Learning-Based Object Detection in Cardiac MR Images

Nicolae Duta[1], Anil K. Jain[1], and Marie-Pierre Dubuisson-Jolly[2]
[1]Department of Computer Science and Engineering, Michigan State University
{dutanico, jain}@cse.msu.edu, http://web.cse.msu.edu/~dutanico
[2]Imaging and Visualization Department, Siemens Corporate Research, Princeton

Abstract

An automated method for left ventricle detection in MR cardiac images is presented. Ventricle detection is the first step in a fully automated segmentation system used to compute volumetric information about the heart. Our method is based on learning the gray level appearance of the ventricle by maximizing the discrimination between positive and negative examples in a training set. The main differences from previously reported methods are feature definition and solution to the optimization problem involved in the learning process. Our method was trained on a set of 1,350 MR cardiac images from which 101,250 positive examples and 123,096 negative examples were generated. The detection results on a test set of 887 different images demonstrate an excellent performance: 98% detection rate, a false alarm rate of 0.05% of the number of windows analyzed (10 false alarms per image) and a detection time of 2 seconds per 256×256 image on a Sun Ultra 10 for an 8-scale search. The false alarms are eventually eliminated by a position/scale consistency check along all the images that represent the same anatomical slice.

I. Introduction

The goal of this study is to automatically learn the appearance of flexible objects in gray level images. Our working definition of appearance is that it is the *pattern* of gray values in the object of interest and its immediate neighborhood. The learned appearance model can be used for object detection: given an arbitrary gray level image, decide if the object is present in the image and find its location(s) and size(s). Object detection is typically the first step in a fully automatic segmentation system for applications such as medical image analysis [1–3], industrial inspection, surveillance systems and human-computer interfaces.

The application of interest here deals with detecting the left ventricle in short axis cardiac MR images. There has been a substantial amount of recent work in studying the dynamic behavior of the human heart using non-invasive techniques such as magnetic resonance imaging [4, 5]. In order to provide useful diagnostic information, a cardiac imaging system should perform several tasks such as segmentation of heart chambers, identification of endocardium and epicardium, measurement of the ventricular volume over different stages of the cardiac cycle, measurement of the ventricular wall motion, etc. Most approaches to segmentation and tracking of heart ventricles are based on deformable templates, which require specification of a good initial position of the boundary of interest. This is often provided manually, which is both time consuming and requires a trained operator.

The main objective of this paper is to automatically provide the approximate scale/position (given by a tight bounding box) of the left ventricle in 2-D cardiac MR images. This information is needed by most deformable template segmentation algorithms which require that a region of interest be provided by the user. This detection problem is difficult because of the variations in shape, scale, position and gray level appearance exhibited by the cardiac images across different slice positions, time instants, patients and imaging devices (see Fig. 1).

We make a distinction between the algorithms designed to detect specific structures in medical images and general methods that can be trained to detect an arbitrary object in gray level images. The dedicated detection algorithms rely on the designer's knowledge about the structure of interest and its variation in the images to be processed as well as on the designer's ability to code this knowledge. On the other hand, a general detection method, would necessitate very little, if any, prior knowledge about the object of interest. The specific domain information is usually replaced by a general learning mechanism based on a number of training examples of the object of interest. Among the domain specific methods for ventricle detection in cardiac images, one can mention Chiu and Razi's mutiresolution approach for segmenting echocardiograms [6], Bosch *et al.*'s dynamic programming based approach [7], and Weng *et al.*'s algorithm based on learning an adaptive threshold and region properties [5]. Most general learning strategies are based on additional cues like color or motion or rely extensively on object shape. As far as we know, the few systems that are based only on raw gray level information have only been applied to the detection of human faces in gray level images [8–12]. We want to emphasize the difference between *object detection* and *object recognition* [13, 14]. The *object recognition* problem [13] typically assumes that a

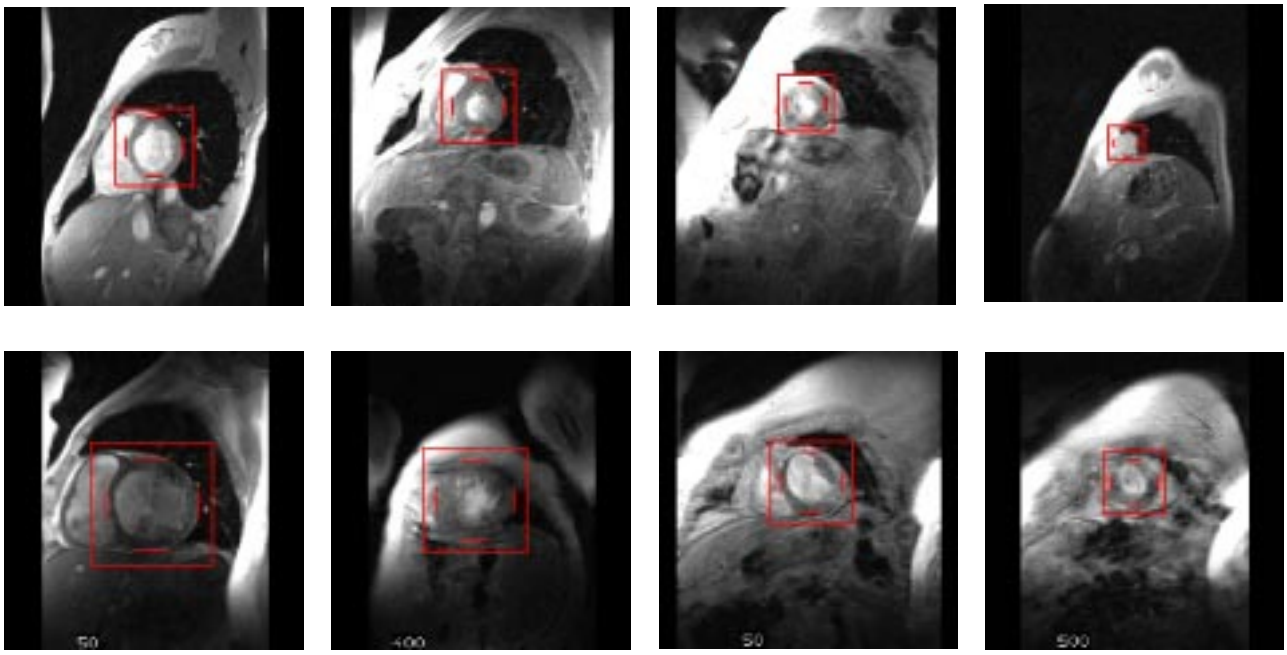

Fig. 1. Several examples of 256 × 256 gradient echo cardiac MR images (short axis view) showing the left ventricle variations as a function of acquisition time, slice position, patient and imaging device. The left ventricle is the bright area inside the square. The four markers show the ventricle walls (two concentric circles).

test image contains one of the objects of interest on a homogeneous background. The problem of object detection does not use this assumption and, therefore, is considered to be more difficult than the problem of isolated object recognition [14].

Most general-purpose detection systems essentially utilize the following detection paradigm: several windows are placed at different positions and scales in the test image and a set of low-level features is computed from each window and fed into a classifier. Typically, the features used to describe the object of interest are the "normalized" gray-level values in the window. This generates a large number of features (of the order of a couple of hundred), whose classification is both time consuming and requires a large number of training samples to overcome the "curse of dimensionality". The main difference among these systems is the classification method: Moghaddam and Pentland [8] use a complex probabilistic measure, Rowley *et al.* [9] use a neural network while Colmenarez and Huang [10] use a Markov model.

One of the main performance indices used to evaluate such systems is the detection time. Most detection systems are inherently very slow since for each window (pixel in the test image), a feature vector with large dimensionality is extracted and classified. A novel way to perform the classification (called *Information-based Maximum Discrimination*) is introduced by Colmenarez and Huang [10]: the pattern vector is modeled by a Markov chain and its elements are rearranged such that they produce maximum discrimination between the sets of positive and negative examples. The parameters of the optimal Markov chain obtained after rearrangement are learned and a new observation is classified by thresholding its log-likelihood ratio. The main advantage of the method is that the log-likelihood ratio can be computed extremely fast, only one addition operation per feature is needed.

We propose to modify and adapt the Maximum Discrimination method [10] for left ventricle detection in MR cardiac images. The ventricle variations shown in Fig. 1 suggest that the ventricle detection problem is even more difficult than face detection. Our proposed method differs from that of Colmenarez and Huang in two significant ways:

1. Definition of the instance space. In [10] the instance space was defined as the set of 2-bit 11×11 non-equalized images of human faces. In our case, the ventricle diameter ranges from 20 to 100 pixels and a drastic subsampling of the image would loose the ventricle wall (the dark ring). On the other hand, even a 20×20 window would generate 400 features and the system would be too slow. Therefore, we used only four profiles passing through the ventricle (see Fig. 2) subsampled to define a total of 100 features.

2. Solution to the optimization problem. An approximate solution to a Traveling salesman type problem is computed in [10] using a minimum spanning tree algo-

rithm. Since the quality of the solution is crucial for the learning performance, we believe simulated annealing to be a better choice for our optimization problem.

II. Mathematical model

In order to learn a *pattern*, one should first specify the instance (feature) space from which the pattern examples are drawn. Since the left ventricle appears as a relatively symmetric object with no elaborate texture, it was not necessary to define the heart ventricle as the entire region surrounding it (the grey squares in Fig. 1). Instead, it was sufficient to sample four cross sections through the ventricle and its immediate neighborhood, along the four main directions (Fig. 2(a)). Each of the four linear cross sections was subsampled as to contain 25 points and the values were normalized in the range 0-7. The normalization scheme used here is a piece-wise linear transformation that maps the average gray level of all the pixels in the cross sections to a value 3, the minimum gray level is mapped to a value 0 and the maximum gray value is mapped to 7. In this way, a heart ventricle is defined as a feature vector $x = (x_1, ..., x_{100})$, where $x_i \in 0..7$ (Fig. 2(b)). We denote by Ω the instance space of all such vectors.

A. Markov Chain-based discrimination

We regard an observation as the realization of a random process $X = \{X_1, X_2, .., X_n\}$, where n is the number of features defining the object of interest and X_i's are random variables associated with each feature. We introduce two probabilities P and N over the instance space Ω:

$P(\mathbf{x}) = P(X = \mathbf{x}) = \text{Prob}(\mathbf{x} \text{ is a heart example})$, and
$N(\mathbf{x}) = N(X = \mathbf{x}) = \text{Prob}(\ \mathbf{x}$ is a non-heart example).

Since P and N can only be estimated from the training set which might be noisy, it is possible that $\mathrm{P}(\mathbf{x}) + N(\mathbf{x}) \neq 1$. In what follows, P and N will be treated as two independent probabilities over Ω. For each instance $\mathbf{x} \in \Omega$, we define its log-likelihood ratio $L(\mathbf{x}) = log\frac{P(\mathbf{x})}{N(\mathbf{x})}$. Note that $L(\mathbf{x}) > 0$ if and only if $\mathbf{x}$ is more probable to be a heart than a non-heart, while $L(\mathbf{x}) < 0$ if the converse is true.

The Kullback divergence between P and N can be regarded as the average of the log-likelihood ratio over the entire instance space [15]:

$$H_{P||N} = \sum_{\mathbf{x}\in\Omega} P(\mathbf{x}) log\frac{P(\mathbf{x})}{N(\mathbf{x})}. \qquad (1)$$

It has been shown that the Kullback divergence is not a distance metric. However, it is generally assumed that the larger $H_{P||N}$ is, the better one can discriminate between observations from the two classes whose distributions are P and N. It is not computationally feasible to estimate P and N taking into account all the dependencies between the features. On the other hand, assuming a complete independence of the features is not realistic because of the mismatch between the model and the data. A compromise is to consider the random process X to be a Markov chain, which can model the dependency in the data with a reasonable amount of computation.

Let us denote by S the set of feature sites with an arbitrary ordering $\{s_1, s_2, .., s_n\}$ of sites $\{1, 2, .., n\}$. Denote by $X_S = \{X_{s_1}, ..., X_{s_n}\}$ an ordering of the random variables that compose X corresponding to the site ordering $\{s_1, s_2, .., s_n\}$. If X_S is considered to be a first-order Markov chain then for $\mathbf{x} = (x_1, x_2, .., x_n) \in \Omega$ one has:

$P(X_S = \mathbf{x}) = P(X_{s_1} = x_1, ..., X_{s_n} = x_n) =$
$= P(X_{s_n} = x_n | X_{s_{n-1}} = x_{n-1}) \times ... \times$
$\times P(X_{s_2} = x_2 | X_{s_1} = x_1) \times P(X_{s_1} = x_1).$

Therefore, the log-likelihood ratio of the two distributions P and N under the Markov chain assumption can be written as follows:

$$L^S(\mathbf{x}) = log\frac{P(X_S=\mathbf{x})}{N(X_S=\mathbf{x})} =$$

$$= log(\frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)} \prod_{i=2}^{n} \frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}) =$$

$$= \sum_{i=2}^{n} log\frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})} + log\frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)} =$$

$$= L^{s_1}(x_1) + \sum_{i=2}^{n} L^{s_i||s_{i-1}}(x_i, x_{i-1}). \qquad (2)$$

The Kullback divergence of the two distributions P and N under the Markov chain assumption can be computed as follows:

$$H^S_{P||N} = H_{P||N}(X_{s_1}, ..., X_{s_n}) =$$

$$= \sum_{(x_1,..,x_n)\in\Omega} P(X_{s_1} = x_1, ..., X_{s_n} = x_n) log\frac{P(X_{s_1}=x_1,...,X_{s_n}=x_n)}{N(X_{s_1}=x_1,...,X_{s_n}=x_n)}$$

$$= \sum_{(x_1,..,x_n)\in\Omega} P(X_{s_1} = x_1, ..., X_{s_n} = x_n) log(\frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)}$$

$$\prod_{i=2}^{n} \frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}) = \sum_{i=2}^{n}$$

$$(\sum_{(x_i,x_{i-1})} P(X_{s_i} = x_i, X_{s_{i-1}} = x_{i-1}) log\frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})})$$

$$+ \sum_{x_1} P(X_{s_1} = x_1) log\frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)} =$$

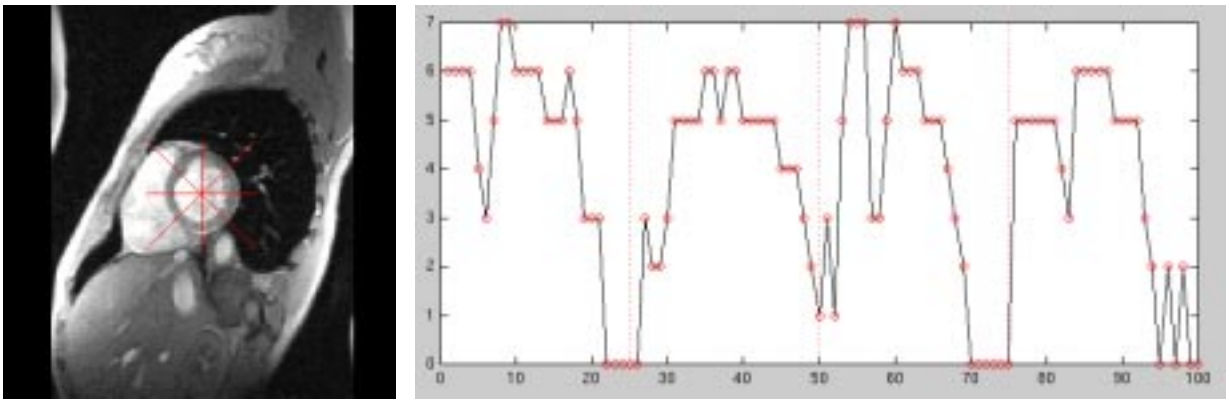

Fig. 2. The feature set defining a heart ventricle. (a) The four cross sections through the ventricle and its immediate surroundings used to extract the features. (b) The 100-element normalized feature vector associated with the ventricle in (a).

$$= H_{P||N}(X_{s_1}) + \sum_{i=2}^{n} H_{P||N}(X_{s_i}||X_{s_{i-1}}). \quad (3)$$

III. Most discriminant Markov chain

One can note that the divergence $H^S_{P||N}$ defined in Eq.(3) depends on the site ordering $\{s_1, s_2, .., s_n\}$ because each ordering produces a different Markov chain with a different distribution. The goal of the learning procedure is to find a site ordering S^* that maximizes $H^S_{P||N}$ which will result in the best discrimination between the two classes. The resulting optimization problem, although related to, is more difficult than the Traveling salesman problem since:

1. It is asymmetric (the conditional Kullback divergence is not symmetric, i.e. $H_{P||N}(X_{s_i}||X_{s_{i-1}}) \neq H_{P||N}(X_{s_{i-1}}||X_{s_i})$).

2. The salesman does not complete the tour, but remains in the last town.

3. The salesman starts from the first town with a handicap ($H_{P||N}(X_{s_1})$) which depends only on the starting point.

Therefore, the instance space of this problem is of the order of $n \times n!$, where n is the number of towns (feature sites), since for each town permutation one has n starting possibilities. It is well known that this type of problem is $NP - complete$ and cannot be solved by brute-force except for a very small number of sites. Although for the symmetric Traveling salesman problem there exist strategies to find both exact and approximate solutions in a reasonable amount of time, we are not aware of any heuristic for solving the asymmetric problem involved here. However, a good approximate solution can be obtained using simulated annealing [16]. Even though there is no theoretical guarantee to find an optimal solution, in practice, simulated annealing does almost always find a solution which is very close to the optimal (see also the discussion in [16]). Comparing the results produced by the simulated annealing algorithm on a large number of trials with the optimal solutions (for small size problems), we found that all the solutions produced by simulated annealing were within 5% of the optimal solutions.

Once S^* is found, one can compute and store tables with the log-likelihood ratios such that, given a new observation, its log-likelihood can be obtained from $n-1$ additions using Eq.(2).

The learning stage, which is described in Algorithm 1, starts by estimating the distributions P and N and the parameters of the Markov chains associated with *all* possible site permutations using the available training examples. Next, the site ordering that maximizes the Kullback distance between P and N is found, and the log-likelihood ratios induced by this ordering are computed and stored.

Algorithm 1: Finding the most discriminating Markov Chain

• Given a set of positive/negative training examples (as preprocessed n-dimensional feature vectors).

1. For each feature site s_i, estimate $P(X_{s_i} = v)$ and $N(X_{s_i} = v)$ for $v = 0..GL-1$ (GL = number of gray levels) and compute the divergence $H_{P||N}(X_{s_i})$.

2. For each site pair (s_i, s_j), estimate $P(X_{s_i} = v_1, X_{s_j} = v_2)$, $N(X_{s_i} = v_1, X_{s_j} = v_2)$, $P(X_{s_i} = v_1 | X_{s_j} = v_2)$ and $N(X_{s_i} = v_1 | X_{s_j} = v_2)$ for $v1, v2 \in 0..GL-1$ and compute $H_{P||N}(X_{s_i}||X_{s_j}) =$
$$= \sum_{v_1, v_2 = 0}^{GL-1} P_X(X_{s_i} = v_1, X_{s_j} = v_2) ln \frac{P_X(X_{s_i} = v_1 | X_{s_j} = v_2)}{N_X(X_{s_i} = v_1 | X_{s_j} = v_2)}$$

3. Solve a traveling salesman type problem over the sites S to find $S^* = \{s^*_1, ..., s^*_n\}$ that maximizes $H_{P||N}(X_S)$.

4. Compute and store $L(X_{s_1^*} = v) = ln \frac{P(X_{s_1^*}=v)}{N(X_{s_1^*}=v)}$ and

$L(X_{s_i^*} = v_1 || X_{s_{i-1}^*} = v_2) = ln \frac{P(X_{s_i^*}=v1 | X_{s_{i-1}^*}=v2)}{N(X_{s_i^*}=v1 | X_{s_{i-1}^*}=v2)}$ for $v, v_1, v_2 \in \{0..GL-1\}$.

IV. Classification procedure

The detection (testing) stage consists of scanning the test image at different scales with a constant size window from which a feature vector is extracted and classified. The classification procedure using the most discriminant Markov chain, detailed in Algorithm 2, is very simple: the log-likelihood ratio for that window is computed as a sum of conditional log-likelihood ratios associated with the Markov chain ordering (Eq.(2)). The total number of additions used is at most equal to the number of features.

Algorithm 2: Classification

• Given S^*, the best Markov chain structure and the learned likelihoods $L(X_{s_1^*} = v)$ and $L(X_{s_i^*} = v_1 || X_{s_{i-1}^*} = v_2)$.

• Given a test example $O = (o_1, ...o_n)$ (as preprocessed n-dimensional feature vector).

1. Compute the likelihood $L_O = L(X_{s_1^*} = o_{s_1^*}) + \sum_{i=2}^{n} L(X_{s_i^*} = o_{s_i^*} || X_{s_{i-1}^*} = o_{s_{i-1}^*})$.

2. If $L_O > T$ then classify O as heart else classify it as nonheart.

Here T is a threshold to be learned from the ROC curve of the training set depending on the desired (correct detect - false alarm) trade-off. In order to make the classification procedure faster, one can skip from the likelihood computation the terms with little discriminating power (associated Kullback distance is small).

V. Experimental results

A. Training Data

A collection of 1,350 MR cardiac images from 14 patients was used to generate positive training examples. The images were acquired using a Siemens Magnetom MRI system. For each patient, a number of slices (4 to 10) were acquired at different time instances (5 to 15) of the heart beat, thus producing a matrix of $2D$ images (in Fig. 4, slices are shown vertically and time instances are shown horizontally). As the heart is beating, the left ventricle is changing its size, but the scale factor between the end of diastolic and the end of systolic periods is negligible compared to the scale factor between slices at the base and the apex of the heart.

On each image, a tight bounding box (defined by the center coordinates and scale) containing the left ventricle was manually identified. From each cardiac image, 75 positive examples were produced by translating the manually defined box up to 2 pixels in each coordinate and scaling it up or down 10%. In this way, a total of 101,250 positive examples were generated. We also produced a total of 123,096 negative examples by uniformly subsampling a subset of the 1,350 available images at 8 different scales. The distributions of the log-likelihood values for the sets of positive and negative examples are shown in Fig. 3. They are very well separated, and by setting the decision threshold at 0, the resubstitution detection rate is 97.5% with a false alarm rate of 2.35%.

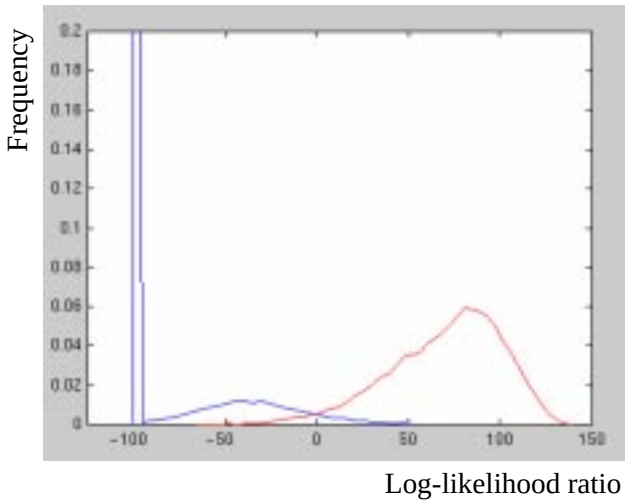

Fig. 3. The distribution of the log-likelihood ratio for heart (right) and non-heart (left) examples computed over the training set.

B. Test Data

We tested our algorithm on a dataset of 887 images (size 256×256) from 7 patients different from those used for training. Each image was subsampled at 8 different scales and scanned with a constant 25×25 pixel window using a step of 2 pixels in each direction. This means that, at each scale, a number of windows equal to a quarter of the number of pixels of the image at that scale was used for feature extraction and classification. All positions that produced a positive *log-likelihood ratio* were classified as hearts. Since several neighboring positions might have been classified as such, we partitioned them into clusters (a cluster was considered to be a set of image positions classified as hearts that had

Resubstitution detection rate	97.5%
Resubstitution false alarm rate	2.35%
Test set size (# of 256×256 images)	887
Test set detection rate	98%
Test set false alarms per image	10
Test set false alarm rate/windows analyzed	0.05%
Detection time/image (Sun Ultra 10)	2 sec

TABLE I

PERFORMANCE OF THE LEFT VENTRICLE DETECTION ALGORITHM.

a distance smaller than 25 pixels to its centroid). At each scale, only the cluster centroids were reported, together with the *log-likelihood ratio* value for that cluster (a weighted average of the *log-likelihood ratio* values in the cluster).

It was not possible to choose the best scale/position combination based on the *log-likelihood* value of a cluster. That is, values of the *log-likelihood* criterion obtained at different scales are *not comparable*: in about 25% of the cases, the largest *log-likelihood* value failed to represent the real scale/position combination. Therefore, we report *all* cluster positions generated at different scales (an average of 11 clusters are generated per image by combining all responses at different scales). Even if we could not obtain a single scale/position combination per image using this method, the real combination was among those 11 clusters reported in 98% of the cases. Moreover, the 2% failure cases came only from the bottom most slice, where the heart is very small (15-20 pixels in diameter) and looks like a homogeneous grey disk. We suspect that these situations were rarely encountered in the training set, so they could not be learned very well. The quantitative results of the detection task are summarized in Table I. The false alarm rate has been greatly reduced by reporting only cluster centroids.

We could select the best hypothesis by performing a consistency check along all the images that represent the same slice: our prior knowledge states that, in time, one heart slice does not modify its scale/position too much, while consecutive spatial slices tend to be smaller. By enforcing these conditions, we could obtain complete spatio-temporal hypotheses about the heart location. A typical detection result on a complete spatio-temporal (8 slice positions, 15 sampling times) sequence of one patient is shown in Fig. 4).

VI. Conclusion

In order to detect the left ventricle in MR cardiac images, we have proposed a new approach based on learning the ventricle gray level appearance. The method has been successfully tested on a large dataset and shown to be very fast and accurate. The detection results can be summarized as follows: 98% detection rate, a false alarm rate of 0.05% of the number of windows analyzed (10 false alarms per image) and a detection time of 2 seconds per 256×256 image on a Sun Ultra 10 for an 8-scale search. The false alarms are eventually eliminated by a position/scale consistency check along all the images that represent the same anatomical slice.

Acknowledgments

This work was supported by a grant from Siemens Corporate Research, Princeton.

References

[1] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 14(11):1061–1075, 1992.

[2] N. Ayache, I. Cohen, and I. Herlin. Medical image tracking. In *Active Vision*, A. Blake and A. Yuille (Eds.), 1992. MIT Press.

[3] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

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

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

[6] C. H. Chiu and D. H. Razi. A nonlinear multiresolution approach to echocardiographic image segmentation. *Computers in Cardiology*, pages 431–434, 1991.

[7] J. G. Bosch, J. H. C. Reiber, Burken G., J. J. Gerbrands, A. Kostov, van de A. J. Goor, M. Daele, and J. Roelander. Developments towards real time frame-to-frame automatic contour detection from echocardiograms. *Computers in Cardiology*, pages 435–438, 1991.

[8] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 19(7):696–710, 1997.

[9] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 20(1):23–38, 1998.

[10] A. Colmenarez and T. Huang. Face detection with information-based maximum discrimination. In *Proceedings of CVPR-'97*, pages 782–787, San Juan, Puerto Rico, 1997.

[11] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 19:1300–1306, 1997.

[12] A. L. Ratan, W. E. L. Grimson, and Wells W. M. Object detection and localization by dynamic template warping. In *Proceedings of CVPR '98*, pages 634–640, Santa Barbara, CA, 1998.

[13] S. K. Nayar, H. Murase, and S. Nene. Parametric Appearance Representation. In *Early Visual Learning*, pages 131–160, S. K. Nayar and T. Poggio (Eds.), 1996. Oxford University Press.

[14] T. Poggio and D. Beymer. Regularization Networks for Visual Learning. In *Early Visual Learning*, pages 43–66, S. K. Nayar and T. Poggio (Eds.), 1996. Oxford University Press.

[15] R. M. Gray. *Entropy and Information Theory.* Springer-Verlag, Berlin, 1990.

[16] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing.* Wiley, Chichester, 1989.

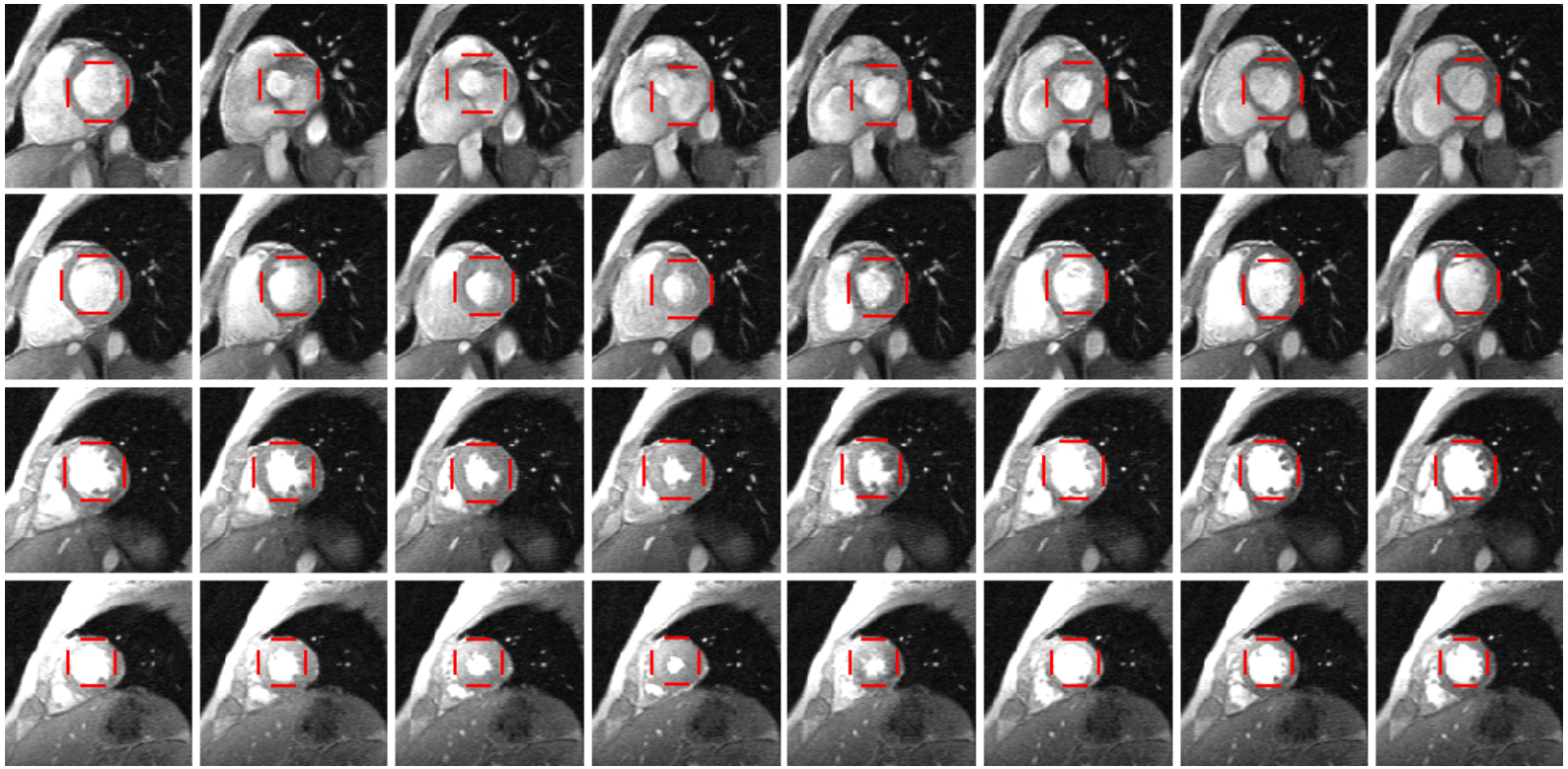

Fig. 4. Results of the detection algorithm on a complete spatio-temporal image sequence.