

## SEMI-AUTOMATIC SNAKE-BASED SEGMENTATION OF CAROTID ARTERY ULTRASOUND IMAGES

Sherif G. Moursi and Mahmoud R. El-Sakka  
*Computer Science Department*  
*University of Western Ontario*  
*London, Ontario, Canada*  
*{smoursi,elsakka}@csd.uwo.ca*

### Abstract

Carotid ultrasound imaging is one of the clinical diagnostic procedures that can be employed to detect plaque buildup at the carotid artery walls. It is an inexpensive and non-invasive procedure that has no known side effects. Yet, the acquired ultrasound images have poor quality and contain a lot of noise.

Active contouring segmentation techniques (also known as snakes or deformable model) are characterized by their robustness to both image noise and boundary gaps. Hence, they are suitable to be used to segment noisy poor quality ultrasound images. One of the major issues of active contouring methods is their sensitivity to the initial contour that is provided by the user. Unless it is drawn close enough to the actual contour, it may lead to unsatisfactory results. Thus, most active contour algorithms require considerable user interaction to provide a good initial contour.

This paper presents an efficient algorithm for extracting carotid artery lumens in ultrasound images. It starts by utilizing a rule-based scheme to generate an initial contour for the lumen. This contour is refined using a snake scheme, after carefully adjusting its energies. Our algorithm reduces the user interaction, as the user is only required to place a seed point inside the region of interest. It is worth mentioning that our proposed initial contour generation scheme can be easily integrated as an independent module with any active contouring algorithm.

Sensitivity, precision rate, and overlap ratio are utilized to assess the performance of the proposed scheme. The results show that the extracted initial and final contours have a good overlap with contours that are manually segmented by an experienced clinician.

**Keywords:** Initial contour, active contouring methods, ultrasound image, carotid artery.

## **1. Introduction**

According to the American Heart Association in 2007 [1], stroke is the third leading cause of death, ranked behind cancer and heart diseases. On average, in the United States of America, every three to four minutes someone dies of a stroke, and each year, about 700,000 people suffer from its effects. The estimated direct and indirect cost of stroke in the United States for 2007 is \$62.7 billion.

According to the Heart and Stroke Foundation of Canada in 2006 [2], over 14,000 Canadians die as a result of strokes every year, and about 300,000 Canadians suffer from its effects.

The carotid arteries supply the head and neck with oxygenated blood. Carotid artery stenosis is a narrowing in the carotid arteries caused by the build-up of plaque (composed of fatty materials, calcium and scar tissues) at the carotid artery walls. This process is called atherosclerosis. When the carotid artery is narrowed, blood flow to the brain can slow down or even stop. If blood flow to the brain is stopped for longer than a few seconds, brain cells could die due to the lack of oxygen, causing permanent damage. Furthermore, plaque causes the blood to flow abnormally, which can cause the blood to clot. Blood clots can break apart and drift through the bloodstream where it is susceptible to cause a blockage at another location causing health problems

Strokes are mainly caused by the interruption of blood flow to the brain due to carotid artery stenosis. Early detection of plaque build up plays an important role in preventing strokes. Currently, carotid angiography is the standard diagnostic technique to detect carotid stenosis. Carotid angiography is an invasive imaging modality where the patient is injected with an X-ray dye. Then the carotid artery is examined using X-ray imaging. It takes about one hour to perform the test, however the patient has to remain under observation for 6 hours after the test is performed. In general, carotid angiography is an uncomfortable procedure for patients and has some risk factors, including allergic reaction to the injected dye, renal failure, the exposure to ionic radiation, as well as arterial puncture site complications, e.g., pseudoaneurysm and arteriovenous fistula formation.

Ultrasound imaging is a non-invasive medical imaging modality. Carotid ultrasound images are obtained using a probe that emits ultrasound waves. This probe is moved back and forth over the patient's neck to obtain different views of the carotid artery. The entire process takes few minutes and is completely non-invasive. There are no known side effects for using ultrasound imaging. Furthermore, the device used in this process is portable and relatively inexpensive. Ultrasound imaging is considered a well-established clinical diagnostic technique for detecting carotid artery stenosis. The main problem with ultrasound imaging is the fact that it produces poor quality images containing a lot of noise.

In order to detect carotid stenosis, clinicians typically use a manual process in which the borders of the carotid artery in ultrasound images are traced. However, due to the poor quality of ultrasound images, it takes a considerable amount of time and effort from clinicians to delineate manually the borders of the carotid artery. Furthermore, carotid artery manual segmentation is generally a non-reproducible process and its results may greatly vary from one clinician to another, due to their experience and subjective judgment. When dealing with a large database of ultrasound carotid artery images, such manual processes become impractical. Hence, it takes considerable time and effort from clinicians to extract information from these images. Because of this, many researchers attempt to develop automated methods for extracting carotid artery boundaries. These methods include texture-based segmentation, edge-based segmentation, region-based segmentation, and active contouring.

## **2. Related Research**

Hamou *et al.* [3] proposed to segment carotid artery contours using histogram equalization and Canny edge detection followed by morphological opening as a post processing operation. However, their method produces many false segmented regions.

Abdel-Dayem *et al.* [4] proposed a carotid artery segmentation scheme that utilizes histogram equalization and median filtering as pre-processing operations followed by uniform quantization to partition the artery into three regions: area inside the artery, artery wall and a plaque layer. The method

produces many false segmented regions and it may not work well on all images since the quantization levels are empirically selected.

Abdel-Dayem *et al.* [5] also proposed another segmentation scheme that utilizes wavelet transforms and watershed segmentation in the multi-resolution domain. However, this method requires the user to specify an image dependent threshold value to merge regions resulting from watershed segmentation.

Active contouring (also known as snakes or deformable model) was first introduced by Kass *et al.* [6]. Active contouring treats the surface of the object as an elastic sheet that stretches and deforms when external and internal forces are applied to it. The behaviour of the active contouring model (snake) mimics the physical laws that govern real-world objects. A snake is an energy minimization problem, where two opposing forces control its behaviour. These forces are *the internal force* (which imposes the smoothness of the snake), and *the external force* (which guides the snake towards the image features). The total energy of the snake is defined as:

$$\text{Total Energy} = \int_{s=0}^{s=1} E_{in}(v(s)) + E_{ex}(v(s)) ds$$

where  $E_{in}$ ,  $E_{ex}$  are the internal and external energies of the snake respectively,  $v(s)$  is the parametric representation of the snake, and  $s$  is a real number between 0 and 1. The snake adjusts itself to fit a desired contour by minimizing its total energy in order to reach equilibrium between the two opposing forces.

The internal energy of the snake is defined as:

$$E_{in}(v(s)) = \alpha(s) \cdot \left( \frac{dv(s)}{ds} \right)^2 + \beta(s) \cdot \left( \frac{d^2v(s)}{ds^2} \right)^2,$$

where  $\alpha(s)$  is called the first order term, which makes the snake act like a membrane, and  $\beta(s)$  is called the second order term, which makes the snake act like a thin plate. The external energy is a function that drive the snake towards the required image features, such as contours. Various internal and external energies can be used to drive the snake [7][8][9].

The main advantage of active contouring is its robustness to both image noise and boundary gaps. Hence, it is widely used in segmenting medical images. However, active contouring has a major drawback, due to its great sensitivity to the initial contour that should be provided by the user. Unless the snake is drawn close enough to the actual contour, it will converge leading to unsatisfactory results. Thus, most active contouring algorithms require considerable user interaction to provide a good initial contour.

Cheng *et al.* [10] utilized a snake model to automatically detect the intimal and adventitial layers of common carotid artery. They modified the balloon snake proposed by Cohen [7], and added spatial criteria to enhance the edges of the adventitia. However the snake requires a long time to converge to the final contour.

Mao *et al.* [11] used a thresholded Euclidean distance map and morphological operations in order to estimate an initial contour for the carotid artery based on a seed point placed by the user. This initial contour is then refined using the active contour's external and internal forces. However, this method is very time consuming. Moreover, it requires the seed point to be placed around the centre of gravity of the actual contour; otherwise, the method will fail to get a good initial contour. In addition, this method is only tested on a very small set of images.

Loizou *et al.* [12] proposed a snake-based segmentation technique for detecting the intima-media layer of the common carotid artery wall in longitudinal ultrasound images. The user must first crop the area of the carotid artery. The method then applies normalization and speckle noise reduction via a linear scaling filter, together with thresholding and morphological operations in order to generate an initial snake that is further refined using snake forces. Yet, this method requires considerable user interaction in order to correctly crop the area around the carotid artery contour.

Hamou *et al.* [13] proposed incorporating ellipse shape enforcement and a modified Canny edge detection into the active contour's external energy. The user has to place at least three points on the carotid artery walls. A least square ellipse fitting technique is used to generate the initial contour that is further refined using the internal and proposed external forces. Yet, unless the initial seed points are placed exactly on the carotid artery walls, the method will fail to generate a good contour.

Moursi *et al.* [14] proposed a preliminary algorithm that generates an initial contour for carotid artery lumen in ultrasound images. The proposed algorithm reduced the active contour's sensitivity to initial contours by reducing user interactions to just supplying a single seed point anywhere inside the artery. It utilizes a rule-based strategy in order to generate a contour that is close enough to the actual lumen edge.

### 3. Problem Definition

Active contouring is characterized by its robustness to both image noise and boundary gaps. Hence, it is suitable to be used as a segmentation technique in noisy poor quality ultrasound images. However, active contouring methods are very sensitive to the initial contour that is provided by the user. Unless it is drawn close enough to the actual contour, it may lead to unsatisfactory results. Thus, most active contouring algorithms require considerable user interaction to provide a good initial contour. Accordingly, having an accurate initial contour without too much user interaction will have a positive impact on the performance of any active contouring algorithm. It will also reduce the massive amount of time required for the user to interact to provide the initial contour.

### 4. Proposed Method

In this paper we utilize and improve our preliminary work [14] to generate a more accurate initial carotid artery contour using a single seed point that can be placed anywhere inside the region of interest. In addition, we extend the algorithm by incorporating a simple active contouring algorithm to refine the generated initial contour. Figure 1 shows the block diagram of our algorithm.

#### 4.1 Initial Contour Generation

This stage utilizes a priori knowledge reflecting the fact that the shape of the carotid artery

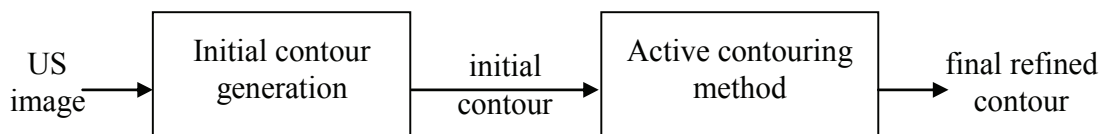


Figure 1: Block diagram of our algorithm.

boundaries is an ellipse. It first removes unnecessary borders from the image. Then a connected region that is a rough estimate of the carotid lumen is extracted. An additional island removal technique is applied to remove islands and noise found inside the extracted connected region. Finally, an ellipse-shaped initial contour that is very close to the carotid artery boundaries is generated. Figure 2 shows a block diagram of our improved initial contour generation algorithm.

#### **4.1.1 Borders Removal**

Carotid ultrasound images usually have a black border around the actual image. This border contains information about the image that facilitates the diagnosis procedure. However, on an automated carotid segmentation process, this border may cause some problems, especially when calculating the histogram of the image. This step of the algorithm locates this border in carotid ultrasound images in order to deal with the actual carotid artery region. To locate the left and right margins, we calculate the number of distinct grey levels in each vertical scan line. Since the border area usually has a low contrast with very few activities (mainly background, letters, symbols, and lines), it will possess very few distinct grey levels in each scan line (not more than three). Hence, by examining the number of distinct grey levels from left and from right sides, we can determine the left and right margins where the activity begins in the image. Similarly, to identify the top and bottom margins, the same procedure mentioned above is performed horizontally. Figure 3 shows a plot for the number of grey levels at each vertical and horizontal scan line in an image, where the dotted lines represent the margins of the actual scan of the carotid artery.

#### **4.1.2 Finding Connected Region**

The goal of this stage is to utilize the seed point placed by the user in order to extract a connected region that is considered a rough estimate of the carotid artery boundary. Figure 4 shows the block diagram of the *finding connected region* algorithm. To find the connected region we do the following:

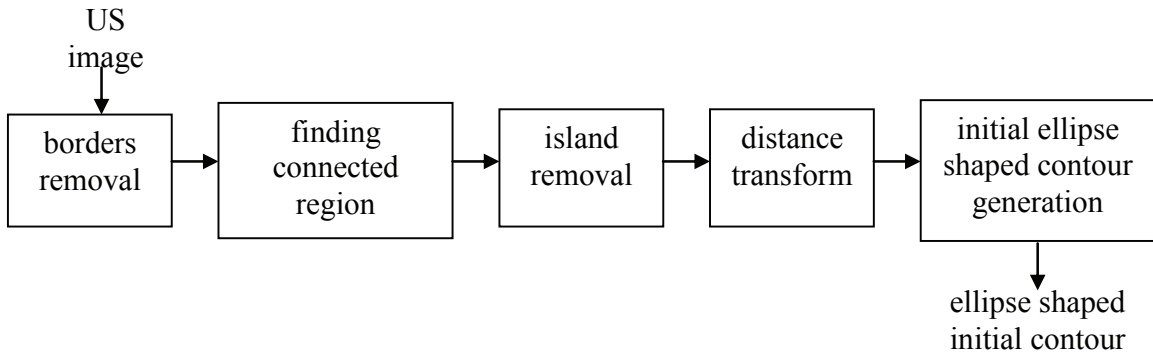


Figure 2: Block diagram of initial contour determination algorithm.

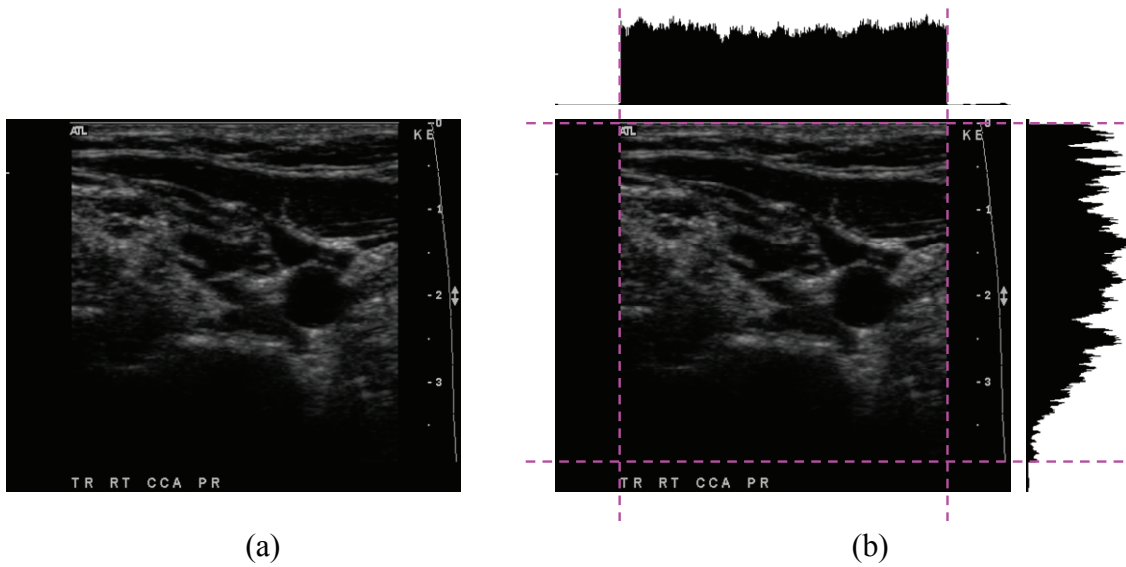


Figure 3: The number of grey levels at each vertical and horizontal scan line, where the dotted lines represent the margins of the actual scan of the carotid artery; (a) original image and (b) a plot for the number of grey levels at each vertical and horizontal scan line, as well as the four margins.

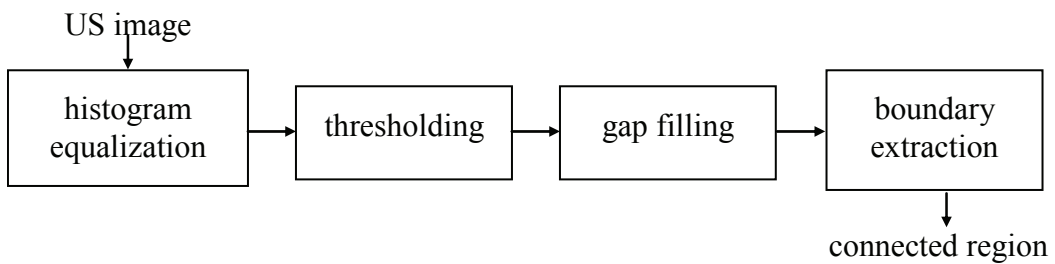
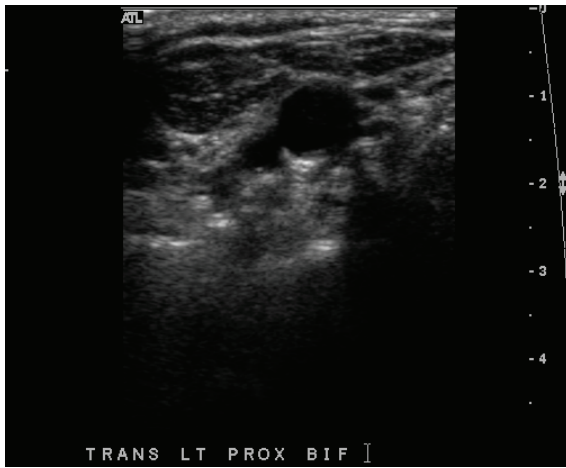


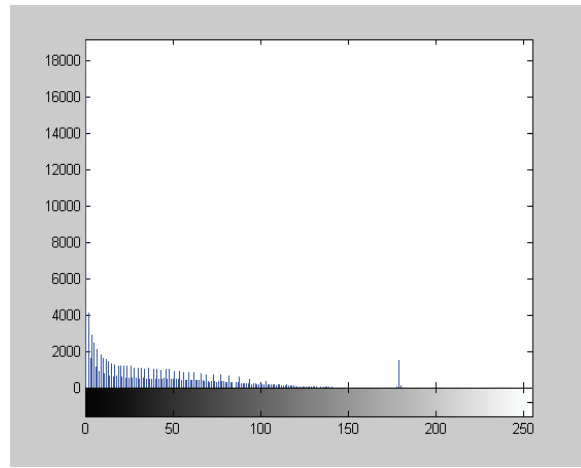
Figure 4: Block diagram of finding connected region algorithm.



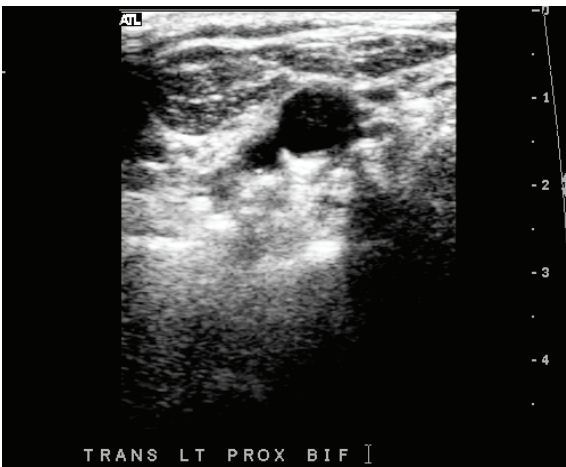
- i) *Histogram equalization*: Carotid ultrasound images usually have poor contrast. The goal of this step is to improve the contrast of the image by increasing the dynamic range of grey levels [15]. Figure 5 presents an example of the effect of histogram equalization on a poor contrast image. Figure 5(b) shows the histogram of the original image shown in Figure 5(a), where most of the grey level values are located at the dark part of the histogram. However, after applying histogram equalization, the contrast of the image has improved and the dynamic range of grey levels has increased, as shown in Figure 5(c) and Figure 5(d), respectively.
- ii) *Thresholding*: The main purpose of this step is to partition the image into foreground (containing the region-of-interest) and background regions. In this work, an image dependent global thresholding scheme is used. The threshold value is calculated as a weighted average of a  $101 \times 101$  block centered at the seed point. The more the pixel is closer to the seed point, the higher the weight is. Hence, the seed point is given the highest weight. The weights given to each pixel in the block is calculated using Gaussian distribution that is sampled with step size  $= \frac{\sigma}{3}$ . Thus, it will take nine steps for the Gaussian distribution to reach 1% of its maximum. After that, the weights are set to a constant equals to that 1% value. At the end, these generated weights are normalized. The value of the step size used in the Gaussian distribution, as well as the value of the block size,, were chosen after careful analysis (See Section 8.1 for details).
- iii) *Gap filling*: This step removes tiny gaps that are less than six pixels, if any. This is done by horizontally and vertically scanning the image with  $6 \times 1$  and  $1 \times 6$  masks, respectively, and filling any detected gaps on the contour, where a gap is any tiny black area between two white pixels. Once a gap is detected, it is filled with white color. Various mask sizes were tested and the best results were obtained by using the  $6 \times 1$  and  $1 \times 6$  masks (See Section 8.2 for details).



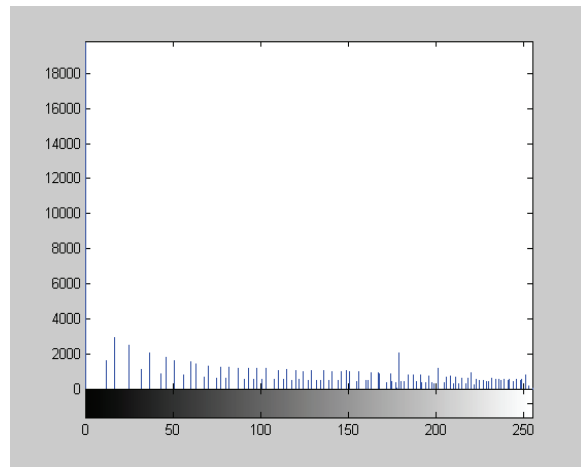
(a)



(b)



(c)



(d)

Figure 5: The effect of histogram equalization; (a) original image, (b) histogram of original image, (c) original image after histogram equalization and (d) histogram of the image after applying histogram equalization.

iv) *Boundary extraction*: A region growing scheme, starting from the seed point placed by the user, is applied to extract the internal boundaries of the region of interest (carotid artery boundaries in our case). The criterion used for region growing is the equality of the grey level value. Figure 6 shows an example of the extracted boundary using the proposed region growing scheme. As shown on Figure 6(a), the region growing scheme expands the region around the seed point by adding neighbouring pixels that have equal grey level (black). Accordingly, the internal boundaries of the connected region around the seed point are extracted as shown on Figure 6(b). Hence, all unneeded information that was outside the connected region is eliminated from the image.

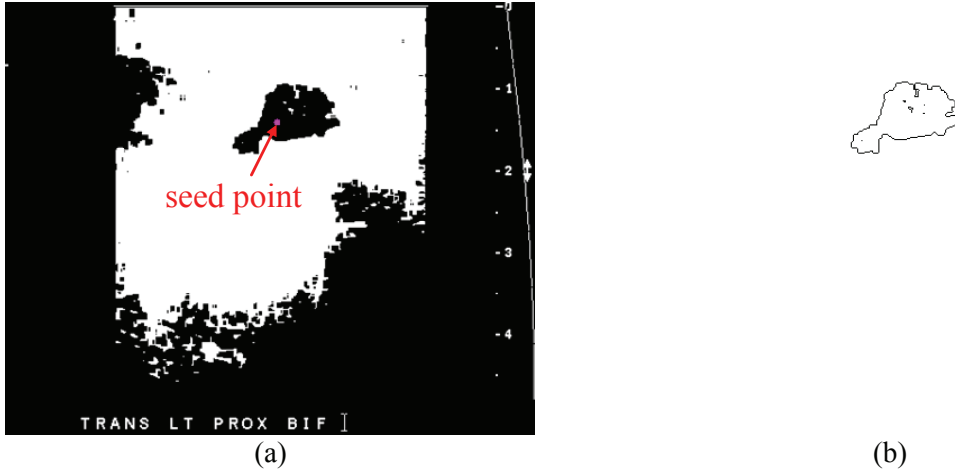


Figure 6: Results of the boundary extraction step; (a) image resulting from gap filling step. The seed point is placed inside the region of interest and (b) extracted boundary using region growing.

The extracted connected region represents a rough estimate of the carotid lumen boundaries. Due to the poor quality of ultrasound images, the connected region may contain some leaks. Furthermore, some islands may appear inside the region of interest. These islands appear either due to noise, or in some cases, it may be actual plaque accumulations.

#### 4.1.3 Island Removal

Islands inside the extracted connected region may cause problems in our initial contour determination algorithm. Hence, we incorporated this step to remove such islands. To do so, we first determine an external point outside the connected region. This is done automatically by calculating the following locations on the connected region:

- *left edge*: represents the x-coordinate of the left most contour point at the connected region,
- *right edge*: represents the x-coordinate of the right most contour point at the connected region,
- *upper edge*: represents the y-coordinate of the upper most contour point at the connected region, and
- *lower edge*: represents the y-coordinate of the lower most contour point at the connected region.

Thus, the following points are guaranteed to be outside the connected region and are considered candidate external points:  $(left\ edge - 1, lower\ edge + 1)$ ,  $(left\ edge - 1, upper\ edge - 1)$ ,  $(right\ edge + 1, lower\ edge + 1)$ , and  $(right\ edge + 1, upper\ edge - 1)$ . Figure 7 shows an example of the candidate external

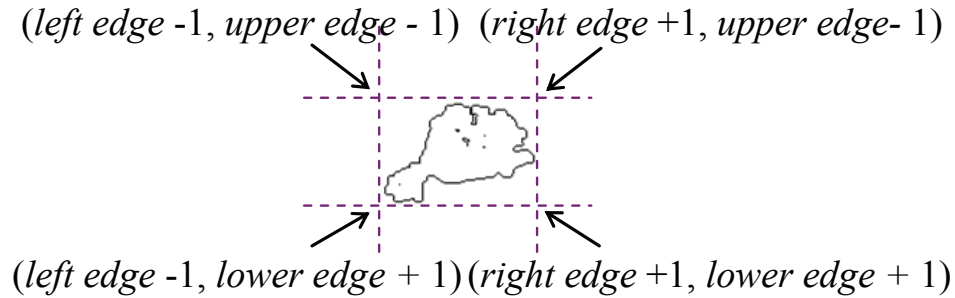


Figure 7: The four candidate external points.

points. Some of these points may lie outside the actual image. Hence, our external point is chosen as any point among these four candidate external points which lies inside the image.

After the external point is determined, we apply the same boundary extraction scheme used before, but starting from the determined external point. By matching external and internal boundary points, any contours inside the connected region (including islands) can be identified and removed leading to an island-free connected region.

#### 4.1.4 Distance Transform

The algorithm proposed by Felzenszwalb et al. [16] is utilized to generate the Euclidean distance map relative to the edges in the connected region. This distance map determines the Euclidean distance from each pixel to the closest point on the contour. The smaller the value of the distance map at a certain pixel, the closer to a contour the pixel is, where the value of the distance map at a contour pixel is zero. The distance map will be used as an input to the initial ellipse shaped contour generation algorithm. Figure 8 shows sample intermediate outputs of the previously discussed steps in our algorithm.

#### 4.1.5 Initial Ellipse Shaped Contour Generation

Since we assumed having a prior knowledge reflecting the fact that the shape of the carotid artery boundary is an ellipse, our goal in this step is to determine the best horizontal and vertical axis lengths that can be used to draw an ellipse that is as close as possible to the carotid artery lumen boundaries.

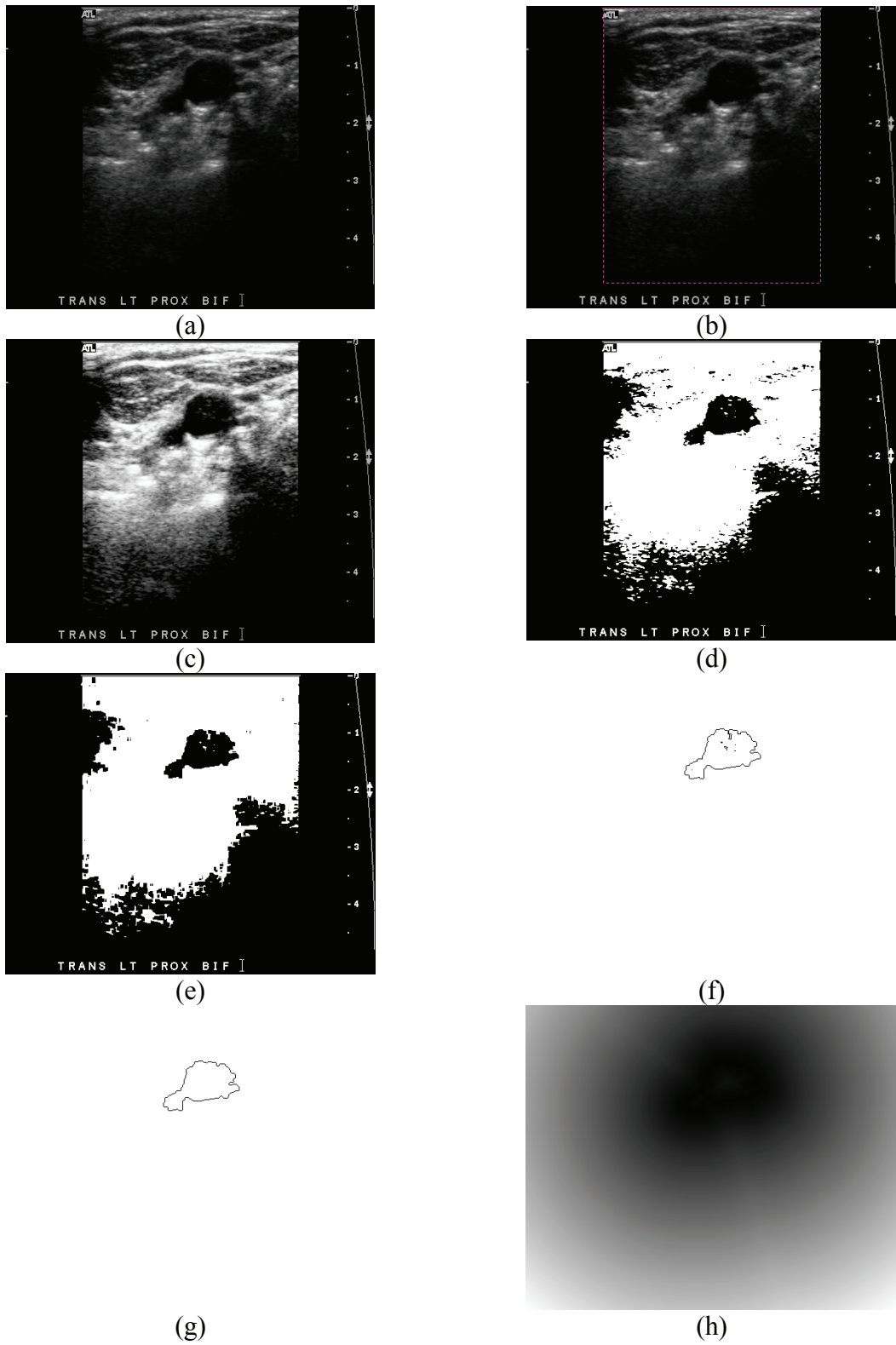


Figure 8: Sample intermediate outputs of various steps of our algorithm; (a) original image, (b) boundary removed, (c) histogram equalized, (d) thresholded, (e) gaps filled, (f) extracted connected region, (g) islands removed and (h) Euclidean distance map.

Centered at the seed point that is provided by the user, we draw an initial circle of diameter 5 pixels. Then, we gradually increase the vertical axis length, while counting the number of ellipse points that touch the contour at each value. The objective here is to determine the best axis length that maximizes this number. Once we achieved this objective, we horizontally repeat this process in order to determine the best horizontal axis length. Figure 9 shows an example of the process of determining the lengths of the ellipse horizontal and vertical axes.

Axis length determination alone is insufficient to generate a good initial contour, since it requires the central point of the ellipse to be located at the center of gravity of the contour. Figure 10 shows an example of an ellipse generated using *generating initial ellipse shaped contour* method when the seed point is placed away from the center of gravity of the contour. Thus, in some situations we have to re-adjust the location of the ellipse central point by moving it towards the center of gravity of the contour to make sure that we reached a stable local energy minimum.

After determining ellipse axis lengths, an ellipse is drawn using these axes. A convergence check is then performed to test whether or not we need to adjust the location of the ellipse central point. In order to do this, we identify four regions in our ellipse (*left, right, upper* and *lower*) as shown on Figure 11 where  $h$  and  $k$  are the horizontal and vertical coordinates of the ellipse central point, respectively,  $a$  and  $b$  are the length of horizontal and vertical axes, respectively. Our convergence check is based on the average distance map value for each region, which is calculated as follows:

$$\text{Average distance (region)} = \frac{\sum_{i=1}^n \text{distance}(p_i)}{n},$$

where *region* can be *left, right, upper, or lower*;  $p_i$  is the  $i^{\text{th}}$  point in the region,  $\text{distance}(p_i)$  is the value of the distance map at point  $p_i$ , and  $n$  is the number of discrete points in the region. The farther the region from the boundaries, the greater the average distance is. In order to have a good ellipse shaped contour that is as close as possible to the carotid artery lumen, each two opposite regions should lie almost at the same distance away from the carotid artery boundary.

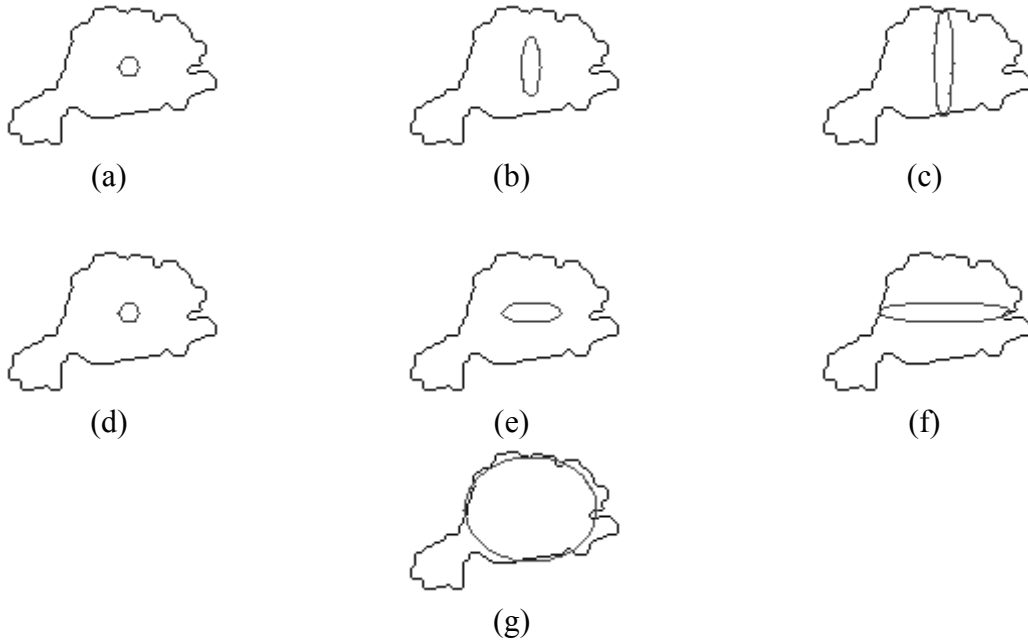


Figure 9: An example of the process of determining the lengths of the ellipse horizontal and vertical axes; (a) initial circle at the seed point, (b) gradually increase vertical axis, (c) vertical axis that maximize the number of ellipse points that touch the contour, (d) initial circle at the seed point, (e) gradually increase horizontal axis, (f) horizontal axis that maximize the number of ellipse points that touch the contour and (g) the produced ellipse.



Figure 10: An ellipse generated using the axis length determination when the seed point is placed away from the center of gravity.

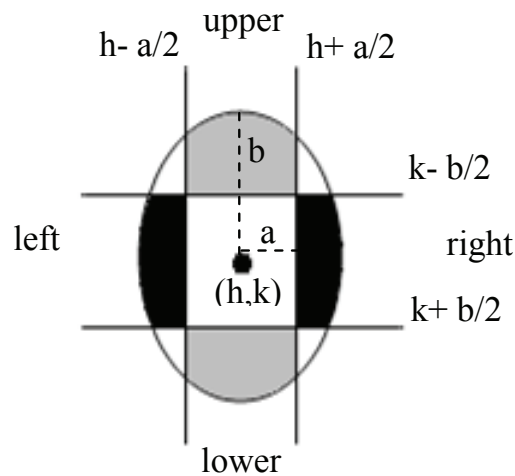


Figure 11: Dividing the ellipse into four regions (left, right, upper and lower).

```

if (Average distance(left) – Average distance(right)) > threshold then
    move the central point one pixel to the left
else
    if (Average distance(right) – Average Distance(left)) > threshold then
        move the central point one pixel to the right
    endif
endif

if (Average distance(upper) – Average distance(lower)) > threshold then
    move the central point one pixel upwards
else
    if (Average distance(lower) – Average distance(upper)) > threshold then
        move the central point one pixel downwards
    endif
endif.

```

Figure 12: Our convergence check algorithm.

Our convergence check is done by calculating the difference between the average distances of each two opposite regions in the ellipse. If the difference is greater than a certain threshold, we move the ellipse central point by one pixel to a location which minimizes that difference as shown in Figure 12.

In the example shown on Figure 10, the left and upper regions of the ellipse are closer to the contour than their opposite right and lower regions, respectively. As a result of this, the ellipse central point is moved one pixel to the right and one pixel downwards.

The threshold value used in the convergence check algorithm has a great effect on the algorithm performance. Low threshold values produce more accurate initial contours, as well as decrease the algorithm's sensitivity to the location of the seed point. However, due to the irregularity of some contours, low threshold values may cause the algorithm to move the central point to a previously visited pixel resulting in entering an endless loop. To prevent this situation, we use a loop prevention technique that keeps track of all visited central points by the algorithm and stops the algorithm when a certain central point is revisited.



1. Set the ellipse central point to the seed point placed by the user.
2. Starting with a circle of diameter 5, determine:
  - 2.1 the best vertical axis length that minimizes the ellipse energy.
  - 2.2 the best horizontal axis length that minimizes the ellipse energy.
3. Draw an ellipse using the determined axis lengths.
4. Do a convergence check to see whether we need to change the location of the ellipse central point or not.
5. If the ellipse central point needs to be adjusted then
  - 5.1 set the ellipse central point to the new location
  - 5.2 if the new central point was not previously visited then
    - 5.2.1 Go to step 2.
6. Draw the final ellipse-shaped contour from the ellipse central point using the determined vertical and horizontal axis lengths.

Figure 13: pseudo-code of *initial ellipse shaped contour generation* algorithm.

We then use the new central point to recalculate the best horizontal and vertical axis lengths and repeat the same process. We continue this iterative process until an ellipse that passes the convergence check is generated. This would be the resulting contour of our algorithm. Figure 13 shows a pseudo-code of the entire Initial Ellipse Shaped Contour Generation step. Figure 14 shows some examples of initial contours generated from our algorithm.

## 4.2 Active contouring

After our initial contour is generated, it is refined using the active contouring (snake) energy model. We use a discrete snake model at which the continuous snake curve is sampled into discrete snake nodes or control points. The internal energy of our snake model is defined as:

$$E_{in} = \alpha \cdot \sum_{i=0}^{n-1} |v_{i+1} - v_i|^2,$$

where  $\alpha$  is the weight of the elasticity of the snake,  $n$  is the number of discrete snake nodes, and  $v_i$  is the  $i^{\text{th}}$  snake node. This force makes the snake acts like a membrane.

On selecting our external energy that drives the snake towards the required contour, many factors were taken into consideration, including:

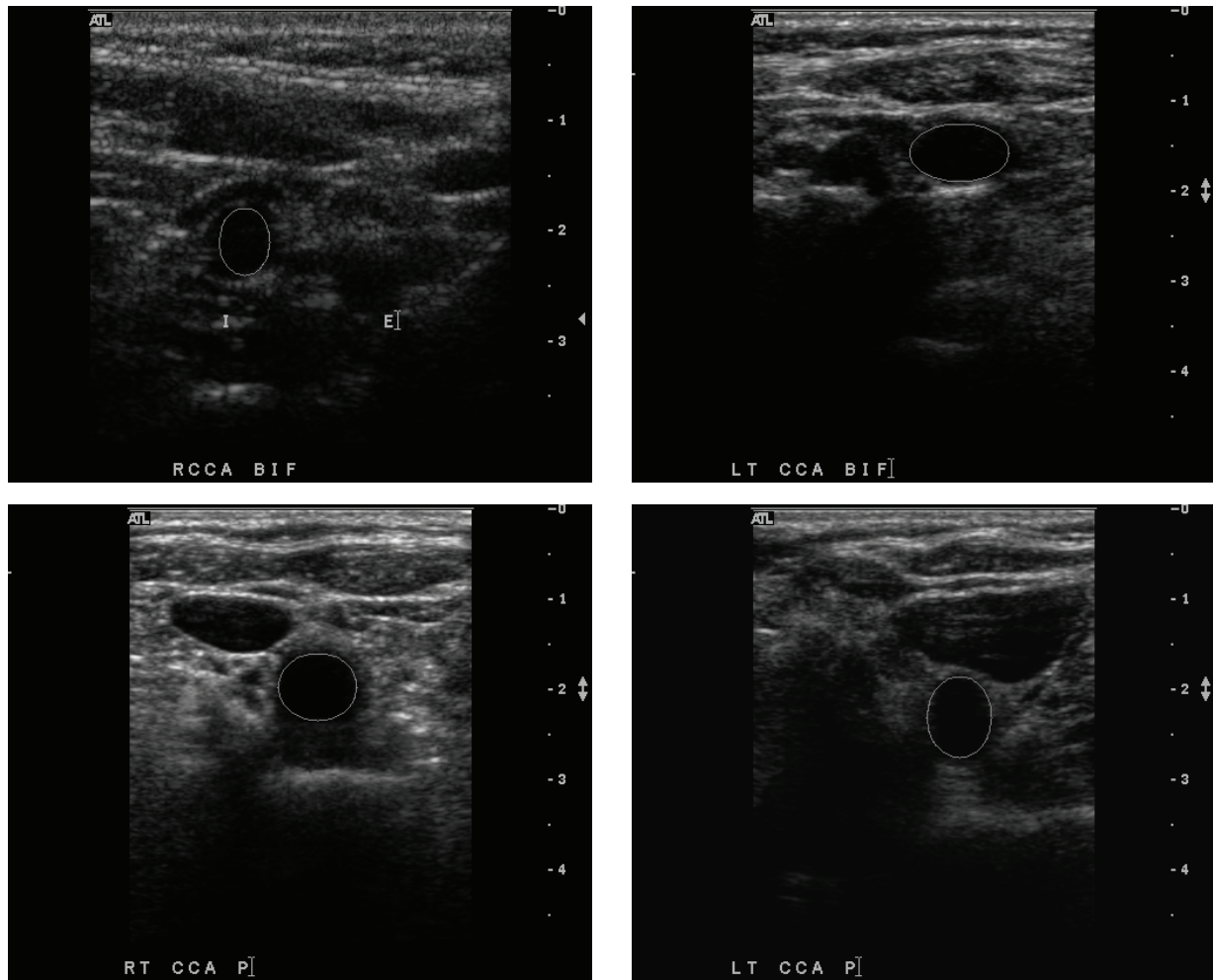


Figure 14: Examples of initial contours generated from our algorithm.

1. The extracted connected region should have a very good overlap with the actual carotid artery contour. The only problem in the connected region is the presence of gaps causing the contour to leak. Hence, the main purpose of using active contouring is to fill the gaps in the connected region.
2. Our generated initial contour is very close to the connected region boundaries. Hence, it will quickly converge to fit the connected region even when using a simple external energy.
3. The image representing the extracted connected region is a binary image with almost no noise.

Accordingly, the external energy of our snake model is chosen as the Euclidean distance map of the extracted connected region and is defined as:

$$E_{ex} = \sum_{i=0}^n \text{Distance}(v_i),$$

where  $n$  is the number of discrete snake nodes,  $v_i$  is the  $i^{\text{th}}$  snake node, and  $\text{Distance}(v_i)$  is the value of the Euclidean distance map of the image at point  $v_i$ . Since our initial contour is so close to the connected region, using the Euclidean distance as an external energy is sufficient to generate a refined contour that fills the gaps in the connected region.

The weight of the internal energy,  $\alpha$ , should be carefully adjusted to accurately refine the initial contour. If a very low weight value is used, the snake will be strongly attracted towards the image features. In addition, it would cause the snake to bend forming non-smooth final contours. On the other hand, if a very high weight value is used, the resulting contour will be more rigid and smooth. However, it may converge away from the required image features, generating inaccurate final contours. Various values of  $\alpha$  have been considered. We empirically found that at  $\alpha = 0.25$ , the algorithm generates the best performance.

#### 4.2.1 Snake deformation

Many techniques can be used to minimize the snake energy. Kass *et al.* [6] proposed viewing the snake energy as a pair of Euler equations and solving these equations iteratively on the discrete grid. Other methods can be used such as gradient descent [17] and dynamic programming [18][19]. Dynamic programming is considered a fast and efficient method for quickly directing the snake towards the desired contour. Hence, for minimizing the snake energy in our algorithm we use dynamic programming.

#### 4.2.2 Snake resampling

During the snake deformation, some snake nodes may either overlap or get too far from each other causing problems for the snake convergence. Thus, it is important to re-adjust the resolution of the snake model by resampling the snake after each iteration. We use the same resampling technique proposed by Lobregt *et al.* [9].

Resampling is implemented as a two-pass process. In the first pass, we check to see if the Euclidean distance between any two successive nodes is less than a minimum length. If this is the case, the two nodes are removed and replaced by the midpoint of the line joining those two nodes. The second pass checks to see if the Euclidean distance between any two successive nodes is greater than a maximum length. In this case, an additional node is added at the midpoint of the line joining the two nodes.

In order to prevent the occurrence of oscillation, where the same nodes are repeatedly removed in the first pass, then inserted on the second, the by *maximum distance* should be at least twice the *minimum distance*. Lobregt *et al.* [9] proposed the relation between the minimum and maximum distance should be:

$$\textit{Maximum distance} = 1.5 \times D,$$

and

$$\textit{Minimum distance} = 0.5 \times D,$$

where  $D$  is the desired spacing between the snake nodes. Various values of  $D$  have been considered. We imperically found that at  $D = 3$ , the algorithm generates the best performance. Hence, the values of the minimum and maximum distances are chosen as 1.5 and 4.5 pixel units, respectively.

Figure 15 shows an example of the contours generated using our algorithm compared to contours that are manually segmented by an experienced clinician.

## 5. Experimental Setup

To test the accuracy of our algorithm, we obtained two set of images as follows:

- **Dataset 1:** consists of 70 ultrasound carotid artery images acquired using a Philips HDI 5000 ultrasound machine. The resolution of these images is 504×412 pixels. Those images represent patients with various levels of plaque on the carotid arteries.
- **Dataset 2:** consists of 39 images acquired using Ultramark 9 HDI ultrasound machine and L10-5 linear array transducer. Images were digitized by a high-speed frame grabber and were saved to disk. The resolution of these images is 400×235 pixels.

In both set of images, the carotid artery lumen boundaries are manually delineated by an experienced clinician.

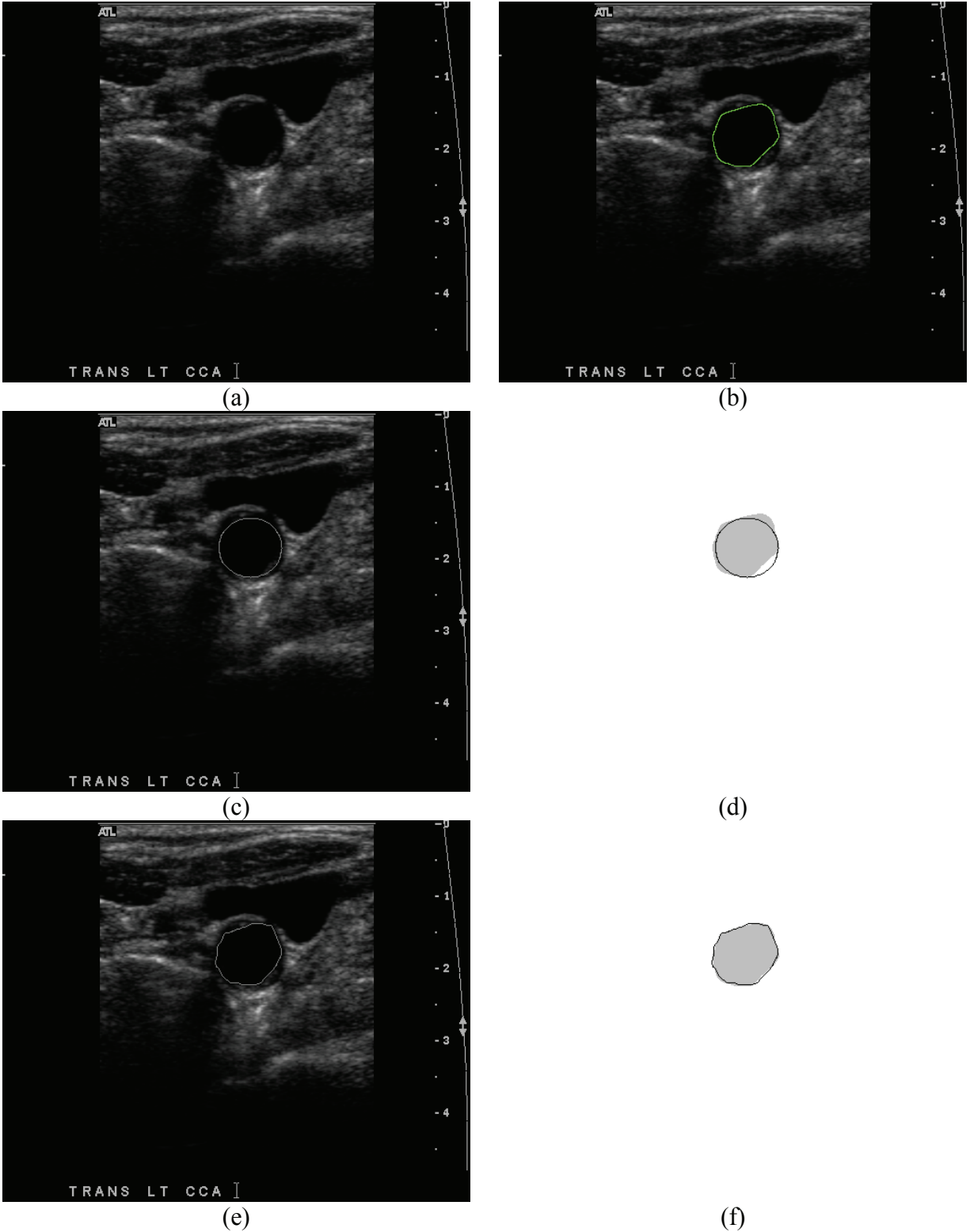


Figure 15: Example of the contours extracted using our algorithm; (a) original image, (b) contour extracted manually by a clinician, (c) generated initial contour, (d) generated initial contour (black) superimposed on the contour extracted manually by a clinician (grey), (e) final refined contour and (f) final refined contour (black) superimposed on the contour extracted manually by a clinician (grey).

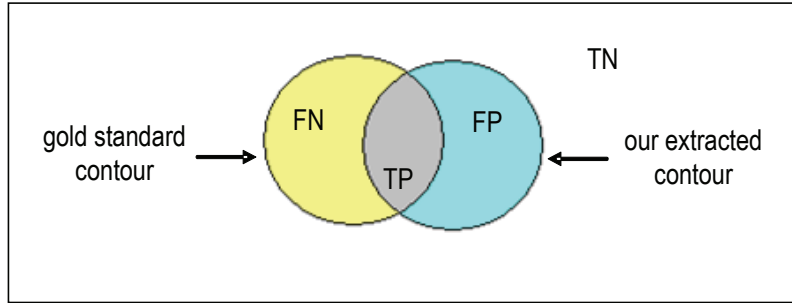


Figure 16: The definition of the true positive (TP), false positive (FP), true negative (TN) and false negative (FN) terms.

## 6. Performance Measures

To measure the accuracy of our extracted contours we compare it with gold standard images using three performance measures, namely: *sensitivity* [20], *precision rate* [21], and *overlap ratio* [22]. To calculate these measures we first determine some areas on the resulting images, namely: true positive (TP), false positive (FP), true negative (TN), and false negative (FN), as shown in Figure 16.

TP indicates the area which our segmentation algorithm correctly detected as being part of the carotid artery. FP is the area which our segmentation algorithm incorrectly detected as being part of the carotid artery. TN is the area which our segmentation algorithm correctly detected as not being a part of the carotid artery. FN is the area which our segmentation algorithm incorrectly detected as not being a part of the carotid artery. Our performance measures are then calculated as follows:

$$Sensitivity = \frac{TPs}{TPs + FN_s} \times 100 \%,$$

$$Precision\ rate = \frac{TPs}{TPs + FP_s} \times 100 \%,$$

and

$$Overlap\ ratio = \frac{TPs}{TPs + FN_s + FP_s} \times 100 \%,$$

where TPs, FP<sub>s</sub>, and FN<sub>s</sub> are the sum of TP, FP, and FN respectively, for all the images in the dataset.

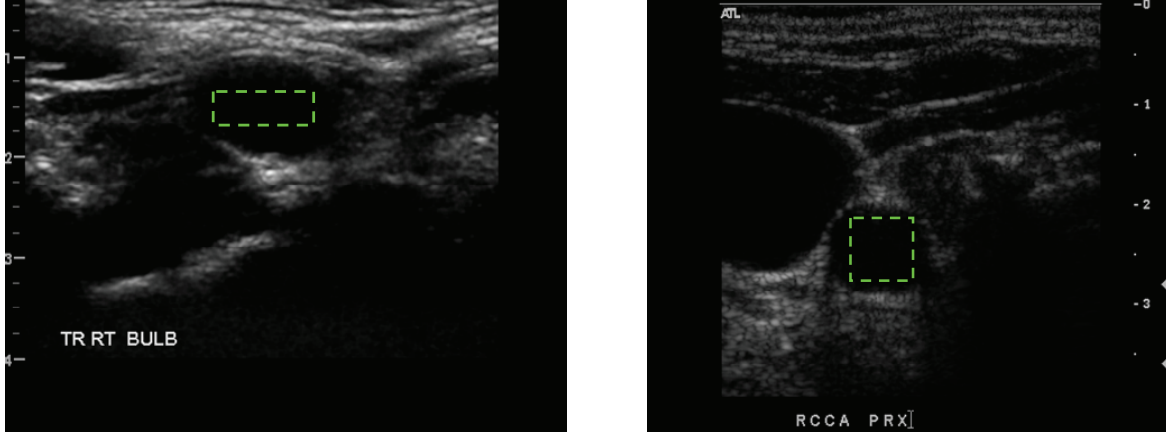


Figure 17: Examples of the seed point grids used in our experiments.

## 7. Experimental Results

To assess the performance of our algorithm, we ran our experiments using both sets of images mentioned in Section 5. In order to take into consideration the effect of changing the location of the seed point, the TP, FP, and FN for each image are measured as the average values when starting from various seed points using a grid of seed points. The size of the grid varies according to the size and shape of the carotid artery. Figure 17 shows examples of the seed point grids used in our experiments.

For our initial contour, using dataset 1, we obtained 82.4% for sensitivity, 89.6% for precision rate, and 75.3% for the overlap ratio. Meanwhile, using dataset 2, we obtained 87.1% for sensitivity, 89.3% for precision rate, and 78.9% for the overlap ratio.

However, after refinement, our final contour shows a great performance improvement. Using dataset 1, we obtained 90.5% for sensitivity, 91.6% for precision rate, and 83.6% for the overlap ratio. Meanwhile, using dataset 2, we obtained 94.4% for sensitivity, 86.7% for precision rate, and 82.5% for the overlap ratio. Table 1 shows the obtained results for both our initial and final contours.

In terms of the computational time, it takes, on average, 80 milli-seconds (with standard deviation of 30 milli-seconds) and 67 milli-seconds (with standard deviation of 37 milli-seconds) to generate a single initial and final contours, respectively (using P4 Core 2 Duo processor 2.4 GHZ, 1 GB of RAM, and Windows XP operating system). The time varies depending on the size of the carotid artery and the location of the seed point.

Table 1: Initial and final results obtained using our algorithm.

		<b>Sensitivity</b>	<b>Precision rate</b>	<b>Overlap ratio</b>
<b>Dataset 1</b>	<b>Initial contour</b>	82.4	89.6	75.3
	<b>Final contour</b>	90.5	91.6	83.6
<b>Dataset 2</b>	<b>Initial contour</b>	87.1	89.3	78.9
	<b>Final contour</b>	94.4	86.7	82.5

## 8. Algorithm Analysis

This section provides analysis of the effect of the various processing steps on the performance of our algorithm. To obtain quantitative performance measures in our comparisons, we used 25 images, which were randomly selected from both datasets. Each image is tested using a grid of seed points.

### 8.1 Thresholding

The image dependent threshold used in Section 4.1.2 has a great influence on the accuracy of our generated contours. A very high threshold value results in over-segmented contours, while a very low threshold value results in under-segmented contours. The threshold value is calculated as a weighted average of a  $101 \times 101$  block centered at the seed point. Since the seed point is placed inside the carotid artery boundaries, more weights should be given to this area compared to the rest of the image.

The block size affects our calculated threshold value, increasing this size would add more influence of pixels that are far away from the seed point. After examining our datasets, we noticed that the average size of the carotid artery is 2500 pixels. Since we want our block to cover both the carotid artery and the background, we chose a block size of  $101 \times 101$ . This will make the carotid artery acquire 25% of our block while the background acquires 75%. It is worth mentioning that more weights will be given to the area around the seed point (where the carotid artery is located).

The step size used in sampling the Gaussian distribution has a great influence in the calculated the threshold value. Larger step sizes would cause the Gaussian distribution to quickly diminish. Thus, the contribution of area around the seed point would get lower weights. On the other hand, smaller step sizes mean that the Gaussian distribution will take many steps to diminish. Thus, similar weights would be



given to areas around and far away from the seed point, without giving enough preference to the area around the seed point. Figure 18 shows the effect of using various step sizes on the performance of our algorithm. It is clear that using large step sizes leads to over-segmented contours (i.e., high sensitivity and low precision rate values). As the step size decrease, the sensitivity decreases and the precision rate increases. The obtained results show that using  $\sigma/3$  as our step size produces the best performance.

## 8.2 Gap Filling

Gaps in the connected region may negatively affect our extracted contours. For example, consider the contour shown in Figure 19. The grey area represents the optimal vertical and horizontal axes that would be used to draw an optimal ellipse. The few pixels wide regions at the endpoints of these axes are called the critical regions. The presence of a gap in the carotid artery boundaries at these critical regions may cause the algorithm to produce false initial contours if the user placed the seed point very close to this gap.

Our gap filling procedure helps in eliminating tiny gaps from the extracted connected region. Furthermore, it smoothes the resulting contour. Figure 20 shows the effect of gap filling on the performance of our extracted contours. Figure 20(a) and Figure 20(b) show the original and thresholded images, respectively. Figure 20(c) shows the connected region obtained without gap filling application, whereas Figure 20(d) shows the connected region obtained after applying the gap filling procedure, where tiny gaps have been filled. Hence, after applying island removal step, as shown in Figure 20(f), some contour leaks were eliminated. The contour obtained without applying gap filling is shown in Figure 20(e). Figure 20(g) shows the final contour, where gaps in the connected region negatively affect the performance of the extracted contour. This issue was solved by applying the gap filling procedure as shown on Figure 20(h).

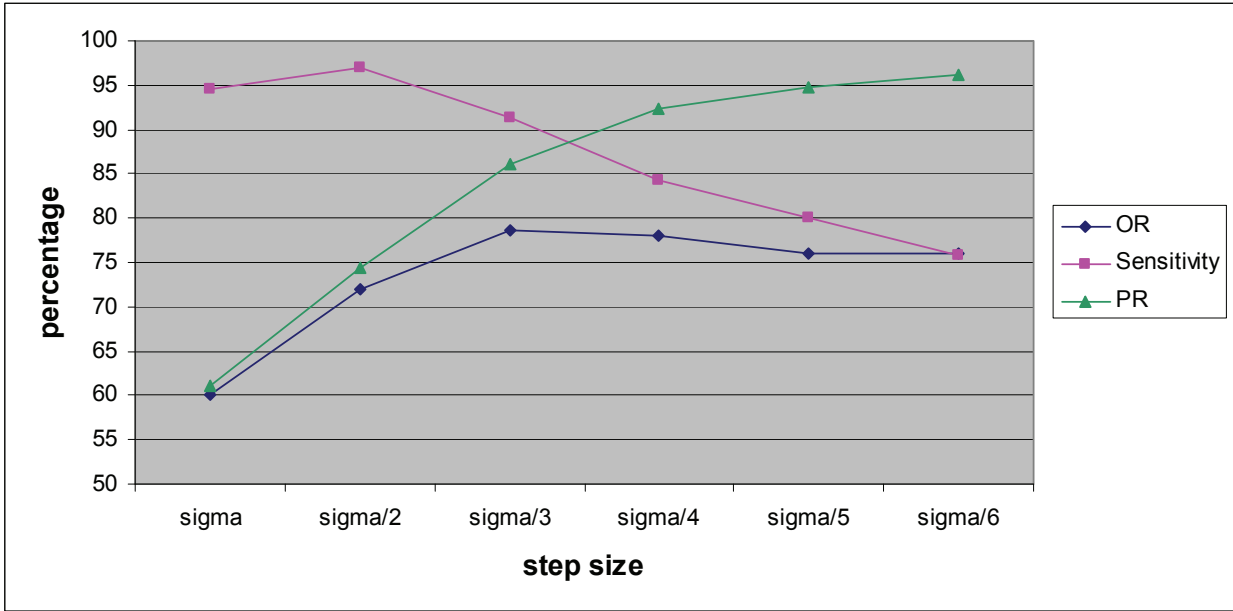


Figure 18: Overlap ratio, sensitivity, and precision rate obtained using various step sizes.

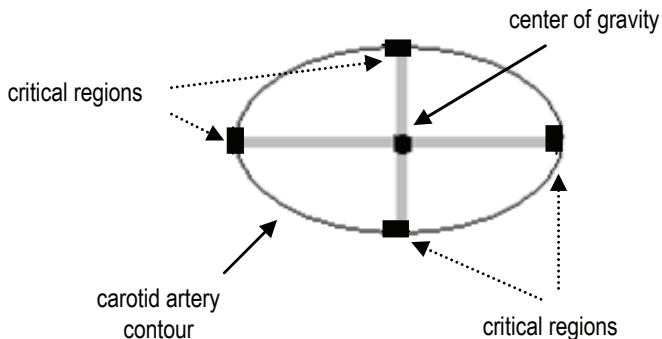
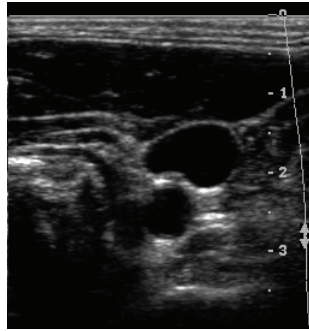


Figure 19: Critical regions.



(a)



(b)



(c)



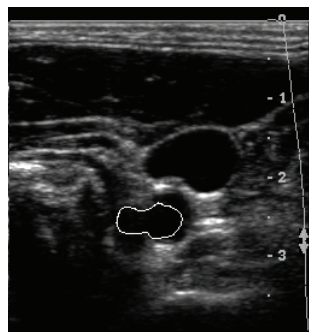
(d)



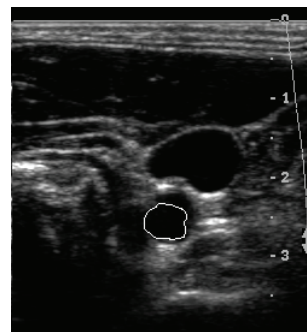
(e)



(f)



(g)



(h)

Figure 20: The effect of gap filling procedure in the performance of the extracted contours; (a) original image, (b) thresholded image, (c) extracted connected region without applying gap filling, (d) extracted connected region after applying gap filling, (e) connected region in (c) after applying island removal, (f) connected region in (d) after applying island removal, (g) final extracted contour without applying gap filling, and (h) final extracted contour after applying gap filling.

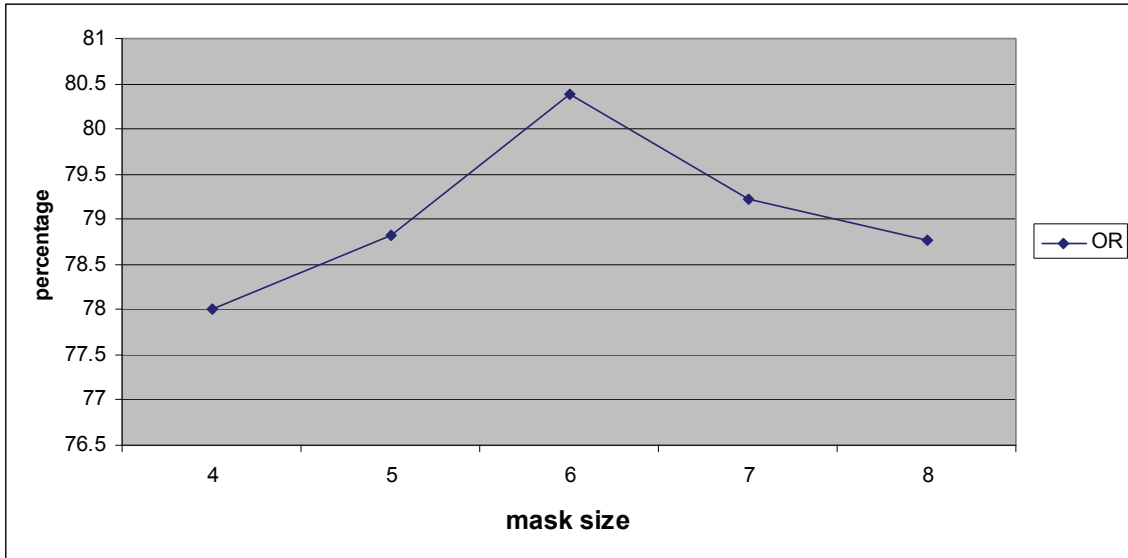


Figure 21: The average overlap ratio obtained using various mask sizes.

In our experimentations, we used  $6 \times 1$  and  $1 \times 6$  masks to eliminate horizontal and vertical gaps that are less than six pixels wide. Figure 21 shows the effect of using various mask sizes on the average overlap ratio, where  $6 \times 1$  and  $1 \times 6$  masks produce the best results.

### 8.3 Island Removal

The determination of ellipse axis is based on counting the number of the ellipse points that touch the contour. In some situations (depending on the location of the seed point), islands present inside the connected region may be incorrectly assumed to be part of the contour, hence blocking the ellipse from growing during the axes length determination process and leading to unsatisfactory results. Figure 22 shows how the presence of islands inside the connected region may cause the algorithm to produce inaccurate initial and final contours. However, this problem was solved by using the island removal process.

### 8.4 Seed point location

Our algorithm starts with a circle centered at the seed point and has a diameter of five pixels. This means that the seed point should be placed at least two pixels away from the carotid artery contour. Otherwise, the initial circle will be drawn outside the carotid artery boundary leading to inaccurate results.

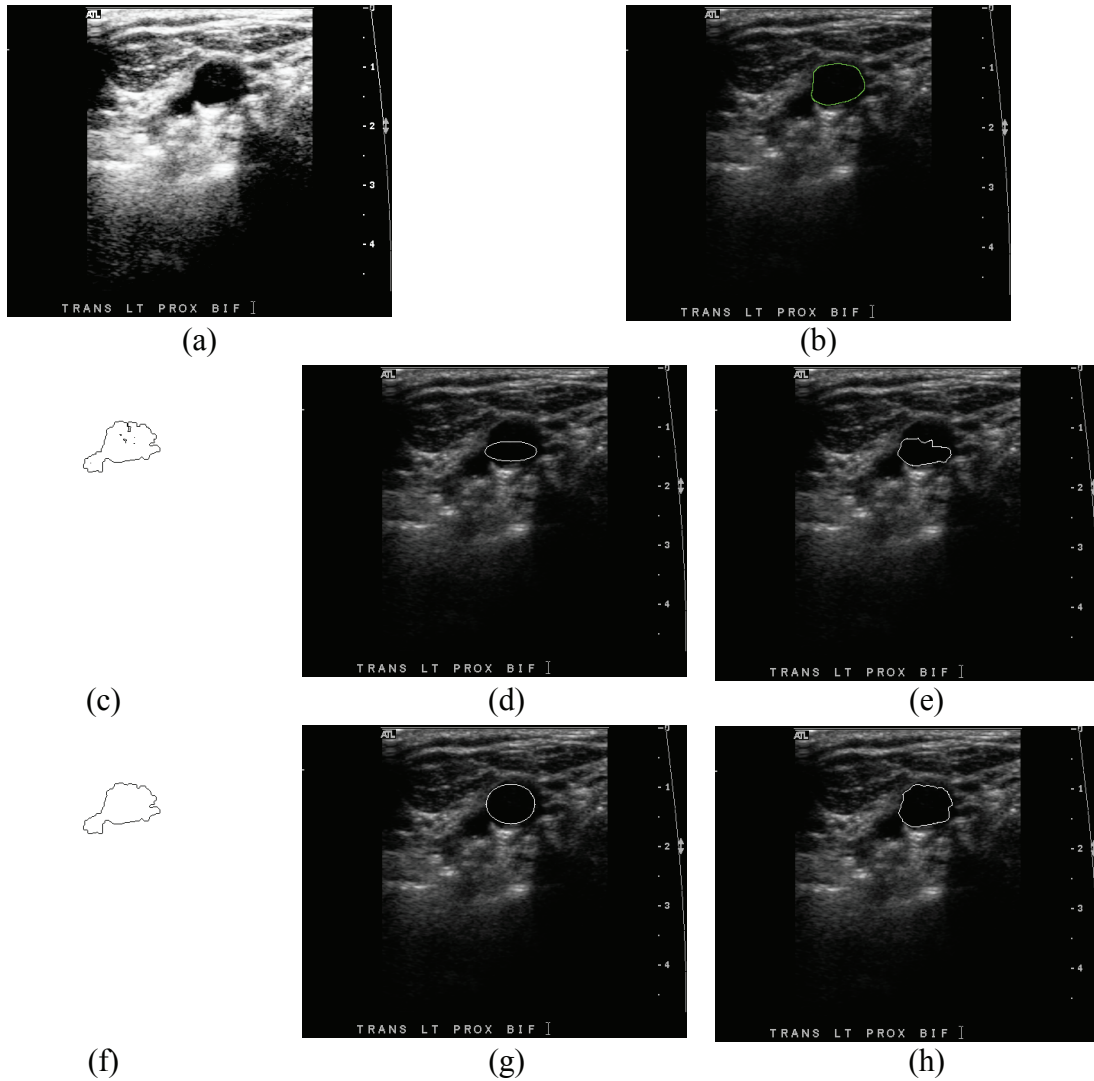


Figure 22: The effect of islands in the accuracy of the extracted contours; (a) histogram equalized image, (b) manually extracted contour by a clinician, (c) extracted connected region without applying island removal, (d) initial contour without applying island removal, (e) final contour without applying island removal, (f) extracted connected region after applying island removal, (g) initial contour after applying island removal and (h) final contour after applying island removal.

### 8.5 Shortcomings

In some odd cases our algorithm may produce inaccurate contours. This may happen when a large gap (more than 6 pixels wide) is present in one of the critical regions of the carotid artery contour (see Section 8.2). Thus, gap filling procedure will fail to close this gap and an inaccurate initial contour will be generated leading to an inaccurate final contour. Figure 23 shows an example of some odd cases where our algorithm fails to produce accurate contours.

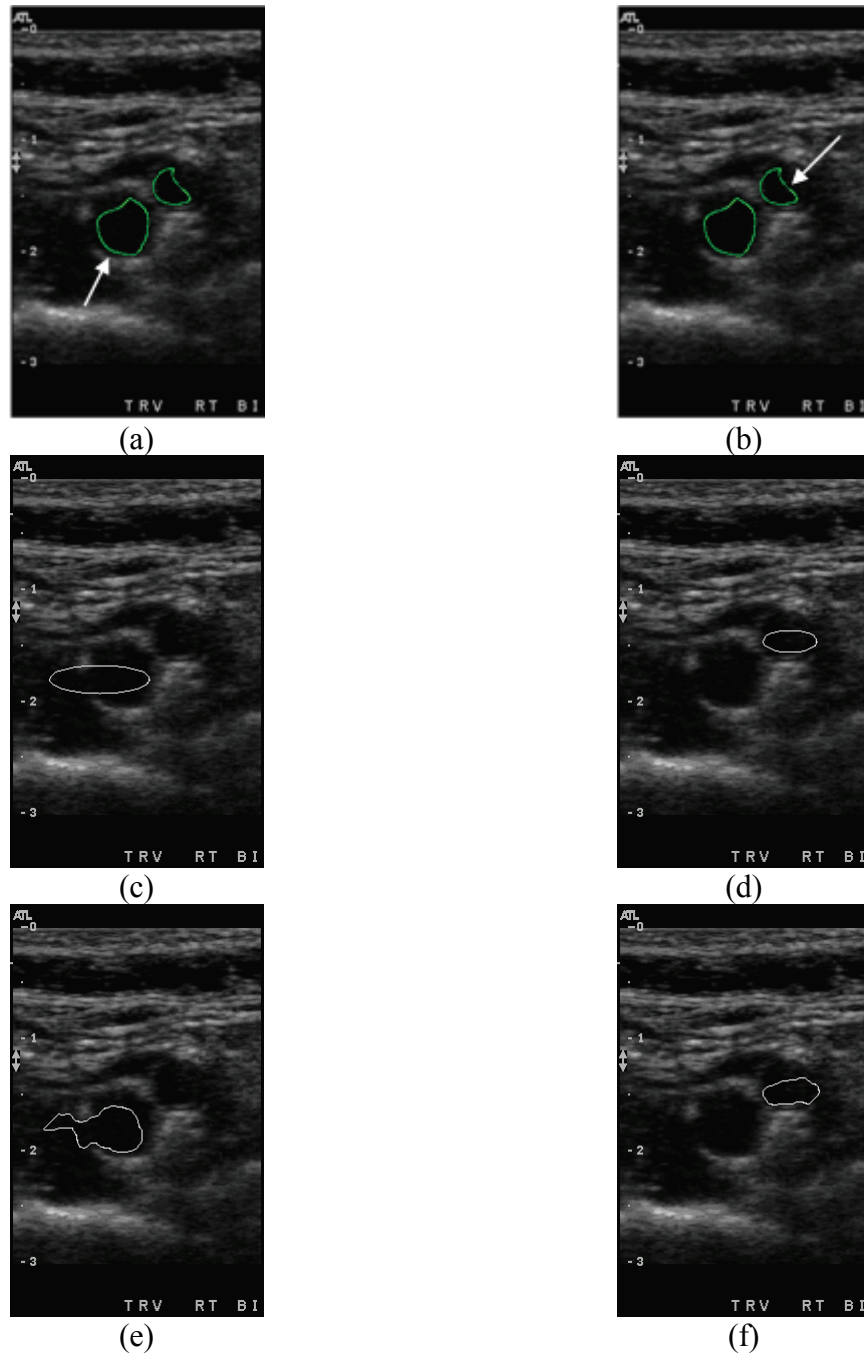


Figure 23: Odd cases where our algorithm fails to produce accurate contours; (a) and (b) are manually segmented contours, (c) and (d) are extracted initial contours, (e) and (f) are extracted final contours.

## 9. Conclusion

In this paper we presented a fast and efficient semi-automatic snake-based segmentation technique for segmenting carotid artery lumen boundaries in ultrasound images. Our algorithm requires only a single seed point to be placed almost anywhere inside the carotid artery boundaries. Hence it reduces user interaction. The proposed technique was tested using a total of 109 ultrasound images and the results show that the extracted initial and final contours have a good overlap with contours that are manually segmented by an experienced clinician.

Furthermore, the proposed initial contour generation algorithm demonstrates that most of the snake nodes are close enough to the carotid artery lumen boundaries. Hence, it can be used as an independent module with any active contouring algorithm to produce accurate results.

## Reference

- [1] American Heart association, Heart diseases and stroke statistics, 2007 update at a glance, [http://www.americanheart.org/downloadable/heart/1166712318459HS\\_StatsInsideText.pdf](http://www.americanheart.org/downloadable/heart/1166712318459HS_StatsInsideText.pdf), pp. 14-16.
- [2] Heart and Stroke Foundation of Canada statistics, <http://www.heartandstroke.com/site/c.ikiQLcM WJtE/b.3483991/k.34A8/Statistics.htm#stroke>.
- [3] A. Hamou and M. El-Sakka, "A Novel Segmentation Technique for Carotid Ultrasound Images", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, May 2004, vol. 3, pp. 521-524.
- [4] A. Abdel-Dayem and M. El-Sakka, "A Novel Morphological-based Carotid Artery Contour Extraction", Proceedings of the Canadian Conference on Electrical and Computer Engineering, May 2004, vol. 2, pp. 1873-1876.
- [5] A. Abdel-Dayem and M. El-Sakka, "Carotid artery contour extraction from ultrasound images using multi-resolution-analysis and watershed segmentation", ICGST International Journal on Graphics, Vision and Image Processing, December 2005, vol. 5, no. 9, pp. 1-10.
- [6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models", International Journal of Computer Vision, 1988, vol. 1, no. 4, pp. 321-331.
- [7] L. Cohen, "On active contour models and balloons", CVGIP: Image Understanding, March 1991, vol. 53, no. 2, pp. 211-218.
- [8] C. Xu and J. Prince, "Gradient Vector Flow: A new External Force for Snakes", Proceedings of Computer Vision and Pattern Recognition, June 1997, pp. 66-71.

- [9] S. Lobregt and M. Viergever, "A Discrete Dynamic Contour Model", IEEE transactions on medical imaging, March 1995, vol. 14, no. 1, pp. 12-24.
- [10] D. Cheng, A. Trucksass, K. Cheng, M. Sandrock, Q. Pu, and H. Burkhardt, "Automatic Detection of the Intimal and the Adventitial Layers of the Common Carotid Artery Wall in Ultrasound B-mode Images using Snakes", Proceedings of the International Conference on Image Analysis and Processing, September 1999, pp. 452-457.
- [11] F. Mao, J. Gill, D. Downey, and A. Fenster, "Segmentation of Carotid Artery in Ultrasound Images", Proceedings of the Annual International Conference on Engineering in Medicine and Biology Society, July 2000, vol. 3, pp. 1734-1737.
- [12] C. Loizou and C. Pattichis, "Snakes Based Segmentation of the Common Carotid Artery Intima Media", Medical and Biological and Computing, Springer, January 2007, vol. 45, no. 1, pp. 35-49.
- [13] A. Hamou and M. El-Sakka, "Carotid Ultrasound Segmentation Using DP Active Contours", Proceedings of the International Conference on Image Analysis and Recognition, August 2007, vol. 4633, pp. 961-971.
- [14] S. Moursi and M. El-Sakka, "Initial Contour for Ultrasound Carotid Artery Snakes", Proceedings of the 7th IEEE International Symposium on Signal Processing and Information Technology, December 2007.
- [15] R. Gonzalez and R. Woods, "Digital Image Processing", 2<sup>nd</sup> Edition, Addison-Wesley Publishing Company, 2002.
- [16] P. Felzenszwalb and D. Huttenlocher, "Distance Transforms of Sampled Functions", Cornell Computing and Information Science, Technical report 1963, 2004.
- [17] H. Park, T. Schoepflin, and Y. Kim, "Active Contour Model with Gradient Directional Information: Directional Snake", IEEE Transactions on Circuits and Systems for Video Technology, February 2001, vol. 11, no. 2, pp. 252-256.
- [18] A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, September 1990, vol. 12, no. 9, pp. 855-867.
- [19] A. Amini, S. Tenran, and T. Weymouth, "Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints", Proceedings of the 2<sup>nd</sup> International Conference on Computer Vision, December 1988, pp. 95-99.
- [20] K. Bowyer, "Validation of Medical Image Analysis Techniques", in M. Sonka *et al.* (Eds.), "Handbook of Medical Imaging", vol. 2, SPIE Press, 2000, chapter 10, pp. 571-575.
- [21] J. Neyman and E. Pearson, "On the Use and Interpretation of Certain Test Criteria for Purposes of Statistical Inference, Part I", Joint Statistical Papers, Cambridge University Press, 1967 (originally published in 1928), pp.1-66.
- [22] E. Vansteenkiste, A. Ledda, G. Stippel, B. Huysmans, P. Govaert, and W. Philips, "Segmenting Leukomalacia using Textural Information and Mathematical Morphology", Proceedings of the Program for Research on Integrated Systems and Circuits, 2003, pp. 441-446.