MULTI-ATLAS SEGMENTATION USING MANIFOLD LEARNING WITH DEEP BELIEF NETWORKS

Jacinto C. Nascimento[(a)] *Gustavo Carneiro*[(b)]

Instituto de Sistemas e Robótica, Instituto Superior Técnico, 1049-001 Lisboa, **Portugal**[(a)]
Australian Centre for Visual Technologies, The University of Adelaide, **Australia** [(b)]

ABSTRACT

This paper proposes a novel combination of manifold learning with deep belief networks for the detection and segmentation of left ventricle (LV) in 2D - ultrasound (US) images. The main goal is to reduce both training and inference complexities while maintaining the segmentation accuracy of machine learning based methods for non-rigid segmentation methodologies. The manifold learning approach used can be viewed as an atlas-based segmentation. It partitions the data into several patches. Each patch *proposes* a segmentation of the LV that somehow must be fused. This is accomplished by a deep belief network (DBN) multi-classifier that assigns a weight for each patch LV segmentation. The approach is thus threefold: (i) it does not rely on a single segmentation, (ii) it provides a great reduction in the rigid detection phase that is performed at lower dimensional space comparing with the initial contour space, and (iii) DBN's allows for a training process that can produce robust appearance models without the need of large annotated training sets.

1. INTRODUCTION

Two sequential stages are typical in current machine learning methodologies for object segmentation [1, 2]: (i) rigid detection (coarse step) and (ii) non-rigid segmentation (fine step). The first step (rigid detection) is of crucial importance, since it reduces the *search running time and training complexities*. This paper achieves a complexity reduction of the rigid detection[1] by using a manifold learning algorithm. This is an atlas-based segmentation, in sense that the manifold partitions (by soft clustering) the data into several patches using two distinct assumptions: (i)preserving the angular within a patch using a smaller number of points and (ii) the distance (i.e. neighborhood) within these points [4]. Each patch in the learned manifold provides a segmentation proposal. Since multiple patches are obtained, multiple segmentations should be combined. In this paper, a novel strategy to accomplish this is developed. More specifically, a DBN multi-classifier for final segmentation is proposed. This means that a multi-atlas segmentation strategy is followed, *i.e.* the different segmentations are fused within patches, in which the weights are given by the deep belief network classifiers.

This work was supported by the FCT project [UID/EEA/50009/2013] and by Australian Research Council, discovery project DP140102794.

[1] State-of-the-art rigid detection produces in practice, translation, rotation and scaling of the visual object, (e.g [3]), i.e. $R = 5$. In this paper, the rigid detection space achieves $M < R$, where M in the intrinsic dimension of the manifold. See Section 6.1.

2. RELATED WORK

There exists works showing that the use of multiple segmentations as classifiers and combining them using fusion rules improve the segmentation accuracy. This has been shown in [5], where it is discussed different techniques for atlas selection and where several individual images are used as atlases and multi-classifier for the final segmentation. In the attempt to improve the reliance on a single segmentation in [6] it is proposed the so called STAPLE, in which the classifiers are weighted using the expectation-maximization (EM) algorithm. It has been shown that Multi-atlas propagation is useful in the context of brain segmentation in MRI. In [7], it is shown the superiority of using multi-atlas propagation and fusion when compared with single atlases for the segmentation of unseen images. The study presented in [8] also avoids the use of a single atlases, using instead, a family of templates. In [9] the LEAP framework for multi-atlas is proposed, where the initial atlases may represent a subset of target images. The goal is then to propagate a small number of atlases through a large set of MRI brain containing a significant amount of variability among the anatomical structures. Another contribution that uses manifold learning as a tool for atlas selection has been presented in [10] where three different manifold learning techniques (ISOMAP, LLE and Laplacian Eigenmaps) are assessed to select the best atlases and to combine in the multi-atlas segmentation context.

In this paper, our multiple atlas segmentation exhibits the following advantages: (i) reduction of the rigid detection space achieving $M < R$, using manifold with low intrinsic dimensionality which allows for a faster inference process (ii) reduction of the training data sets, since now the positives and negatives samples lie on the learned low-dimensional manifold and (iii) multiple segmentations fusion that allows for accuracy segmentation. We illustrate the performance of the proposed approach in the segmentation of the LV in 2D US data.

3. PROBLEM STATEMENT

The goal is to produce a non-rigid segmentation, say $\mathbf{y} \in \mathbb{R}^{2S}$ containing S 2-D points, representing the segmentation contour. We will represent the training set by $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_j\}_{j=1}^{|\mathcal{D}|}$,

where $\mathbf{x}_j : \Omega \rightarrow \mathbb{R}$ denotes the training images, $\mathbf{y}_j$ denotes the corresponding manual annotations and Ω stands for the image domain. The segmentation is achieved using the following function:

$$\mathbf{y}^* = E_{p(\mathbf{y}|\mathbf{x},\mathcal{D})}[\mathbf{y}] = \int_{\mathbf{y}} \mathbf{y} p(\mathbf{y}|\mathbf{x},\mathcal{D}) d\mathbf{y}. \quad (1)$$

The high dimensionality of $\mathbf{y}$ makes the computation of (1) challenging. The usual solution is the introduction of a coarse search step that can be solved in lower dimensionality, where the solutions are used to constrain and initialize an optimization process that can produce samples $\mathbf{y}$, which are then used in a Monte Carlo approximation of (1). This coarse step involves the use of a hidden variable $\mathbf{t} \in \mathbb{R}^R$, with $R << 2S$, as follows [11, 12, 13]:

$$p(\mathbf{y}|\mathbf{x},\mathcal{D}) = \int_{\mathbf{t}} p(\mathbf{t}|\mathbf{x},\mathcal{D}) p(\mathbf{y}|\mathbf{t},\mathbf{x},\mathcal{D}) d\mathbf{t}. \quad (2)$$

where the variable $\mathbf{t}$ is used to transform linearly the coordinates of a window that encloses the mean segmentation contour. This linear transform is obtained from the variable $\mathbf{t}$ as follows: $\mathbf{A_t} = f(\mathbf{t})$, where $\mathbf{A_t} \in \mathbb{R}^{3\times3}$ (see [11, 12, 13] for details). The meaning of each term in (2) is as follows: (i) $p(\mathbf{t}|\mathbf{x},\mathcal{D})$ represents the rigid detection classifier that outputs the probability of having the visual object within the boundaries of the window transformed by $\mathbf{t}$; (ii) $p(\mathbf{y}|\mathbf{t},\mathbf{x},\mathcal{D})$ is the non-rigid segmentation classifier denoted by the probability of finding the contour $\mathbf{y}$ in image $\mathbf{x}$ given the value of $\mathbf{t}$. This means that $\mathbf{t}$ denotes an initial guess for $\mathbf{y}$ and at the same time it constrains the search space of $\mathbf{y}$ to be around the mean segmentation contour transformed by $\mathbf{t}$.

Assuming that the original rigid search space represented by the variable $\mathbf{t}$ has dimension $R = 5$ (i.e. accounting for translation, rotation and scaling, e.g. [3]), one of the objectives of this paper is the introduction of a new space for $\mathbf{t}$ with dimension $M < R$, based on a manifold learning, where this rigid search will take place with gradient descent search mechanism on the manifold.

4. LEARNING THE MANIFOLD

The framework presented herein uses the manifold learning algorithm proposed in [4]. Basically, from a training samples $\{\mathbf{y}_j\}_{j=1}^{|\mathcal{D}|}$, this framework finds a manifold $\mathcal{M}$ contained in $\mathbb{R}^{2S}$, associated with a set of one-to-one mappings $\zeta_i : \mathcal{P}_i \rightarrow \mathcal{U}_i$ (i.e. the *charts*) and invertible functions $\mathbf{y}_{i,j} = \xi_i(\mathbf{t}_{i,j})$, (or *parameterizations* $\xi_i : \mathcal{U}_i \rightarrow \mathcal{P}_i$), where $\mathcal{P}_i \subset \mathcal{M}$ and $\mathcal{U}_i \subset \mathbb{R}^n$. The manifold $\mathcal{M}$ is covered by the union of the overlapped $\mathcal{P}_i$, with $i = 1, .., |P|$. The $\mathcal{P}_i$ are called *patches* and $\mathcal{U}_i$ are the *parametric domains* of $\mathcal{M}$. Locally, $\mathcal{M}$ it is at least homeomorphic to $\mathbb{R}^M$ having an *intrinsic dimension* of M. The set of charts $\mathcal{A} = \{\zeta_i\}$ is called an *atlas*. See [4] for details. Fig. 1 (top) illustrates the main concepts used in the manifold learning algorithm. Thus, one innovation of this paper is the execution of the segmentation in (2) directly on the manifold $\mathcal{M}$. Fig.1 (bottom) illustrates the learned manifold for the LV segmentation. In this figure, each circle is a LV annotation after a PCA reduction.

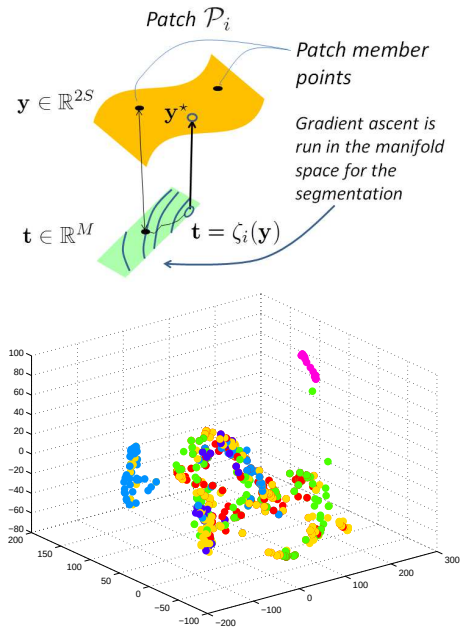

Fig. 1. (Top) Partition of the data (contours) into patches (only one patch is shown) and the corresponding tangent hyperplane. The forward-backward mappings are shown in black arrows. The black dots are the contours on the manifold (in orange) and their respective low dimensional representation (in green tangent space). (Bottom) Manifold obtained for the LV segmentation problem. The graph illustrates the patch member points after a PCA reduction (the first three dimensions are shown). Each color denotes the patch-members in a different patch.

5. TRAINING AND INFERENCE ON THE MANIFOLD USING DEEP BELIEF NETWORKS

The rigid detection classifier in (2) is modeled by the parameter vector γ_{MAP} (learned with a MAP learning algorithm), which means that $p(\mathbf{t}|\mathbf{x},\mathcal{D})$ is herein denoted by $p(\mathbf{t}|\mathbf{x},\gamma_{\text{MAP}})$. The parameter vector γ_{MAP} is estimated using a set of training samples taken from the *patch member* points $\mathbf{t}_{i,j} = \zeta_i(\mathbf{y}_{i,j})$ (for $j \in \{1, ..., |\mathcal{P}_i|\}$) of each learned patch $\mathcal{P}_i$ produced by the manifold learning algorithm. Specifically, the generation of positive and negative samples involves the following steps: 1) estimate the contour in the original image space from $\widehat{\mathbf{y}}_{i,j} = \zeta_i^{-1}(\mathbf{t}_{i,j})$; and 2) find the transformation matrix $\mathbf{A}_{\mathbf{t}_{i,j}}$ of the image window enclosing the segmentation contour $\widehat{\mathbf{y}}_{i,j}$ produced in step (1) (see Fig. 1 top).

For training the classifier, we take the patch members $\mathbf{t}_{i,j} \in \mathcal{M}$ to build the positives and negatives sets, as follows,

$$Dist(\mathcal{P}_i) = U(R(\mathbf{t}_{i,j})) \quad (3)$$

where U denotes an uniform distribution over the range R of the patch-member points in $\mathcal{P}_i$. More specifically, for the i-th patch the positives and negative are generated as follows,

$$\begin{aligned} \mathcal{P}os_{(i,j)} &= \left\{ \mathbf{t} \middle| \mathbf{t} \sim \text{Dist}(\mathcal{P}_i), |\mathbf{t} - \mathbf{t}_{i,j}| \prec \mathbf{r}_i \right\} \\ \mathcal{N}eg_{(i,j)} &= \Big\{ \mathbf{t} | \mathbf{t} \sim \text{Dist}(\mathcal{P}_i), |\mathbf{t} - \mathbf{t}_{i,j}| \succ 2 \times \mathbf{r}_i, \\ &\qquad \text{for all } j \in \{1, ..., |\mathcal{P}_i|\} \Big\} \end{aligned} \quad (4)$$

where $\mathbf{r}_i$ is the margin between positive and negative samples and where $|.|$ returns the absolute value of the difference. The samples drawn in (4) are used to learn the rigid classifier by maximizing the following cost function [14]

$$\phi_{\text{MAP}} = \arg\max_{\phi} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} \Big(\prod_{\mathbf{t} \in \mathcal{P}os_{(i,j)}} p(\mathbf{t}|\mathbf{x}, \phi) \Big) \times \Big(\prod_{\mathbf{t} \in \mathcal{N}eg_{(i,j)}} (1 - p(\mathbf{t}|\mathbf{x}, \phi)) \Big). \quad (5)$$

where $|\mathcal{P}_i|$ is the number of patch members in the i-th patch. For training the non-rigid classifier (see second term in (2)) we follow our previous work [15], that is

$$\psi_{\text{MAP}} = \arg\max_{\psi} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} \prod_{l=1}^{L} p(\mathbf{y}_{i,j}(l)|\mathbf{t}_{i,j}, \mathbf{x}, \psi) \quad (6)$$

where ψ represents the deep belief network (DBN) weights and $\mathbf{y}_{i,j}(l) \in [0, .., C]$, is the l-th orthogonal line of the contour represented by $\mathbf{y}_{i,j}$ with C-length. More specifically, $p(\mathbf{y}_{i,j}(l)|\mathbf{t}_{i,j}, \mathbf{x})$, is a regressor that receives as the input a profile of the image gray levels taken at the orthogonal line from each contour point $\mathbf{y}_{i,j}(l)$, and outputs an image location at that l-th orthogonal line. Notice that, the training strategy follows the same strategy as in [15] with the following key difference: the inference procedure to generate the segmentation contour in the image $\mathbf{x}$, takes each patch-member $\mathbf{t}_{i,j}$ from each learned patch $\mathcal{P}_i$ as an initial guess for the gradiente procedure on the output of the rigid classifier $p(\mathbf{t}|\mathbf{x}, \phi_{\text{MAP}})$ in the manifold $\mathcal{M}$. Whereas in [15], the initial guess of the gradient is taken at $\mathbf{t} \in \mathbb{R}^5$ that represents the parameters of an affine transformation that aligns the contour in a canonical coordinate system. This approach herein proposed has the advantage of providing $\mathbf{t}_{i,j} \in \mathbb{R}^M$, with the lower intrinsic M-dimensionality of the manifold[2].

For the gradient ascent, a number of iterations is used[3]. Once the gradient ascent is reached for each patch-member $\mathbf{t}_{i,j}$, the estimate $\widehat{\mathbf{y}}$ is obtained by the following Monte-Carlo approximation

$$\widehat{\mathbf{y}} = \frac{1}{Z} \sum_{i=1}^{|\mathcal{P}|} \sum_{j=1}^{|\mathcal{P}_i|} \mathbf{y}\, p(\widetilde{\mathbf{t}}_{i,j}|\mathbf{x}, \phi_{\text{MAP}})\, p(\mathbf{y}|\widetilde{\mathbf{t}}_{i,j}, \mathbf{x}, \psi_{\text{MAP}}) \quad (7)$$

estimate in the gradient process.

6. EXPERIMENTAL EVALUATION

In this section we illustrate the performance of the proposed approach targeted to the two goals mentioned in Section 1. To accomplish this, we provide a study concerning the use of a relatively small annotated training sets, providing segmentation results for several configurations of positive and negative sets. Also, we provide a comparison of running time figures with other methods to observe the running time improvement.

[2]In the experiments shown in Section 6 we obtained an intrinsic dimension of $M = 2$.

[3]In this work, and from the experiments conducted, we concluded that above five iterations no changes were observed.

6.1. Experimental setup

For the segmentation of the LV from US sequences [16], 14 sequences taken from 14 different subjects are considered. We worked with a cardiologist, who annotated 480 images in 14 sequences. We stress that the annotations in the training set contain the same number of keypoints, and that the base and apical points are explicitly identified in order for us to determine the rigid transformation between each annotation and the canonical location of such points in the reference patch.

The dimensionality of the representation for the LV contour is $S = 21$. The manifold learning algorithm produces: (i) $|\mathcal{P}| = 7$ patches, with a total of 496 patch member points, and (ii) $M = 2$ for the dimensionality of the rigid search space (i.e., the intrinsic dimensionality of the manifold). It is worth mentioning that the original dimensionality of the rigid search space is $R = 5$ (representing two translation, one rotation and two scale parameters), which is the dimensionality usually found in current state-of-the-art methods [17, 11, 13, 3]. See Fig. 1 (bottom) for an illustration of the results of our manifold learning algorithm on the LV segmentation problem. In this figure, the circles (each color for each patch-member point) are the annotations after PCA reduction.

The training and inference methods used in this paper follows a coarse-to-fine rigid detector $p(\mathbf{t}|\mathbf{x}, \gamma_{\text{MAP}})$ and a non-rigid classifier $p(\mathbf{y}|\mathbf{t}, \mathbf{x}, \lambda_{\text{MAP}})$ based on deep belief networks (DBN) [14]. The key issue is the use of low-dimensional manifolds to represent the rigid detection space, which means that we re-trained the coarse-to-fine rigid detector to run on the learned manifold. Also, the proposed classifier is estimated using training sets of different sizes, where the number of additional training samples can be reduced with the use of our low-dimensionality manifold. Specifically, we vary the size of the set of positive samples by varying the number of additional positive and negative samples per training image, as follows $|\mathcal{P}os(p, j)| \in \{1, 5, 10, 15, 20\}$, and the size of negative samples as $|\mathcal{N}eg(p, j)| \in \{10, 50, 100, 150, 200\}$, as explained in (4).

6.2. Experimental results and comparison evaluation

The performance of our approach is assessed with a quantitative comparison over the test sets. The experiments are divided in two main parts: (i) we perform a n-fold validation over the 14 sequences, and (ii) we perform a comparison with state-of-the-art methods, e.g. [11, 13] and [17]. To accomplish the later, we train our method in 12 sequences performing the test in the remaining two test sequences, termed herein as $\mathcal{T}_1$ and $\mathcal{T}_2$. This is because we have available the segmentation results of the above state-of-the art methods in these two sequences.

Quantitative comparison is performed using (i) Jaccard distance (JCD) and (ii) average distance (AV) [16].

We start by illustrating an example of a test frame segmentation provided by our multi-atlas approach in Fig. 2. It is shown the segmentation of each patch found in the manifold along with the confidences given by the DBN. In the left most image, we see that the segmentation estimated by this patch is a bit under the apex location. In the right most image, the left side of the contour and the base points are slightly above the

correct locations. In these two situations the DBN correctly provides low confidences. The other two segmentations are accurate and receive high confidence of the DBN providing, in this way, accuracy robustness in the final segmentation, that is accomplished using (7).

Fig. 3 shows the error metric statistics of the segmentations. In the first row, each box-plot is a mean value over the segmentations performed in the n-fold validation. From the figure, remarkable performance is achieved. The Jaccard distance seems the metric that most distinguish the results. For all configurations of the training sets the performance is similar, being the best result below 0.20 (see configuration $\{20, 200\}$).

The $2nd$ and the $3rd$ rows of the Fig. 3 shows a comparison with [11, 13] and [17] approaches using the two measures already mentioned. For this comparison, we also varied training set sizes as in the previous experiment. In this figure, we observe that all the methods exhibit a $d_{\text{JCD}} < 0.2$ and $d_{\text{AV}} < 4$ pixel for the test sequence $\mathcal{T}_1$ (see top row). For the second test sequence ($\mathcal{T}_2$), only the method in [17] is able to produce $\text{JCD} < 0.2$ and all the methods provide an average error $\text{AV} < 4$ (see bottom row and for the configuration $\{20, 200\}$). We stress that, however, this comparison is a bit unfair since in [17] a dynamical model is used, whereas our framework is purely for segmentation without any time information.

Fig. 4 shows a qualitative comparison displaying the segmentation results from our method and also from [11, 13] and [17] illustrating some snapshots of the LV test sequences.We compare the running time figures of our approach with [17]. The mean running time of our approach for both sequences is 2.77 sec., while for [17] is 7.4 seconds. From the obtained running time of the proposal, 2.60 sec. is needed for the rigid part. The non-rigid part takes 0.17 sec. All these running time figures were obtained with unoptimized Matlab implementations on a computer with the following configuration: Intel Core i5, with $4GB$ of RAM.

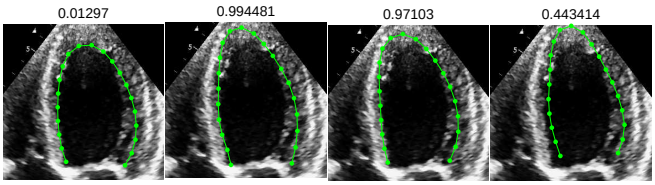

Fig. 2. Muti-atlas segmentation illustrating each patch proposal segmentation along with the confidence provided by the deep belief network. From the 7 patches in the manifold (see Section 6.1), it is shown the results in four patches.

7. CONCLUSIONS

We presented a multi-atlas segmentation methodology for non-rigid object segmentation. The methodology combines the deep learning architecture with the use of manifold learning. The main contribution and focus of the paper is the dimensionality reduction of the segmentation contour parametrization for the rigid segmentation stage. A manifold learning based approach has been proposed and allows to reduce the dimension of the rigid space. Each patch in the manifold proposes a segmentation of the LV, that is further fused using the confidences of the deep belief network through monte carlo approximation. This allow us to conclude, that the framework provides faster running time in both training and segmentation stages. This is because, the training and parameters search are both reformulated directly in terms of the manifold parametrization at low dimensionality.

Further directions will include a dynamical model in the manifold, where the object dynamics is learned directly in the low dimensional manifold parameter space. This will allow for a reduction of the computational cost in the prediction step. As explained above we also plan to parallelize the segmentation process given that the landmarks represent independent initial guesses for the search process.

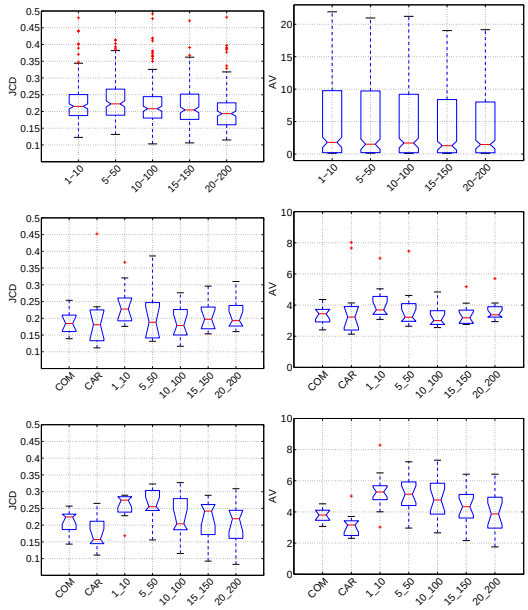

Fig. 3. (Top row) N-fold validation of the proposed approach in 14 LV sequences. Comparison with state-of-the-art for the test sequence $\mathcal{T}_1$ (middle row) and for the test sequence $\mathcal{T}_2$ (bottom row). Comparison with COM (approaches in [11, 13]) and CAR (method in [17]) are also shown.

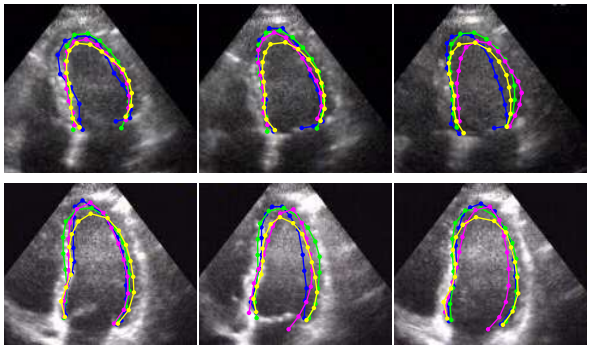

Fig. 4. Qualitative comparison between the expert annotation (GT in blue) and the results of our approach (green), [17] (yellow), and [11, 13] (purple). The results show the segmentations for the test sequence $\mathcal{T}_1$ (top row) and for $\mathcal{T}_2$ (bottom row).

8. REFERENCES

[1] Yiqiang Zhan, Xiang Sean Zhou, Zhigang Peng, and Arun Krishnan, "Active scheduling of organ detection and segmentation in whole-body medical images," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pp. 313–321. Springer, 2008.

[2] Shaoting Zhang, Yiqiang Zhan, Maneesh Dewan, Junzhou Huang, Dimitris N Metaxas, and Xiang Sean Zhou, "Towards robust and effective shape modeling: Sparse shape composition," *Medical image analysis*, vol. 16, no. 1, pp. 265–277, 2012.

[3] S. K. Zhou, "Shape regression machine and efficient segmentation of the left ventricle endocardium from 2-D B-mode echocardiogram," *Med. Imag. Analysis*, , no. 14, pp. 563–581, 2010.

[4] J. C. Nascimento and J. G. Silva, "Manifold learning for object tracking with multiple motion dynamics," in *ECCV*, 2010.

[5] T. Rohlfing, R. Brandt, R. Menzel, and C.R. Maurer, "Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains," *Neuroimage*, vol. 21, no. 4, pp. 1428–42, 2004.

[6] S. K. Warfield, K. H. Zou, and W. M. Wells, "Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation," *IEEE Trans. Med. Imag.*, vol. 23, no. 7, pp. 903–921, 2004.

[7] R. A. Heckemann, J. V. Hajnal, P. Aljabar, D. Rueckert, and A. Hammersc, "Automatic anatomical brain mri segmentation combining label propagation and decision fusion," *Neuroimage*, vol. 33, pp. 115–126, 2006.

[8] M. Wu, C. Rosano, P. Lopez-Garcia, C.S. Carter, and H.J. Aizenstein, "Optimum template selection for atlas-based segmentation," *Neuroimage*, vol. 34, no. 4, pp. 1612–1618, 2007.

[9] R. Wolz, P. Aljabar, J.V. Hajnal, A. Hammers, and D. Rueckert, "Leap: Learning embeddings for atlas propagation," *Neuroimage*, vol. 49, no. 2, pp. 1316–1325, 2010.

[10] A.K. Hoang Duc, M. Modat, K.K. Leung, M.J. Cardoso, J. Barnes, T. Kadir, and S. Ourselin, "Using manifold learning for atlas selection in multi-atlas segmentation," *PLOS one*, vol. 8, no. 8, pp. 1–15, 2013.

[11] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta, "Databased-guided segmentation of anatomical structures with complex appearance," in *CVPR*, 2005.

[12] 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.

[13] 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.

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

[15] G. Carneiro and J. C. Nascimento, "Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, no. 11, pp. 2592–2607, 2013.

[16] 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.

[17] 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 *CVPR*, 2010, pp. 2815–2822.