Multiple Dynamic Models for Tracking the Left Ventricle of the Heart from Ultrasound Data using Particle Filters and Deep Learning Architectures

Gustavo Carneiro* and Jacinto C. Nascimento*
Instituto de Sistemas e Robótica
Instituto Superior Técnico, Lisbon, Portugal.

Abstract

The problem of automatic tracking and segmentation of the left ventricle (LV) of the heart from ultrasound images can be formulated with an algorithm that computes the expected segmentation value in the current time step given all previous and current observations using a filtering distribution. This filtering distribution depends on the observation and transition models, and since it is hard to compute the expected value using the whole parameter space of segmentations, one has to resort to Monte Carlo sampling techniques to compute the expected segmentation parameters. Generally, it is straightforward to compute probability values using the filtering distribution, but it is hard to sample from it, which indicates the need to use a proposal distribution to provide an easier sampling method. In order to be useful, this proposal distribution must be carefully designed to represent a reasonable approximation for the filtering distribution. In this paper, we introduce a new LV tracking and segmentation algorithm based on the method described above, where our contributions are focused on a new transition and observation models, and a new proposal distribution. Our tracking and segmentation algorithm achieves better overall results on a previously tested dataset used as a benchmark by the current state-of-the-art tracking algorithms of the left ventricle of the heart from ultrasound images.

1. Introduction

The automatic tracking and segmentation of the endocardial border of the Left Ventricle (LV) of the heart from ultrasound data is an important tool to analyze the health of the heart. For instance, with such segmentation in time, it is possible for a clinician to provide a quantitative functional analysis of the heart, such as the ejection fraction estimation. The automation of this procedure is desirable in a clinical setting mainly due to the following reasons: 1) it can improve the workflow in a typical clinical environment by increasing the patient throughput; and 2) it can decrease the variability between user measurements. However, in order to be useful, automatic LV tracking and segmentation systems have to handle several problems inherent to ultrasound imaging, such as: fast motion during systole (contraction) phase, low signal-to-noise ratio, edge dropout caused by motion, and the presence of shadows produced by the dense muscles.

The tracking algorithm to solve this problem can be formulated to compute the expected segmentation given the previous and current observations over the space of segmentation parameters [6]. In this framework, the segmentation parameters constitutes the state vector while the image represents the observation vector. The expected segmentation described above is computed using the filtering distribution, which calculates the probability of a certain segmentation given previous and current observations. The computation of this expected value is usually not tractable given the high number of dimensions of the space of segmentation parameters. As a result, it is common to approximate this expected value using Sequential Monte Carlo (SMC) sampling techniques, which means that only a few weighted samples (each sample represents a segmentation) are used to produce the expected value. The weights in the samples are computed using the observation and transition models, while the samples are obtained from sampling the filtering distribution. Another usual problem is the difficulty in sampling this filtering distribution, which is solved by sampling another distribution (called the proposal distribution) that provides a reasonable approximation to the filtering distribution, but is much simpler to sample. Then the probability of the proposal distribution has to be taken into account when calculating the sample weights. Finally, using the samples and their respective weights, it is possible to compute the expected segmentation mentioned above. Below we provide a brief discussion on the observation and transition models and on the proposal distributions.

The solutions proposed for the observation model can be categorized into two classes: 1) low-level methods that use prior models of the LV appearance, and 2) pattern recognition methods based on appearance models automatically built from manually annotated LV images. Low-level methods [5, 10, 13] consist of segmentation algorithms that use a prior model of the LV based on the assumptions that the

*This work was supported by the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds and Project HEARTTRACK (PTDC/EEA-CRO/103462/2008). This work was partially funded by EU Project IMASEG3D (PIIF-GA-2009-236173).

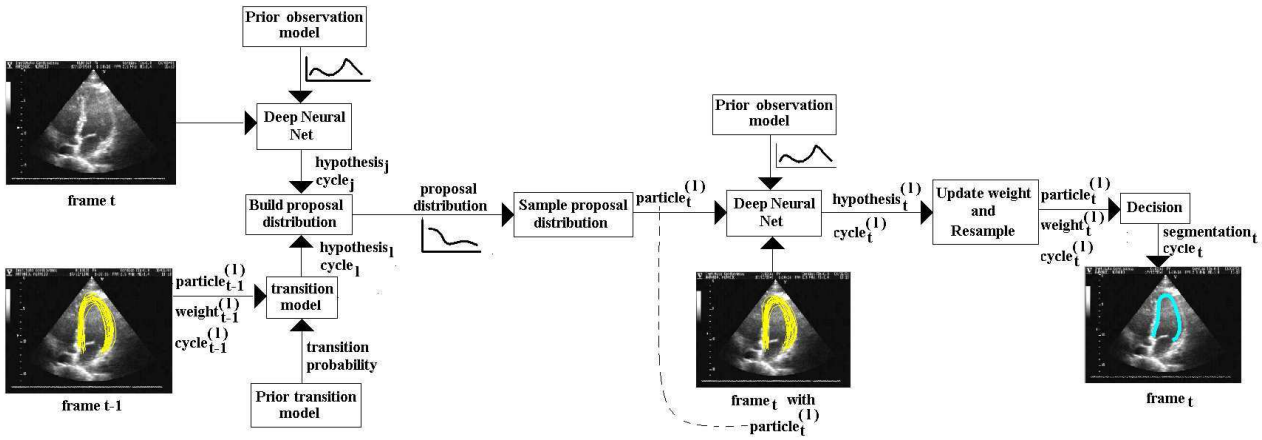


Figure 1. Block diagram containing all steps of the tracking algorithm proposed in this paper.

myocardium displays brighter, and the blood pool in the LV displays darker than other structures in the image. The main problem with this approach is that the violation of these assumptions may lead to incorrect segmentations. Pattern recognition methods involve the use of a database of annotated LV images (*i.e.*, a training set) to automatically build a model of the LV appearance [4, 7]. Even though this approach currently holds the most competitive results [11], it still faces a few challenges, such as: the need of a large training set, robustness to imaging conditions unseen in the training set, and the run-time complexity of the search process. Lately, there has been a significant effort to reduce the search complexity. For instance, the marginal space learning (MSL) [20], which partitions the search space into subspaces of increasing complexity, achieves a significant complexity reduction, but the search methods proposed by our paper are orthogonal to it, meaning that our search methods can be easily integrated into MSL. Another contribution [21] was a pattern recognition approach that, given any position in the search space, the method outputs a gradient vector that optimizes the LV segmentation function. This approach is likely to work as long as the searching region is sufficiently close to a local optimum of the objective function. In addition, the training procedure is likely to need a larger training set due to the much higher number of parameters to be learned in the gradient vector.

Though receiving relatively less attention than the observation model, the transition model plays an important role in the computation of the filtering distribution. The most typical transition model is the prediction estimated from the Kalman filtering [18], but the assumption of a Gaussian distribution is not appealing given the complex motion patterns of the heart. More interesting transition models are built when providing more degrees of freedom to explain those motion patterns. For instance, Sun *et al*. [17] introduce a transition model that is learned from training data using an information-theoretic criterion, but the lack of a prior distribution in the model imposes the need of a large and thorough training set to build a reliable transition model. A similar approach is proposed by Yang *et al*. [19] consisting of a transition model that depends not only on the previous state vector, but also on all state vectors up to current time instant. This model is also automatically learned from training data and consists of a manifold describing the motion pattern of the heart. Note that the dependence on a large and complete training set is even more blatant in this work given the large number of parameters in the model. Models based solely on prior information [16] also seem inadequate given that there might be information present in training data that may not have been captured by the prior. The transition model proposed by Nascimento [11] consisting of a mixture of two models (one for systole and another for diastole) is in the right direction, and inspired us to implement our transition model. The main difference is that we use both a prior information on the motion patterns, assuming the existence of two cardiac cycles (*i.e*., systole and diastole), and a learned model from data instead of a transition model containing only prior distributions [11].

Finally, for tracking methods based on SMC sampling techniques, it is necessary to use a proposal distribution that approximates reasonably well the filtering distribution. Senegas *et al*. [16] propose an SMC sampling method using a proposal distribution based only on the observation model, which is a limitation because it does not take into account the transition model. At each new time step, each sample is re-weighted, and if its weight falls below a threshold, then it gets discarded. Sun *et al*. [17] introduce a proposal distribution based only on the transition model, which is certainly a limitation given that it does not take into account the observation model. The work that inspired our model was proposed by Okuma *et al*. [12], who proposed a tracking algorithm (*i.e*., not LV tracking) combining discriminative classifier detections and particle filtering to track multi-target non-rigid objects, but note that the requirements in terms of segmentation precision is not as high as in the LV segmentation. Their main contribution is the fact that the proposal

distribution is built based on the transition and observation models. We adapt this idea to the LV tracking problem and extend it with the addition of prior models.

In this paper, we propose a new LV tracking algorithm based on SMC methods (Fig. 1). Our main contributions are a new transition model, a new observation model based on deep learning architectures and the formulation of a new proposal distribution. The transition model proposed in this paper makes use of the prior information that at each time instant, the heart is either expanding (diastole) or contracting (systole). The deformation caused by these motion patterns are described by a linear transform, whose parameters are learned from the training data. The new observation model is based on deep learning architectures, which involves a statistical pattern recognition model, where we address two of the problems present in pattern recognition methods, namely: 1) robustness to imaging conditions unseen in training data, and 2) run-time complexity of the search process. In order to handle the robustness to imaging conditions, we move away from the use of boosting classifiers [7], and rely on the use of deep neural network classifiers [15]. The main advantage of deep neural networks is its ability to produce more abstract feature spaces for classification and to automatically generate optimum feature spaces directly from image data. In order to tackle the complexity issue, we study the use of optimization algorithms of first and second orders [2]. The main difference compared to the work by Zhou and Comaniciu [21] is that we compute the gradient vector and Hessian matrix directly from the output of the classifiers, imposing no additional requirements for the training set. Finally, our proposal distribution is inspired by the work of Okuma *et al*. [12], which combines the detection results from the deep learning architecture with the transition model. This combination provides precise segmentation, and robustness to imaging conditions and drifting. We show quantitative comparisons between our method and state-of-the-art approaches [4, 7, 11] and with a previous version of this system [3], which uses only the observation model for segmenting the LV (*i.e*., the dynamic model was not implemented), and the results not only show a superior performance of our approach, but they also display that efficient search methods maintain the original accuracy of the method while reducing ten times the runtime complexity.

2. Tracking of the Left Ventricle

The main problem we wish to solve in this paper is the tracking and segmentation of the left ventricle in an ultrasound sequence $I_{1:t}$, which denotes the images from time instant 1 to t. The segmentation is denoted by a set of points $\mathcal{S}_t = \{\mathbf{s}_{i,t}\}_{i=1..N}$, with $\mathbf{s}_{i,t} \in \Re^2$ representing the the i^{th} of the N points forming the LV contour. Recall that in our system, the images represent the observations and the segmentation denotes the state vector. We assume the existence of a training set $\mathcal{D} = \{(I, \theta, \mathcal{S}, K)_i\}_{i=1..M}$ containing M training images I_i of the imaging of LV using ultrasound, a respective manual annotation $\mathcal{S}$, the parameters

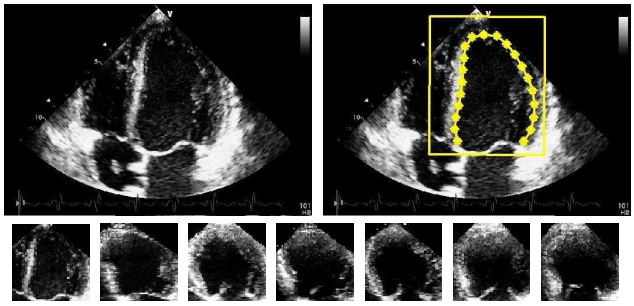


Figure 2. Original training image (top left) with the manual LV segmentation in yellow line and star markers (top right) with the rectangular patch representing the canonical coordinate system for the segmentation markers. The bottom images display several patches extracted from the annotated training images, which will be used to train the rigid classifier.

of a rigid transformation $\theta \in \Re^5$ (position $\mathbf{x} \in \Re^2$, orientation $\gamma \in [-\pi, \pi]$, and scale $\sigma \in \Re^2$) that aligns rigidly the annotation points to a canonical coordinate system (see Fig.2) and $K \in \{\text{systole,diastole}\}$, which denotes the type of cardiac cycle.

The goal of our tracking algorithm is to produce the LV contour $\mathcal{S}_t$ for current time instant t given all previous observations $I_{1:t}$, as follows:

$$\mathcal{S}_t^* = \int_{\mathcal{S}_t} \mathcal{S}_t p(\mathcal{S}_t | I_{1:t}, y_1, \mathcal{D}) d\mathcal{S}_t, \tag{1}$$

where y_1 is a random variable indicating the presence of the LV in the image region represented by the contour (we provide more details below). The integral in (1) cannot be computed in a reasonable amount of time, so we use particle filtering to estimate it. With particle filtering, the filtering distribution $p(\mathcal{S}_t | I_{1:t}, y_1, \mathcal{D})$ is approximated with a set of weights and particles: $\{w_t^{(l)}, \mathcal{S}_t^{(l)}\}$, which can approximate the segmentation as [6]:

$$\mathcal{S}_t^* \approx \sum_{\mathcal{S}_t^{(l)}} \mathcal{S}_t^{(l)} w_t^{(l)}, \tag{2}$$

with $w_t^{(l)} \approx p(\mathcal{S}_t^{(l)} | I_{1:t}, y_1, \mathcal{D})$ and $\sum_l w_t^{(l)} = 1$. One of the issues of particle filtering is that while it is easy to compute $p(\mathcal{S}_t^{(l)} | I_{1:t}, y_1, \mathcal{D})$ for a contour $\mathcal{S}_t^{(l)}$, it is hard to sample from this distribution, so it is necessary to have a reasonable proposal distribution (to sample from) that approximates well the filtering distribution.

The particle filtering algorithm used in this paper is the sampling importance re-sampling (SIR) [1], which has the following steps (Fig. 1):

1. Draw P samples from proposal distribution using:
   for $l = 1 : P$, sample
   $\mathcal{S}_t^{(l)} \sim q(\mathcal{S}_t | \mathcal{S}_{1:t-1}^{(l)}, I_{1:t}, y_1, \mathcal{D})$;
2. Update sample weights:
   $\tilde{w}_t^{(l)} = w_{t-1}^{(l)} \frac{p(I_t | \mathcal{S}_t^{(l)}, y_1, \mathcal{D}) p(\mathcal{S}_t | \mathcal{S}_{t-1}^{(l)}, \mathcal{D})}{q(\mathcal{S}_t^{(l)} | \mathcal{S}_{1:t-1}^{(l)}, I_{1:t}, y_1, \mathcal{D})}$;

3. Normalize sample weights: $w_t^{(l)} = \frac{\tilde{w}_t^{(l)}}{\sum_l \tilde{w}_t^{(l)}}$;

4. Compute effective number of particles $N_{eff} = 1/\sum_l (w_t^{(l)})^2$;

5. If $N_{eff} < K_{Neff} \times P$, then re-sample by drawing P particles from current particle set proportionally to weight and replace particle, and set $w_t^{(l)} = 1/P$.

In this section, we first discuss the filtering and predictive distributions. Then we explain how the transition and observation models work, followed by an explanation of the proposal distribution.

2.1. Filtering and Predictive Distributions

The filtering distribution is the posterior distribution in (1) that can be expanded as in:

$$p(\mathcal{S}_t|I_{1:t}, y_1, \mathcal{D}) = \frac{p(I_t|\mathcal{S}_t, y_1, \mathcal{D})p(\mathcal{S}_t|I_{1:t-1}, y_1, \mathcal{D})}{p(I_t|I_{1:t-1}, y_1, \mathcal{D})} \tag{3}$$

where the denominator can be re-written as:

$$p(I_t|I_{1:t-1}, y_1, \mathcal{D}) = \int_{\mathcal{S}_t} p(I_t|\mathcal{S}_t, y_1, \mathcal{D})p(\mathcal{S}_t|I_{1:t-1}, y_1, \mathcal{D})d\mathcal{S}_t. \tag{4}$$

Finally, we also need to expand the predictive model in (3) as follows:

$$p(\mathcal{S}_t|I_{1:t-1}, y_1, \mathcal{D}) = \int_{\mathcal{S}_{t-1}} p(\mathcal{S}_t|\mathcal{S}_{t-1}, I_{1:t-1}, y_1, \mathcal{D})p(\mathcal{S}_{t-1}|I_{1:t-1}, y_1, \mathcal{D})d\mathcal{S}_{t-1}. \tag{5}$$

2.2. The Transition Model

One of our contributions resides in the definition of the transition model, which takes into account the cardiac cycle observed in the previous instant and the prediction for the next step in the cycle. We simplify the problem by assuming that the cycle is composed of two phases, namely the systole (contraction) and diastole (relaxation). The transition model in the predictive model (5) is defined as follows:

$$p(\mathcal{S}_t|\mathcal{S}_{t-1}, I_{1:t-1}, y_1, \mathcal{D}) = \sum_{K_{t-1}} p(\mathcal{S}_t|\mathcal{S}_{t-1}, I_{1:t-1}, K_{t-1}, y_1, \mathcal{D})p(K_{t-1}|\mathcal{S}_{t-1}, I_{t-1}, y_1, \mathcal{D}), \tag{6}$$

with $K_{t-1} \in \{\text{systole}, \text{diastole}\}$. The first term on the right hand side (RHS) of (6) represents the probability of a segmentation $\mathcal{S}_t$ given the cycle in $t-1$. Noting that cycle changes can be considered not frequent in a cardiac cycle, we provide a transition distribution which tends to maintain the current cycle using a Gaussian model learned from the training set that indicates the expected contour given the previous cycle value. This means that in (6), we have:

$$p(\mathcal{S}_t|\mathcal{S}_{t-1}, I_{1:t-1}, K_{t-1}, y_1, \mathcal{D}) = G(\mathcal{S}_{t-1}|M(K_{t-1})\mathcal{S}_{t-1}, \Sigma_{\mathcal{S}}), \tag{7}$$

where $M(K_{t-1})$ is a linear transform applied to $\mathcal{S}_{t-1}$ learned from the training data that expands or contracts the contour depending on phase K_{t-1} of the cycle, as shown in Fig. 4-(b), $\Sigma_{\mathcal{S}}$ is the covariance of the annotations $\mathcal{S}$ learned from the training data, and $G(\mathbf{x}|\mu, \Sigma)$ computes the probability of $\mathbf{x}$ using a Gaussian distribution of mean μ and covariance Σ. In (6), the term $p(K_{t-1}|\mathcal{S}_{t-1}, I_{1:t-1}, y_1, \mathcal{D})$ corresponds to the probability of cycle K_{t-1} in $t-1$ given the contour was detected. We show in the observation model below how this value can be computed.

2.3. The Observation Model

The observation model in (3) can be defined as

$$p(I_t|\mathcal{S}_t, y_1, \mathcal{D}) \propto p(\mathcal{S}_t|I_t, y_1, \mathcal{D})p(I_t|y_1, \mathcal{D}), \tag{8}$$

where $p(I_t|y_1, \mathcal{D}) = C$ (here, assume $C = 1$), and

$$p(\mathcal{S}_t|I_t, y_1, \mathcal{D}) = \int_\theta p(\mathcal{S}_t|\theta, I_t, y_1, \mathcal{D})p(\theta|I_t, y_1, \mathcal{D})d\theta. \tag{9}$$

The first RHS term in (9), representing the non-rigid part of the detection, is defined as follows:

$$p(\mathcal{S}_t|\theta, I_t, y_1, \mathcal{D}) = \prod_i p(\mathbf{s}_{i,t}|\theta, I_t, y_1, \mathcal{D}), \tag{10}$$

where $p(\mathbf{s}_{i,t}|\theta, I_t, y_1, \mathcal{D})$ represents the probability that the point $\mathbf{s}_{i,t}$ is located at the LV contour. Assuming that ψ denotes the parameter vector of the classifier for the non-rigid contour, we compute

$$p(\mathbf{s}_{i,t}|\theta, I_t, y_1, \mathcal{D}) = \int_\psi p(\mathbf{s}_{i,t}|\theta, I_t, y_1, \mathcal{D}, \psi)p(\psi|\mathcal{D})d\psi. \tag{11}$$

In practice, we run a maximum a posteriori learning procedure of the classifier parameters, which produces ψ_{MAP}, meaning that in the integral (11) we have $p(\psi|\mathcal{D}) = \delta(\psi - \psi_{\text{MAP}})$, where $\delta(.)$ denotes the Dirac delta function. Also, instead of computing the probability $p(\mathbf{s}_{i,t}|\theta, I_t, y_1, \mathcal{D})$, we train a regressor that indicates the most likely edge location (see Fig.3); this roughly means that $p(\mathbf{s}_{i,t}|\theta, I_t, y_1, \mathcal{D}) = \delta(\mathbf{s}_{i,t} - \mathbf{s}^r_{i,t}(\theta, I_t, y_1, \mathcal{D}))$, with $\mathbf{s}^r_{i,t}(.)$ being the regressor result for the i^{th} contour point, so Eq. 9 is effectively $\int_\theta \left(\prod_i \delta(\mathbf{s}_{i,t} - \mathbf{s}^r_{i,t}(\theta, I_t, y_1, \mathcal{D}))\right) p(\theta|I_t, y_1, \mathcal{D})d\theta$.

The second RHS term in (9) represents the rigid detection, which is denoted as

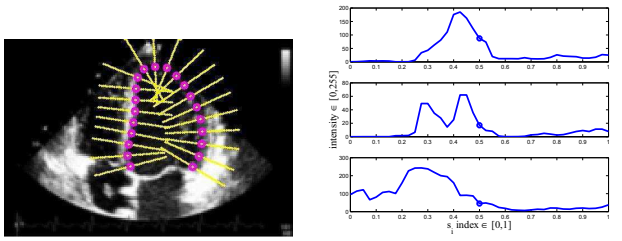
$$p(\theta|I_t, y_1, \mathcal{D}) = \mathcal{Z}p(y_1|\theta, I_t, \mathcal{D})p(\theta|I_t, \mathcal{D}) \tag{12}$$

where $\mathcal{Z}$ is a normalization constant, $p(\theta|I_t, \mathcal{D})$ is a prior on the parameter space, and

$$p(y_1|\theta, I_t, \mathcal{D}) = \int_\gamma p(y_1|\theta, I_t, \mathcal{D}, \gamma)p(\gamma|\mathcal{D})d\gamma, \tag{13}$$

with γ being the vector of classifier parameters, which are estimated through a maximum a posteriori learning procedure, producing γ_{MAP}. This means that in (13) $p(\gamma|\mathcal{D}) = \delta(\gamma - \gamma_{\text{MAP}})$.

The observation model introduced in this section is used in the following steps of the SIR algorithm:



a) lines perpendicular to annotation points

(b) intensity value profiles

Figure 3. Intensity value profiles (from inside to outside the LV) of the lines drawn perpendicularly to annotation points. These profiles represent the training data used to learn the regressor parameters.

- Proposal distribution generation for $t > 1$ (see Sec. 2.4). This step involves a search in the space of segmentations $\mathcal{S}_t$ to find a set of M segmentations that maximizes the probability (9), as follows:

$$\{\tilde{\mathcal{S}}_t\} = \text{top}_M \arg\max_{\mathcal{S}_t} p(\mathcal{S}_t|I_t, y_1, \mathcal{D}), \quad (14)$$

  where the operator top_M returns the arguments $\mathcal{S}_t$ that produce the M highest values for the probability $p(\mathcal{S}_t|I_t, y_1, \mathcal{D})$.

- Initial proposal distribution generation at $t = 1$. Given the set $\{\tilde{\mathcal{S}}_1\}$ in (14) denoting the top M detections for $t = 1$, the initial proposal distribution, from which the first P samples are drawn from, is defined as:

$$\mathcal{S}_1^{(l)} \sim \sum_{\{\tilde{\mathcal{S}}_1\}} \mathcal{Z} p(\tilde{\mathcal{S}}_1|I_1, y_1, \mathcal{D}) G(\mathcal{S}_1|\tilde{\mathcal{S}}_1, \Sigma_{\mathcal{S}}), \quad (15)$$

  where $l \in \{1, ..P\}$ denotes the index to the l^{th} particle, the constant $\mathcal{Z}$ guarantees that $\sum_{\{\tilde{\mathcal{S}}_1\}} \mathcal{Z} p(\tilde{\mathcal{S}}_1|I_1, y_1, \mathcal{D}) = 1$, and $G(.|\mu, \Sigma)$ is a Gaussian distribution with mean μ and covariance Σ, with $\Sigma_{\mathcal{S}}$ learned from the training data.

- Calculation of the probability values $p(I_t|\mathcal{S}_t^{(l)}, y_1, \mathcal{D})$ for each particle $\mathcal{S}_t^{(l)}$ using the filtering distribution (3). This step involves the calculation of the probability of $p(\mathcal{S}_t|I_t, y_1, \mathcal{D})$ according to (9), which does not involve any search process.

- Computation of the probability of the cycle $p(K_{t-1}|\mathcal{S}_{t-1}, I_{t-1}, y_1, \mathcal{D})$ in the transition model (6) given the segmentation $\mathcal{S}_{t-1}$. Notice that our training set $\mathcal{D}$ includes the annotation of the cycle in each training image, which means that the classifier (9) also computes $p(K_{t-1}|\mathcal{S}_{t-1}, I_{t-1}, y_1, \mathcal{D})$, where $\sum_{K_{t-1}} p(K_{t-1}|\mathcal{S}_{t-1}, I_{t-1}, y_1, \mathcal{D}) = 1$.

Finally, it is important to note that the observation model also uses a shape model consisting of a principal component analysis (PCA) shape space for systole, and another PCA space for diastole. These PCA spaces are learned from the annotated training set $\mathcal{D}$. During detection, given $\mathcal{S}_t$, we first find the detected cycle as

$$K_t^* = \arg\max_{K_t} p(K_t|\mathcal{S}_t, I_t, y_1, \mathcal{D}), \quad (16)$$

and project $\mathcal{S}_t$ onto the PCA space indicated by $K_t^* \in \{\text{systole,diastole}\}$. This projection smooths the segmentation $\mathcal{S}_t$.

2.3.1 Deep Neural Network

The effective use of large-scale conventional neural network classifiers (with several hidden layers and thousands of nodes) is limited because backpropagation [14] (algorithm to estimate the classifier parameters) converges only when the initial guess for the parameter values are close to a local optimum of the optimization function. Hinton et al. [15] found a way to provide such initial guesses through unsupervised training of multiple layers of restricted Boltzmann machines (RBM), which are represented by a hidden and a visible layer of stochastic binary units with connections only between layers (*i.e.*, no connections within layers). After the parameters of several layers of RBMs were learned, the whole network is trained using backpropagation to adjust the weights to a local maximum for the regressor and classifier functions. For the regressor in (11), we find the solution for the maximization function $\psi_{\text{MAP}} = \arg\max_\psi p(\{\mathcal{S}_i\}_{i=1..N}|\{(I,\theta)_i\}_{i=1..N}, \psi)$, where $(I, \theta, \mathcal{S})_i \in \mathcal{D}$. For the classifier (13), we find the solution for $\gamma_{\text{MAP}} = \arg\max_\gamma p(y = 1|\{(I,\theta)_i\}_{i=1..N}, \gamma)$.

2.3.2 Efficient Search Methods

For the detection of the LV in the generation of proposal distributions, there is a five dimensional space for the rigid detection and N dimensions for the non-rigid search space, resulting in a search space of E^{5+N_E} samples, which is too high for most of practical values of $E \in [10^2, 10^3]$ and $N_E \in \{10, ..., 25\}$. Running the search procedure on the image pyramid, with one classifier per image scale, reduces the search space significantly. The advantage here is to reduce the number of samples in the coarsest scale to E_{coarse}, and move to finer scales only the best $E_{\text{fine}} \in [10, 30]$ candidates. Note that the search procedure in fine scales needs to happen only around the current search point, meaning 3^5 (3 points in 5 dimensions) samples for each of the E_{fine} positions. Moreover, performing the non-rigid search only after the rigid search is done means a total search space of $E_{\text{coarse}}^5 + (\#\text{scales} - 1) \times E_{\text{fine}} \times 3^5 + N_E \times E_{\text{fine}}$.

In order to reduce the search space we assume a prior distribution on the coarse search space, and sample E_{coarse} times from this distribution (Monte-Carlo sampling), which means a search space of $E_{\text{coarse}} + (\#\text{scales} - 1) \times E_{\text{fine}} \times 3^5 + N_E \times E_{\text{fine}}$. Our second contribution is the implementation of efficient search procedures in order to reduce the exhaustive search of 3^5 points around the hypotheses.

We use two methods that are widely used in optimization algorithms, which are: gradient descent and Newton step [2]. These methods work for convex functions, and their use in non-convex functions, such as the ones produced by the deep neural net classifiers, only works with a sufficiently large number of E_{coarse}. In gradient descent, $\nabla p(y = 1|\theta, I, \mathcal{D}, \gamma_{MAP})$ is computed numerically using central difference, representing a computation of the classifier in 10 points of the search space (five parameters times two points) plus the line search in 10 points. By limiting the number of iterations between one and five for each hypothesis, the search space is then reduced to 20 to 100 points, which is smaller than $3^5 = 243$. In theory, a faster convergence can be achieved with the Newton step, but the computation of the Hessian matrix, gradient and line search involves 25+10 search space points. Limiting the number of iterations between one and five means that the complexity of this step for one hypothesis is between 35 to 175, which is also smaller than $3^5 = 243$.

2.4. The Proposal Distribution

The proposal distribution is another important contribution of this work. It consists of multiple dynamic models, where each model is built for each particle at each time instant t based on the observation model (*i.e.*, the detections from the deep neural network) and the particle from time $t-1$. The proposal distribution at time t is defined with a mixture of Gaussians, as follows [12] (Fig 4-(a)):

$$q(\mathcal{S}_t|\mathcal{S}^{(l)}_{1:t-1}, I_{1:t}, y_1, \mathcal{D}) \sim \\ \alpha q_{obs}(\mathcal{S}_t|K_t, y_1, I_{1:t}, \mathcal{D}) + (1-\alpha)p(\mathcal{S}_t|\mathcal{S}_{t-1}, \mathcal{D}), \tag{17}$$

where the observation model is represented by

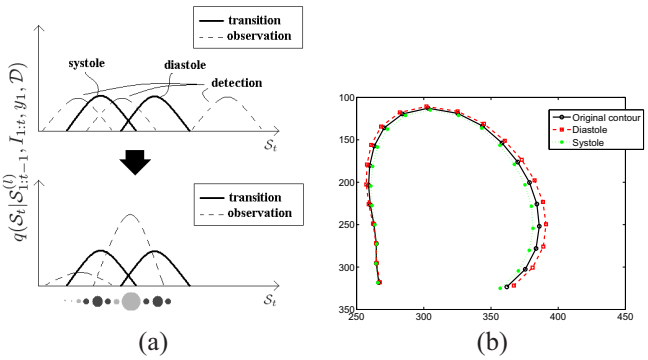
$$q_{obs}(\mathcal{S}_t|K_t, y_1, I_{1:t}, \mathcal{D}) = \sum_{\tilde{\mathcal{S}}_t} \mathcal{Z} p(\tilde{\mathcal{S}}_t|I_t, y_1, \mathcal{D}) G(\mathcal{S}_t|\tilde{\mathcal{S}}_t, \Sigma_{\mathcal{S}}), \tag{18}$$

which is a Gaussian mixture model indicating the probability distribution taking the top M detections $\{\tilde{\mathcal{S}}_t\}$ from the observation model defined in (14), where the normalization constant $\mathcal{Z}$ assures that $\sum_{\tilde{\mathcal{S}}_t} \mathcal{Z} p(\tilde{\mathcal{S}}_t|I_t, y_1, \mathcal{D}) = 1$, and $p(\tilde{\mathcal{S}}_t|I_t, y_1, \mathcal{D})$ is the probability response from the observation model given a specific segmentation (*i.e.*, it does not involve a search process). In Fig. 4-(a), the distribution $q_{obs}(.)$ is represented by the dashed curves. The transition model is then represented by:

$$p(\mathcal{S}_t|\mathcal{S}_{t-1}, \mathcal{D}) = \\ \sum_{K_{t-1} \in \{\text{systole,diastole}\}} 0.5 \times G(\mathcal{S}_t|M(K_{t-1})\mathcal{S}_{t-1}, \Sigma_{\mathcal{S}}), \tag{19}$$

where $M(K_{t-1})$ is the linear transform (Fig. 4-(b)) learned from the training set that describes the parameters of a transformation during the systole and diastole phases (solid lines in Fig. 4-(a)). Finally, the parameter α in (17) denotes how each detection $\tilde{\mathcal{S}}_t$ result should be trusted given the particle $\mathcal{S}_{t-1}$, as follows:

$$\alpha = \max_{\tilde{\mathcal{S}}_t} \exp\{-K_\alpha (\tilde{\mathcal{S}}_t - \mathcal{S}_{t-1})^\top \Sigma_{\mathcal{S}}^{-1} (\tilde{\mathcal{S}}_t - \mathcal{S}_{t-1})\}, \tag{20}$$



(a) (b)

Figure 4. Graph (a) shows the mixture of Gaussians forming the proposal distribution. Notice that the detections parameters closer to the particle parameters at $t-1$ (roughly at the center between the two solid thicker curves) have higher weights in the proposal distribution. Graph (b) displays the application of the learned linear transformations for the systole (contraction) and diastole (relaxation) cycles.

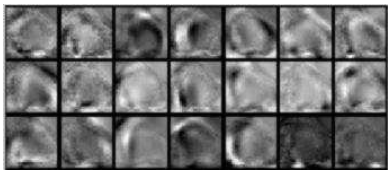


Figure 5. Subset of learned features for classifier at $\sigma = 4$.

where K_α is a parameter of the model to be determined through cross-validation.

3. Training Parameters

The training of the observation model involved the use of a set of 400 ultrasound images (from 12 sequences) of left ventricles annotated by experts. For the rigid classifier, we build an image scale space $L(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) * I(\mathbf{x})$, where $G(\mathbf{x}, \sigma)$ is the Gaussian kernel, $*$ is the convolution operator, $I(\mathbf{x})$ is the input image, σ is the image scale parameter, and $\mathbf{x}$ is the image coordinate. We train three separate classifiers (13); one for each scale $\sigma = \{4, 8, 16\}$. The positive and negative training sets are defined based on a scale-dependent margin m_σ that increases by a factor of two after each octave. Positives for $L(\mathbf{x}, \sigma)$ are randomly generated *inside* the range $[\theta - m_\sigma/2, \theta + m_\sigma/2]$, and negatives are randomly generated *outside* the range $[\theta - m_\sigma, \theta + m_\sigma]$, where θ is the parameter vector representing the rigid transformation of the LV annotation. Notice in Fig. 5 that the type of features automatically learned from the this training process resembles wavelets. The non-rigid regressor is trained at $\sigma = 4$, where each training sample is a line of 41 pixels of length extracted perpendicularly from the LV contour points (see Fig. 3) and the label to learn is the pixel index in $\{1, ..., 41\}$ that is closest to the LV contour. Running a cross-validation procedure with 200 images for training and 200 images for validation, the following pa-

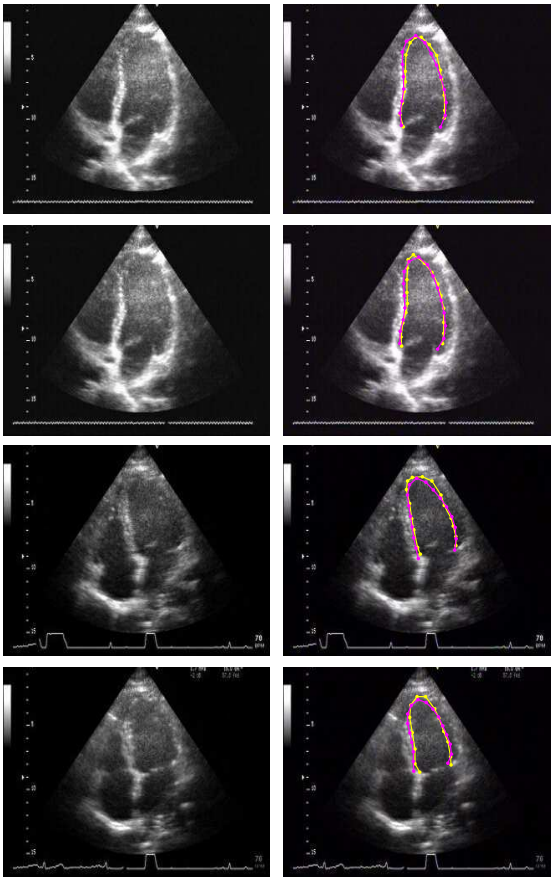


Figure 6. Example of the first (top two rows) and second (bottom two rows) test sequences. The yellow, solid line displays the manual annotation, while the magenta dashed line shows the results from our system.

rameters were estimated: 1) number of nodes per layer of regressor network: 41 (visible), 50 (hidden 1), 50 (hidden 2), 250 (hidden 3), 1 (output); 2) number of nodes per layer of the classifier networks: 16, 49, 196 (visible layers at $\sigma = \{16, 8, 4\}$, respectively), 50 (hidden 1), 50 (hidden 2), 100 (hidden 3), 3 (output); 3) the prior distribution $p(\theta|I, \mathcal{D})$ used in (12): Gaussian with mean and covariance computed from the training parameters of the rigid transform; 4) $E_{\text{coarse}} = 10^3$ and $E_{\text{fine}} = 10$. For the SIR algorithm, the following parameters are determined through cross validation: 1) number of particles: 100; 2) $K_\alpha = 0.1$ in (20); and 3) the rate of effective particles for re-sampling is $K_{Neff} = 0.1$.

4. Experiments

We use the following three metrics to compare the output of the detector with the reference contours, namely [11]: the Hausdorff distance, the average distance, and the Hammoude distance [8]. Assuming that $\mathcal{X} = \{\mathbf{x}_i\}_{i=1..N}$ is the automatically estimated contour from a system and $\mathcal{S} = \{\mathbf{s}_i\}_{i=1..N}$ is the manual segmentation, we first define the smallest point to curve distance as $d(\mathbf{x}_i, \mathcal{S}) = \min_j \|\mathbf{s}_j - \mathbf{x}_i\|_2$. The average distance between two curves is defined by:

$$d_{avg}(\mathcal{X}, \mathcal{S}) = \frac{1}{N} \sum_i d(\mathbf{x}_i, \mathcal{S}), \tag{21}$$

and the Hausdorff distance is defined as follows [9]:

$$d_{max}(\mathcal{X}, \mathcal{S}) = \max \left(\max_i \{d(\mathbf{x}_i, \mathcal{S})\}, \max_j \{d(\mathbf{s}_j, \mathcal{X})\} \right). \tag{22}$$

Finally, the Hammoude distance [8] is defined by:

$$d_H(\mathcal{X}, \mathcal{S}) = \frac{\#((R_\mathcal{X} \cup R_\mathcal{Y}) - (R_\mathcal{X} \cap R_\mathcal{Y}))}{\#(R_\mathcal{X} \cup R_\mathcal{Y})}, \tag{23}$$

where $R_\mathcal{X}$ represents the image region delimited by the contour $\mathcal{X}$, and similarly for $R_\mathcal{S}$.

The performance of the tracker was measured by comparing the contour estimates with reference contours provided by a cardiologist of Hospital Fernando Fonseca, Portugal. Note that these images were not included in the 400 images of the training set. Two sequences from two different subjects were used (see Fig. 6), where the cardiologist provided manual segmentations for 40 images from each sequence. Seq. 1 has in total 490 frames containing 26 cardiac cycles, while seq. 2 has 470 frames with 19 cycles. For the comparison, we present the results obtained with state-of-the-art trackers for the left ventricle recently proposed by Comaniciu et al. [4, 7], by Nascimento [11] and by a previous version of our method [3] that contains only the observation model (*i.e.*, we did not implement a dynamic model, only the static segmentation was present in the algorithm), applied on the same data. Table 1 shows the comparisons for the two sequences with the results of our approach in rows "D.Full" (original search for the detection process), "D.GradDes" (gradient descent search), and "D.Newton" (Newton step search). The rows "S.Full", "MMDA" and "COM" show the respective results by the previous version of our model with only the static segmentation [3], Nascimento [11] and Comaniciu [4, 7]. In this table the best value for each measure and sequence is highlighted. We computed the statistical significance of these results with the student's t-test. This is a reasonable choice because the error measures contain the same number of points, the variances are similar independently of the methodology, and the error distributions are roughly Gaussian. We obtain a p-value < 0.05 in the following cases: 1) using Hamm.(23) w.r.t. COM [4, 7] and MMDA [11] in both sequences; 2) using aver. (21) w.r.t. COM and MMDA in seq. 1 and MMDA in seq. 2; and 3) using Hausd. (22) w.r.t. MMDA in seq. 2. Recall that the p-value indicates whether the averages from the 2-sample sets differ significantly (note that the null hypothesis is that the means of the two normally distributions are equal).

The run-time complexity is dominated by the LV detection, using the deep neural networks, that is run during the

Table 1. Comparisons in the sequences (Fig. 6). Each cell shows the mean value and the standard deviation in parentheses.

Sequence One			
Approach	Hamm. (23)	Aver. (21)	Hausd. (22)
D.Full	0.17(0.04)	**3.2(0.8)**	19.9(1.9)
D.GradDes	0.17(0.04)	3.3(0.8)	**19.1(2.2)**
D.Newton	**0.16(0.04)**	3.3(0.8)	19.7(1.6)
S.Full[3]	0.18(0.06)	3.2(0.8)	20.0(2.6)
MMDA[11]	0.24(0.03)	4.8(0.9)	22.4(2.1)
COM[4, 7]	0.20(0.03)	3.8(0.5)	20.4(1.1)
Sequence Two			
Approach	Hamm. (23)	Aver. (21)	Hausd. (22)
D.Full	0.16(0.02)	3.0(0.5)	19.6(0.7)
D.GradDes	**0.15(0.03)**	**2.9(0.5)**	19.4(1.4)
D.Newton	0.21(0.13)	4.1(3.9)	20.5(5.3)
S.Full[3]	0.17(0.02)	3.0(0.5)	19.8(1.1)
MMDA[11]	0.24(0.03)	4.8(0.7)	20.2(1.4)
COM[4, 7]	0.18(0.03)	3.3(0.5)	**17.2(1.3)**

construction of the proposal distribution, which have the following numbers of floating point multiplications for the classifier at $\sigma = 16$ is $O(8 \times 10^6)$, at $\sigma = 8$ is $O(2.5 \times 10^7)$, at $\sigma = 4$ is $O(9.8 \times 10^7)$, and the regressor is $O(2.6 \times 10^7)$. Given these numbers, the "Full" search average complexity is $O(3.5 \times 10^{11})$, while the average complexity for "GradDes" is $O(2 \times 10^{10})$ and for "Newton" is $O(3 \times 10^{10})$. All other steps of the algorithm present negligible run-time complexity.

5. Conclusion and Future Work

In this paper we propose a new LV tracking algorithm from ultrasound data. Using Sequential Monte Carlo sampling algorithm (particle filtering), our main contributions are a new transition and observation models, and a new proposal distribution. The experiments show competitive tracking results, which are compared quantitatively to state-of-the-art methods. These results show empirical evidence that SMC sampling methods are useful in LV tracking, but the design of all models and distributions must be done carefully. More specifically, we see that the combination of different types of models in the proposal distribution provides accuracy and robustness to imaging conditions and drifting. Also, the principled combination of prior and learned models help alleviate the need of having an extensive and thorough training set.

Acknowledgements: We would like to thank G. Hinton and R. Salakhutdinov for making the deep neural network code available online. We also would like to thank Dr. António Freitas for providing the manual LV annotations.

References

[1] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE T. Sig. Pr.*, 50(2):174–188, 2002. 3

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. 3, 6

[3] G. Carneiro, J.C. Nascimento, and A. Freitas. Robust left ventricle segmentation from ultrasound data using deep neural networks and efficient search methods. In *ISBI*, 2010. 3, 7, 8

[4] D. Comaniciu, X. Zhou, and S. Krishnan. Robust real-time myocardial border tracking for echocardiography: An information fusion approach. *IEEE TMI*, 23(7):849–860, 2004. 2, 3, 7, 8

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV*, 1998. 1

[6] A. Doucet, N. de Freitas, N. Gordon, and A. Smith. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001. 1, 3

[7] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta. Databased-guided segmentation of anatomical structures with complex appearance. In *CVPR*, 2005. 2, 3, 7, 8

[8] A. Hammoude. *Computer-assited Endocardial Border Identification from a Sequence of Two-dimensional Echocardiographic Images*. PhD thesis, Univ. Washington, 1988. 7

[9] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using Hausdorff distance. *IEEE TPAMI*, 15(9):850–863, 1993. 7

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 4(1):321–331, 1987. 1

[11] J. C. Nascimento and J. S. Marques. Robust shape tracking with multiple models in ultrasound images. *IEEE TIP*, 17(3):392–406, 2008. 2, 3, 7, 8

[12] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004. 2, 3, 6

[13] N. Paragios. A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE TMI*, 21(9):773–776, 2003. 1

[14] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, (323):533–536, 1986. 5

[15] R. Salakhutdinov and G. Hinton. Learning a non-linear embedding by preserving class neighbourhood structure. In *AI and Statistics*, 2007. 3, 5

[16] J. Sénégas, T. Netsch, C. Cocosco, G. Lund, and A. Stork. Segmentation of medical images with a shape and motion model: a Bayesian perspective *MMBIA*, 2004. 2

[17] W. Sun, M. Cetin, R. Chan, V. Reddy, G. Holmvang, V. Chandar, and A. Willsky. Segmenting and tracking the left ventricle by learning the dynamics in cardiac images. *IPMI*, 2005. 2

[18] D. Terzopoulos and R. Szeliski. *Tracking with Kalman Snakes*. MIT Press, 1993. 2

[19] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu. 3-D ultrasound tracking of the left ventricle using one-step forward prediction and data fusion of collaborative trackers. In *CVPR*, 2008. 2

[20] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. *IEEE TMI*, 27(11):1668–1681, 2008. 2

[21] S. Zhou and D. Comaniciu. Shape regression machine. In *IPMI*, 2007. 2, 3