

Automatic Recovery of the Left Ventricular Blood Pool in Cardiac Cine MR Images

Marie-Pierre Jolly

Siemens Corporate Research, Imaging and Visualization Dept., Princeton, NJ
`marie-pierre.jolly@siemens.com`

Abstract. We present a method for automatic localization and rough segmentation of the left ventricle blood pool in cardiac cine magnetic resonance images. The method first detects the whole heart using time-based Fourier analysis. It then segments the left ventricle blood pool by grouping connected components across slices using isoperimetric clustering. The system was tested on 253 datasets and failed in only 2 cases.

1 Introduction

Cardiovascular disease has become the largest cause of death in the modern world and is an important health concern. Imaging technologies such as magnetic resonance (MR) imaging allow physicians to non-invasively observe the behavior of the heart. Physicians are particularly interested in the left ventricle (LV) because it pumps oxygenated blood out to the rest of the body. Ideally, they would like to quantify the volume of the blood pool over time and estimate its ejection fraction, cardiac output, peak ejection rate, filling rate, and myocardial thickening. These quantities are easy to compute once the left ventricle is outlined in all images. Manual outlining is very cumbersome however, and most physicians limit it to the end-diastolic and end-systolic phases. This allows the system to calculate ejection fraction and cardiac output, but it is not enough information to estimate peak ejection rate or filling rate. Therefore, for complete patient care, it is very important to provide an automatic segmentation system.

This paper is concerned with the localization of the LV blood pool in the images of the 3D+T dataset (10 slices with 20 phases on the average). This is an important problem because the localized blood pool can be used to initialize a more elaborate LV segmentation algorithm. Therefore, the solution should be very robust. It is a difficult problem because MR intensities are not consistent across acquisitions and blood pixels cannot easily be identified in the images. In addition, most acquisitions cover slices beyond the LV itself to guarantee that it is seen in all phases. This means that some slices can be below the apex and contain no blood pool, and some slices can be above the mitral valve and contain the left atrium blood pool.

There has been many publications in the field of cardiac MR image segmentation. Some researchers have constructed models to help in the initialization process. Lorenzo-Valdés *et al.* [1] use a 4D probabilistic atlas of the heart and

a 3D intensity template which they propose to register to the ED frame to localize the left and right ventricles. Mitchell *et al.* [2] use a hybrid active shape and appearance model and locate the heart using the Hough transform. Unfortunately, both these methods are too slow for clinical practice. In addition, models have difficulty capturing variability outside the training sets. This means that pathological cases which fall outside the standard set of shapes might not be recognized and appearance models have to be re-trained for new acquisition protocols and sequences. The solution proposed by Jolly [3] is faster, and actually used in clinical practice, but it still depends on a learned appearance represented by a Markov chain. Kaus *et al.* [4] combine a statistical model with coupled mesh surfaces. They originally had 169 datasets for testing but had to drop 48 of them because they exhibited breathing artifacts and through-plane motion which violated their model continuity assumption. They initialize their model by assuming that the heart is located in the center of the image which is definitely not a valid assumption.

To avoid the restriction imposed by a model, other researchers have chosen to use simple image processing techniques. These solutions however have been minimally tested [5,6] and tend to be less robust. Cocosco *et al.* [7] make a very restricting assumption that the coverage of the short-axis image stack should stop at the mitral valve and not go into the atrium. This is quite unreasonable as physicians tend to increase the coverage of the image stack to correct for potential motion after the acquisition of the localizer images and make sure that the LV is completely seen during all phases. Indeed, in 5 of their 32 cases, the top slice extended into the atrium and the algorithm could not separate the LV from the right ventricle (RV) without user intervention at the mitral valve. Lin *et al.* [8] also use simple image processing techniques and report only one failure out of 330 cases. This is the type of outstanding performance that is required for clinical practice. Unfortunately, we were not able to reproduce their results on our noisier datasets.

In this paper, we propose a new method to automatically detect and roughly segment the LV blood pool. We do not make any assumption on the quality of the acquisition or position of the heart in the images. First, the heart is localized in the images using motion information. Second, the LV blood pool is localized and roughly segmented within the heart. We tested our algorithm on 253 datasets and compared it with Argus (commercialized by Siemens Medical Solutions). Note that the goal of this work is not to segment the LV in all the images, but just to recover it in enough slices to be able to initialize a more sophisticated segmentation process.

2 Heart Localization

The first part of the algorithm consists in localizing the heart in the images and generating a region of interest to be used in the rest of the processing. We follow the recommendation from Lin *et al.* [8] and compute the first harmonic of the Fourier transform over time for each slice. These $H_1^s(x, y)$ images highlight

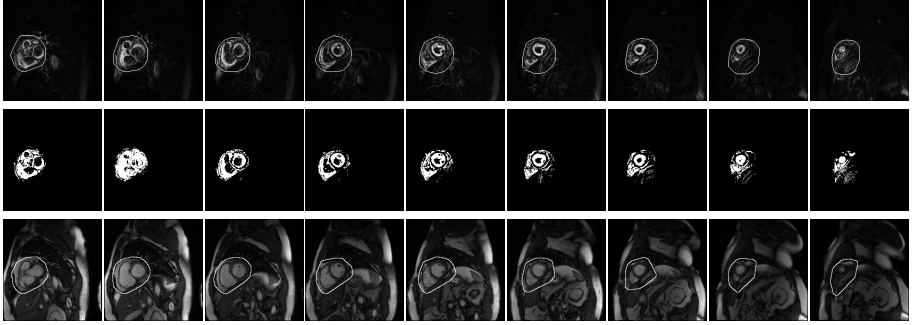


Fig. 1. Automatic heart localization: first row: first harmonic of the Fourier transform with the region extracted after 3D line fitting and distant artifact removal; second row: white connected components were kept and gray ones were discarded; third row: region of interest localizing the heart

the moving structures (cardiac chambers and great vessels). We then want to eliminate the distant moving artifacts. As in [8], a 3D line is fitted through the 2D centroids of the H_1^s images. Then, the distances weighted by the $H_1^s(x, y)$ values are histogrammed and thresholded to remove the farthest points. This procedure is repeated until the 3D centroid of the H_1^s images becomes stable.

Then, for each slice, we compute the average response $\overline{H}_1^s$ of the first harmonic. Each $H_1^s(x, y)$ image is thresholded at $2\overline{H}_1^s$ to retain only the strongly moving areas. Connected components (CC) are extracted and the next step consists of grouping them between slices to generate regions of interest that are consistent in space. To this end, the CC with largest average motion ($H_1^s(x, y)$ over all pixels) is identified in each slice and the relative motion of the other CCs is computed. In the same manner, we compute the relative size of each CC. The confidence is then defined as the relative motion times the relative size. The CCs with smallest confidence are removed one at a time until a slice containing a single CC (denoted $\hat{C}$) is identified. In the other slices, the 2D overlap between each CC and $\hat{C}$ is computed and the confidence becomes the relative size times this overlap. Connected components with a confidence lower than 0.1 are discarded. The final region of interest (ROI) is defined as the convex hull of the retained CCs in each of the slices.

Fig. 1 shows the process of heart localization on the first phase of the 9 slices of an example dataset. The first row shows the first harmonic image with the region that was extracted after 3D line fitting and distant artifact removal. The second row shows the connected components (the gray ones had low confidence and were discarded). Finally, the third row shows the detected ROI.

3 Left Ventricle Blood Pool Segmentation

In the second part of the algorithm, the system isolates the blood pool of the left ventricle inside the region of interest. The process is divided into 2 steps:

1) thresholding and connected component analysis, and 2) clustering to find the best set of connected components to describe the blood pool.

In the first step, each slice is processed one at a time. The images are thresholded inside the ROIs using Otsu's algorithm [9]. Then, the bright pixels are grouped into connected components. Here, we decided to extract 2D+T connected components for each slice rather than the 3D+T connected components proposed by Cocosco *et al.* [7] because the first slices of many of our datasets contained the left atrium (LA). If we had built 3D+T connected components, the bright pixels of the LV would have been connected above to the bright pixels of the LA which would have been connected below to the bright pixels of the RV, and the left and right ventricles would have been connected. With 2D+T connected components, the LV and RV are connected only in the mitral valve slice. This first steps highlights bright objects in the regions of interest, usually the LV blood pool, the RV blood pool, the aorta and other noisy regions.

In the second step, we want to group connected components between slices and identify the group that corresponds to the LV blood pool. For the grouping, we use the isoperimetric clustering algorithm proposed by Grady and Schwarz [10]. The goal of the algorithm is to partition a weighted graph by minimizing the perimeter to area ratio (i.e.: the isoperimetric ratio $h(S) = \frac{|\partial S|}{|S|}$). A graph is a pair $G = (V, E)$ with vertices $v \in V$ and edges $e \in E \subseteq V \times V$. An edge between two vertices v_i and v_j is denoted e_{ij} . In our case, the 4 largest connected components in each slice are each associated with a vertex in the graph and edges are defined between vertices in neighboring slices.

Let $A_p(v_i)$ be the area in phase p of the connected component associated with vertex v_i . Let $A^m(v_i)$ and $A^M(v_i)$ be the minimum and maximum areas over time, respectively. For each connected component, the following measures are computed:

- 1) Shrinking: $\mathcal{S}(v_i) = \frac{A^m(v_i)}{A^M(v_i)}$, the amount that the object contracts over time.
- 2) Roundness: $\mathcal{R}(v_i)$, the ratio of the smallest eigenvalue to the largest eigenvalue in 2D principal components analysis in each phase, averaged over time.
- 3) Connectedness: $\mathcal{C}(v_i)$, connected components when observed in each phase can be composed of multiple pieces. The relative size of each of these pieces is denoted $r_p^j(v_i), j = 1, \dots, n_i$. Then, $\mathcal{C}(v_i) = \frac{1}{P} \sum_{p=1}^P \sum_{j=1}^{n_i} r_p^j(v_i)$.
- 4) Concavity: $\mathcal{D}(v_i)$, the maximum distance (normalized between 0 and 1) between the object and its convex hull, averaged over time.

Finally, the overall confidence of a connected component is defined as $\mathcal{L}(v_i) = \frac{1}{50} \mathcal{S}(v_i) (1 - \mathcal{R}(v_i)) \mathcal{C}(v_i)^{10} (1 - \mathcal{D}(v_i))^{10}$.

Edge weights, denoted $w(e_{ij})$, should indicate the similarity between vertices. The normalized area of a connected component in phase p is defined as $a_p(v_i) = \frac{A_p(v_i) - A^m(v_i)}{A^M(v_i) - A^m(v_i)}$. For each pair of connected components in neighboring slices, the following measures are computed:

- 1) Overlap: $\mathcal{O}(v_i, v_j)$, the intersection of the two 2D+T CCs divided by their union (if $\mathcal{O}(v_i, v_j) < 0.001$, the edge e_{ij} is discarded).

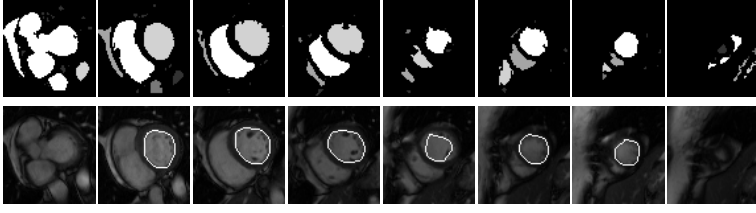


Fig. 2. Results of the LV blood pool recovery: first row: 2D+T connected components in the first phase of each slice, second row: contours for the LV blood pool cluster

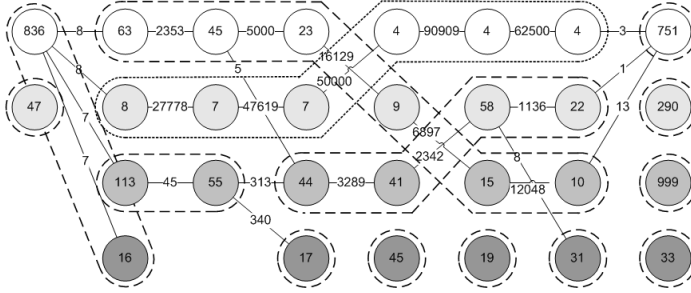


Fig. 3. Graph and isoperimetric clustering for LV blood pool recovery. The edge weights are multiplied by 10000 and the connected component confidences are multiplied by 100. Each column of vertices corresponds to a slice. The vertices and corresponding connected components in Fig. 2 have the same gray color.

- 2) Distance: $\mathcal{D}(v_i, v_j)$, the distance between the centers of the CCs averaged over time.
- 3) Resemblance of their area-time curves: $\mathcal{T}(v_i, v_j) = \frac{1}{P} \sum_p \|a_p(v_i) - a_p(v_j)\|$.
- 4) Difference in size: $\mathcal{S}(v_i, v_j) = \frac{1}{P} \sum_p \max(1, \frac{A_p(v_i)}{A_p(v_j)})$ where v_i is on the slice below v_j , it is expected that the components get smaller as they go down the slices closer to the apex, so $\mathcal{S}(v_i, v_j)$ should stay close to 1 for CCs in the LV blood pool.

The edge cost is then defined as $c(v_i, v_j) = \frac{\mathcal{D}(v_i, v_j)\mathcal{T}(v_i, v_j)\mathcal{S}(v_i, v_j)}{\mathcal{O}(v_i, v_j)} \mathcal{L}(v_i)\mathcal{L}(v_j)$ and the edge weight is $w(e_{ij}) = \frac{1}{c(v_i, v_j)}$.

The isoperimetric clustering algorithm is as follows. Let x be an indicator vector which takes a binary value at each vertex and encodes the partition S :

$$x_i = \begin{cases} 0 & \text{if } v_i \in S, \\ 1 & \text{if } v_i \in \bar{S}. \end{cases}$$

Then, the perimeter and area of the partition are defined as

$$|\partial S| = x^T L x \quad \text{and} \quad |S| = x^T \mathbf{1},$$

where $\mathbf{1}$ is the unit vector and L is the Laplacian matrix defined as

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w(e_{ij}) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise.} \end{cases} \quad \text{and} \quad d_i = \sum_{e_{ij}} w(e_{ij}).$$

The indicator vector is recovered by solving the linear system $Lx = \mathbf{1}$ which results in a real-valued solution for x . This can be converted to a binary partition by ranking the x_i 's and choosing the threshold which yields the minimum value for the isoperimetric ratio $h(S)$. We choose the ground node as the center of the graph, i.e.: the vertex for which the shortest path to the farthest vertex is the smallest, which can easily be recovered using Floyd-Warshall's algorithm. This performs better to recover elongated clusters rather than the node with the largest degree as suggested in [10]. The graph is recursively partitioned until the isoperimetric ratio of the subpartitions is larger than a stopping criterion.

Once the clustering is determined, the blood pool cluster should be large and its connected components should shrink nicely over time, should be round, and should contain one main piece in each phase. The clusters are ranked in decreasing order of size, and we examine at least the first two clusters, as well as any cluster as large as the first two and any cluster of size larger than 3. We then choose the cluster $\hat{K}$ for which the function $\frac{1}{N_K} \sum_{v_i \in K} \mathcal{L}'(v_i)$ is minimum, where N_K is the number of vertices in cluster K and $\mathcal{L}'(v_i) = \mathcal{S}(v_i)(1 - \mathcal{R}(v_i))\mathcal{C}(v_i)^{10}$ is the confidence of a connected component. The 2D convex hulls of the connected components in $\hat{K}$ define the left ventricle blood pool.

Fig. 2 shows an example of the blood pool recovery. The first row shows the connected components and the second row shows the segmented LV blood pool. The graph that was constructed along with edge weights $w(e_{ij})$ and confidences $\mathcal{L}'(v_i)$ is shown in Fig. 3. The partitions are shown in dashed lines and the LV blood pool partition in a dotted line.

4 Experiments

We collected 253 datasets from 20 different clinical sites around the world. These datasets include both patients and volunteers and were all acquired on Siemens MR scanners. They contained between 2 and 16 slices, with an average of 10 slices and between 7 and 75 phases, with an average of 20 phases. 125 (49%) datasets started above the mitral valve plane, 7 (3%) of which were well inside the atrium and 72 (28%) datasets ended below the apex.

We ran the algorithm on all the datasets. For an average dataset of 200 images, the whole process (implemented in non-optimized C++) takes 8 seconds on a dual core laptop. By visual inspection, we consider that the algorithm succeeds if the resulting contours are around the blood pool in some slices in the dataset. We observed only 2 failures out of the 253 datasets. Both cases are shown in Fig. 4. In the first case, the strongest cluster corresponded to the aorta. In the second case, the system picked a totally wrong structure which was included in the region of interest due to some motion artifacts.

In most cases, no LV was segmented in the first and last slices. This was expected since the first slice was often above the mitral valve and the last slice

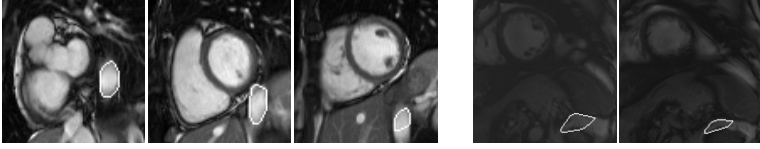


Fig. 4. The 2 failures out of 253 datasets for the blood pool segmentation algorithm

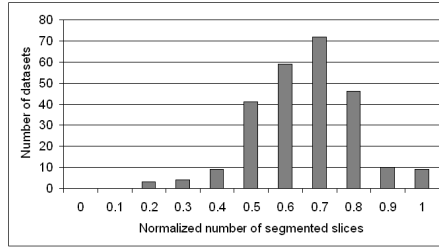


Fig. 5. Normalized number of segmented slices

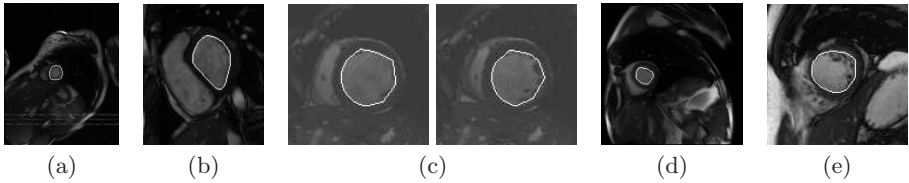


Fig. 6. Good results despite: (a) Large motion artifacts; (b) Not perfectly round LV blood pool; (c) Non contracting blood pool; (d) Heart not in the center of the image; (e) Very thin septum

was often below the apex. Fig. 5 shows the distribution of the normalized number of slices in which the LV was roughly segmented. On the average, the LV was localized in 63% of the slices. This is more than adequate to initialize a refined segmentation algorithm.

Some of the datasets were provided to us as failure cases from Argus, the commercial package from Siemens Medical Solutions (described in [3]). For 16 datasets, Argus was not able to locate the LV, and for 25 others, it identified another structure as the LV, thus causing 41 (16%) failures in total. When the system performs a fully automatic segmentation and identifies the wrong structure, the user has to redo the segmentation for the entire dataset, which can be very time consuming. This happened for 10% of the cases with Argus and for only 0.8% of the cases with our solution.

To illustrate the robustness of our method, Fig. 6 shows some examples of good results when some of the traditional assumptions are violated. In Fig. 6(a), there is a large motion artifact (streak at the bottom of the image). Fig. 6(b)

shows that the LV blood pool is not always round. In Fig. 6(c) the blood pool does not contract much at ES (right) compared to ED (left). Fig. 6(d) shows that the heart is not always in the center of the image. In Fig. 6(e), the myocardium is very thin in the septum.

5 Conclusions

We have proposed a system to localize and roughly segment the left ventricular blood pool in cardiac cine MR images. In the first phase, the algorithm uses a time-based Fourier analysis to identify the moving heart. In the second phase, isoperimetric clustering is used to group 2D+T connected components and form the left ventricle blood pool. Even though the system uses simple image segmentation techniques, we have demonstrated on many different datasets that it is very robust.

At this time, we are focusing on segmenting both the endocardium (more accurately and in all slices) and the epicardium so that clinically meaningful measurements can be computed. In the near future, we will see if the same technique can be used to localize the right ventricle. Indeed, it can be seen in Figs. 2 and 3 that the other large cluster containing 6 vertices actually corresponds to the right ventricle.

Acknowledgements

We would like to thank Leo Grady at Siemens Corporate Research for his advice on the isoperimetric clustering algorithm. Also, this large study would not have been possible without the help of Carmel Hayes and Michaela Schmidt from Siemens Medical Solutions who provided all the test datasets.

References

1. Lorenzo-Valdés, M., Sanchez-Ortiz, G.I., Elkington, A.G., Mohiaddin, R.H., Rueckert, D.: Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm. *Medical Image Analysis* 8, 255–265 (2004)
2. Mitchell, S.C., Lelieveldt, B.P.F., van der Geest, R.J., Bosch, H.G., Reiber, J.H.C., Sonka, M.: Multistage hybrid active appearance model matching: Segmentation of the left and right ventricles in cardiac mr images. *IEEE Trans. Medical Imaging* 20(5), 415–423 (2001)
3. Jolly, M.P.: Automatic segmentation of the left ventricle in cardiac MR and CT images. *International Journal of Computer Vision* 70(2), 151–163 (2006)
4. Kaus, M.R., von Berg, J., Weese, J., Niessen, W., Pekar, V.: Automated segmentation of the left ventricle in cardiac MRI. *Medical Image Analysis* 8, 245–254 (2004)
5. Gering, D.: Automatic segmentation of cardiac MRI. In: Ellis, R.E., Peters, T.M. (eds.) *MICCAI 2003*. LNCS, vol. 2878, pp. 524–532. Springer, Heidelberg (2003)
6. Phumeechanya, S., Pluempitwiriyawej, C.: Left ventricular segmentation using double region-based snakes. In: *Proc. ISBI*, pp. 840–843 (2007)

7. Cocosco, C.A., Niessen, W.J., Netsch, T., Vonken, E.P.A., Viergever, M.A.: Automatic image-driven segmentation of cardiac ventricles in cine anatomical MRI. In: Proc. SPIE Applications of Digital Image Processing, vol. 5909, pp. 1–11 (1995)
8. Lin, X., Cowan, B.R., Young, A.A.: Automated detection of left ventricle in 4D MR images: Experience from a large study. In: Larsen, R., Nielsen, M., Sparring, J. (eds.) MICCAI 2006. LNCS, vol. 4190, pp. 728–735. Springer, Heidelberg (2006)
9. Otsu, N.: A threshold selection method from gray level histograms. IEEE Trans. Systems, Man and Cybernetics 9, 62–66 (1979)
10. Grady, L., Schwartz, E.L.: Isoperimetric partitioning: A new algorithm for graph partitioning. SIAM Journal on Scientific Computing 27(6), 1844–1866 (2006)