Combining Multiple Dynamic Models and Deep Learning Architectures for Tracking the Left Ventricle Endocardium in Ultrasound Data

Gustavo Carneiro*, Jacinto C. Nascimento, *Member, IEEE*.

Abstract

We present a new statistical pattern recognition approach for the problem of the left ventricle endocardium tracking in ultrasound data. The problem is formulated as a sequential importance resampling algorithm such that the expected segmentation of the current time step is estimated based on the appearance, shape, and motion models that take into account all previous and current images and previous segmentation contours produced by the method. The new appearance and shape models decouple the affine and non-rigid segmentations of the left ventricle in order to reduce the running time complexity. The proposed motion model combines the systole and diastole motion patterns and an observation distribution built by a deep neural network. The functionality of our approach is evaluated using a data set of diseased cases containing 16 sequences and another data set of normal cases comprising four sequences, where both sets present long axis views of the left ventricle. Using a training set comprising diseased and healthy cases, we show that our approach produces more accurate results than current state-of-the-art endocardium tracking methods in two test sequences from healthy subjects. Using three test sequences containing different types of cardiopathies, we show that our method correlates well with inter-user statistics produced by four cardiologists.

Index Terms

Left Ventricle Segmentation, Deep Belief Networks, Particle Filters, Dynamical Model, Discriminative Classifiers.

This work was supported by project the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds and by project PTDC/EEA-CRO/098550/2008. This work was also supported by project 'HEARTRACK' - PTDC/EEA-CRO/103462/2008. *This work was partially funded by EU Project IMASEG3D (PIIF-GA-2009-236173).

Gustavo Carneiro (corresponding author) is with the Australian Centre for Visual Technologies at the University of Adelaide, SA 5005, Australia. Email: gustavo.carneiro@adelaide.edu.au. **Phone:** +61-883136164, **Fax:** +61-883034366. Jacinto C. Nascimento is with the *Instituto de Sistemas e Robótica*, *Instituto Superior Técnico*, 1049-001 Lisboa, Portugal.

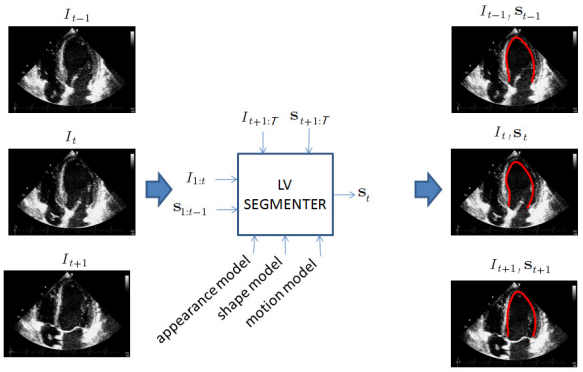


Fig. 1. General LV segmentation and tracking method based on appearance, shape and motion models, that uses current, past and future images I and contours $\mathbf{s}$ (the subscript t denotes the time index, where it is assumed that the sequence has T frames).

I. Introduction

One of the most important steps in estimating the health of the heart is the tracking and segmentation of the left ventricular (LV) endocardial border from an ultrasound sequence, which is used for measuring the ejection fraction and to assess the regional wall motion [1]. A fully automatic LV segmentation system has the potential to streamline the clinical work-flow and reduce the inter-user variability. Current state-of-the-art automatic LV segmentation and tracking methodologies are based on a function that takes into account the image to be segmented (and perhaps all previous and future images), the contours produced in the past (and possibly the future contours), and the appearance, shape and motion models (see Fig. 1). This is a dynamical model, where the state is represented by the LV contour, and the observation comprises the input sequence frames.

In general, the LV appearance can be characterized by a dark region, representing the blood pool inside the chamber, enclosed by the endocardium, myocardium, and epicardium, which are roughly depicted by a brighter region. The specific spatial texture and gray value distribution of each region vary substantially among different sequences (and even within each sequence) because of the following issues in the LV imaging using ultrasonic devices: fast motion during systole phase, low signal-to-noise ratio, edge dropout, the presence of shadows produced by the dense muscles, the specific properties and settings of the ultrasound machine, and the anisotropy of the ultrasonic image formation [2]. The large variation of the LV appearance forced researchers to impose constraints on the LV segmentation process using shape and motion models. A shape model typically constrains the mean shape of the LV and the main modes of variation based on a collection of manually annotated LV images, and some hand-designed priors. However, the characterization of all possible shape patterns and variations has proven to be a difficult task given the large variability of LV shapes due to the heart anatomy (especially regarding the

TABLE I

DIFFERENT SEGMENTATION AND TRACKING MODELS

Model (1)	$\mathbf{s}_t = f_{(1)}(\mathbf{s}_{t-1}, I_t;$ appearance,shape$)$	[3]–[6]
Model (2)	$\mathbf{s}_t = f_{(2)}(\mathbf{s}_{t-1}, I_t;$ appearance,shape,motion$)$	[7]–[11]
Model (3)	$\mathbf{s}_t = f_{(3)}(\mathbf{s}_{t-K:t-1}, I_{t-K:t};$ appearance,shape,motion$)$	[12]
Model (4)	$\mathbf{s}_t = f_{(4)}(\mathbf{s}_{t-1:t+1}, I_{t-1:t+1};$ appearance,shape,motion$)$	[13]–[16]
Model (5)	$\mathbf{s}_{1:T} = f_{(5)}(\mathbf{s}_{1:T}, I_{1:T};$ appearance,shape,motion$)$	[2,10,17,18]
Model (6)	$\mathbf{s}_t = f_{(6)}(\mathbf{s}_{1:t-1}, I_{1:t};$ appearance,shape,motion$)$	[19]–[23]

hearts presenting some kind of cardiopathy), and the inter-user variability of the LV manual annotation. For this reason, the motion model restricts the search space to a small region of the state space where the LV contour is expected to be found at each time instant of the sequence. Below, we review the foremost ideas in this field of research followed by our contributions.

A. Literature Review

Table I shows the different models used by several authors for the problem of LV segmentation and tracking. The functions can receive the following inputs: (previous, current, or future) contours $\mathbf{s}$, images I, and the models of appearance, shape, and motion. In general, the main differences lie in the use of a motion model, in the image and contour representations, and the types of appearance, shape and motion models. Model (1) represents the simplest tracker, where no motion model is used, and the segmentation result produced by the appearance and shape models is initialized by the contour produced in the previous frame [3]–[6]. Although simple and efficient, this method lacks robustness with respect to 1) the large contour motion and deformation between two frames in a sequence, and 2) the missegmentation produced by the shape and appearance models due to poor image conditions. The use of a motion model in order to predict the contour in the current frame, given the contour in the previous frame, gets around the first issue mentioned above (model (2) in Tab. I) [7]–[11]. It is interesting to note that the heart motion can be regarded as bi-modal (in the systole and diastole phases of the cardiac cycle), and the incorporation of this prior information in the motion model can be considered to be one more tool to fix the problem (1) mentioned above [24]–[26], if the method infers correctly the mode. Model (3) increases the robustness to large cardiac deformation by predicting $\mathbf{s}_t$ using the past K segmentation contours [12], but note that the time resolution of the training and test sequences must be similar for this approach to work. Another alternative studied, represented by the model (4), is to consider not only past contours, but also future contours [13]–[16]. Notice that models (1)-(4) only solve the first issue mentioned above (i.e., the large contour motion and deformation between frames).

In order to let the method recover from incorrect segmentations using the motion model, researchers

have proposed the use of the full segmentation history of the sequence, such that erroneous segmentations should appear as outliers having little weight in the prediction process. This idea is present in models (5) and (6), where model (5) [2,10,17,18] produces the segmentation for all the frames in parallel of a sequence in an off-line fashion, and model (6) [19]–[22] outputs the segmentation on-line as new frames are displayed. Model (5) appears to be the most robust approach of all, but the large parameter space generated by the joint appearance, shape and motion models represents a challenge (in terms of running time complexity) that is usually solved by restricting the search space with simple probability distributions containing few parameters, which is usually too restrictive to provide a reliable representation. Model (6) offers the best compromise between robustness and efficiency by looking only at the past segmentation contours using particle filtering methods [19]–[21] or a manifold of low intrinsic dimensionality representing clusters of contour sequences [22]. A variant of model (6) is proposed by Zhu et al. [23], who use not only the forward prediction (i.e., predict current contour based on previous segmentations), but also the backward prediction. One of the main advantages of models (5) and (6) compared to the other models reside not only in their ability to constrain the state search space, but also in their capability of incorporating non-linear motion dynamics, increasing the potential to represent complex heart motion patterns.

The appearance models proposed in the literature are often tightly integrated with a specific shape model, which can be characterized by one of the following methodologies: 1) active contours methods [27], 2) level set approaches [8,28]–[37], 3) deformable templates [12,26,38]–[41], 4) active shape and appearance models (ASM and AAM) [2,42]–[44], and 5) database-guided (DB-guided) segmentation [7,19,45]–[48]. Active contours [27] is the seminal approach that consists of an optimization problem that moves a parametrized curve towards image regions with strong edge information. Though successful at several tasks, its issues with respect to initialization and imaging conditions motivated the development of level-set methods [34] and deformable templates [12,26,38]–[41]. For the level-set, the LV contour is represented by the zero level-set of a distance function, while for the deformable model, the contour is represented by a parametrized curve. In general, level-set approaches reduce significantly the sensitivity to initial conditions, but have issues with imaging conditions [8,28]–[33,35]–[37,39], while deformable templates present robustness to imaging conditions, but have issues with initialization conditions [26,47]. Although level-sets and deformable templates have shown outstanding results in medical image analysis, they present a drawback, which is the prior knowledge defined in the optimization function about the LV border, shape, and texture distribution. This prior knowledge can assume the form of a hand-designed function or a probability distribution used to represent the priors above. As a result, the effectiveness of such approaches is limited by the validity of this prior knowledge, which is usually unlikely to capture all possible LV shape variations and nuances present in the ultrasound imaging of the LV [46].

The issue about these priors motivated the development of supervised learning approaches depending on

large annotated training sets and machine learning algorithms to estimate the parameters of the appearance, motion and shape models. Active shape and appearance models [2,42]–[44] estimate the parameters of a joint distribution representing the shape and appearance of the LV. The main issues with these models are the need of a large set of annotated training images, the condition that the initialization must be close enough to a local optimum, and the fact that the model assumes a Gaussian distribution of the shape and appearance information derived from the training samples. The use of a supervised learning model that does not assume Gaussian distributions was proposed in the form of a database-guided (DB-guided) segmentation [7,46], where the authors designed a discriminative learning model based on boosting techniques [49] to segment LV from ultrasound images. DB-guided approaches replace the dependence on an initial guess by an exhaustive search of the parameter space, which guarantees the reproducibility of the final result, but increases considerably the search complexity. Moreover, DB-guided approaches require a large number of training images (usually between hundreds and thousands) for a reliable estimation of the model parameters. Finally, DB-guided methods are in general sensitive to imaging conditions absent from the training set.

B. Contributions

We propose a new fully automatic LV endocardial border delineation and tracking methodology based on a sampling importance re-sampling (SIR) [50] formulation that is an instance of model (6) in Tab. I. In our new probabilistic model, the state space consists of the LV contour and the cardiac phase, where the current observation (image) distribution depends on the current values of both states, the current LV contour distribution depends on the previous values of contour and phase, and the current phase distribution depends on the previous phase value. Note that in this model, the filtering distribution of a cardiac phase and LV contour never commits to any specific heart dynamical regime at any time step because each element in the set of particles (used to estimate the filtering distribution) has its own LV contour and cardiac phase values. Another novelty of our approach is a new proposal distribution that combines a discriminative classifier proposal mechanism (based on a deep neural network) and a transition distribution based on the LV contour and the cardiac phase. The last contribution of our methodology is the observation model based on a deep neural network [51], which is a novel type of discriminative classifier that has the advantage of reducing the need of large and rich training sets given its better abstraction capabilities. The efficiency of our methodology is improved with an appearance model that decouples the affine and non-rigid contour detections, and with a gradient-based search approach [52]. All these contributions have been proposed by Carneiro and Nascimento [19], but in this paper, we provide a more comprehensive literature review, explanations, and experimental results, which not only compares the segmentation results with the state of the art, but also with inter-user statistics. Moreover, we show results on heart sequences containing several types of diseases.

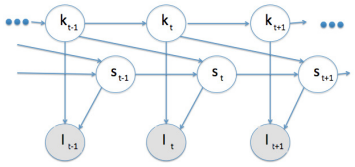


Fig. 2. Graphical model of our state-space model with the hidden nodes representing the cardiac phase k and contour $\mathbf{s}$, and the observed nodes denoting the image I.

II. Statistical Model

We assume a non-Gaussian state-space model, where the state sequence is a process represented by $\{k_t, \mathbf{s}_t | t \in \mathbb{N}\}$, where $k \in \{\text{systole}, \text{diastole}\}$ indicates the cardiac phase, and $\mathbf{s} \in \Re^{2S}$ denotes the explicit contour representation with S key points. In this model, the initial state distribution is represented by $p(k_0, \mathbf{s}_0)$ and the transition distribution takes into consideration the previous cardiac phase and contour representation with $p(k_t, \mathbf{s}_t | k_{t-1}, \mathbf{s}_{t-1})$. The observations consist of the images $\{I_t | t \in \mathbb{N}\}$, which are conditionally independent given the process $\{k_t, \mathbf{s}_t | t \in \mathbb{N}\}$, with marginal distribution $p(I_t | k_t, \mathbf{s}_t)$. Figure 2 shows the graphical model of our proposal. We assume the availability of a training set (for estimating the parameters of the models), which is represented by $\mathcal{D} = \{(I, \theta, \mathbf{s}, k)_j\}_{j=1}^{|\mathcal{D}|}$, where I denotes the training images containing the ultrasound imaging of different left ventricles, $\theta = [\mathbf{x}, \gamma, \sigma] \in \Re^5$ represents the parameters of an affine transformation (with position $\mathbf{x} \in \Re^2$, orientation $\gamma \in [-\pi, \pi]$, and scale $\sigma \in \Re^2$) that aligns the base and apical points of the LV annotation (see Fig.3) to specific locations in a canonical coordinate system to form the canonical LV annotation $\mathbf{s}$, k represents the cardiac phase, and $|\mathcal{D}|$ is the cardinality of the set $\mathcal{D}$. Finally, at each time step of a test sequence, our main goal is to estimate the current contour and its respective cardiac phase.

A. Observation Model

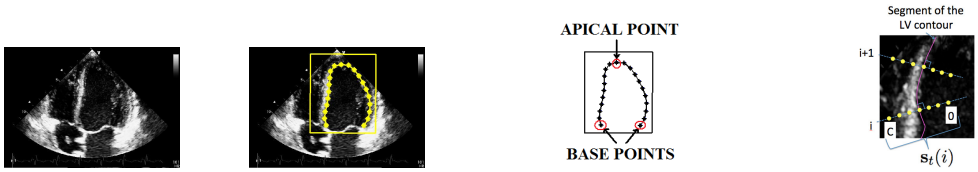
The observation model is the process of image formation given the cardiac phase and the contour, defined as:

$$p(I_t | k_t, \mathbf{s}_t) \propto p(k_t, \mathbf{s}_t | I_t) p(I_t), \tag{1}$$

where $p(I_t)$ =constant, and

$$p(k_t, \mathbf{s}_t | I_t) = \int_\theta p(k_t | \theta, I_t) p(\mathbf{s}_t | \theta, k_t, I_t) p(\theta | I_t) d\theta. \tag{2}$$

Equation (2) decouples the affine detection $p(k_t | \theta, I_t)$, the non-rigid segmentation $p(\mathbf{s}_t | \theta, k_t, I_t)$ and the prior distribution of the affine parameters $p(\theta | I_t)$. This means that in order to estimate the probability of the LV contour $\mathbf{s}_t$ and cardiac phase k_t, this model marginalizes out the affine transformation θ using



a) Training image b) Manual annotation c) Canonical LV contour d) Non-rigid Annotation

Fig. 3. Original training image (a) with the manual LV segmentation in yellow line and star markers (b) with the rectangular patch representing the canonical coordinate system for the segmentation markers. The image (c) shows the reference patch with the base and apical points highlighted and located at their canonical locations within the patch, and (d) displays the possible locations of contour points $\{\mathbf{s}_t(i)\}_{i=1}^{S}$, with $\mathbf{s}_t(i) \in [0, C]$.

its prior distribution, which is estimated from the training set $\mathcal{D}$. This decoupled formulation [19,46,48] is important in order to reduce the number of joint parameters for the learning and inference processes, which allows for more efficient search strategies during the LV segmentation and for smaller training sets.

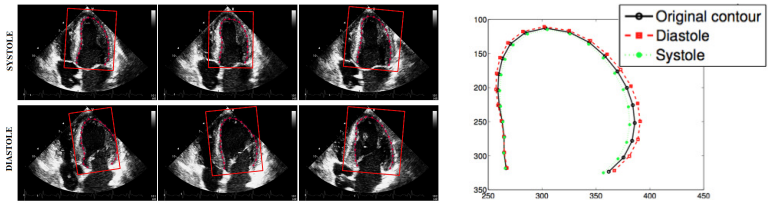
The first term in (2) represents the affine detection (computed by a discriminative classifier), which receives as input an image patch extracted from image I_t using θ and outputs the probability of $k_t \in \{\text{systole, diastole}\}$. Fig. 3-(b,c) displays how the image patch is extracted given the aligned base and apical points. The second term in (2), representing the non-rigid detection, is defined as follows:

$$p(\mathbf{s}_t|\theta, k_t, I_t) = \prod_{i=1}^{S} p(\mathbf{s}_t(i)|\theta, k_t, I_t), \tag{3}$$

where $p(\mathbf{s}_t(i)|\theta, k_t, I_t)$ represents the probability that the i^{th} contour key-point $\mathbf{s}_t(i) \in \Re^2$ is located at the LV contour. Fig. 3-(d) shows that for each $i \in \{1, ..., S\}$ the possible $\mathbf{s}_t(i)$ is confined to points lying on a line orthogonal to the LV contour, varying from 0 to C. Notice in (3) that we make the strong assumption that the LV contour keypoints are independent, but later in the paper we show how to alleviate such assumption with the use of a shape model (see Sec. III-A). Finally, the third term in (2) is defined as $p(\theta|I_t) = g(\theta|\bar{\theta}, \Sigma_\theta)$, where $\bar{\theta}$ and Σ_θ are the mean and covariance values of the training set values for θ, and $g(.)$ denotes the multivariate Gaussian density function.

B. Transition Model

The new transition model proposed in this paper estimates the distribution of current values for the LV contour and cardiac phase based on their previous values (see examples of contours and cardiac phases in Fig. 4-(b)). Our main assumption is that the current cardiac phase only depends on previous cardiac phase, while the current LV contour depends on the previous LV contour and the previous cardiac phase



(a) Annotations (b) Affine transformation

Fig. 4. Examples of systole and diastole annotations (a), and the linear transformations for each cardiac phase using as reference the mean contour (in black, labeled 'original contour'), which is defined by $\bar{\mathbf{s}} = \frac{1}{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{D}|}\mathbf{s}_j$ (6), as shown in (b).

(see Fig. 2). Based on this assumption, we define the transition distribution as follows:

$$p(k_t, \mathbf{s}_t|k_{t-1}, \mathbf{s}_{t-1}) = p(k_t|k_{t-1})p(\mathbf{s}_t|k_{t-1}, \mathbf{s}_{t-1}), \tag{4}$$

with $p(k_t|k_{t-1})$ represented by a 2×2 table (denoting the probabilities of staying at the same phase or switching phases at each time step), and

$$p(\mathbf{s}_t|k_{t-1}, \mathbf{s}_{t-1}) = g(\mathbf{s}_t|f(\mathbf{s}_{t-1}, \mathbf{M}(k_{t-1})), \Sigma_{\mathbf{s}}), \tag{5}$$

where $g(.)$ represents a multivariate Gaussian density function with mean $f(\mathbf{s}_{t-1}, \mathbf{M}(k_{t-1}))$ and covariance $\Sigma_{\mathbf{s}}$. The function $f : \Re^{2S} \times \Re^{3\times3} \rightarrow \Re^{2S}$ produces the affine transformation of the LV contour points $\mathbf{s}_{t-1}$ given by the matrix $\mathbf{M}(k_{t-1})$. The affine matrix $\mathbf{M}$(systole) (equivalently for $\mathbf{M}$(diastole)) is learned from the training data with the following optimization function (Fig. 4-(b)):

$$\mathbf{M}(\text{systole}) = \arg\min_{\mathbf{M}} \|\bar{\mathbf{s}}_{\text{systole}} - f(\bar{\mathbf{s}}_{\text{diastole}}, \mathbf{M})\|^2, \tag{6}$$

where $\mathbf{M}$ is an affine transformation matrix, $\bar{\mathbf{s}}_{\text{systole}} = \frac{1}{\sum_{j=1}^{|\mathcal{D}|}\delta(k_j-\text{systole})}\sum_{j=1}^{|\mathcal{D}|}\mathbf{s}_j\delta(k_j - \text{systole})$ (i.e., this is the mean contour of the training samples where $k_j = $ systole), and equivalently for $\bar{\mathbf{s}}_{\text{diastole}}$. The matrix $\Sigma_{\mathbf{s}} = \frac{1}{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{D}|}(\mathbf{s}_j - \bar{\mathbf{s}})(\mathbf{s}_j - \bar{\mathbf{s}})^T$ in (5) denotes the covariance of the annotations $\mathbf{s}$ learned from the training data, where $\bar{\mathbf{s}} = \frac{1}{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{D}|}\mathbf{s}_j$.

C. Filtering Distribution

Using the transition and observation models defined above, and assuming that the state and observation vectors up to time t are represented by $k_{0:t} \triangleq \{k_0, ..., k_t\}$ (similarly for $\mathbf{s}_{0:t}$ ad $I_{1:t}$), the filtering distribution is defined by:

$$p(k_t, \mathbf{s}_t|I_{1:t}) = \frac{p(I_t|k_t, \mathbf{s}_t)\sum_{k_{t-1}}\int p(k_t, \mathbf{s}_t|k_{t-1}, \mathbf{s}_{t-1})p(k_{t-1}, \mathbf{s}_{t-1}|I_{1:t-1})d\mathbf{s}_{t-1}}{\sum_{k_t}\int p(I_t|k_t, \mathbf{s}_t)p(k_t, \mathbf{s}_t|I_{1:t-1})d\mathbf{s}_t}, \tag{7}$$

where $p(k_t, \mathbf{s}_t|I_{1:t-1}) = \sum_{k_t}\int p(k_t, \mathbf{s}_t|k_{t-1}, \mathbf{s}_{t-1})p(k_{t-1}, \mathbf{s}_{t-1}|I_{1:t-1})d\mathbf{s}_{t-1}$. Notice that (7) computes the distribution of cardiac phases and LV contours, which means that at each time step we never commit

to any of the cardiac phases or contour values. This aspect of our model increases the robustness to the issues mentioned in Sec. I-A, related to the large contour motion and deformation between consecutive frames and the missegmentation produced by the observation model.

D. Particle Filtering

The posterior $p(k_t, \mathbf{s}_t|I_{1:t})$ in (7) is approximated with a finite set of P particles, as follows:

$$p(k_t, \mathbf{s}_t|I_{1:t}) \approx \sum_{l=1}^{P} w_t^{(l)} \delta(k_t - k_t^{(l)}) \delta(\mathbf{s}_t - \mathbf{s}_t^{(l)}), \tag{8}$$

where $\delta(.)$ is the delta function. The particles $\{k_t^{(l)}, \mathbf{s}_t^{(l)}\}_{l=1}^{P}$ are sampled from a proposal distribution with $(k_t^{(l)}, \mathbf{s}_t^{(l)}) \sim q(k_t, \mathbf{s}_t|k_{0:t-1}^{(l)}, \mathbf{s}_{0:t-1}^{(l)}, I_{1:t})$ in (10), where each particle is weighted by

$$\tilde{w}_t^{(l)} = w_{t-1}^{(l)} \frac{p(I_t|k_t^{(l)}\mathbf{s}_t^{(l)}) p(k_t^{(l)}, \mathbf{s}_t^{(l)}|k_{t-1}^{(l)}\mathbf{s}_{t-1}^{(l)})}{q(k_t^{(l)}, \mathbf{s}_t^{(l)}|k_{0:t-1}^{(l)}\mathbf{s}_{0:t-1}^{(l)}, I_{1:t})}, \tag{9}$$

where $\tilde{w}_t^{(l)}$ and $w_t^{(l)}$ represent the un-normalized and normalized weights, respectively.

E. Deep Particle Filter

The proposal distribution is another important contribution of this work. It consists of multiple dynamic models, where a separate model is built for each particle at each time instant t using the detections of the discriminative classifier based on a deep neural network and the transition model (4) applied to the particle from time $t-1$. The proposal distribution at time t is defined with a mixture of Gaussians, as follows [53]:

$$q(k_t, \mathbf{s}_t|k_{0:t-1}^{(l)}, \mathbf{s}_{0:t-1}^{(l)}, I_{1:t}) = q(k_t, \mathbf{s}_t|k_{t-1}^{(l)}\mathbf{s}_{t-1}^{(l)}, I_t) = (1-\alpha) p(k_t, \mathbf{s}_t|k_{t-1}^{(l)}, \mathbf{s}_{t-1}^{(l)}) + \alpha q_D(k_t, \mathbf{s}_t|I_t), \tag{10}$$

with $p(k_t, \mathbf{s}_t|k_{t-1}^{(l)}, \mathbf{s}_{t-1}^{(l)})$ defined in (4),

$$q_D(k_t, \mathbf{s}_t|I_t) = \sum_h p(\tilde{k}_t^{(h)}, \tilde{\mathbf{s}}_t^{(h)}|I_t) g(\mathbf{s}_t|\tilde{\mathbf{s}}_t^{(h)}, \Sigma_{\mathbf{s}}) p(k_t|\tilde{k}_t^{(h)}), \tag{11}$$

where $p(\tilde{k}_t^{(h)}, \tilde{\mathbf{s}}_t^{(h)}|I_t)$ is defined in (2), $h \in \{1, ..., H\}$ is the index to the hypotheses representing the local maxima of the observation distribution (2), $g(.)$ is the multivariate Gaussian density function with mean $\tilde{\mathbf{s}}_t^{(h)}$ and covariance $\Sigma_{\mathbf{s}}$ defined in (6), and $p(k_t|\tilde{k}_t^{(h)})$ is the 2×2 matrix describing the transition between cardiac phases defined in (4). The parameter $\alpha \in [0, 1]$ in (10) is used to weight the contribution of the observation and transition models. Note that $\alpha = 0$ represents a distribution that takes into account only the transition model, while $\alpha = 1$ denotes a proposal distribution built based on the observation model only. Since α represents a free parameter, we provide a study in the experiments that shows how the performance is altered for different values of α. Finally, assuming $p(k_0, \mathbf{s}_0)$ uniform, then at time step $t = 1$, we have:

$$q(k_1, \mathbf{s}_1|k_0^{(l)}, \mathbf{s}_0^{(l)}, I_1) = q_D(k_1, \mathbf{s}_1|I_1). \tag{12}$$

Note that the main advantage of this proposal distribution is that when the motion model fails, the observation model has a chance to recover the problem. For the LV tracking this is important because it is hard to obtain a faithful model of the LV motion. Nevertheless, the presence of the transition model is still quite important to deal with the detection and segmentation failures of the observation model.

F. Segmentation Algorithm

Using the approximation in (8) for the filtering distribution, we estimate the values of the state variables at each time step t for a test sequence frame at time step t as follows:

$$\mathbf{k}_t^* = \arg \max_{k \in \{systole, diastole\}} E_{p(k_t, \mathbf{s}_t | I_{1:t})}[k], \tag{13}$$

where $E_{p(k_t,\mathbf{s}_t|I_{1:t})}[k] \approx \sum_{l=1}^{P} w_t^{(l)} \delta(k - k_t^{(l)})$, and

$$\mathbf{s}_t^* = E_{p(k_t,\mathbf{s}_t|I_{1:t})}[\mathbf{s}_t | k_t^*] \approx \frac{1}{\sum_{l=1}^{P} w_t^{(l)} \delta(k_t - k_t^*)} \sum_{l=1}^{P} w_t^{(l)} \mathbf{s}_t^{(l)} \delta(k_t - k_t^*). \tag{14}$$

The full segmentation algorithm is summarized in Alg. 1.

Algorithm 1 SIR Algorithm.

for $t = 1$ to T **do**
  sample $\{k_t^{(l)}, \mathbf{s}_t^{(l)}\}_{l=1}^{P}$ using (10);
  Update sample weights $\{\tilde{w}_t^{(l)}\}_{l=1}^{P}$ with (9) ;
  Normalize sample weights: $w_t^{(l)} = \frac{\tilde{w}_t^{(l)}}{\sum_l \tilde{w}_t^{(l)}}$ for $l \in \{1, ..., P\}$;
  Compute effective number of particles $N_{eff} = 1/\sum_l (w_t^{(l)})^2$;
  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 $w_t^{(l)} = 1/P$ for $l \in \{1, ..., P\}$
  end if
  Compute LV segmentation for I_t with k_t^* (13) and $\mathbf{s}_t^*$ (14).
end for

III. Training and Inference of the Observation Model

In this section, we describe the training and segmentation processes for the affine and non-rigid classifiers (2) present in the observation model described in (1). These classifiers are essentially artificial neural networks (ANN) with a relatively large number of hidden layers (and nodes), which is generally known as deep neural networks (DNN). The larger number of hidden layers in a DNN, compared to the original ANN, is usually associated with better representation capabilities [54], but the parameter

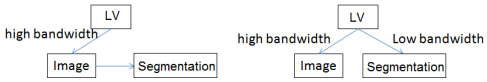


Fig. 5. Comparison between current machine learning approaches (left diagram) and deep learning methodologies (right) [59]. Bandwidth in this context means the number of possible ways an instance of the variable can be generated. For example, given the left ventricle of a heart, there is a large number of ways that it can be imaged using an ultrasound device, but there is only a small number of possible LV contours (in fact, there should be only one possible LV contour, representing the golden standard).

estimation with back-propagation from a random initialization [55] is usually ineffective due to slow convergence and failure to reach good local optima. Hinton and colleagues have recently proposed a two-stage learning methodology to train a DNN [56]–[58], where the first step consists of an unsupervised generative learning that builds incrementally an auto-encoder (as new hidden layers are added to the network), and the second step comprises a supervised discriminative learning that uses the parameters learned for the auto-encoder as an initialization for the back-propagation algorithm [55]. Fig. 5 motivates the aforementioned new learning methodology, where the left diagram displays the current supervised learning paradigm, where it is assumed that the LV segmentation to an image is independent of the original cause (i.e., the imaging of the LV of the heart) given the image. Therefore, current learning models (e.g., boosting) need to collect a large training set in order to confidently learn the parameters of the statistical model, representing the probability of segmentation given image. On the other hand, the right diagram shows the deep learning approach, where an unsupervised generative model learns the LV image generation process, and then a discriminative model is trained based on this generative model [59]. Hence, leveraging the generative model in the learning of the discriminative model is the key that makes deep learning less dependent on large and rich training sets.

A. Training Procedure

For the affine classifier in the first term of (2), we follow the multi-scale implementation of Carneiro et al. [45] and build an image scale space $L(\mathbf{x}, \sigma)$ produced from the convolution of the Gaussian kernel with the input image $I(\mathbf{x})$, as follows:

$$L(\mathbf{x}, \sigma) = \left(\frac{1}{2\pi\sigma^2} e^{-\frac{\mathbf{x}^2}{2\sigma^2}} \right) * I(\mathbf{x}), \tag{15}$$

where σ is the scale parameter, $\mathbf{x}$ is the image coordinate, and $*$ is the convolution operator. Assuming that our multi-scale implementation uses a set of image scales represented by $\{\sigma_1, ..., \sigma_Q\}$, we train Q affine classifiers. In order to train each affine classifier, it is necessary to build a set of positive and negative image samples. An image sample is built using the extraction function $u(I, \sigma_q, \theta)$ that takes the image I, the scale σ_q, and the affine parameter θ to produce a contrast normalized [60] image patch of

size $\kappa_q \times \kappa_q$, where κ_q represents a vector indexed by $q \in \{1, ..., Q\}$ with the sizes of the image patch at each scale. The contrast normalization makes our approach more robust to brightness variations by taking each pixel of the $\kappa_q \times \kappa_q$ patch, subtracting it by the mean gray value of the patch and dividing it by the standard deviation of the patch values. The sets of positives $\mathcal{P}(k, q, j)$ and negatives $\mathcal{N}(q, j)$, from each training image I_j at each scale q and $k \in \{\text{systole,diastole}\}$, are formed by sampling the distribution over the training affine parameters $\{\theta_j\}_{j=1}^{|\mathcal{D}|}$, which can be respectively defined as

$$\begin{array}{c} \mathcal{P}(k, q, j) = \{\theta | \theta \sim \mathcal{U}(r(\Theta)), d(\theta, \theta_j) \prec \mathbf{m}_q, k_j = k\} \\ \mathcal{N}(q, j) = \{\theta | \theta \sim \mathcal{U}(r(\Theta)), d(\theta, \theta_j) \succ 2\mathbf{m}_q\} \end{array}, \tag{16}$$

where $\mathcal{U}(r(\Theta))$ represents the uniform distribution such that the range of possible values for θ is denoted by $r(\Theta) = [\max_{row}(\Theta) - \min_{row}(\Theta)] \in \Re^5$ (with $\Theta = [\theta_1..\theta_{|\mathcal{D}|}] \in \Re^{5\times|\mathcal{D}|}$ being a matrix with the training vectors $\theta_j \in \mathcal{D}$ in its columns and the functions $\max_{row}(\Theta) \in \Re^5$ and $\min_{row}(\Theta) \in \Re^5$ representing, respectively, the maximum and minimum row elements of the matrix Θ), $\prec$ and $\succ$ denote the element-wise "less than" and "greater than" vector operators, respectively,

$$\mathbf{m}_q = r(\Theta) \times \sigma_q \times v_{\mathcal{U}} \tag{17}$$

represents the margin between positive and negative cases with $v_{\mathcal{U}}$ defined as a constant, and

$$d(\theta, \theta_j) = |\theta - \theta_j| \in \Re^5 \tag{18}$$

denotes the dissimilarity function in (16), where $|.|$ returns the absolute value of the vector $\theta - \theta_j$. Note that according to the generation of positive and negative sets in (16)-(18) one can notice a margin between these two sets, where no samples are generated for training. The existence of this margin facilitates the training process by avoiding similar examples with opposite labels, which could generate over-trained classifiers. The affine DNN at scale σ_q is trained by first stacking several hidden layers to reconstruct the input patches in $\mathcal{P}$ and $\mathcal{N}$ (unsupervised training). Then three nodes are added to the top layer of the DNN, which indicate $p(k = \text{systole}|\theta, I)$, $p(k = \text{diastole}|\theta, I)$, and $1 - \sum_k p(k|\theta, I)$ (note that this last term represents the probability that the image patch formed by θ, I_j, and σ_q does not contain an LV). The discriminative training finds the following maximum posterior at each scale σ_q:

$$\gamma_{\text{MAP}}(\sigma_q) = \arg\max_{\gamma} \prod_{j=1}^{|\mathcal{D}|} \left[\prod_k \prod_{\theta \in \mathcal{P}(k,q,j)} p(k|\theta, I_j, \gamma) \right] \left[\prod_{\theta \in \mathcal{N}(q,j)} (1 - \sum_k p(k|\theta, I_j, \gamma)) \right], \tag{19}$$

where γ denotes the DNN weights.

The training of the non-rigid classifier (second term in (2)) is based on the following optimization function:

$$\psi_{\text{MAP}}(\sigma_q) = \arg\max_{\psi} \prod_{j=1}^{|\mathcal{D}|} \prod_{i=1}^{S} \prod_k \prod_{\theta \in \mathcal{P}(k,q,j)} p(\mathbf{s}_j(i)|k, \theta, I_j, \psi), \tag{20}$$

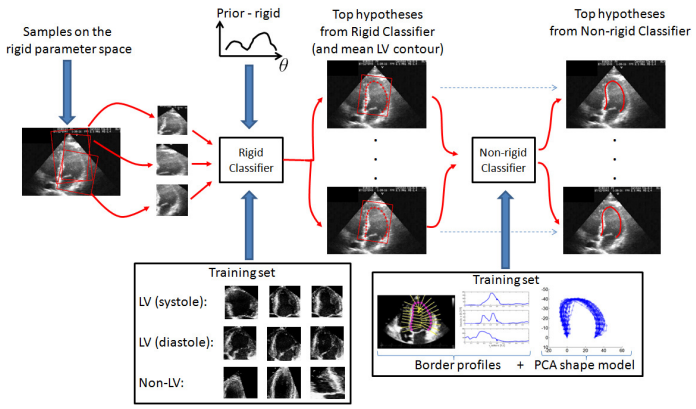


Fig. 6. Block diagram of the inference procedure of the observation model.

where ψ represents the DNN weights, and $\mathbf{s}_j(i) \in [0, C]$ with C being the length of the normal to the LV contour represented by $\mathbf{s}_j$ at the point $i \in \{1, ..., S\}$ (see Fig. 3-(d) and Fig. 6). In practice, $p(\mathbf{s}_j(i)|k, \theta, I_j)$ is a regressor that receives as input a profile of the gray values taken from an orthogonal line from each key point of the canonical LV contour and returns a value between 0 and C, representing the most likely location of the LV in that orthogonal line. Therefore, the training is realized by taking the profiles from the training images using only the positive set $\mathcal{P}(k, q, j)$ in (16). Furthermore, since the non-rigid classifier is run only at the finest scale Q, the training is run only at σ_Q.

Finally, we build a shape model based on principal component analysis (PCA) [61,62] that is used to project the final result from the non-rigid classifier. The goal of this step is to alleviate the strong independence assumption made in (3), which in practice suppresses the noisy results from the non-rigid classifier. Assuming that $\mathbf{X} = [\mathbf{s}_1, ..., \mathbf{s}_{|\mathcal{D}|}] \in \Re^{2S \times |\mathcal{D}|}$ is a matrix that contains in its columns all the annotations in the training set $\mathcal{D}$, where the mean shape $\bar{\mathbf{s}} = \frac{1}{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{D}|} \mathbf{s}_j$ has been subtracted from each column, then we can decompose $\mathbf{X}$ using eigenvalue decomposition, as follows: $\mathbf{X}\mathbf{X}^\top = \mathbf{W}\mathbf{\Sigma}\mathbf{W}^\top$. Given a new annotation produced by the non-rigid classifier, say $\mathbf{s}$, we obtain its new value by first projecting it onto the PCA space $\mathbf{y}^\top = (\mathbf{s}^\top - \bar{\mathbf{s}}^\top)\widetilde{\mathbf{W}}\widetilde{\mathbf{\Sigma}}^{-0.5}$, where $\widetilde{\mathbf{W}}$ contains the first E eigenvectors, and $\widetilde{\Sigma}$ is a diagonal matrix containing the first E eigenvalues in the diagonal. Then the final shape $\mathbf{s}^*$ is obtained by re-projecting $\mathbf{y}$ onto the original shape space and adding back the mean shape, as in $\mathbf{s}^* = \left(\mathbf{y}^\top \widetilde{\mathbf{\Sigma}}^{0.5} \widetilde{\mathbf{W}}^\top\right)^\top + \bar{\mathbf{s}}$.

B. Inference Procedure

The first step of the inference procedure described in Alg. 2 consists of running the affine classifier at scale σ_1 on H_{coarse} samples drawn from $\mathcal{U}(r(\Theta))$ defined in (16). The samples $\theta^{(h)}$ ($h \in \{1, ..., H_{\text{coarse}}\}$)

for which $\sum_k p(k|\theta^{(h)}, I) > 0$ are used to build a Gaussian mixture model distribution, using the expectation maximization algorithm [63], as follows:

$$\text{Dist}(\theta; \sigma_1) = Z \sum_{h=1}^{H_{\text{fine}}} (\sum_k p(k|\theta^{(h)}, I)) g(\theta|\theta^{(h)}, \Sigma_{\mathbf{s}}), \tag{21}$$

where Z is a normalization constant, $H_{\text{fine}} << H_{\text{coarse}}$, $p(k|\theta^{(h)}, I)$ is the affine classifier, and $g(\theta|\theta^{(h)}, \Sigma_{\mathbf{s}})$ is the Gaussian density with mean $\theta^{(h)}$ and covariance $\Sigma_{\mathbf{s}}$ defined in (6). Then, we draw H_{fine} samples from $\text{Dist}(\theta; \sigma_1)$ to be used as initial guesses for the search procedure for the affine classifier trained at σ_2, resulting in at most H_{fine} samples (again, we only keep the samples for which $\sum_k p(k|\theta^{(h)}, I) > 0$), which are used to build $\text{Dist}(\theta; \sigma_2)$. This process of sampling/searching/building distribution is repeated for each scale $q \in \{2, ..., Q\}$, until we reach σ_Q. The final H_{fine} samples are used by the non-rigid classifier to produce the set of output contours $\{\mathbf{s}_t^{(h)}\}_{h=1}^{H_{\text{fine}}}$, which are projected onto the PCA space explained in Sec. III-A.

Algorithm 2 Inference Procedure of the Observation Model (see Fig. 6).

1: sample $\{\theta^{(h)}\}_{h=1}^{H_{\text{coarse}}} \sim \mathcal{U}(r(\Theta))$ defined in (16)
2: compute $\{\sum_k p(k|\theta^{(h)}, I)\}_{h=1}^{H_{\text{coarse}}}$ using DNN trained at σ_1
3: build $\text{Dist}(\theta; \sigma_1)$ using the set $\{\theta^{(h)} | h = 1..H_{\text{coarse}}, p(k|\theta^{(h)}, I_t) > 0\}$, as defined in (21)
4: **for** $q = 2$ to Q **do**
5: sample $\{\theta^{(h)}\}_{h=1}^{H_{\text{fine}}} \sim \text{Dist}(\theta; \sigma_{q-1})$
6: search using $\{\theta^{(h)}\}_{h=1}^{H_{\text{fine}}}$ as initial guesses for one of the search procedures (full or gradient descent) with DNN $\sum_k p(k|\theta, I_t)$ trained at σ_q (each initial guess $\theta^{(h)}$ generates a local optimum $\widetilde{\theta}^{(h)}$)
7: build $\text{Dist}(\theta; \sigma_q)$ using the set $\{\widetilde{\theta}^{(h)} | h = 1..H_{\text{fine}}, \sum_k p(k|\widetilde{\theta}^{(h)}, I_t) > 0\}$
8: **end for**
9: **for** $h = 1$ to H_{fine} **do**
10: run the non-rigid classifier $p(\mathbf{s}|k, \widetilde{\theta}^{(h)}, I)$ trained at σ_Q to generate the contour $\mathbf{s}^{(h)}$
11: $\mathbf{y}^\top = ((\mathbf{s}^{(h)})^\top - \bar{\mathbf{s}}^\top)\widetilde{\mathbf{W}}\widetilde{\mathbf{\Sigma}}^{-0.5}$
12: $\widetilde{\mathbf{s}}_t^{(h)} = \left(\mathbf{y}^\top \widetilde{\mathbf{\Sigma}}^{0.5} \widetilde{\mathbf{W}}^\top\right)^\top + \bar{\mathbf{s}}$.
13: **end for**

The search process that uses the DNN classifier is based on one of the following two different search approaches: *1) full search, and 2) gradient descent* [64]. For the full search, we run the DNN classifier at σ_q at all the 243 points in $\theta^{(h)} + [-\mathbf{m}_p, 0, +\mathbf{m}_p]$ for $h \in \{1, ..., H_{\text{fine}}\}$ and $\mathbf{m}_p$ in (22) (note that $243 = 3^5$, that is the five dimensional parameter space of the affine classifier with three points per dimension). Assuming that $p(\theta) = \sum_k p(k|\theta, I)$, the gradient descent algorithm [64] uses the Jacobian,

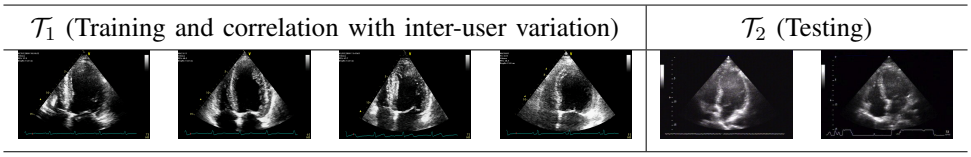


Fig. 7. Images of a subset of the sequences $\mathcal{T}_1$ and $\mathcal{T}_2$.

which is computed numerically using central difference, with the step size $\mathbf{m}_p$ (17), as follows:

$$\frac{\partial p(\theta)}{\partial \mathbf{p}_1} = \frac{p(\theta + \mathbf{v}_1) - p(\theta - \mathbf{v}_1)}{\mathbf{m}_p(1)} \tag{22}$$

where the subscript indicates the dimension (i.e., $\mathbf{p}_1$ denotes the first dimension of $\mathbf{p} \in \theta$), and $\mathbf{v}_1 = [\frac{\mathbf{m}_s(1)}{2}, 0, 0, 0, 0]^\top$. The first order partial derivatives for the other dimensions of θ are computed similarly to (22).

IV. Experimental Setup

In this section, we first examine how the experimental data sets have been set up, and then we explain the technical details involved in the training and segmentation procedures. We also introduce the quantitative comparisons to measure the performance of our approach.

A. Data sets and Manual Annotation Protocol

We extend the sets of annotated data introduced by Nascimento et al. [26]. In this paper, we use 20 sequences for training and testing (20 sequences from 20 subjects with no overlap), from which 16 present some kind of cardiopathy. According to the cardiologist's report[1], the following cardiopathies/abnormalities are considered:

§1. Dilation of the LV, which can be mild, moderate or severe;
§2. Presence of hypertrophy of the LV, which can be classified into mild, moderate or severe;
§3. Wall motion abnormalities, which can be global, affecting all the LV segments, or localized, affecting some of the LV segments;
§4. Dysfunction of the LV, which may be preserved, mild, depressed, or severe;
§5. Presence of valvular heart disease; and
§6. Presence of a pacemaker device.

We divide the data set into two sets: $\mathcal{T}_1$ and $\mathcal{T}_2$. The set $\mathcal{T}_1$ contains 16 sequences presenting some cardiopathy and two sequences from healthy subjects (each sequence is represented by a letter from A to R), while the set $\mathcal{T}_2$ comprises two sequences of healthy subjects (each sequence is represented by

[1]This was done in collaboration with Dr. Freitas from Hospital Fernando Fonseca who detailed each of the sequences

the letters A or B). We propose this data set division because our training set only contains images (and annotations) from $\mathcal{T}_1$, while our test sets consist of images from both sets, depending on the experiment (as explained below). We worked with four cardiologists and one technician. The technician annotated 496 images in the 18 sequences of the set $\mathcal{T}_1$, providing roughly an average of 27 contours per sequence. Four cardiologists annotated three sequences from $\mathcal{T}_1$, each one providing 52 annotations (an average of 17 images per sequence per cardiologist), and we denote these three sequences as $\mathcal{T}_{1,\{A,B,C\}}$. Additionally, one of the cardiologists annotated 80 images of the set $\mathcal{T}_{2,\{A,B\}}$ (40 per sequence). For the manual annotations, the cardiologists could use any number of points to delineate the LV, but they had to explicitly identify the base and apical points in order for us to determine the affine transformation between each annotation and the canonical location of such points in the reference patch (see Fig. 3).

Using the 496 images (18 sequences) of the annotated set $\mathcal{T}_1$, we assess the performance of our system on the sets $\mathcal{T}_{2,\{A,B\}}$ with 80 annotated images (i.e., 40 annotations for each of the two sequences), and compare the results against state-of-the-art methodologies. Furthermore, we measure how the results produced by our system correlates with inter-user variation on the three sequences $\mathcal{T}_{1,\{A,B,C\}}$ containing 52 annotated images by four cardiologists. Therefore, in total we test our system on five sequences, and compute results on 132 images.

B. Training and Segmentation Procedure Details

For training the affine classifiers at each scale $q \in \{1, ..., Q\}$, we produce 100 positive and 500 negative patches per training image to be inserted in the sets $\mathcal{P}$ and $\mathcal{N}$ in (16), respectively (Fig. 6 shows examples of positive and negative patches for one training image). This unbalance in the number of positive and negative samples can be explained by the much larger volume covered by the negative regions [65]. This initial training set is divided into 80% of $\mathcal{P}$ and $\mathcal{N}$ for training and 20% for validation, where this validation set is necessary to determine several parameters, as described below. The multi-scale implementation (15) used in the training and segmentation procedures used three scales $\sigma_q \in \{16, 8, 4\}$ for $q \in \{1, 2, 3\}$, where the images $L(.)$ are down-sampled by a factor of two after each octave. The values for these scales have been determined from the scale set $\{32, 16, 8, 4, 2\}$ using the validation set, from which we observe that $\sigma > 16$ (i.e., coarser scales) prevents the inference procedure to converge, and $\sigma < 4$ (i.e., finer scales) does not improve the accuracy of the method. The original patches used for training the affine classifier (see Fig. 6) have size 56×56 pixels, but the sizes used for scales $\{16, 8, 4\}$ are $\{4\times4, 7\times7, 14\times14\}$, respectively. For the uniform distributions in (16), the constant $v_{\mathcal{U}} = \frac{1}{400}$ in (17) has been empirically determined from the set $\{\frac{1}{100}, \frac{1}{200}, \frac{1}{400}, \frac{1}{800}\}$ based on the segmentation performance on the validation set. For the DNN, the validation set is used to determine the following parameters: a) number of nodes per hidden layer, and b) number of hidden layers. The number of nodes per hidden layer varies from 50 to 500 in intervals of 50. The number of hidden layers varies from 1 to 4 (we did not

TABLE II

LEARNED CONFIGURATION FOR THE DEEP BELIEF NETWORKS (TABLE FROM [66]).

Affine Classifier						
σ	Visible Layer	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Hidden Layer 4	Output Layer
4	196 (14×14 pix.)	100	100	200	200	3
8	49 (7×7 pix.)	50	100	-	-	3
16	16 (4×4 pix.)	100	50	-	-	3
Non-rigid Classifier						
σ	Visible Layer	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Hidden Layer 4	Output Layer
4	41	50	50	-	-	1

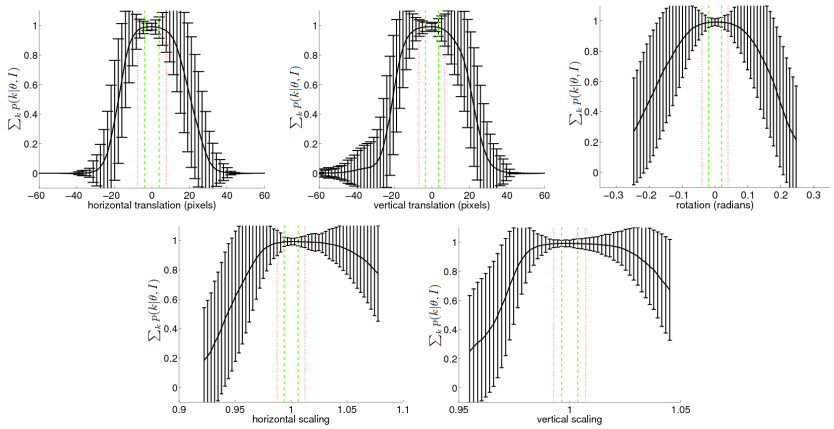


Fig. 8. Performance of the affine classifier trained at $\sigma = 4$, showing the mean and standard deviation of $\sum_k p(k|\theta, I)$ as a function of the variation of each one of the affine transformations (translation, rotation, and scaling) with respect to the manual annotation for all training images (i.e., only one transformation is varied while the others are kept fixed with respect to the manual annotation), where the vertical green dashed lines indicate the upper bound of the parameters used for the positive set and the vertical red dotted lines show the lower bound of the negative parameters (graphs from [66]).

notice any boost in performance with more than 4 layers). Using all annotated images from set $\mathcal{T}_1$, we achieved the configurations displayed in Table II. Figure 8 shows the performance of the affine classifier as a function of the affine transformations from the manual annotation. For the transition distribution, the validation set is used to estimate the 2×2 table representing $p(k_t|k_{t-1})$ in (4), where the following results have been obtained: if $k_t = k_{t-1}$, then $p(k_t|k_{t-1}) = 0.8$, and if $k_t \neq k_{t-1}$, then $p(k_t|k_{t-1}) = 0.2$.

The non-rigid classifier (3) is trained using the method described in Sec. III-A, where $C = 40$ in (20), which means that the profiles perpendicular to the LV contour have 41 pixels. In order to increase the robustness of the non-rigid classifier, we use same set $\mathcal{P}$ as the one mentioned before, with 100 positives

per training image defined in (20). Using 80% of $\mathcal{P}$ for training and 20% for validation, we have achieved the configuration displayed in Table II. Finally, for the PCA model, we cross validated E (number of eigenvectors) with the validation set, and selected $E = 10$.

The detection procedure in Alg. 2 uses $H_{\text{coarse}} = 1000$ (at $\sigma = 16$, this means that the initial grid has around four points in each of the five dimensions of $\text{Dist}(\theta; \mathcal{D})$) and $H_{\text{fine}} = 10$ based on the trade off between segmentation accuracy and running time (i.e., the goal was to reduce H_{coarse} and H_{fine} as much as possible without affecting the results on the validation set).

Using the training parameters defined above, the run-time complexity of the different search approaches (full and gradient descent) is presented in terms of the number of calls to the DNN classifiers, which represents the bottleneck of the segmentation algorithm. The *full search* approach has a search complexity of $H_{\text{coarse}} + (\#\text{scales} - 1) \times H_{\text{fine}} \times 3^5 + H_{\text{fine}} \times N$, where H_{coarse} is $O(10^3)$, H_{fine} is $O(10)$, and for the non-rigid classifier, the detection of each contour point is independent of the detection of other contour points (see Eq. 3). From Table II, we notice that the complexity of the affine classifier at $\sigma = 16$ is $O(16 \times 100 \times 50 \times 3) = O(2.4 \times 10^5)$, at $\sigma = 8$ is $O(49 \times 50 \times 100 \times 3) = O(7.3 \times 10^5)$, at $\sigma = 4$ is $O(196 \times 100 \times 100 \times 200 \times 200 \times 3) = O(2.35 \times 10^{11})$, and the non-rigid classifier is $O(41 \times 50 \times 50 \times 1) = O(1 \times 10^5)$. This means that the full search method (using 243 samples in fine scale for each of the H_{fine} samples) needs roughly the following number of multiplications: $1000 \times 2.4 \times 10^5 + 10 \times 3^5 \times 7.3 \times 10^5 + 10 \times 3^5 \times 2.35 \times 10^{11} + 10 \times 21 \times 1 \times 10^5 \approx 5.7 \times 10^{14}$.

For the *gradient descent* search procedure, each iteration above (at $\sigma_q \in \{8, 4\}$) represents a computation of the classifier in 10 points of the search space (five parameters times two points) plus the line search computed in 10 points as well. The gradient descent search needs roughly the following number of multiplications: $1000 \times 2.4 \times 10^5 + 10 \times [20, 100] \times 7.3 \times 10^5 + 10 \times [20, 100] \times 2.35 \times 10^{11} + 10 \times 21 \times 1 \times 10^5 \in [4.7 \times 10^{13}, 2.3 \times 10^{14}]$, where $[20, 100]$ means that by limiting the number of iterations to be between one and five, the complexity of this step for each hypothesis θ_i is between 20 and 100.

C. Error Measures

In order to evaluate our algorithm, we use the following error measures: Hammoude distance (HMD) (also known as Jaccard distance) [67], average error (AV) [26], mean absolute distance (MAD) [68], and average perpendicular error (AVP) between the estimated and ground truth contours.

Let $\mathbf{s}_1 = [\mathbf{s}_1^\top(i)]_{i=1}^S$, and $\mathbf{s}_2 = [\mathbf{s}_2^\top(i)]_{i=1}^S$, with $\mathbf{s}_1(i), \mathbf{s}_2(i) \in \Re^2$ be two vectors of points representing the automatic and manual LV contours, respectively. The smallest point $\mathbf{s}_1(i)$ to contour $\mathbf{s}_2$ distance is:

$$d(\mathbf{s}_1(i), \mathbf{s}_2) = \min_j ||\mathbf{s}_2(j) - \mathbf{s}_1(i)||_2, \tag{23}$$

which is the distance to the closest point (DCP). The average error between $\mathbf{s}_1$ and $\mathbf{s}_2$ is

$$d_{\text{AV}}(\mathbf{s}_1, \mathbf{s}_2) = \frac{1}{S} \sum_{i=1}^{S} d(\mathbf{s}_1(i), \mathbf{s}_2). \tag{24}$$

The Hammoude distance is defined as follows [67]:

$$d_{\mathrm{HMD}}(\mathbf{s}_1, \mathbf{s}_2) = \frac{\#((R_{\mathbf{s}_1} \cup R_{\mathbf{s}_2}) - (R_{\mathbf{s}_1} \cap R_{\mathbf{s}_2}))}{\#(R_{\mathbf{s}_1} \cup R_{\mathbf{s}_2})}, \tag{25}$$

where $R_{\mathbf{s}_1}$ represents the image region delimited by the contour $\mathbf{s}_1$ (similarly for $R_{\mathbf{s}_2}$), $\cup$ is the set union operator, $\cap$ is the set intersection operator, and $\#(.)$ denotes the number of pixels within the region described by the expression in parenthesis. The error measure MAD [69] is defined as follows:

$$d_{\mathrm{MAD}}(\mathbf{s}_1, \mathbf{s}_2) = \frac{1}{S} \sum_{i=1}^{S} \|\mathbf{s}_1(i) - \mathbf{s}_2(i)\|_2. \tag{26}$$

Note that MAD (26) is defined between corresponding points (not DCP).

Finally, the average perpendicular error (AVP) between estimated (say $\mathbf{s}_2$) and reference ($\mathbf{s}_1$) contours is the minimum distance between $\mathbf{s}_2(i)$ and $\mathbf{s}_1(i)^\star$ using a line perpendicular to the contour at $\mathbf{s}_2$ at $\mathbf{s}_2(i)$. Let us represent the line tangent to the curve at the point $\mathbf{s}_2(i)$ as $\mathcal{L} = \{\mathbf{s}_2(i-1) + t(\mathbf{s}_2(i+1) - \mathbf{s}_2(i-1)) | t \in \Re\} = \{\mathbf{s}_2 | \mathbf{a}^\top \mathbf{s}_2 + b = 0\}$ with $\mathbf{a}^\top(\mathbf{s}_2(i+1) - \mathbf{s}_2(i-1)) = 0$ and $b = -\mathbf{a}^\top \mathbf{s}_2(i-1)$. Let us also denote the curve sampled at points $\mathbf{s}_1 = [\mathbf{s}_1^\top(i)]_{i=1}^S$ with the following implicit representation: $h(\mathbf{s}_1, \theta_{\mathbf{s}_1}) = 0$, where $\theta_{\mathbf{s}_1}$ denotes the parameters of this representation. Hence, we can find the point $\mathbf{s}_1(i)^\star = \arg\min_{\mathbf{s}(i) \in \mathbf{s}_1} (\|\mathbf{s}(i) - (s^* \mathbf{a} + \mathbf{s}_2(i))\|_2$, where $s^* = \arg\min s$ subject to $h(s\mathbf{a} + \mathbf{s}_2(i), \theta_{\mathbf{s}_1}) = 0$. The AVP error measure is defined as:

$$d_{\mathrm{AVP}}(\mathbf{s}_1, \mathbf{s}_2) = \frac{1}{S} \sum_{i=1}^{S} \|\mathbf{s}_1(i)^\star - \mathbf{s}_2(i)\|. \tag{27}$$

D. Comparison with the State of the Art

We compare the segmentations produced by two state-of-the-art methods [26,46,68] with those by our method (labeled '496 train img-F'), which has been trained with 496 annotated images from $\mathcal{T}_1$ (Sec IV-A) and uses the full search scheme (Sec III-B) with $\alpha = 0.5$ (10).

The model proposed by Nascimento et al. [26] (labeled 'MMDA') consists of a deformable template approach that uses multiple dynamic models to deal with the two LV motion regimes (systole and diastole), where the filtering approach is based on probabilistic data association (which deals with measurement uncertainty), and the shape model (that defines the LV shape variation) is based on a hand-built prior. The main differences between our model and MMDA are the following: MMDA is a fundamentally different approach based on deformable template model using an LV shape prior with a simple appearance model that is learned for each new test sequence based on a manual initialization of the LV contour. The model proposed by Comaniciu et al. [46,68] (labeled 'COM') is a supervised learning approach (i.e., it is a DB-guided approach) relying on a large annotated training set (in the order of hundreds of annotated images), using a discriminative classifier based on boosting techniques for the affine detector and a shape inference based on a nearest neighbor classifier for the non-rigid detection, and the motion

model is based on a shape tracking methodology that fuses shape model, system dynamics and the observations using heteroscedastic noise. Compared to our model, COM uses a different type of classifier for the affine and non-rigid classifiers, and a different type of motion model. The methods 'MMDA' and 'COM' have been run on the data set of normal cases $\mathcal{T}_{2,\{A,B\}}$ by the original authors of those methods. Moreover, in order to assess the robustness of our method to small training sets, we randomly select a subset of the 496 annotated images from $\mathcal{T}_1$ to train our method, where the subset size varies from $\{20, 50, 100, 200\}$ (labeled '$\{20, 50, 100, 200\}$ train img-F'), and compare the error measures obtained with the segmentations from the DNN classifier trained with 496 images (labeled '496 train img-F'). Notice that we re-trained all parameters of the system with these subsets in order to provide a reliable estimate of performance in case only a small training set is available (even the range $r(\Theta)$ is re-defined for the smaller training sets). The only values that are fixed throughout this experiment are: the *range of possible number* of hidden layers in the training of the DNN, the *range of number* of nodes per layer in the training of the DNN, and the transition distribution $p(k_t|k_{t-1})$ defined in (4). We also compare the segmentations of the gradient descent search scheme (labeled '496 train img-G') with that of the full search. Furthermore, we study the influence of α in (10), which determines the weight between the transition and observation models for building the proposal distribution using '496 train img-F'. Finally, we include in this comparison the performance of our own method without the use of the dynamical model (i.e., a separate segmentation is performed in each new frame using Alg. 2, without any information from the previous frame or the motion model, and the final segmentation is achieved by computing the Monte-Carlo estimation of the LV contour using the final H_{fine} hypotheses). This method is labeled as 'STATIC' in the experiments.

E. Comparison with Inter-user Statistics

The assessment of the performance of our method ('496 train img-F') against the inter-user variability follows the methodology proposed by Chalana and Kim [70] (revised by Lopez *et al.* [71]), using the gold standard LV annotation computed from the manual segmentations [70]. The measures used are the following: *modified Williams index*, *Bland-Altman plot* [72], and *scatter plot*. These comparisons are performed on the diseased sets $\mathcal{T}_{1,\{A,B,C\}}$, for which we have four LV manual annotations per image produced by four different Cardiologists (Sec. IV-A). In these sequences, we have an average of 17 images annotated for each sequence, so in total we have 52 images annotated by four experts. In order to have a fair comparison, we train three separate DNN classifiers using the following training sets: 1) $\mathcal{T}_1 \setminus \mathcal{T}_{1,A}$, 2) $\mathcal{T}_1 \setminus \mathcal{T}_{1,B}$, and 3) $\mathcal{T}_1 \setminus \mathcal{T}_{1,C}$, where $\setminus$ represents the set difference operator. These three classifiers are necessary because when testing any image inside each one of these three sequences, we cannot use any image of that same sequence in the training process.

1) Modified Williams Index: Assume that we have a set $\{\mathbf{s}_{j,u}\}$, where $j \in \{1..M\}$ indexes the images in one of the disease sets $\mathcal{T}_{1,\{A,B,C\}}$, and $u \in \{0..U\}$ indexes the manual annotations, where the index $u = 0$ denotes the computer-generated contour (i.e., each one of the M images has U manual annotations). The function $D_{u,u'}$ measures the disagreement between users u and u', which is defined as

$$D_{u,u'} = \frac{1}{M}\sum_{j=1}^{M} d_{-}(\mathbf{s}_{j,u}, \mathbf{s}_{j,u'}), \tag{28}$$

where $d_{-}(.,.)$ is an error measure between two annotations $\mathbf{s}_{j,u}$, $\mathbf{s}_{j,u'}$, which can be any of the measures defined previously in (24)-(27). The modified Williams index is defined as

$$I^{'} = \frac{\frac{1}{U}\sum_{u=1}^{U}\frac{1}{D_{0,u}}}{\frac{2}{U(U-1)}\sum_{u}\sum_{u':u'\neq u}\frac{1}{D_{u,u'}}}. \tag{29}$$

A confidence interval (CI) is estimated using a jackknife (leave one out) non-parametric sampling technique [70] as follows:

$$I^{'}_{(.)} \pm z_{0.95} se, \tag{30}$$

where $z_{0.95} = 1.96$ represents 95^{th} percentile of the standard normal distribution, and

$$se = \left\{ \frac{1}{M-1}\sum_{j=1}^{M}[I^{'}_{(j)} - I^{'}_{(.)}] \right\} \tag{31}$$

with $I^{'}_{(.)} = \frac{1}{M}\sum_{j=1}^{M} I^{'}_{(j)}$, and $I^{'}_{(j)}$ is the Williams index (29) calculated by leaving image j out of computation of $D_{u,u'}$. A successful measurement for the Williams index is to have the average and confidence interval (30) close to one.

2) Bland-Altman and Scatter Plots: We also present quantitative results using the Bland-Altman [72] and scatter plots (from which it is possible to compute a linear regression, the correlation coefficient and the p-value). To accomplish this we have: (i) the gold standard LV volume computed via an iterative process using the manual annotations [70]; (ii) the Cardiologists' LV volumes, and (iii) the computer generated LV volume. To estimate the LV volume from 2-D contour annotation we use the area-length equation [73,74] with $V = \frac{8A^2}{3\pi L}$, where A denotes the projected surface area, L is the distance from upper aortic valve point to apex, and V is expressed in cubic pixels. The p-values are computed as follows: 1) compute several independent p-values from 3 samples, each taken from separate sequence; and then 2) combine the p-values using the Fisher's method into a single result by assuming independence among the p-values [75].

V. Experimental Results

Figure 9 shows the error measures (24)-(27) in sequences $\mathcal{T}_{2,\{A,B\}}$ using box plot graphs labeled as described in Sec. IV-D, where we compare the segmentation results of 'COM' [46,68], 'MMDA' [26],

TABLE III

COMPARISON OF THE COMPUTER GENERATED CURVES TO THE USERS' CURVES WITH RESPECT TO ALL THE ERROR MEASURES FOR THREE SEQUENCES USING THE AVERAGE AND 95% CONFIDENCE INTERVAL (IN PARENTHESIS) OF THE WILLIAMS INDEX.

measure	d_{HMD}	d_{AV}	d_{MAD}	d_{AVP}
Average(CI)	$0.83\ (0.82, 0.84)$	$0.91\ (0.90, 0.92)$	$0.94\ (0.93, 0.95)$	$0.83\ (0.82, 0.84)$

and 'STATIC' against those of $\{20, 50, 100, 200, 496\}$ train img-$\{$F,G$\}$ with $\alpha = 0.5$, see (10), and those of 496 train img-$\{$F$\}$ with varying $\alpha \in \{0, 0.1, 0.25, 0.50.75, 0.9, 1\}$. In order to measure the statistical significance of the results of '496 train img-F' compared to 'COM', 'MMDA' and 'STATIC', we use the t-test, where the null hypothesis is that the difference between two responses has mean value of zero (we used the Welch's t-test, which assumes normal distributions with different variances). For all tests, a value of $p < 0.05$ was considered statistically significant. In both sequences $\mathcal{T}_{2,\{A,B\}}$, we obtained $p < 0.05$ with respect to 'MMDA' and 'COM' for all measures. Figure 10 displays a qualitative comparison of the results of '496 train img-F', 'MMDA, 'COM', and the expert annotation. Compared to STATIC, we obtained $p < 0.05$ only in sequence $\mathcal{T}_{2,A}$ for all measures. In terms of running time, using a non-optimized Matlab implementation, the full search takes around 20 seconds to run per frame, and the gradient descent search runs in between 5 to 10 seconds on a laptop computer with the following configuration: Intel Centrino Core Duo (32 bits) at 2.5GHz with 4GB.

In terms of inter-user statistics, Table III shows the average and confidence intervals of the Williams index defined in (29)-(30) for all ultrasound sequences considered for the comparison with inter-user statistics. Finally, Fig. 11 shows the scatter and Bland-Altman plots. In the scatter plot, notice that the correlation coefficient between the users and gold standard is 0.99 with p-value$= 3.11 \times 10^{-68}$ (see graph Inter-user) and for the gold standard versus computer the correlation is 0.95 with p-value$= 1.9 \times 10^{-4}$ (graph Gold vs Computer). In the Bland-Altman plots, the Inter-user plot produced a bias of 4.9×10^4 with confidence interval of $[-5 \times 10^5, 5 \times 10^5]$, while the Gold vs Computer plot shows a bias of 1.8×10^5 and confidence interval of $[-4 \times 10^5, 7 \times 10^5]$.

VI. DISCUSSION

The main objective of this work is to propose a new DB-guided methodology for the fully automatic LV segmentation problem, which produces state-of-the-art results that correlates well with inter-user statistics. Our new dynamical model generates a proposal distribution combining the results of an observation model based on a deep neural classifier and a transition model that estimates the current values for the LV contour and cardiac phase using their previous values. The use of deep neural classifier is justified because of its abstraction capabilities that can reduce the need of a large and rich training set. The

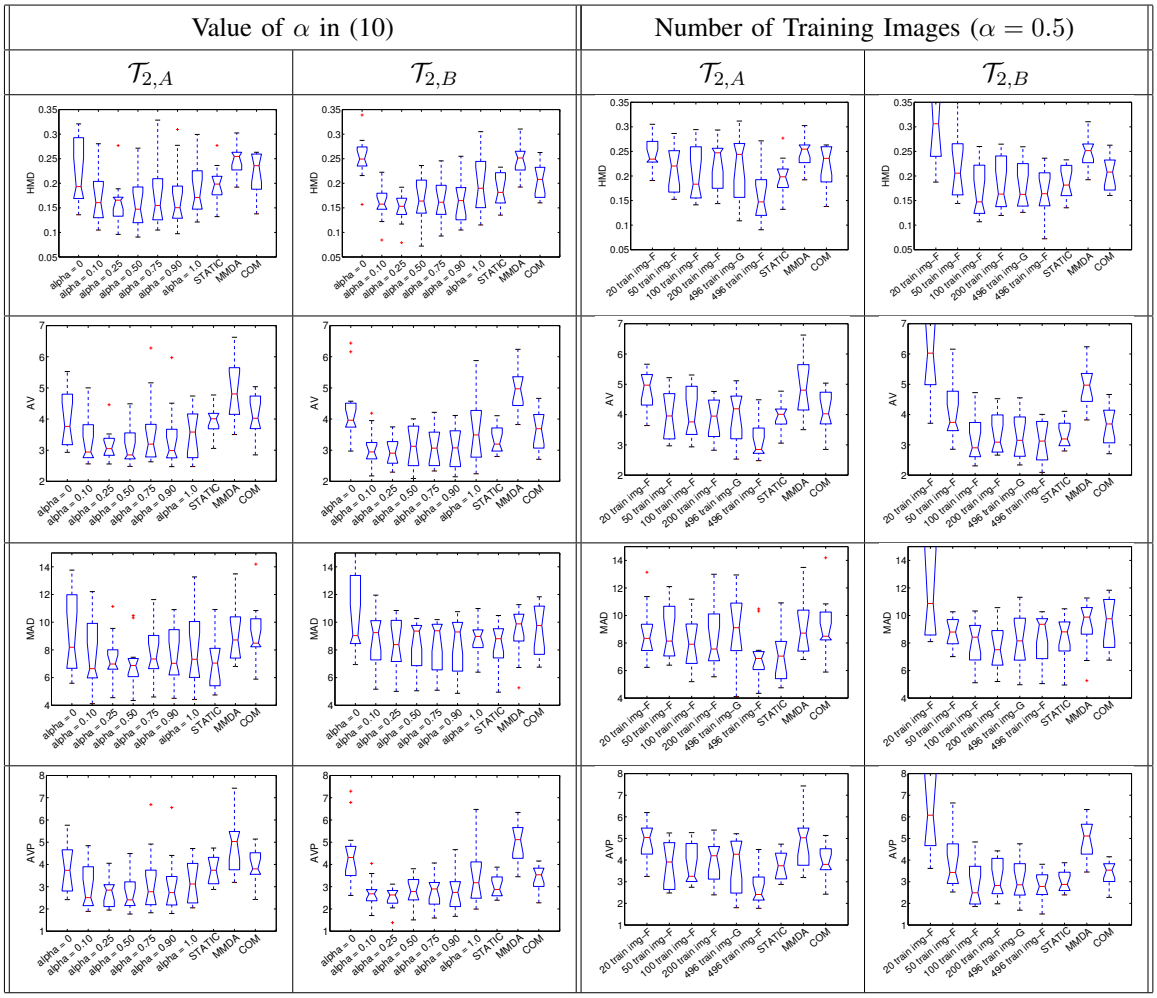


Fig. 9. Box plot results for all error measures explained in Sec. IV-C (the measures are denoted in the vertical axis of each graph). Using the sequences $\mathcal{T}_{2,A}$ (columns 1 and 3) and $\mathcal{T}_{2,B}$ (columns 2 and 4), we compare the segmentation of our method with varying values for α (columns 1 and 2), and varying values of training images and search approaches (columns 3 and 4) with the segmentation produced by 'STATIC' (i.e., our own method without the use of the dynamical model proposed in this paper), 'MMDA' [26] and 'COM' [46,68].

efficiency of the observation model is improved by decoupling the affine and non-rigid detections and by using gradient-based search with multiple hypotheses. The results shown in Sec. V support our claims.

For instance, the comparison between our approach and other state-of-the-art methods [26,46,68] on the data set of normal cases shows that our approach trained with 496 images and using the full search scheme (i.e., the '496 train img-F') produces significantly more precise results than 'MMDA' and 'COM' in the sequences $\mathcal{T}_{2,\{A,B\}}$ for all error measures. The experiment that studies the value of α in the proposal

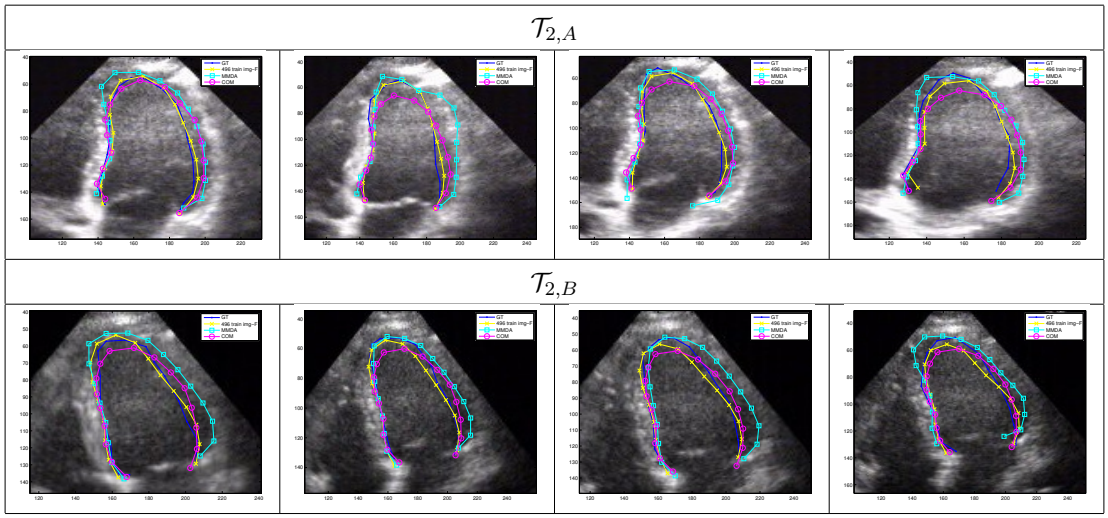


Fig. 10. Qualitative comparison between the expert annotation (GT in blue with point markers) and the results of '496 train img-F' (yellow with 'x' markers), 'MMDA' (cyan with square markers), and 'COM' (purple with 'o' markers).

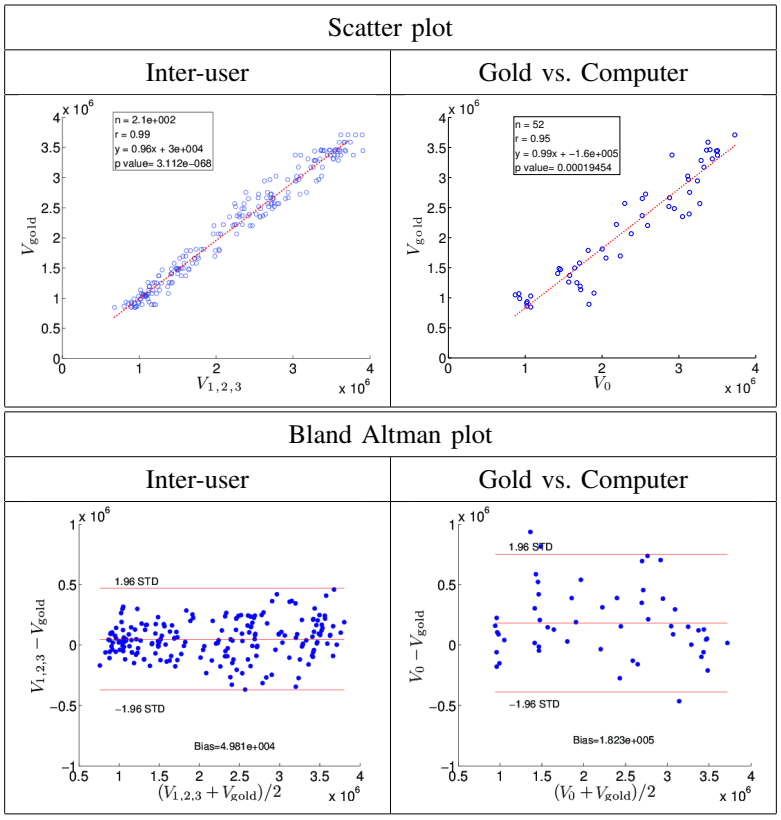


Fig. 11. Scatter plots with linear regression and Bland-Altman bias plots (V denotes LV volume).

distribution (Fig. 9) shows that the best results are achieved when $\alpha \in [0.25, 0.75]$, which means that the combination of both models is an important aspect of the algorithm. The results in Fig. 9 also show that our method is robust to a severe reduction of the training set size (notice that a training set of 50 images produces competitive results). Another interesting point shown in Fig. 9 is the influence of the observation model in the process, which seems to be highly significant given that for both sequences it produces results that are quite comparable to the proposed approach with the dynamical model, but notice that in one of the sequences, the use of the dynamics improves significantly the accuracy. Finally, the qualitative comparison in Fig. 10 shows that our approach is more precise in the detection of the right border of the LV than 'MMDA', which tends to overshoot this border detection; also, the apical border detection (upper part of the LV) produced by our method is consistently more accurate than the result by 'COM', which tends to undershoot that border detection. All three approaches seem to be equally precise in the detection of the left border of the LV.

All implementations proposed in this paper enable significant run-time complexity reductions. For instance, a naive search over the $5 + 42$ dimensions of the affine and non-rigid spaces would imply a run-time complexity of at least $O(10^{47} \times 10^{11})$, where $O(10^{11})$ is the complexity of a typical deep DNN classifier (see Sec. IV-B). The separation between affine and non-rigid classifier reduces this figure to $O(10^{42} \times 10^{11})$, and the independence assumption of the contour points, further reduces this complexity to $O(10^{5} \times 10^{11})$. Finally, the coarse-to-fine search used allows for a complexity in the order of $O(10^{14})$, and the gradient based search can reduce the complexity to $O(10^{13})$ without showing any significant deterioration in terms of segmentation accuracy. In practice, we believe that an efficient C++ implementation of our algorithm can reduce the running time of the method to well under one second on a modern desktop computer. Moreover, our derivative-based search process can be easily combined with MSL [48] to improve even more the search efficiency.

Finally, the inter-user statistics on the data set of diseased cases shows that the results produced by our approach are within the variability of the manual annotations of four cardiologists using several error metrics (four error measures) and statistical evaluations (Williams index, Bland-Altman plot and scatter plot). Nevertheless, notice from Fig. 11 that the automated contours vs. the manual gold standard present slightly larger variance and bias than the individual users vs. the gold standard, which means that there is still room for improvement for our algorithm.

A. Limitations of the Method

The main limitations of the proposed approach can be summarized as follows. Even though a small training set can be used to train the DNN classifiers, it is important to have a reasonably rich initial training set (for instance, it is better to have 50 annotated images collected from different sequences than to have 50 images from the same sequence). Another issue with our approach is with the range of

values to search in the first step of the inference procedure (i.e., the range $r(.)$ of the uniform distribution $\mathcal{U}(r(\Theta))$ in Alg. 2), which is estimated from the training set, as defined in (16). For small training sets, this range may not cover the location of the LV in the test image, so a solution for this problem (adopted in this work) is to increase this range artificially. This can have undesirable consequences, such as unnecessarily large search spaces that lead not only to inefficient search procedures, but also to a larger number of local minima. Finally, we also observe some instability in the tracking results of some areas of the LV border. What happens is that, due to the clutter and low signal-to-noise ratio of the ultrasound images, the locations of these (noisy) observations in consecutive normal lines often occur at distinct locations. The consequence is that the state estimate (i.e., the cardiac phase and LV contour) is directly influenced by these local perturbations, which produces the visually observed instability. This is a classical behavior of the filtering step in tracking systems based on Kalman and particle filters, for Gaussian and non-Gaussian assumptions.

VII. Conclusion and Future Work

We presented a new fully automatic methodology for the LV tracking and segmentation using ultrasound data. The novelties of our approach are centered in a new dynamical model based on a SIR formulation, where the main novelties are with respect to the new observation and transition models. We show that the proposed observation model, based on deep neural network, can be learned with training sets of limited size (state-of-the-art results are obtained with training sets of 50 images). Also decoupling the affine and non-rigid detections and using a gradient-based search scheme prove to be simple but effective approaches to reducing the running time complexity. Finally a motion model that does not commit to a specific heart dynamical regime, and that combines the transition and observation model results to build a proposal distribution shows effective tracking accuracy. According to the results, our approach is more accurate than other state-of-the-art DB-guided [46,68] and deformable template [26] methodologies and correlates well with inter-user statistics. In the future, we plan to address the issues mentioned in Sec. VI, with the development of a semi-supervised approach [76] to reduce the dependence on a rich initial training set, and with the implementation of an approach to automatically determine the likely positions, rotation and scales of the LV in the test image. We also plan to work on a shape model that is less dependent on the training set, similarly to the DNN used for the appearance model. Moreover, we plan to apply this approach to other anatomies and other medical imaging techniques.

References

[1] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: A survey," *IEEE Trans. Med. Imag.*, vol. 25, no. 8, pp. 987–1010, 2006.

[2] J. Bosch, S. Mitchell, B. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models," *IEEE Trans. Med. Imag.*, vol. 21, no. 11, pp. 1374–1383, 2002.

[3] E. Bardinet, L. Cohen, and N. Ayache, "Tracking and motion analysis of the left ventricle with deformable superquadratics," *Med. Image Anal.*, vol. 2, no. 1, pp. 129–149, 1996.

[4] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking and matching deformable contours," *IEEE Trans. Pat. Anal. Mach. Int.*, vol. 3, no. 17, pp. 294–302, 1995.

[5] T. McInerney and D. Terzopoulos, "A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to 4-D image analysis," *Comput. Med. Imag. Graph.*, vol. 1, no. 19, pp. 69–83, 1995.

[6] J. Montagnat and H. Delingette, "4D deformable models with temporal constraints: Application to 4D cardiac image segmentation," *MeDIA*, vol. 9, pp. 87–100, 2005.

[7] D. Comaniciu, X. Zhou, and S. Krishnan, "Robust real-time myocardial border tracking for echocardiography: An information fusion approach," *IEEE Trans. Med. Imag.*, vol. 23, no. 7, pp. 849–860, 2004.

[8] M. Lynch, O. Ghita, and P. Whelan, "Segmentation of the left ventricle of the heart in 3-D+T MRI data using an optimized nonrigid temporal model," *IEEE Trans. Med. Imag.*, vol. 27, no. 2, pp. 195–203, 2008.

[9] N. Paragios, "4-D deformable models with temporal constraints: Application to 4-D cardiac image segmentation," *Med. Imag. Anal.*, vol. 9, no. 1, pp. 87–100, 2005.

[10] J. Sénégas, T. Netsch, C. Cocosco, G. Lund, and A. Stork, "Segmentation of medical images with a shape and motion model: A Bayesian perspective," in *ECCV Workshops CVAMIA and MMBIA*, 2004, pp. 157–168.

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

[12] G. Jacob, J. A. Noble, C. Behrenbruch, A. D. Kelion, and A. P. Banning, "A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography," *IEEE Trans. Med. Imag.*, vol. 21, no. 3, pp. 226–238, 2002.

[13] J. Dias and J. Leitão, "Wall position and thickness estimation from sequences of echocardiograms images," *IEEE Trans. Med. Imag.*, vol. 15, no. 1, pp. 25–38, 1996.

[14] N. Friedland and D. Adam, "Automatic ventricular cavity boundary detection from sequential ultrasound images using simulated annealing," *IEEE Trans. Med. Imag.*, vol. 8, no. 4, pp. 344–353, 1989.

[15] J. Montagnat, M. Sermesant, H. Delingette, G. Maladain, and N. Ayache, "Anisotropic filtering for model-based segmentation of 4-D cylindrical echocardiographic images," *Patt. Rec. Lett.*, vol. 24, no. 4-5, pp. 815–828, 2003.

[16] S. K. Setarehdan and J. J. Soraghan, "Automatic cardiac LV boundary detection and tracking using hybrid fuzzy temporal and fuzzy multiscale edge detection," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 11, pp. 1364–1378, 1999.

[17] M. Lorenzo-Valdes, G. Sanchez-Ortiz, A. Elkington, R. Mohiaddin, and D. Rueckert, "Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm," *Medical Image Analysis*, vol. 8, 2004.

[18] W. Sun, M. Cetin, R. Chan, and A. Willsky, "Learning the dynamics and time-recursive boundary detection of deformable objects," *IEEE Trans. Imag. Proc.*, vol. 17, no. 11, pp. 2186–2200, 2008.

[19] G. Carneiro and J. C. Nascimento, "Multiple dynamic models for tracking the left ventricle of the heart from ultrasound data using particle filters and deep learning architectures," in *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2010.

[20] Z. Qian, D. Metaxas, and L. Axel, "Boosting and nonparametric based tracking of tagged MRI cardiac boundaries," in *MICCAI*, 2006.

[21] 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," *Inf Process Med Imaging*, vol. 19, pp. 553–565, 2005.

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

[23] Y. Zhu, X. Papademetris, A. Sinusas, and J. Duncan, "Segmentation of the left ventricle from cardiac MR images using a subject-specific dynamical model," *IEEE Trans. Med. Imag.*, vol. 29, no. 3, pp. 669–687, 2010.

[24] V. Chalana, D. Linker, D. Haynor, and Y. Kim, "A multiple active contour model for cardiac boundary detection on echocardiographic sequences," *IEEE Trans. Med. Imag.*, vol. 15, no. 3, pp. 290–298, 1996.

[25] M. Jolly, H. Xue, L. Grady, and J. Guehring, "Combining registration and minimum surfaces for the segmentation of the left ventricle in cardiac cine MR images," in *MICCAI*, 2009.

[26] J. C. Nascimento and J. S. Marques, "Robust shape tracking with multiple models in ultrasound images," *IEEE Trans. Imag. Proc.*, vol. 17, no. 3, pp. 392–406, 2008.

[27] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 321–331, 1987.

[28] O. Bernard, B. Touil, A. Gelas, R. Prost, and D. Friboulet, "A RBF-based multiphase level set method for segmentation in echocardiography using the statistics of the radiofrequency signal," in *ICIP*, 2007.

[29] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser, "Variational B-spline level-set: A linear filtering approach for fast deformable model evolution," *IEEE Trans. Imag. Proc.*, vol. 18, no. 6, pp. 1179–1991, 2009.

[30] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti, "Left ventricular volume estimation for real-time three-dimensional echocardiography," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1202–1208, 2002.

[31] D. Cremers, S. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.

[32] E. Debreuve, M. Barlaud, G. Aubert, I. Laurette, and J. Darcourt, "Space-time segmentation using level set active contours applied to myocardial gated SPECT," *IEEE Trans. Med. Imag.*, vol. 20, no. 7, pp. 643–659, 2001.

[33] N. Lin, W. Yu, and J. Duncan, "Combinative multi-scale level set framework for echocardiographic image segmentation," *Medical Image Analysis*, vol. 7, no. 4, pp. 529–537, 2003.

[34] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 158–175, 1995.

[35] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, no. 3, pp. 223–247, 2002.

[36] N. Paragios, "A level set approach for shape-driven segmentation and tracking of the left ventricle," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 773–776, 2003.

[37] A. Sarti, C. Corsi, E. Mazzini, and C. Lamberti, "Maximum likelihood segmentation of ultrasound images with Rayleigh distribution," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 52, no. 6, pp. 947–960, 2005.

[38] T. Chen, J. Babb, P. Kellman, L. Axel, and D. Kim, "Semiautomated segmentation of myocardial contours for fast strain analysis in cine displacement-encoded MRI," *IEEE Trans. Med. Imag.*, vol. 27, no. 8, pp. 1084–1094, 2008.

[39] Q. Duan, E. D. Angelini, and A. Laine, "Real time segmentation by active geometric functions," *Comput. Methods Programs Biomed.*, vol. 98, no. 3, pp. 223–230, 2010.

[40] M. Mignotte, J. Meunier, and J. Tardif, "Endocardial boundary estimation and tracking in echocardiographic images using deformable template and Markov random fields," *Pattern Analysis and Applications*, vol. 4, no. 4, pp. 256–271, 2001.

[41] V. Zagrodsky, V. Walimbe, C. Castro-Pareja, J. X. Qin, J.-M. Song, and R. Shekhar, "Registration-assisted segmentation of real-time 3-D echocardiographic data using deformable models," *IEEE Trans. Med. Imag.*, vol. 24, no. 9, pp. 1089–1099, 2005.

[42] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, 1995.

[43] T. Cootes, C. Beeston, G. Edwards, and C. Taylor, "A unified framework for atlas matching using active appearance models," in *Information Processing in Medical Imaging*, 1999, pp. 322–333.

[44] S. Mitchell, B. Lelieveldt, R. van der Geest, H. Bosch, J. Reiber, and M. Sonka, "Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac MR images," *IEEE Trans. Med. Imag.*, vol. 20, no. 5, pp. 415–423, 2001.

[45] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu, "Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree," *IEEE Trans. Med. Imaging*, vol. 27, no. 9, pp. 1342–1355, 2008.

[46] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta, "Databased-guided segmentation of anatomical structures with complex appearance," in *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2005.

[47] J. Weng, A. Singh, and M. Chiu, "Learning-based ventricle detection from cardiac MR and CT images," *IEEE Trans. Med. Imag.*, vol. 16, no. 4, pp. 378–391, 1997.

[48] 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 Trans. Med. Imaging*, vol. 27, no. 11, pp. 1668–1681, 2008.

[49] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[50] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, pp. 174–188, 2002.

[51] R. Salakhutdinov and G. Hinton, "Learning a non-linear embedding by preserving class neighbourhood structure," in *AI and Statistics*, 2007.

[52] G. Carneiro, J. C. Nascimento, and A. Freitas, "Robust left ventricle segmentation from ultrasound data using deep neural networks and efficient search methods," in *IEEE Int. Symp. on Biomedical Imaging, from nano to macro (ISBI)*, 2010, pp. 1085–1088.

[53] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *ECCV*, 2004.

[54] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[55] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, no. 323, pp. 533–536, 1986.

[56] M. Carreira-Perpinan and G. Hinton, "On contrastive divergence learning," in *Workshop on Artificial Intelligence and Statistics*, 2005.

[57] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[58] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[59] G. Hinton. http://videolectures.net/nips09_hinton_dlmi/.

[60] R. Gonzalez and R. Woods, *Prenticel Hall*. Digital Image Processing, 2008.

[61] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley and Sons, 2001.

[62] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Elsevier, 1990.

[63] A. Dempster, M. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

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

[65] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2001, pp. 511–518.

[66] G. Carneiro, J. C. Nascimento, and A. Freitas, "The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods," *Image Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 968–982, 2012.

[67] A. Hammoude, "Computer-assited endocardial border identification from a sequence of two-dimensional echocardiographic images," Ph.D. dissertation, University Washington, 1988.

[68] X. S. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 1, pp. 115–129, 2005.

[69] I. Mikić, S. Krucinki, and J. D. Thomas, "Segmentation and tracking in echocardiographic sequences: Active contours guided by optical flow estimates," *IEEE Trans. Med. Imag.*, vol. 17, no. 2, pp. 274–284, 1998.

[70] V. Chalana and Y. Kim, "A methodology for evaluation of boundary detection algorithms on medical images," *IEEE Trans. Med. Imag.*, vol. 16, no. 10, 1997.

[71] C. Alberola-Lopez, M. Martin-Fernandez, and J. Ruiz-Alzola, "Comments on: A methodology for evaluation of boundary detection algorithms on medical images," *IEEE Trans. Med. Imag.*, vol. 23, no. 5, pp. 658–660, 2004.

[72] J. Bland and A. Altman, "Statistical methods for assessing agreement between two methods of clinical measurement," *Lancet*, vol. 1, no. 8476, pp. 307–310, 1986.

[73] J. C. Reiber, A. R. Viddeleer, G. Koning, M. J. Schalij, and P. E. Lange, "Left ventricular regression equations from single plane cine and digital X-ray ventriculograms revisited," *Clin. Cardiology*, vol. 12, no. 2, pp. 69–78, 1996, kluwer Academic Publishers.

[74] H. Sandler and H. T. Dodge, "The use of single plane angiocardiograms for the calculation of left ventricular volume in man," *Amer. Heart J.*, vol. 75, no. 3, pp. 325–334, 1968.

[75] R. Fisher, "Questions and answers number 14," *The American Statistician*, vol. 2, no. 5, pp. 30–31, 1948.

[76] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

Gustavo Carneiro received the BS and MSc degrees in computer science from the Federal University of Rio de Janeiro, and the Military Institute of Engineering, Brazil, in 1996 and 1999, respectively. Dr. Carneiro received the PhD degree in Computer Science from the University of Toronto, Canada, in 2004. Currently he is a senior lecturer at the School of Computer Science of the University of Adelaide in Australia (this position is equivalent to associate professor in North America). Previously, Dr. Carneiro worked at the Instituto Superior Técnico (IST), Technical University of Lisbon from 2008 to 2011 as a visiting researcher and assistant professor, and from 2006-2008, he worked at Siemens Corporate Research in Princeton, USA. He was the recipient of a Marie Curie International Incoming Fellowship and has authored more than 35 peer-reviewed publications in international journals and conferences. His research interests include medical image analysis, image feature selection and extraction, content-based image retrieval and annotation, and general visual object classification.

Jacinto C. Nascimento (M'06) received the EE degree from Instituto Superior de Engenharia de Lisboa, in 1995, and the MSc and PhD degrees from Instituto Superior Técnico (IST), Technical University of Lisbon, in 1998, and 2003, respectively. Currently, he is a posdoctoral researcher with the Institute for Systems and Robotics (ISR) at IST. His research interests include image processing, pattern recognition, tracking, medical imaging, video surveillance, machine learning and computer vision. Dr. Nascimento has co-authored over 80 publications in international journals and conference proceedings (many of which of the IEEE), has served on program committees of many international conferences, and has been a reviewer for several international journals.