievable Segmentation Accuracy and processing time (in seconds) comparisons between SLIC, ERGC and ERGC$_g$ on the training part of the MICCAI-2007 Grand Challenge dataset. Processing times are given for one slice.

	SLIC	ERGC	ERGC$_g$
BR			
mean	0.8736	0.9464	0.9464
std	0.0317	0.0209	0.0209
UE	0.15	0.15	0.15
ASA	98.03	99.12	99.12
Processing time	1.56	0.57	0.37

The *Achievable Segmentation Accuracy* result is the more important in our application since the semi-supervised segmentation proposed below considers superpixels/supervoxels as units.

4. Semi-supervised organ segmentation

In order to simplify the images and to reduce the complexity of the semi-supervised segmentation algorithm, ERGC is performed on CT images. The segmentation is then processed on the underlying Region Adjacency Graph (RAG) created from this oversegmentation.

Feature of each vertex is simply the mean color of the underlying superpixel/supervoxel. An edge connecting two vertices u and v is weighted with the function

$$w(u, v) = \|\boldsymbol{C}_u - \boldsymbol{C}_v\|_2$$

Figure 4 shows a 512×512 slice extracted from a CT volume of the MICCAI-2007 Grand Challenge dataset, the oversegmentation performed by ERGC containing approximately 1000 superpixels, and the RAG created from

Table 1: Comparison of ERGC to several superpixels algorithms for approximately 500 superpixels on different metrics. Values of algorithms NC05, GCa10, TP09, and QS08 have been extracted from [1] courtesy of the authors.

	Graph–based			Gradient–ascent–based			
	NC05	GCa10$_g$	GCb10$_g$	TP09	QS08	SLIC12	ERGC
	[18]	[21]	[21]	[12]	[20]	[1]	
BR	0.68	0.69	0.70	0.61	0.79	0.82	**0.88**
UE	0.22	0.22	0.22	0.24	0.20	**0.19**	**0.19**
ASA	N/A	N/A	N/A	N/A	N/A	97.94	**98.22**
Segmentation speed							
320×240 image	×494	×14	×11	×22	×13	×1	×0.33
2048×1536 image	N/A	×21	×15	×53	×12	×1	×0.49
Control over amount of superpixels	Yes	Yes	Yes	Yes	No	Yes	Yes
Control over superpixels compactness	No	No	No	No	No	Yes	Yes
Supervoxel extension	No	Yes	Yes	No	No	Yes	Yes

this oversegmentation. In this example, no spatial constraint has been added to F, that is $m = 0$.

Once the graph is created, a user can manually label a superpixel and then its corresponding vertex in the graph. Each time a superpixel/supervoxel is labeled, a label diffusion on graph is performed using the Dijkstra algorithm [8]. Since the graph contains very few vertices (compared to the number of pixels of the initial image), the diffusion is very fast. The labeled graph then gives a label to each superpixel/supervoxel, which leads to the segmentation.

In our tests, the Dijkstra diffusion on graph that contains 1000 vertices processes in approximately 0.01 seconds on a standard laptop. This processing time allows a real-time segmentation by the radiologist, and a bad labeling can be corrected on the fly. Figure 5 shows the semi-supervised segmentation of the liver on a $2D$ slice with few manual labels.

Processing $3D$ CT images operates in the same manner. Supervoxels are computed from a CT volume and the underlying RAG is created. However, instead of fixing the number of desired K, the initial seeds of ERGC are sampled according to a fix sampling step S. In our experiments on the MICCAI-2007 Grand Challenge dataset, S is fixed to 10 pixels along x and y axis, and 5 pixels along z axis, to take into account the inter-slices distance which is higher than inter-pixels distance in the transverse plane. The second advantage of fixing S along each axis instead of K is to deal with images of various depth.

Figure 6 shows transverse, sagittal and coronal views of a CT volume containing 183 slices of size 512×512. About 96000 ERGC supervoxels have been computed with the sampling steps detailed above. Fixing a high value for the spatial constraint m gives more compact supervoxels, hence simplifying the semi-supervised segmentation. In our experiments, m is fixed to the product of sampling steps which is 500.

Since the graph contains more vertices than in the $2D$ example above, the Dijkstra diffusion processing time is quite longer especially after the firsts supervoxels seeds have been placed. For the underlying RAG of figure 6, the first Dijkstra iteration processes in about 0.3 seconds but the following quickly decreases to 0.01 seconds or less, since distances for only a few number of vertices have to be updated.

Placing seeds into transverse, sagittal and

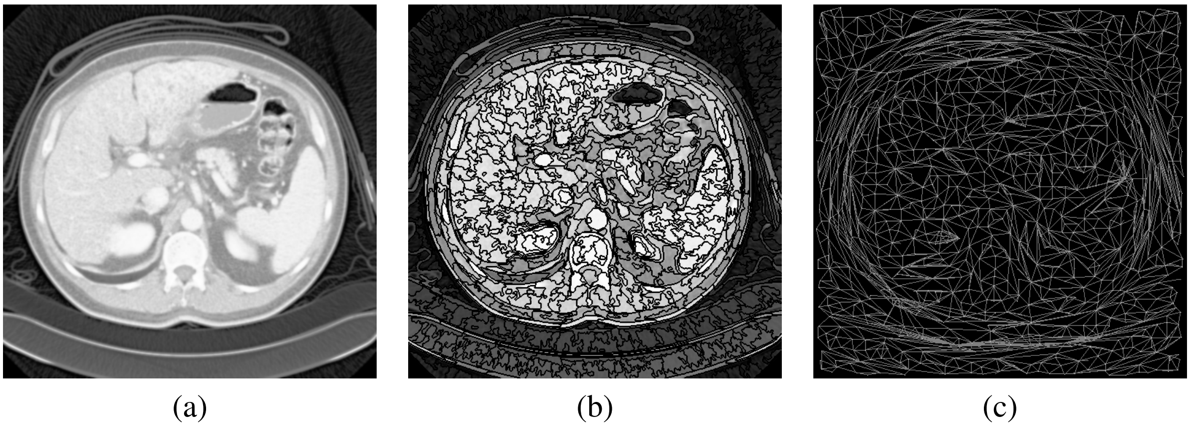


Figure 4: (a) Initial slice, (b) approximately 1000 ERGC superpixels without spatial constraint ($m = 0$), and (c) the underlying RAG.

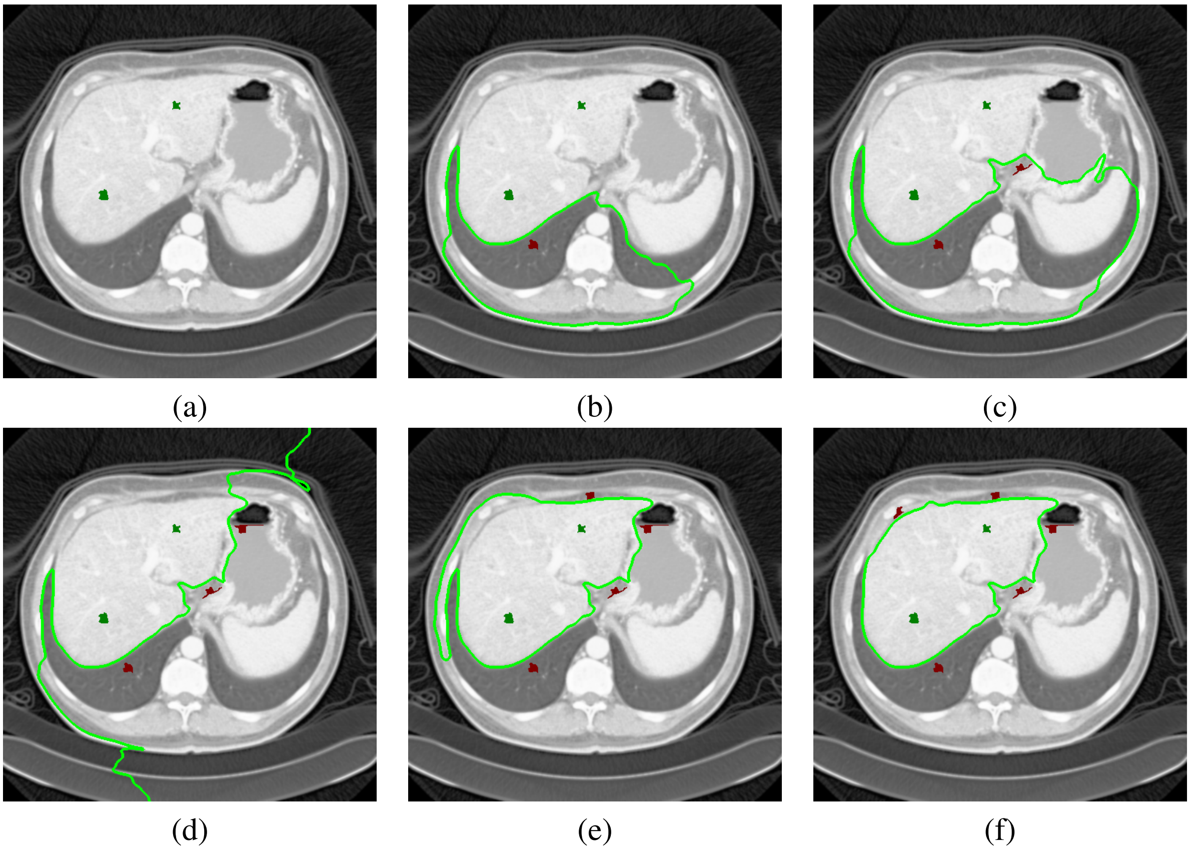


Figure 5: Example of a semi-supervised segmentation of the liver. Two superpixels have been labeled as *liver* in dark green (a), (b,c,d,e,f) annotation of superpixels as *background* in dark red. The liver contour is computed on the fly and appears in green. For clarity purposes, only the liver is segmented.

coronal planes is natural for a radiologist. The whole process acts in the same manner as in $2D$ except that seeds can be placed in any of the three planes. Result segmentation is then updated in the three planes and the radiologist can add/remove seeds to refine/correct the segmentation. Figure 6 shows a segmented volume with liver, lungs and background belonging to different classes (i.e. having different labels). Only a few supervoxel seeds are needed to obtain such a result. Note that edges of lungs and liver are smoothed with gaussian kernels of different strengths.

5. Conclusion

In this paper, we proposed a method based on the Eikonal equation that quickly creates accurate superpixels/supervoxels. Outperforming other oversegmentation algorithms on classical metrics, ERGC is used as a first step in a graph-based semi-supervised image segmentation. Applied to Computed Tomography images, it allows a real-time segmentation by processing the underlying graph, and gives the control of the segmentation to the radiologist.

Further works will take into account color and shape prior to ease the organs segmentation.

6. Acknowledgements

This work is part of the SIRTDose project funded by the grant “Physique Cancer” 2012 (INSERM - Plan Cancer). The authors would like to thank the authors of [1] for the enlightenments on their algorithm.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[2] Hojjat Ahmadzadehfar, Hans-Jürgen Biersack, and Samer Ezziddin. Radioembolization of liver tumors with yttrium-90 microspheres. In *Seminars in nuclear medicine*, volume 40, pages 105–121. Elsevier, 2010.

[3] Pablo A Arbeláez and Laurent D Cohen. A metric approach to vector-valued image segmentation. *International Journal of Computer Vision*, 69(1):119–126, 2006.

[4] Pablo Andrés Arbeláez and Laurent D Cohen. Energy partitions and image segmentation. *Journal of Mathematical Imaging and Vision*, 20(1-2):43–57, 2004.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[6] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131. IEEE, 2005.

[7] Arnaud Dieudonné, Etienne Garin, Sophie Laffont, Yan Rolland, Rachida Lebtahi, Dominique Leguludec, and Isabelle Gardin. Clinical feasibility of fast 3-dimensional dosimetry of the liver for treatment planning of hepatocellular carcinoma with 90y-microspheres. *Journal of Nuclear Medicine*, 52(12):1930–1937, 2011.

[8] Edsger Wybe Dijkstra. *A short introduction to the art of programming*, volume 4. 1971.

[9] S.T. Fan, C.M. Lo, C.L. Liu, B.H. Yong, J.K.F. Chan, and I.O.L. Ng. Safety of donors in live donor liver transplantation using right lobe grafts. *Archives of surgery*, 135(3):336, 2000.

[10] Tobias Heimann, Bram van Ginneken, Martin A Styner, Yulia Arzhaeva, Volker Aurich, Christian Bauer, Andreas Beck, Christoph Becker, Reinhard Beichel, György Bekes, et al. Comparison and evaluation of methods for liver segmentation from ct datasets. *Medical Imaging, IEEE Transactions on*, 28(8):1251–1265, 2009.

[11] M. Kobashi and L.G. Shapiro. Knowledge-based organ identification from ct images. *Pattern Recognition*, 28(4):475–491, 1995.

[12] A. Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2290–2297, 2009.

[13] Hua Li, Anthony Yezzi, and Laurent D Cohen.

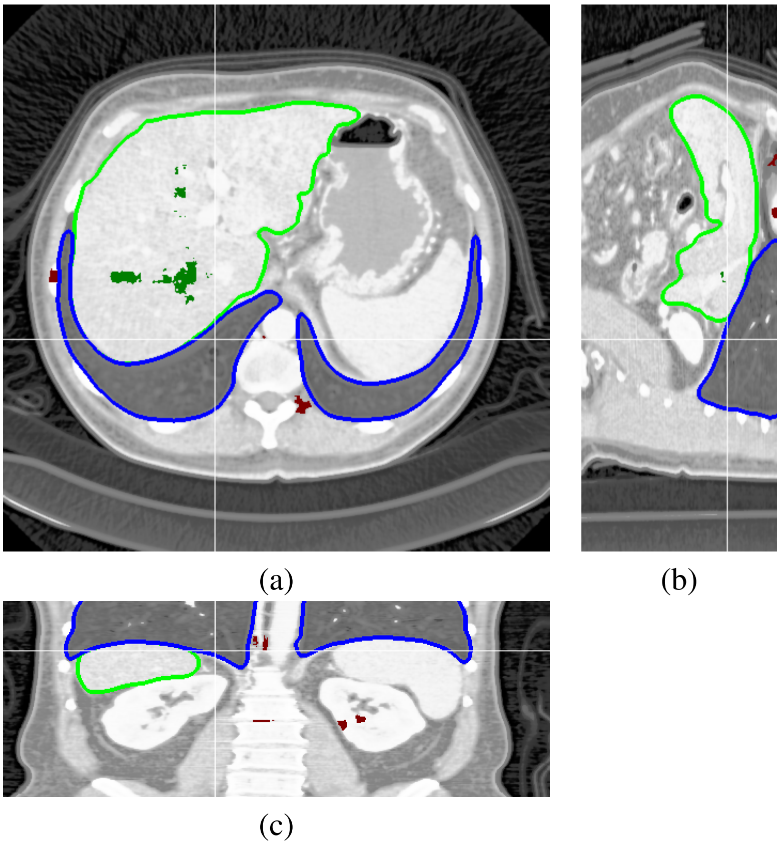


Figure 6: Segmentation of lungs (blue) and liver (green) in transverse (a), sagittal (b), and coronal (c) planes. Supervoxels seeds for the *lungs*, the *liver* and the *background* are colored dark blue, dark green and dark red respectively.

Fast 3d brain segmentation using dual-front active contours with optional user-interaction. In *Computer Vision for Biomedical Image Applications*, pages 335–345. Springer, 2005.

[14] Y. Nakayama, Q. Li, S. Katsuragawa, R. Ikeda, Y. Hiai, K. Awai, S. Kusunoki, Y. Yamashita, H. Okajima, Y. Inomata, et al. Automated hepatic volumetry for living related liver transplantation at multisection ct1. *Radiology*, 240(3):743–748, 2006.

[15] HP Ng, SH Ong, KWC Foong, PS Goh, and WL Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE, 2006.

[16] G. Peyré, M. Péchaud, and R. Keriven. *Geodesic methods in computer vision and graphics*. Now publishers Inc, 2010.

[17] J.A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[19] A. Shimizu, T. Kawamura, Y. Mekada, Y. Hayashi, and D. Deguchi. Preliminary report of cad system competition for liver cancer extraction from 3d ct images and fusion of the cads. *International Journal of Computer Assisted Radiology and Surgery*, 1:525, 2006.

[20] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. *Computer Vision–ECCV 2008*, pages 705–718, 2008.

[21] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. *Computer Vision–ECCV 2010*, pages 211–224, 2010.