Energy minimization and region growing based interactive image segmentation

Domagoj Kovacevic and Sven Loncaric
Faculty of Electrical Engineering, University of Zagreb
Unska 3, 10000 Zagreb, Croatia

ABSTRACT

In this work, two novel methods for semi-automatic image segmentation of medical images are presented. The developed techniques are applied to the problem of medical image segmentation. The two developed approaches are compared and the results are discussed. The first method is based on interactive boundary detection where the user is assisted by the computer in selection of border of the desired anatomical region. The user performs segmentation by selecting a sparse series of points along the desired region border. The optimal path (border segment) is calculated based on the minimization of the energy function. The second method is based on interactive region growing process. The algorithm computes image features along the user-defined path and the computed features are used in the region-growing process. The region selection process is repeated until the user is satisfied with the look of the segmented region. Those two methods are compared to two semi automatic edge and region based image segmentation techniques.

Keywords: image segmentation, image analysis, medical imaging, energy minimization, region growing

1. INTRODUCTION

Image segmentation is a process in which pixels are grouped into regions according to some uniformity criteria. In medical images, regions usually represent organs, for instance, lungs in X-ray image, etc. This is a low-level operation, which is necessary in order to perform high-level operations such as analysis of the shape and size of the organs, 3-D volume visualization, and so on. There are two principal ways to perform segmentation. The first is manual segmentation, performed by medical expert, in which case he or she has to manually outline region of interest, using a pointer device, usually mouse. The other approach is to perform as much as possible of the segmentation automatically. In this case, the whole process is performed by means of a segmentation algorithm, with minimal user interaction. In medical imaging, there are several approaches to automatic image segmentation. Traditionally, they may be divided into region based and edge based.[1] In addition, in the last decade there is a lot of research in the field of active contours[2] which combine good properties of both region and edge based methods.

Obviously, automatic methods have many advantages over manual approach; they are much faster, require much less medical expert time, thus saving money. Unfortunately, there are also many problems with automatic methods. It is hard to design an algorithm that performs well on all atypical cases, so manual inspection and correction of results obtained by automatic segmentation is necessary. For these reasons it makes sense to develop segmentation techniques that are interactive but utilize more advanced image analysis concepts so that the user is assisted in a way that enables fast interactive (semi-automatic) segmentation of images.

In this work we propose two methods for semi-automatic image segmentation. The first method uses edge detection and energy minimization approach. The second method is region based approach. In both methods the user performs segmentation by manually selecting points but the user does not have to select all of the points on the region border. Instead, only few points on the border of the region of interest are required in the first approach, and few points inside region of interest in the second approach. The rest of the procedure is performed by segmentation algorithm. In this approach, the user is still completely in charge as in manual segmentation approach, but the segmentation task is easier, since only few points on the border or inside the region of interest must be selected.

The rest of the paper is organized as follows: In the second section two semi-automatic algorithms published in the literature are presented, one region based, and one edge based. The proposed semi-automatic segmentation algorithms are described in the third section. Implementation details and results are presented in the fourth section. The conclusion is given in the fifth section.

Send correspondence to Domagoj Kovacevic(E-mail: domagoj.kovacevic@zesoi.fer.hr)

2. PREVIOUS APPROACHES TO SEMI-AUTOMATIC SEGMENTATION

2.1. Region-based methods

An algorithm for semi-automatic region-based segmentation is presented by Adams and Bischof,[3] with improvements by Mehnert and Jackway.[4] In this algorithm, it is necessary to manually segment one or more seed points for the each region in the image. After that, for all points that are neighboring to the segmented points similarity to its neighboring segmented regions is measured. After that, neighboring points are sorted in a list in a descending order of the similarity. In the next step, the first point from the list is segmented as its neighboring segmented region, and removed from list. All neighboring non segmented points of the segmented point are added to the list in positions according to the similarity to the neighboring segmented region, and this process is repeated while there is still non segmented points in image. If a point is adjacent to more than one region, similarity is measured among this point and the region to which it is most similar (similarity is highest). For details see.[3]

This algorithm is very fast and performs well if image is not heavily textured, which is the case for many classes of medical images. Also it is not very dependent of the position of the selected seed points. However, a problem in some classes of medical images, such as MRI scans of abdominal region of the human body, is that there is usually vast number of different regions (organs, vessels, bones), and it is necessary to manually mark all of those regions. Using centers of gravity of regions from previous slices as initial points in next slice in case of 3D imaging can reduce this problem.

2.2. Edge-based methods

An edge-based approach to semi-automatic segmentation was presented by Udupa et all.[5] In this approach, the user selects start point on the border by clicking the mouse when pointer is on that point. After that the user moves pointer to another point on the border. Note that the user does not have to follow border while moving the mouse. When pointer is moved to a new position, the algorithm tries to find the best border path from the start point to the current mouse point by forming a graph based on image gradient values and finding the path with the lowest cost in the graph. This path corresponds to the border segment from the start to the current point. If the user is satisfied with the computed border segment, he or she presses the mouse button to accept the offered border segment and repeats the process until entire border is segmented. In the first version of the algorithm, after the start point is selected, optimal paths to all other points in the image were calculated, which was inherently slow. Recently, an improvement of the algorithm was presented by the same authors[6] which is faster, without loss of segmentation quality.

3. TWO SEMI-AUTOMATIC SEGMENTATION METHODS

In this section we present two novel semi-automatic segmentation methods.

3.1. Edge-based semi-automatic segmentation method

The first algorithm is semi-automatic edge segmentation based on the energy minimization function. The method is based on interactive search for the border between two regions. The image is segmented in the following way. The user selects a point on the border between the region of interest (ROI) and the rest of the image by pressing the button on the mouse when pointer is positioned on that point. After that the operator moves the mouse to some other point of the border. The advantage over manual method is that the operator does not have to move pointer accurately along the border. For every new position of the pointer the computer calculates the optimal connection path and displays it by superimposing it on the original image. This connection path is optimal due to some criteria which usually involves the following elements: gradient values along the path, the curvature of the path, and its length. In addition to these elements some other elements can also be taken into consideration. If the operator is satisfied with the border segment from start to the current point the mouse button is pressed again to accept that part of the object border. At this moment the current point becomes the new fixed point and the optimal connection is now calculated from that point to any new pointer position. When after some number of points the border is close to the initial (first selected) point the operator closes the contour by pressing another button on the mouse. In our approach only local information is used to calculate the path from the beginning to the end point. That means that when user moves the pointer to an image point the optimal path is calculated only for that point. Although results are not as good as in the global case speed improvement is significant.

More formally, the user must select point on the border of the region of interest (X_a, Y_a), and move the mouse to another border point (X_p, Y_p). The optimal border segment from the first to the second point is determined using the following algorithm: Let(X, Y) be any point of the image. Let (X_a, Y_a) and (X_p, Y_p) be the initial and the final point of connection path, respectively. Let (X_i, Y_i) be a point that is 8-connected to (X, Y) and $W(X, Y)$ be a set of all (X_i, Y_i) for a given point (X, Y). Let M be a matrix of size equal to the image size, that is used for result storage. Let $E(X, Y)$ be the value of the energy function at point (X, Y).

Algorithm 1 Edge-based semi-automatic segmentation method.

```
1: M(X_a, Y_a) = TRUE
2: (X, Y) = (X_a, Y_a)
3: repeat
4:    Find (X_w, Y_w) ∈ W(X, Y) such that E(X_w, Y_w) ≤ E(X_i, Y_i), i ≠ w and M(X_w, Y_w) = FALSE
5:    M(X_w, Y_w)= TRUE
6:    (X, Y) = (X_w, Y_w)
7: until (X, Y) = (X_p, Y_p)
```

The key element of this method is the energy function $E(X, Y)$. The energy function chosen here is the sum of the three energy terms. Each of these terms imposes some conditions on the shape of the curve from the beginning to the end point. The first term imposes constraints on the curvature of the border segment. Since the shapes that are segmented in this work are biological ones they don't tend to have sharp edges so the border should not have sharp edges. The first term is equal to the absolute difference between the angle between previous point of the curve and the current point and the current point of the curve and a possible next point of the curve. It has minimum value if the curve continues in the same direction. The second term is the function of the distance between selected point and the end point of the curve.

$$E_2(X, Y) = ((X - Xp)^2 + (Y - Yp)^2)^{0.5} \tag{1}$$

This term ensures that we will finally reach point that was selected to be the end point. The third term defined for a point at the coordinates (X, Y) is the sum of values of selected features calculated from the image at the point (X, Y) scaled by some constants.

$$E_3(X, Y) = \sum_{i=1}^{N} K_i f_i(X, Y) \tag{2}$$

where N is the number of features used and $f_i(X, Y)$ is value of the i-th feature at the point (X, Y) and K_i is the scaling factor for i-th feature. If the scaling factor is greater than zero than it is more likely that contour will follow pixels with larger value of that feature and it will avoid pixels having larger value of the features for which the scaling factor is set to negative number. In our algorithm two features were used. The first feature is the absolute value of the Roberts gradient. The value of the gradient is usually high in the border pixels and low inside regions. Because of that the scaling factor for the gradient is set to positive value. The second feature used is brightness. If the region of interest is brighter than the background, it is convenient to set scaling factor associated with brightness to a negative value, since this means that the border will be on the outside of the region of interest. Each of these terms is used for finding the optimal next point scaled by some real constant $c_i, i = 1, ..., 3$

$$E(X, Y) = c_1 E_1(X, Y) + c_2 E_2(X, Y) + c_3 E_3(X, Y) \tag{3}$$

Those constants vary from one type of image to another, and are determined experimentally.

3.2. Region-based semi-automatic segmentation method

The second proposed method is region-based. The user selects one point in the region, and moves the mouse inside the region. For each new mouse position segmentation of the image is dynamically performed based on the selected point, and the current mouse position, and the result is displayed superimposed on the original image. In addition, at the same time the user can increase or decrease segmentation threshold using the keyboard, thus having additional degree of freedom. This algorithm is similar to the method mentioned in the section 2.1 since it is also region-based, but implementation is more similar to the algorithm mentioned in section 2.2, since it is iterative by nature, meaning that by moving mouse, or changing threshold different possible segmentations are presented to the user.

In the region based approach the user selects points inside the region of interest. When the user selects a point in the image that is inside desired region, the point is used as a starting point $S(X_s, Y_s)$. After that, the user moves mouse pointer to any other position within same region $P(X_p, Y_p)$. A line is drawn from S to P, and some predefined number N_p of points along the line is used as a feature set $\mathbf{T} = \{T_1, T_2, ..., T_{N_p}\}$, where $T_1 = S$ and $T_{N_p} = P$, and the other points are linearly interpolated. After that a feature vector set $\mathbf{F} = \{F_1, F_2, ..., F_{N_p}\}$ is formed from each of the points in $\mathbf{T}$. F_i is feature vector with features calculated from point T_i.

Different features can be used in this vector. The first choice is brightness of the selected point T_i. This feature does not require computation, and it is useful if region of interest has relatively constant brightness. Another feature that can be used is standard deviation of the pixels in some window W around point T_i. If the image is noisy, the value of the image processed with some noise removal filter such as mean or median filter can be used as the additional feature. Another useful feature can be value of the gradient at the selected point. If the user selects points inside the region, they are likely to have small value of gradient, and the points on edge of the regions tend to have high gradient value, so when in the next step region growing is performed, it will stop at the end of the regions, since gradient value will be different then in the selected points $\mathbf{T}$. It is possible to use one or more of the features described here. But since it is important that algorithm performs segmentation in real time, meaning that it can be executed as fast as user moves pointer to the new point, only small number of features can be used in the feature vector.

In the next step distance from the all points $C(X, Y)$ in the image to the means of the feature vectors $\mathbf{F}$ in the feature space normalized by the standard deviations of the feature vectors is calculated:

$$d(C) = \frac{\|F(C) - mean(\mathbf{F})\|}{SD(\mathbf{F})} \tag{4}$$

Here, $F(C)$ is feature vector for image point $C(X, Y)$. Feature vectors $F(C)$ are calculated in advance for all of the points in the image to save computational time. After that, distances $d(C)$ are subject to threshold:

$$b(C) = \begin{cases} 1 & if d(C) \leq k \\ 0 & otherwise \end{cases} \tag{5}$$

In Equation (5) k is a numerical constant which can be adjusted during the segmentation process. A higher value of k results in a larger region. User can adjust value of k during the segmentation using keyboard. $b(C)$ are points in the binary image B. Binary image B obtained using threshold has ones in the position of points which are closer in the feature space to the mean of feature vectors from selected points then the given threshold. In the next step region growing is performed on B with points $\mathbf{T}$ used as seed points, which produces new binary image B_2. Since B is a binary image, this region growing step is trivial.

In the next step of the algorithm the region of interest R is computed as

$$R = B_2 - B \tag{6}$$

Borders of the region R are displayed superimposed to the original image and if the user is satisfied with the result of the segmentation he can finish it, and if not he can move mouse to new position P or change value of parameter k, and the whole process is repeated.

3.3. Post processing

When the user is satisfied with general shape of the region, he or she can finish segmentation process. As it can be seen from Figure 2 the shape of the region of interest is not correct, and there are holes inside the region. To solve this problem, morphological operations are applied to binary image R. The algorithm which is used is described in Algorithm 2. E is structuring element. With smaller E algorithm is slower, but region borders are better then if larger E is used. Euler number of the binary image is equal to number of objects minus number of holes in object, and result is object with NR_HOLES. NR_HOLES is known in advance. The morphological operations (dilation, erosion and determining of the Euler number) are performed by convolutions with small kernels and they are not computationally intensive.

Algorithm 2 Post processing algorithm

$n = 0$
repeat
  Dilate R with structural element E
  $n = n + 1$
until Euler number of $R \leq 1 - NR\_HOLES$
Erode R with structural element E n times

3.4. Discussion

If regions to be segmented have sharp, distinct borders, it is better to use edge based approach, but if the region borders are soft and blur then region based algorithm performs better. Region based approach is also easier to use, since it is easier to select point inside of the region, then on the border. On the other hand, edge based approach gives more precise segmentation results.

Advantage of our second (region based) approach over method described in[3] is that it is not necessary to mark all of the regions in the image. This is important if image has many different regions, but we are only interested in few. On the other hand disadvantage is that it is necessary to set fixed threshold on segmentation, while in their approach all regions are grown in parallel, so they limit each other in growth. Also our approach is more computationally demanding, especially if more features, besides image brightness are used.

The border selection procedure used in our region-based method is the same as the method used by Udupa et all. in their work.[5] The difference between these two methods is in the way the optimal connection between fixed point (X_n, Y_n) and the current pointer position (X_p, Y_p) is calculated. We minimize our energy only in the local neighborhood, while they calculate global optimizations. Their approach is more accurate, but it is much more computationally demanding.

4. EXPERIMENTAL RESULTS

4.1. Region-based algorithm

For any interactive algorithm to be useful, it has to work in real time, meaning that segmentation is performed 'instantly' following the movement of the mouse . That means that only few features which have low computational complexity can be used. The proposed algorithm is implemented in Matlab programming language, since it is still in the development phase. If rewritten in compiled language such as C it would be faster, meaning more of the features could be used with same response time. The graphical user interface to the algorithm is shown in Figure 1.

In Figure 2, the steps in segmentation of the brain region from the MRI image are shown. In the left figure original MRI slice is shown. In the middle results of the segmentation described in the section 3.2 are shown. Straight line shown inside the region is the user drawn line from which **T** set of points is sampled. As it can be seen region has holes and it's borders are not correct. In the right figure results of the post-processing as described in section 3.3 are shown. It can be seen that segmentation is much better.

4.2. Edge-based algorithm

The edge-based algorithm presented in this work is implemented in C programming language. For development of the graphic user interface the Motif API on top of the Xlib was used. The software was developed on the Sun Ultra 1 workstation. The graphical user interface is shown in Figure 3.

The comparison of segmentation results for the edge-based and the region based interactive segmentation of the hemorrhage region of the human brain is shown in Figure 4. This result for region-based method is shown before postprocessing step, and it can be seen that borders are not as accurate as in the edge based approach.

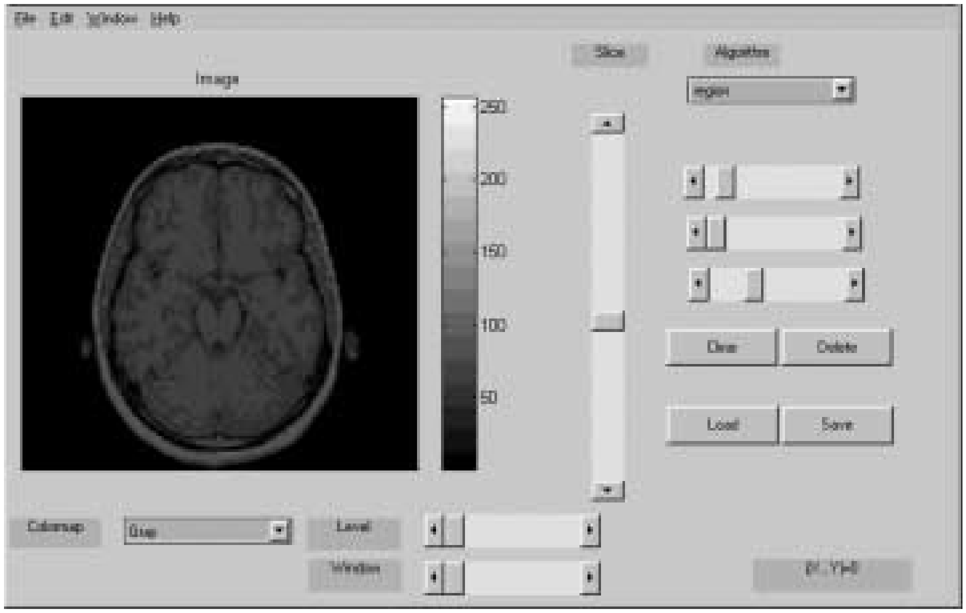

Figure 1. The graphical user interface for the region-based method.

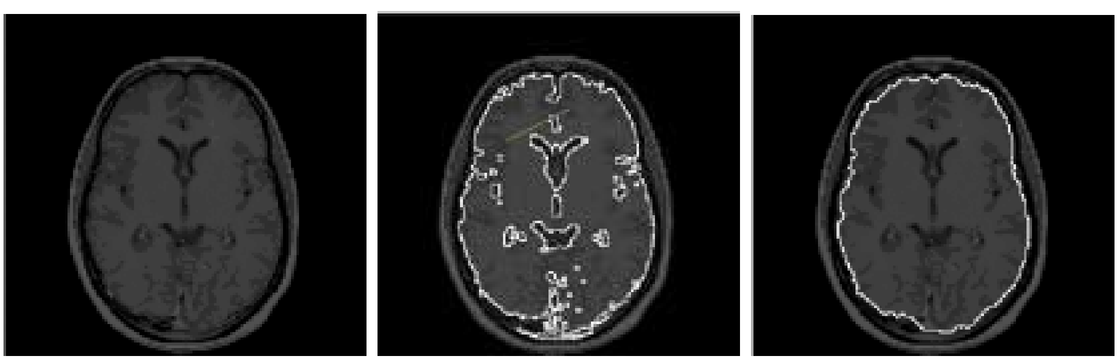

Figure 2. Segmentation steps of the region-based method: original MRI image (left), result of the segmentation (middle), and result after postprocessing step (right)).

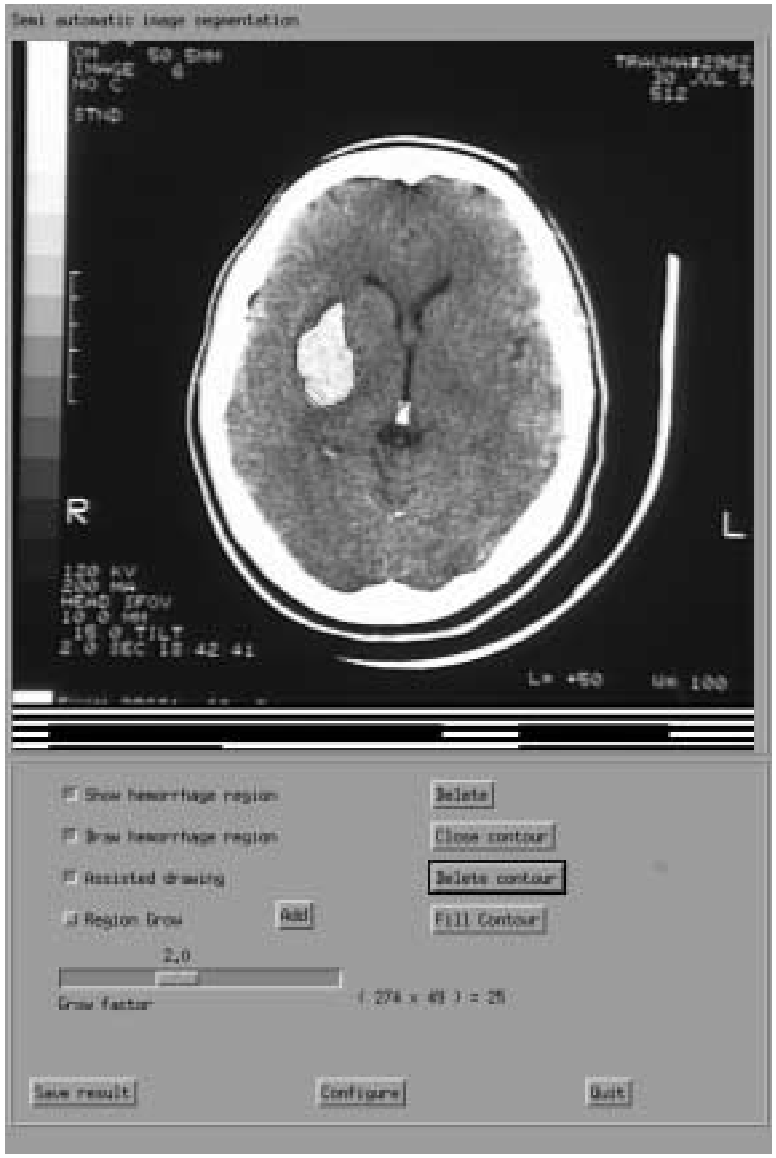

Figure 3. The graphical user interface for the edge-based method.

Figure 4. Segmentation results of the hemorrhage region: edge-based (left) and region-based (right).

5. CONCLUSION

In this work, a novel region and edge-based based methods for semi-automatic image segmentation are presented. In these methods the user interactively selects few points in the region he or she wants to segment (region-based), or on the edge of the region (edge-based). Possible segmentation results are displayed in real time as the mouse pointer is moved. These methods are compared with previously developed similar methods. The advantage of semi-automatic approach over fully automatic segmentation algorithms is that in this approach the user has full control over the segmentation process, but also segmentation is faster then it would be if performed manually. This is important in cases when user has to segment large sets of data, as it is case in medical imaging.

REFERENCES

1. R. Gonzales and P. Wintz, *Digital Image Processing*, Addison Wesley, 1987.
2. T. McInerey and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis* **1**, pp. 91–108, 1996.
3. R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Inteligence* **16**(6), pp. 641–647, 1994.
4. A. Mehnert and P. Jackway, "An improved seeded region growing algorithm," *Pattern Recognition letters* **18**(10), pp. 1065–1071, 1997.
5. A. X. Falcao, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. A. Lotufo, "User-steered image segmentation paradigms: Live-wire and live lane," *Graphical Models Image Processing* **60**, pp. 349–384, 1998.
6. A. X. Falcao, J. K. Udupa, and F. K. Miyazava, "An ultra-fast user-steered image segmentation paradigm: Live wire on the fly," *IEEE Transactions on Medical Imaging* **19**(1), pp. 55–61, 2000.