Segmentation of the Left Ventricle in Cardiac MR Images

Marie-Pierre Jolly, Nicolae Duta[1], and Gareth Funka-Lea
Imaging and Visualization Department
Siemens Corporate Research, Princeton, NJ
{jolly,lea}@scr.siemens.com

Abstract

This paper describes a segmentation technique to automatically extract the myocardium in 4D cardiac MR images for quantitative cardiac analysis and the diagnosis of patients. Three different modules are presented. The automatic localization algorithm is able to approximately locate the left ventricle in an image using a maximum discrimination technique. Then, the local deformation algorithm can deform active contours so that they align to the edges in the image to produce the desired outlining of the myocardium. Finally, the global localization algorithm is able to propagate segmented contours from one image in the data set to all the others. We have experimented with the proposed method on a large number of patients and present some examples to show the strengths and pitfalls of our algorithm.

1. Introduction

Cardiovascular disease is the leading cause of death in the United States. Mortality has been declining over the years as lifestyle has changed, but the decline is also due to the development of new technologies to diagnose disease. One of these techniques is magnetic resonance imaging (MRI) which provides time-varying three-dimensional imagery of the heart. To help in the diagnosis of disease, the physicians are interested in identifying the heart chambers, the endocardium and epicardium, and measuring the ventricular blood volume, the ventricular wall motion and wall thickening properties over various stages of the cardiac cycle. The left ventricle is of particular interest since it pumps oxygenated blood out to distant tissue in the entire body.

There has been a large amount of research on the analysis of medical images [5] and the segmentation problem has been particularly challenging. It was not until the mid-eighties that people have realized that tracking the cardiac wall motion was important to characterize meaningful functional changes. At that time, though, this was done for echocardiograms and nuclear medicine only. The computer analysis of magnetic resonance cardiac images did not come until the early nineties.

The system proposed by Fleagle *et al.* [6] was able to delineate the borders of the myocardium using a minimum cost path graph search algorithm after the user indicated the center of the left ventricular cavity and the area of interest with a few mouse clicks. Geiger *et al.* [8] also asked the user to specify an approximate location of the contours of interest and used a dynamic programming approach to modify these contours so that they correspond to image edges. Goshtasby and Turner [9] proposed a two step algorithm combining intensity thresholding to recover the bright blood and local gradient to outline the strong edges using elastic curves. Weng *et al.* [14] developed an algorithm to threshold the image based on parameters estimated during a learning phase and get a good approximate of the segmentation. Although we have not directly used any of these techniques, all of these papers have greatly influenced our work.

As part of the Siemens MR workstation, we propose a complete system, known as Argus, consisting of drawing tools and automatic segmentation algorithms to allow the physician to outline the myocardium in each image in the patient data set, compute volumes, and perform a thickening analysis. The remainder of this paper describes the segmentation module, including automatic localization, local deformations, and propagation. We also present a number of experimental results.

2. Automatic Localization

The goal of the automatic localization module is to approximately locate the myocardium in an image without any assistance from the user.

The appearance of a left ventricle in a short axis view of the heart is usually similar in temporal and spatial views as well as in many different subjects. It can be roughly summarized as "The myocardium is a dark area between two concentric circles enclosing a bright area corresponding to the blood in the left ventricle. On its left side is a bright region corresponding to the blood in the right ventricle. On its right side is a very dark area corresponding to the lungs". We have developed a method to encode this variable appearance and quickly locate ventricle instances in new images based on the maximum discrimination method proposed by Colmenarez and Huang [3] for human face detection.

Due to the relative ventricle symmetry and computational constraints, we have only used the gray

[1] Nicolae Duta is now at BBN Technologies, Cambridge, MA

0-7695-1143-0/01 $10.00 © 2001 IEEE

values of the pixels along the main four cross sections through the ventricle (see Figure 1) to derive the feature set. Each cross section was sampled to contain 25 points, so that a heart ventricle is defined as a 100-dimensional feature vector $\boldsymbol{x}=\{x_1,x_2,\ldots,x_{100}\}$. A ventricle instance (observation) is regarded as the realization of a random process $\boldsymbol{X}=\{X_1,X_2,\ldots,X_n\}$. Let P and N denote the two class-conditional probabilities over the instance space Ω: $P(\boldsymbol{x})=\mathrm{P}(\boldsymbol{X}=\boldsymbol{x} \mid \boldsymbol{x}$ is a ventricle instance) and $N(\boldsymbol{x})=\mathrm{P}(\boldsymbol{X}=\boldsymbol{x} \mid \boldsymbol{x}$ is not a ventricle instance).

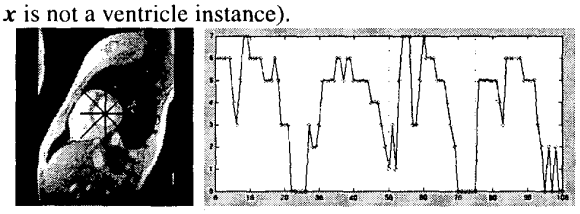


Figure 1: The feature set defining a heart ventricle. (a) The four cross sections through the ventricle used to extract the features. (b) The 100-element normalized feature vector associated with the ventricle in (a).

To train the system, we used a collection of 1,350 MR cardiac images from 14 patients. A tight bounding box was placed around the left ventricle in each image and slightly translated and scaled to generate a total of 101,250 positive examples. We also generated 123,096 negative examples by uniformly sampling the images.

It is commonly assumed that the Kullback distance between P and N, defined by:

$$H_{P\|N}(X)=\sum_{x\in\Omega}P(X=x)\log\frac{P(X=x)}{N(X=x)} \quad (1)$$

is a measure of the discrimination between the two classes whose feature distributions are P and N. In order to include some feature dependency in the model as well as to compute $H_{P\|N}$ and the log-likelihood ratio $L(o)$ very fast, we assume that $\boldsymbol{X}$ is a Markov chain with unknown ordering. In this case, the Kullback distance for a given ordering $S=\{s_1,s_2,\ldots s_n\}$ can be computed as:

$$H^{S}_{P\|N}(X_{s_1},\ldots,X_{s_n})=H_{P\|N}(X_{s_1})+\sum_{i=2}^{n}H_{P\|N}(X_{s_i}\mid X_{s_{i-1}}) \quad (2)$$

The goal of the learning procedure is to find a site ordering S^* for which the Kullback distance is maximum and estimate the transition probabilities of the Markov chain induced by this ordering. The resulting optimization task is equivalent to an asymmetric traveling salesman problem and can be solved using simulated annealing.

During the detection stage, the test image is scanned at different scales with a 25x25 window from which a feature vector $O=(o_1,\ldots,o_n)$ is extracted. The pattern is classified as ventricle if the log-likelihood ratio:

$$L(o)=\ln\frac{P(X_{s_1}=o_{s_1})}{N(X_{s_1}=o_{s_1})}+\sum_{i=2}^{n}\ln\frac{P(X_{s_i}=o_{s_i}\mid X_{s_{i-1}}=o_{s_{i-1}})}{N(X_{s_i}=o_{s_i}\mid X_{s_{i-1}}=o_{s_{i-1}})} \quad (3)$$

is larger than a threshold that sets the trade-off between correct detection and false alarm. The distributions of the log-likelihood ratios for positive and negative examples are shown in Figure 2. By setting the decision threshold to 0, the resubstitution error rate is 2.17% and the false alarm rate is 2.94%.

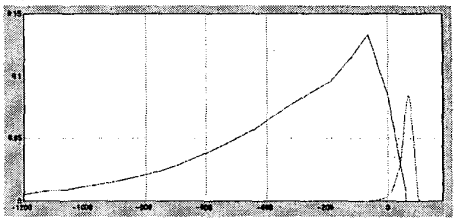


Figure 2: The distribution of the log-likelihood ratio for ventricle (right) and non- ventricle (left) examples computed over the training set.

Since several neighboring positions can be classified as left ventricle, we partition them into clusters. An example of ventricle detection is shown in Figure 3(a) in which 14 clusters were detected. The final decision as to which of the cluster candidates (if any) actually corresponds to the left ventricle is made using a voting procedure. We define 8 salient points in the gray level profiles as the intersection of the 4 cross sections with the ventricle's medial axis. One of the salient points is particularly visible on the left side of the horizontal profile as the deep and narrow valley (see Figure 3(d)).

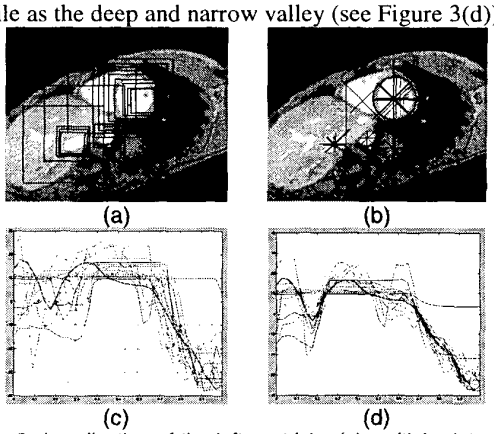


Figure 3: Localization of the left ventricle: (a) multiple detections produced by the maximum discrimination method; (b) the voting procedure and the winning myocardium; (c) horizontal feature profiles corresponding to each detected box in (a); (d) the profiles in (c) aligned to the average profile (bold).

We collected 84 training sample profiles from 14 patients and computed the average signal for each of the 4 cross sections. Note that if the signals are not aligned, the punctual average is meaningless. Therefore, we used the curve warping method proposed by Ramsay and Xi [13] to align the profiles.

The cross sections of each of the cluster candidates are warped onto their corresponding average profiles. Figures

3(c) and (d) show these profile before and after warping. The locations of the salient points in the image are then accumulated using a Hough transform array to vote for the most likely center position and radius for the myocardium centerline. If the number of votes for the best cell is too small, the solution is discarded and the process is considered to have failed. The estimated medial axis is shown as a circle in Figure 3(b).

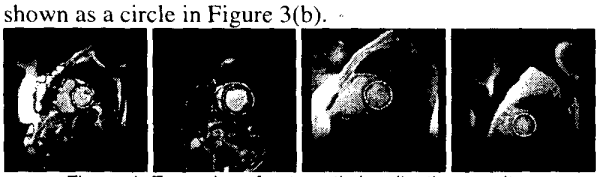


Figure 4: Examples of automatic localization results.

Figure 4 shows some examples of the automatic localization process. It can be seen that the system can recover the location and size of the myocardium for a wide range of cases.

3. Local Deformations

In this section, we describe the procedure that deforms approximate contours to align them with the edges of the myocardium in the image.

3.1. Active contours

Active contours, or snakes, first proposed by Kass *et al.* [10] were designed to handle this very task. Snakes are defined as contours $u(s)=(x(s),y(s))$ that are pushed or pulled towards image features by constraining forces. A typical energy function for a snake is expressed as:

$$E(u)=\int\left\{\alpha\left|u_s(s)\right|^2+\beta\left|u_{ss}(s)\right|^2-\lambda\left|\nabla I\right|^2\right\}ds \quad (4)$$

where the first two terms measure the smoothness of the contour (in terms of the first derivative $u_s(s)$ and second derivative $u_{ss}(s)$ of the contour) and the third term measures the amount of edge strength along the contour using the image gradient ∇I. The goal is to minimize this energy function to find a contour which is smooth and which coincides with high gradient magnitude points in the image. Kass *et al.* proposed to minimize this energy using Euler equations and gradient descent.

To avoid the sensitivity of the algorithm to the initial contour, Geiger *et al.* [8] use dynamic programming to solve for the global minimum of the discretize the energy:

$$E(p_1,\ldots,p_n)=\sum_{i=1}^{n}\frac{1}{\left\|\nabla I(p_i)\right\|+\varepsilon}+\alpha\sum_{i=2}^{n}\left|\bar{\nabla}I(p_i)-\bar{\nabla}I(p_{i-1})\right| \quad (5)$$

where $\left\|\nabla I(p)\right\|$ is the magnitude and $\bar{\nabla}I(p)$ the direction of the gradient of the image at point p. This is equivalent to finding the shortest path in a graph where nodes correspond to pixels and the cost of a link between two neighboring pixels is defined as:

$$e(p_1,p_2)=\frac{1}{\left\|\nabla I(p_2)\right\|+\varepsilon}+\alpha\left|\bar{\nabla}I(p_2)-\bar{\nabla}I(p_1)\right| \quad (6)$$

The same problem can be solved using Dijkstra's algorithm as demonstrated by Mortensen and Barrett [11].

If the position of the desired contour is approximately known, the system can look for the best contour in a search area around the approximate position as shown in Figure 5. Dijkstra's algorithm can be used in the case of multiple sources by connecting with a zero link all the possible source nodes to one "pseudo" source node.

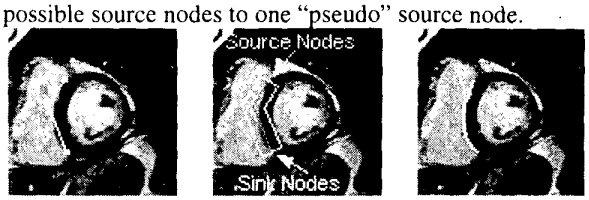


Figure 5: Active contour segmentation: (a) Initial contour; (b) Search area where Dijkstra's algorithm propagates; (c) Final contour.

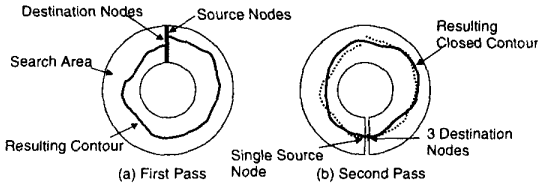


Figure 6: Two passes of Dijkstra's algorithm are enough to recover a closed contour.

If the desired contour has to be closed, there is no guarantee that, when the contour reaches a destination node, it connects with its starting node (see Figure 6(a)). Thus, we run a second pass of Dijkstra's algorithm with a single source node (in the middle of the previous contour) and three destination nodes (neighbors of the source node) as illustrated in Figure 6(b). This way, we guarantee that the resulting contour is closed. This process does not guarantee that the resulting contour has the smallest energy but we have experienced on a large number of examples that the resulting contour is very close (and most of the time equal) to the absolute best contour.

3.2. Size of the search area

One of the key elements to the success of this process is the size of the search area. If the search area is not large enough, it might not overlap with the desired contour and the final segmentation will not be accurate. If the search area is too large, however, the process might be distracted by other strong edges in the region and the resulting contour will be wrong.

To solve this problem, we have decided to do multiple iterations of Dijkstra's algorithm for increasing sizes of the search area. At each iteration, a new contour is recovered and the contribution of every pixel to the total energy is recorded. As the size of the search area increases, if the resulting contour outlines the same pixels over and over again, it means that these pixels are part of

a very stable edge and should belong to the final segmentation. If the contour keeps jumping around as the size of the search area increases, it means that the edges in that area are not stable.

3.3. Local deformations of the endocardium

The first step is to apply the local deformations to recover the inner contour (endocardium). The energy function for the contour is defined by the cost of a link between two pixels:

$$e_1(p_1,p_2)=\begin{cases}\dfrac{1}{\|\nabla I(p_2)\|^2+\varepsilon} & \text{if } z(p_2,\bar{\nabla}I)\geq 0\\ 1/\varepsilon & \text{otherwise}\end{cases} \quad (7)$$

where ε is a small number (0.01) to bound the energy and

$$\begin{pmatrix}0\\0\\z(p_2,\bar{\nabla}I)\end{pmatrix}=\begin{pmatrix}x_2-x_1\\y_2-y_1\\0\end{pmatrix}\times\begin{pmatrix}\cos(\bar{\nabla}I(p_2))\\\sin(\bar{\nabla}I(p_2))\\0\end{pmatrix} \quad (8)$$

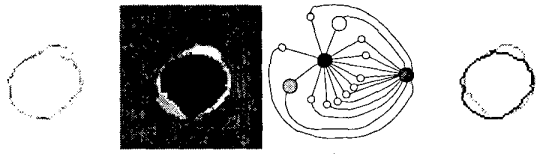
In our case, the contour is built clockwise by the Dijkstra process and the image gradient points from bright to dark. The sign of the z component of the cross product between the image gradient and the contour direction indicates the relationship between the two vectors. By setting the energy to a very large number when $z<0$, we force the contour to separate a bright region inside from a dark region outside.

A search region is placed symmetrically around the input approximate contour. Dijkstra's algorithm is then applied for increasing size of a search area within this search region and the confidence at every pixel is determined. The confidence $C(p)$ is then modified to push the contour as far from the centroid (m_x,m_y) of the input contour as possible:

$$C(p)=C(p)\left(\frac{\sqrt{(x-m_x)^2+(y-m_y)^2}}{\max_{x,y}\sqrt{(x-m_x)^2+(y-m_y)^2}}\right)^3 \quad (9)$$

because Dijkstra's algorithm favors smaller contours and the contour should be behind floating papillary muscles.

The confidence image then needs to be analyzed to extract a closed contour. Figure 7(a) shows a typical example of a confidence image, where the darker pixels indicate more confidence. It can be seen that in some areas, the system "hesitates" between 2 or 3 paths. We would like to find the best closed contour that separates the center of the object from the boundary of the image. Let us build a graph where each node corresponds to a connected component defined by the regions between boundary pixels as seen in Figure 7(b). There is an edge between two nodes if the two corresponding connected components are separated by a single boundary and the weight on the edge encodes the confidence of the pixels on the common boundary. Figure 7(c) shows the graph corresponding to the confidence image in (a). Then, we want to find the minimum cut between the center node and the outside node. We use the max flow algorithm [7] to determine the minimum cut.



(a) (b) (c) (d)

Figure 7: Minimum cut: (a) confidence image; (b) connected components between boundary subparts; (c) corresponding graph; (d) the final segmentation is shown in dark.

Figure 8(a) shows an image with the two input contours generated by dilating the myocardium centerline from the automatic localization process. Figure 8(b) shows the segmented endocardium. It can be seen that the result is not quite satisfactory because papillary muscles that are not attached to the myocardium should not be outlined. To remedy this, we apply a second pass of the local deformation process for the inner contour after adding a region component to the energy function. We compute a mean μ and variance σ^2 for the gray levels of the myocardium using the pixels on the outside of the endocardium between the 7 o'clock and the 11 o'clock position on the contour (because papillary muscles are on the right side of the blood pool). Then, a homogeneity image H is computed as:

$$H(p)=\exp\left\{-\frac{(I(p)-\mu)^2}{2\sigma^2}\right\} \quad (10)$$

The homogeneity image is shown in Figure 8(c). The cost of a link between two pixels is then defined as:

$$e_2(p_1,p_2)=\begin{cases}\dfrac{1}{\left(\dfrac{|\nabla I|+|\nabla H|}{2}\right)^2+\varepsilon} & \text{if } z(p_2,\bar{\nabla}H)<0\\ 1/\varepsilon & \text{otherwise}\end{cases} \quad (11)$$

Note that the edges of the input image are combined with the edges of the homogeneity image, and the gradient direction of H is used in the cross product to force the contour inside the myocardium (where H is bright). The confidence is then modified using Eq. (9) and the minimum graph cut algorithm is applied. This results in the final endocardium contour seen in Figure 8(d).

3.4. Local deformations of the epicardium

Once the endocardium is recovered, a new homogeneity image can be computed, now using all the pixels around the inner contour to compute the mean and variance, as seen in Figure 8(c). To segment the outer contour (epicardium), we apply Dijkstra's algorithm in a search region increasing outwards from the inner contour. The energy function is defined by the cost of a link between two pixels:

$$e_3(p_1,p_2)=\begin{cases}\dfrac{1}{\left(\dfrac{|\nabla I|+|\nabla H|}{2}\right)^2+\varepsilon} & \text{if } z(p_2,\bar{\nabla}H)\geq 0\\ 1/\varepsilon & \text{otherwise}\end{cases} \quad (12)$$

This time, we want the contour to stay outside the bright pixels in the homogeneity image. The confidence at every pixel is then modified by:

$$C(p)=C(p)\left(1-\frac{\left|\sqrt{(x-m_x)^2+(y-m_y)^2}-R_{\text{med}}\right|}{R_{\max}-R_{\min}}\right)^3 \quad (13)$$

where (m_x,m_y) is the centroid of the input contour and $R_{\min}$, R_{med}, and $R_{\max}$ are the minimum, median, and maximum radii of the input contour. This constrains the contour to stay as circular as possible.

The segmentation of the epicardium is more difficult than the endocardium because there is usually no clear edge between the myocardium and the liver, and the right ventricle myocardium appears to merge into the left ventricle myocardium. So, the circularity and smoothness constraints greatly help the segmentation.

To produce a smooth contour, the pixels in the confidence image are fitted with a spline. To fit a spline (X_j,Y_j) with m control points to an ordered curve points $\{P_i=(x_i,y_i),i=0,\ldots,p\}$ we want to minimize the least squares expression [1]:

$$R=\sum_{i=0}^{p}\left\{\left(\sum_{j=0}^{m}X_jB_j(U_i)-x_i\right)^2+\left(\sum_{j=0}^{m}Y_jB_j(U_i)-y_i\right)^2\right\} \quad (14)$$

by solving the linear system:

$$\begin{cases}\sum_{j=0}^{m}\left\{\sum_{i=0}^{p}B_j(U_i)B_n(U_i)\right\}X_j-\sum_{i=0}^{p}x_iB_n(U_i)=0\\ \sum_{j=0}^{m}\left\{\sum_{i=0}^{p}B_j(U_i)B_n(U_i)\right\}Y_j-y\sum_{i=0}^{p}y_iB_n(U_i)=0\end{cases} \quad \forall n=0,\ldots,m \quad (15)$$

where the B_j's are the spline basis functions. Also, U_i is the curve index parameter associated with data point (x_i,y_i) and defined as:

$U_0=3$

$$U_{i+1}=U_i+(m-3)\frac{|P_{i+1}-P_i|}{\sum_{i=1}^{p}|P_{i+1}-P_i|} \quad (16)$$

In general, the optimal number of spline control points is not known. We can repeat the least squares procedure for various values of m in a dichotomy search until a desired residual (say 0.5 pixels) is reached.

However, the data points in the confidence image are not in linear order. Thus, we fit a spline with m control points to the input approximate contour and generate the resulting contour using d points per spline segment to produce the 2D coordinates $\{(\xi_n,\eta_n),\ n=0,\ldots,d(m\text{-}3)\}$ and their corresponding linear coordinates $\{u_n=j+k/d,\ j=0,\ldots,m,\ k=0,\ldots,d\text{-}1\}$. Then, the parametrization for a data point is defined by the linear coordinate of the spline point closest to it:

$$U_i=u_{n_{\min}}$$
$$n_{\min}=\underset{n=0,\ldots,d(m-3)}{\arg\min}\left\{(\xi_n-x_i)^2+(\eta_n-y_i)^2\right\} \quad (17)$$

The final segmentation combining the endocardium and epicardium is shown in Figure 8(f).

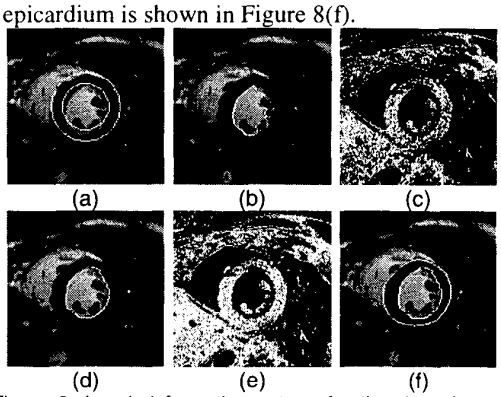


Figure 8: Local deformations steps for the detection of the myocardium: (a) input image and approximate contours; (b) endocardium after the first pass; (c) homogeneity image after the first pass; (d) endocardium after the second pass; (e) homogeneity image after the second pass; (f) epicardium and final segmentation.

4. Propagation

The purpose of the propagation functionality in our system is to segment an entire row or column of the 4D dataset from one segmented image. Typically, the user will segment the upper left image (ED base) and propagate the segmented contours to all the slices in the ED phase (ED propagation). Then, all the ED contours will be propagated to the ES phase (ES propagation). This would be enough to compute the ejection fraction and perform some thickening analysis. For a more detailed analysis, the user can also propagate all the ED contours to all the images in all the phases using temporal propagation.

The local deformations module is able to modify an approximate contour so that it fits the edges in the image. It is the responsibility of the propagation to provide this approximate contour. We have investigated three propagation modes.

4.1. Copy template propagation

The simplest propagation method consists of simply copying the contours being propagated. This method is very appropriate for temporal propagation because the location and shape of the contours do not change much from one phase to the next.

4.2. Automatic localization

The second mode of propagation that we have explored is to run the automatic localization process

described in Section 2. This method gives the best results for ED and ES propagation. Since the ventricle size is approximately known from the template image, we limit the scale search to 0.85 to 1.15 times the size of the template. We also limit the search space for the location of the ventricle. In the case of ED propagation, the location of the left ventricle might change quite a bit due to a different breath hold position by the patient. Thus, we extend the search area to 50 pixels around the location of the template. In the case of ES propagation, the location of the left ventricle barely changes, so we allow a displacement of only 10 pixels.

4.3. Global template matching

The most involved propagation mode is global template matching. We use this technique if the automatic localization technique fails for ED and ES propagation. The template is characterized by the thickness of the myocardium for 12 evenly spaced sectors and the distribution of the gray levels of the myocardium pixels. Given a new image, the process considers each location in a search area around the location of the template. For each location, scaled versions of the contour are evaluated. The process picks the best pair of contours based on the fit of the contours in the image while enforcing that the endocardium stays inside the epicardium and the thickness of the myocardium remains about the same for most sectors. The best location is then chosen by comparing the gray level histograms of the hypothesis and the template. Since this process is very time consuming, only a very small search space both in scale and translation can be considered.

5. Best Set of Parameters

Throughout the design of our segmentation algorithm, we have experimented with a number of solutions (e.g. the form of the energy function or the propagation method). In addition, there were a number of parameters that had to be set, such as the size of the search space for local deformations or propagation.

We have tried to design a systematic way to choose the best set of parameters so that a user of our system in a clinical setting would not have to know the details of the segmentation algorithm to obtain good results. For this purpose, we have established a methodology of black box testing, similar to the one proposed in [2].

For each parameter, we have identified a set of meaningful values. We have collected a number of example images to cover as much of the diversity in patients and acquisition setups as possible. For each example, we have obtained the "true" segmentation from radiologists, cardiologists, and MR physicists. Ideally, ground truth should be obtained from multiple experts because the community does not always agree on the true location of the left ventricle boundaries. Unfortunately, we were not able to collect more than one set of contours for each patient.

For each image, for each set of parameter values, we ran the system and produced a segmentation result. We compared each segmentation result to the ground truth segmentation using the modified Hausdorff distance between two sets of points [4]:

$$D(A,B) = \frac{1}{|A|+|B|}\left\{\sum_{a\in A}\min_{b\in B} d(a,b) + \sum_{b\in B}\min_{a\in A} d(b,a)\right\} \qquad (18)$$

The "best" set of parameters should be such that the segmentation results are better on most datasets, acknowledging the fact that the results might be worse for some datasets. In clinical setting, the radiologist is given the opportunity to correct any of the generated contours before the ejection fraction or the volumes are computed. Thus, it is better to make gross mistakes once in a while rather than small mistakes all the time. For this reason, we are not interested in computing the average distance over all test patients.

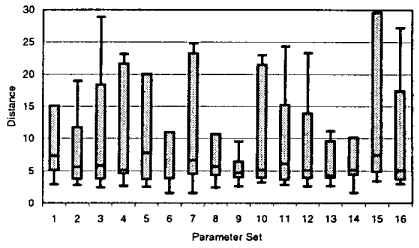


Figure 9: Evaluation chart for 16 parameter sets and 11 patients.

Once the segmentation results were obtained for all patients and all images, and the distances between the resulting contours and the ground truth contours were computed, we plotted charts similar to the one in Figure 9. A box and whiskers plot is a useful tool to look at the distribution of data. The range of the values is indicated by the extent of the vertical line. The rectangular box shows the values between the first and the third quartiles. The horizontal bar inside the rectangular box indicates the median of the values. It can be seen that, in this example, Parameter Set 9 performs the best overall.

6. Results

All the results that are shown in this section were generated completely automatically on entire studies. The image for the base slice of ED was segmented using the automatic localization algorithm followed by the local deformations. Then, this segmentation was propagated to the entire ED phase. Finally, the contours in each slice at ED were propagated temporally across the entire study or to the ES phase only. Rather than showing the segmentation results on entire studies which would require a lot of space, we have chosen to show individual

segmentation results on cropped images and identify the strengths and pitfalls of the algorithm.

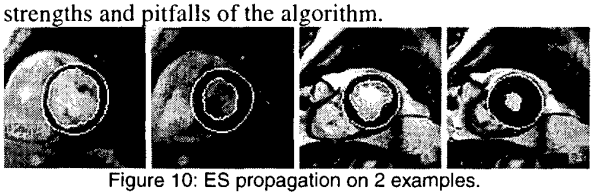
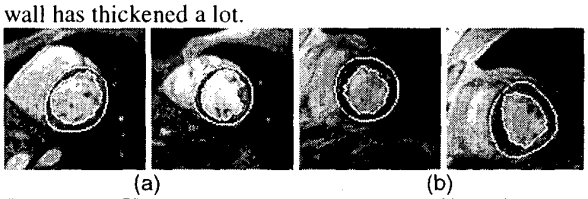


Figure 10: ES propagation on 2 examples.

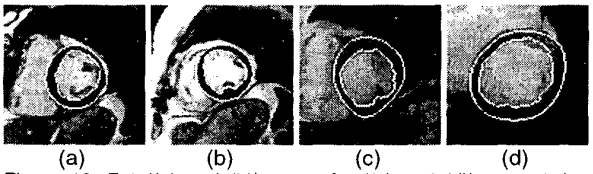
Figure 10 shows the results of ES propagation on two examples. It can be seen that the location of the left ventricle has not changed but the shape of the left ventricle is very different. In particular, the myocardial wall has thickened a lot.



(a) (b)

Figure 11: ED propagation on 2 example. Note the large displacement in (b).

Figure 11 shows the results of ED propagation from one slice to the next on two examples. The distance between the two planes is 20mm. In the (a) case, it can be seen that the location of the left ventricle has remained the same. In the (b) case, however, there was a very large jump (greater than 20 pixels) due to a different breath hold position by the patient. Also, we notice that in both cases, the shape of the myocardium has changed quite a bit between the two slices.

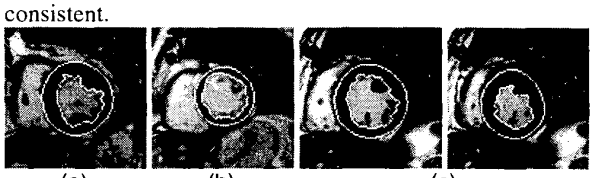


(a) (b) (c) (d)

Figure 12: Fat ((a) and (b)) or no fat ((c) and (d)) around the myocardium.

Figure 12 demonstrates the behavior of the algorithm in the presence or absence of fat around the myocardium. Fat creates a bright thin region between the myocardium and the lung. The problem is that the edge between the fat and the lung is stronger than the desired edge between the myocardium and the fat. Fleagle *et al.* [6] had to provide a heuristic to handle the presence of fat as a special case. Our algorithm is able to handle the presence or absence of fat equally well because the recovery of the epicardium is based on both the image edges and the edges of the homogeneity image. The fat is not detected by the homogeneity criterion because its gray level is very different from the expected gray level of the myocardium.

Figures 13 and 14 show the behavior of the algorithm in the presence of papillary muscles. In Figure 13(a), we show an example where the papillary muscles that are attached to the myocardium are detected as so and included in the endocardium and the papillary muscles that are floating in the blood pool are left inside the endocardium. Figure 13(b) shows an example where the papillary muscles are floating and the endocardium is correctly detected. Figure 13(c) shows an example of papillary muscles that are floating in the blood pool during the ED phase and get attached to the myocardium during the ES phase. The algorithm is able to handle these correctly. In Figure 14, we show an example where the concentration of papillary muscles is so dense that the algorithm is not able to detect the endocardium correctly. It is difficult to see which papillary muscles are attached to the myocardium and which are floating, but the automatic segmentation result does not seem very consistent.



(a) (b) (c)

Figure 13: Papillary muscles.

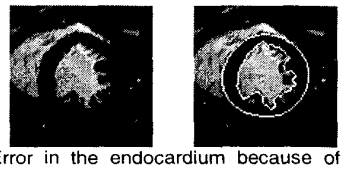


Figure 14: Error in the endocardium because of the papillary muscles.

Figure 15 shows segmentation results in the valve plane. As the heart beats, it slightly descends in the chest. Thus, at end-systole, it might happen that the acquisition plane intersects the heart at the mitral valve when the myocardium is not in the shape of two concentric circles. The automatic localization task becomes more difficult and the local deformation process cannot find the contours in the upper left part of the myocardium. Nevertheless, it can be seen from the results that the algorithm behaves very well.

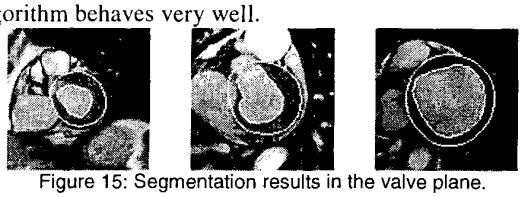
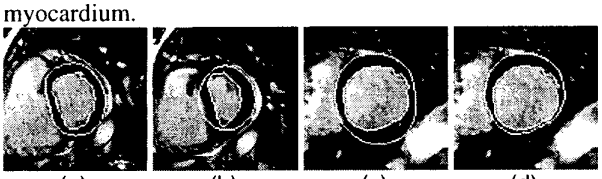


Figure 15: Segmentation results in the valve plane.

Figure 16 shows examples of a diseased heart. The patient in Figures 16(a) and (b) exhibit an oval-shaped left ventricle. This is a difficult task for the automatic localization technique because it is looking for a relatively circular object. The system can recover very well for the endocardium, but it has trouble for the epicardium in Figure 16(b). Figures 16(c) and (d) show an example where the bottom left part of the myocardium is infarcted.

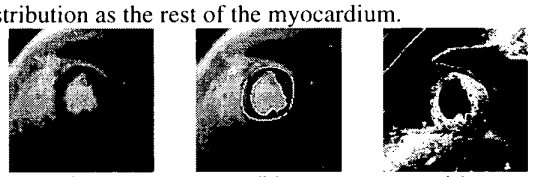
It can be seen that our algorithm overestimated the size of the myocardium as seen in Figure 16(c) compared to the ground truth shown in Figure 16(d). The location of this particular infarct makes it more difficult to segment because there is no clear edge between the liver and the myocardium.



(a) (b) (c) (d)

Figure 16: Diseased heart.

Figure 17 shows another example where the algorithm does not find the right solution. The input image is shown in Figure 17(a) and the segmentation result is shown in Figure 17(b). It can be seen that the size of the myocardium was greatly underestimated. This is due to the inhomogeneous magnetic field in some MR acquisitions. The average gray level of the pixels around the endocardium is about 20 levels smaller than the average gray level of the pixels at the bottom right of the myocardium. Figure 17(c) shows the homogeneity image and it can be seen that the pixels in the bottom right part of the myocardium do not follow the same Gaussian distribution as the rest of the myocardium.



(a) (b) (c)

Figure 17: Error on the estimate of the gray level distribution of the myocardium.

Overall, the segmentation process takes between 0.5 and 1.5 seconds per image on a 700MHz Pentium III. The automatic localization takes between 0.5s and 1s depending on the size of the search area. The copy template process takes virtually no time and the global template matching algorithm (used rarely) takes 1s. The local deformations take about 0.5s for two passes for the inner contour and one pass for the outer contour, with multiple passes of Dijkstra's algorithm for each.

7. Conclusions

We have presented a segmentation module to automatically extract the contours of the myocardium in MR images of the heart left ventricle. The module consists of three different parts: an automatic localization process, a local deformation process using graph algorithms for active contours, and a strategy to propagate contours between images.

We have tested our algorithm on a large number of patients and demonstrated its strengths on many examples. Our partners in the clinical field have been very pleased with the results. This algorithm has been shipping out to customers since March 2001 with the VA20A version of the cardiac analysis package Argus as part of MRease VA12A, an extension of Syngo on the Siemens Magnetic Resonance MAGNETOM systems.

Nevertheless, we are working to improve the algorithm and address the pitfalls shown in Section 6. In particular, we want to estimate the gray level variation inside the myocardium better so that the homogeneity image is more accurate. Also, we are trying to incorporate shape information into the local deformations process.

Bibliography

[1] R. H. Bartles, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics & Geometric Modeling*, Morgan Kaufmann Publishers, 1987.

[2] V. Chalana and Y. Kim, "A methodology for evaluation of boundary detection algorithms on medical images", *IEEE Trans. Medical Imaging*, 16(5):642-652, 1997.

[3] A. Colmenarez and T. Huang, "Face detection with information-based maximum discrimination", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp 782-787, 1997.

[4] M.-P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching", *Proc. International Conference on Pattern Recognition*, Jerusalem, Israel, pp 566-568, 1994.

[5] J. S. Duncan and N. Ayache, "Medical image analysis: Progress over two decades and the challenges ahead", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):85-106, 2000.

[6] S. R. Fleagle, D. R. Thedens, J. C. Ehrhardt, T. D. Scholz, and D. J. Skorton, "Automated identification of left ventricular borders from spin-echo resonance images", *Investigative Radiology*, 26:295-303, 1991.

[7] L. Ford and D. Fulkerson, *Flow in Networks*, Princeton University Press, 1962.

[8] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(3):294-302, 1995.

[9] A. Goshtasby and D. A. Turner, "Segmentation of cardiac cine MR images for extraction of right and left ventricular chambers", *IEEE Trans. Medical Imaging*, 14(1):56-64, 1995.

[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models", *International Journal of Computer Vision*, 2:321-331, 1988.

[11] E. N. Mortensen and W. A. Barrett, "Interactive segmentation with intelligent scissors", *Graphical Models and Image Processing*, 60:349-384, 1998.

[12] M. Plass and M. Stone, "Curve-fitting with piecewise parametric cubics", *Computer Graphics*, 17(3):229-239, 1983.

[13] J. O. Ramsay and X. Li, "Curve registration", *Journal of the Royal Statistical Society Series B*, 60:351-363, 1998.

[14] J. Weng, A. Singh, and M. Y. Chiu, "Learning-based ventricle detection from cardiac MR and CT images", *IEEE Trans. Medical Imaging*, 16(4):378-391, 1997.