Deep Learning on Sparse Manifolds for Faster Object Segmentation

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

This work was supported by the FCT project [PEst-OE/EEI/LA0009/2013] by project 'HEARTTRACK' - PTDC/EEA-CRO/098550/2008. This work was partially supported by the Australian Research Council's Discovery Projects funding scheme (project DP140102794).

Jacinto C. Nascimento (corresponding author) is with the *Instituto de Sistemas e Robótica*, *Instituto Superior Técnico*, 1049-001 Lisboa, Portugal. Email: `jan@isr.ist.utl.pt`. **Phone:** +351-218418270, **Fax:** +351-218418291.

Abstract

We propose a new combination of deep belief networks and sparse manifold learning strategies for the 2D segmentation of non-rigid visual objects. With this novel combination, we aim to reduce the training and inference complexities while maintaining the accuracy of machine learning based non-rigid segmentation methodologies. Typical non-rigid object segmentation methodologies divide the problem into a *rigid detection* followed by a *non-rigid segmentation*, where the low dimensionality of the rigid detection allows for a robust training (i.e., a training that does not require a vast amount of annotated images to estimate robust appearance and shape models) and a fast search process during inference. Therefore, it is desirable that the dimensionality of this rigid transformation space is as small as possible in order to enhance the advantages brought by the aforementioned division of the problem. In this paper, we propose the use of sparse manifolds to reduce the dimensionality of the rigid detection space. Furthermore, we propose the use of deep belief networks to allow for a training process that can produce robust appearance models without the need of large annotated training sets. We test our approach in the segmentation of the left ventricle of the heart from ultrasound images and lips from frontal face images. Our experiments show that the use of sparse manifolds and deep belief networks for the rigid detection stage leads to segmentation results that are as accurate as the current state of the art, but with lower search complexity and training processes that require a small amount of annotated training data.

I. Introduction

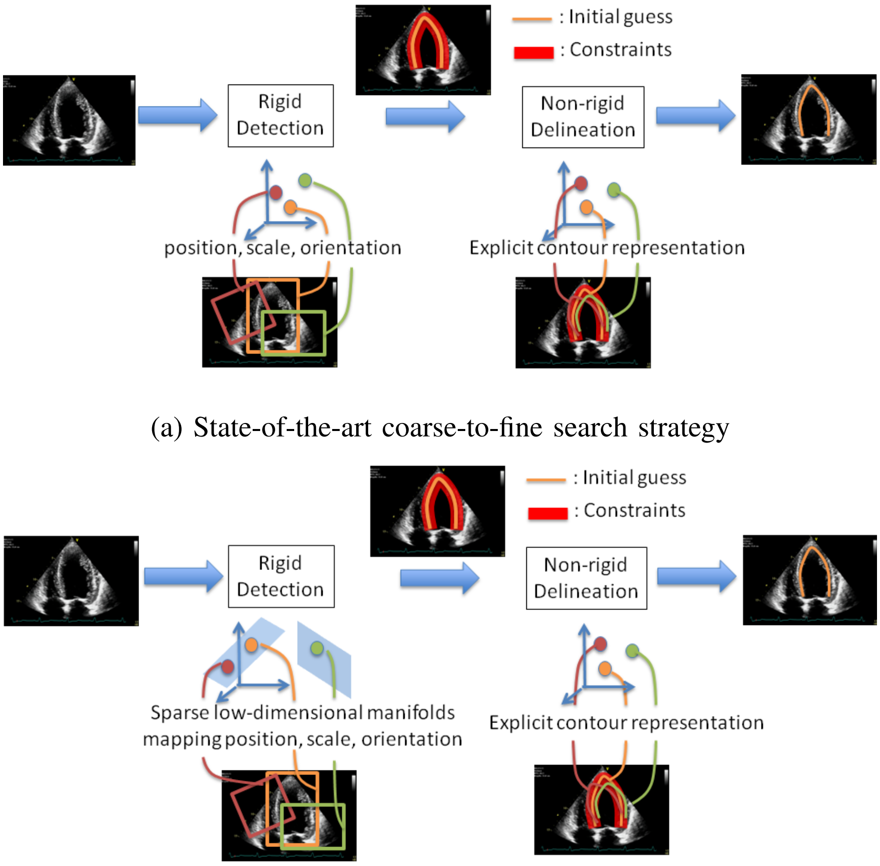
Current methodologies for top-down segmentation of deformable objects using machine learning techniques address the learning and inference tasks with a coarse-to-fine strategy based on the following two consecutive stages [1]–[6]: (i) rigid detection and (ii) non-rigid segmentation. The rigid detection (i.e., coarse step) produces the rotation, scale and translation of the visual object, which are used to initialize and constrain the non-rigid segmentation stage (i.e., fine step). Assuming that the contour of the visual object is represented by S keypoints (or S 2-D points) and the rigid detection is performed in a space with $R << 2S$ dimensions, then the introduction of this coarse step allows for a more efficient inference and less complex training processes.

The improvement in the inference process efficiency stems from the following two facts: 1) faster search in the R-dimensional rigid space (compared to the original S-dimensional non-rigid space) because of the much smaller dimensionality of R; and 2) efficient fine step in the $2S$-dimensional non-rigid space given the initial guess and constraint produced by the rigid detection step (see Fig. 1). The smaller training complexity is achieved because the R-dimensional rigid problem requires smaller training sets and the training for the non-rigid segmentation in the 2S-dimensional space is also simplified because of the constraints produced by the rigid detection stage. Note however, that this strategy imposes strong requirements on the rigid detector, in the sense that it has to be efficient and robust to the appearance and shape variations of the visual object of interest and the size of the training set. The efficiency of this detector depends mainly on the dimensionality of the rigid search space (i.e., lower dimensionality leads to more efficient rigid detectors) and robustness also depends on this dimensionality (for effectively modeling the shape variations), but also depends on the ability of the classifier to model the appearance of the object using a limited number of annotated images. It is important to note that the usual solution to increase the robustness of the classifier when the number of annotated images is small is to increase the training set by artificially perturbing these training images and annotations (e.g., by adding image noise or applying small rigid transformations) in order to generate new images to be added to the training set. However, given the random nature of this perturbation, it is not possible to guarantee whether the generated image can actually exist in practice, which ultimately can lead to ineffective classification problems.

This paper introduces a rigid search space of very low dimensionality with the use of sparse manifolds, where the problem of classifier robustness is dealt with the use of deep learning mechanisms, which has shown unique robustness particularly with respect to the size of training set. More specifically, we propose the use of sparse manifolds with low intrinsic dimensionality for the rigid detection stage [1,2,5]–[7], which allows for a faster inference process that produces competitive segmentation results. Another aspect of our framework, is that by restricting the positive and negative samples to lie in the learned low-dimensional sparse manifold, it is possible to reduce significantly the need for additional artificial positive and negative samples during the training process, and at the same time guarantee that the additional samples are more likely to exist in practice. Consequently, this produces a less complex and faster training process that is as robust to shape and appearance variations as the current state of the art.

We illustrate the performance of the proposed low-dimensional rigid search space using sparse manifolds approach in two different segmentation problems: the left ventricle (LV) endocardium segmentation from ultrasound (US) imagery and lip segmentation using the extended Cohn-Kanade dataset (CK+) [8] consisting of several facial expressions from frontal views. Note that all datasets presented in the paper share the conditions where the object of interest undergoes a rigid transformation followed by a non-rigid deformation. Also note that we are interested in segmenting the object using an explicit representation, where neighboring keypoints in the segmentation are strongly correlated.

We demonstrate that our framework reduces the search complexity without a negative impact on the


: Initial guess
: Constraints
Rigid Detection
Non-rigid Delineation
position, scale, orientation
Explicit contour representation
: Initial guess
: Constraints
Rigid Detection
Non-rigid Delineation
Sparse low-dimensional manifolds mapping position, scale, orientation
Explicit contour representation

(a) State-of-the-art coarse-to-fine search strategy

(b) Proposed coarse search using sparse manifolds in the rigid detection

Fig. 1. (a) Illustration of the two-stage strategy for the non-rigid segmentation used in the state-of-the-art methodologies. (b) Proposed methodology, where the sparse manifold is used in the rigid detection step.

segmentation accuracy, when compared to the state of the art. Moreover, we also show that the our proposed low-dimensional sparse manifolds allows for the use of smaller training sets than the current state-of-the-art methods.

II. Literature Review

The segmentation of non-rigid visual objects is perhaps one of the most studied problems in the field of computer vision. In this literature review, we classify the proposed methodologies as follows: bottom-up approaches [9,10], active contour methods [11]–[23], deformable templates [24]–[29], and data-driven segmentation [6,30]–[48]. The vast majority of the these methodologies breaks down the non-rigid segmentation into two sub-problems, comprising a first stage that selects the location (and usually the scale and orientation too) of the sought visual object, followed by a second stage that searches for the boundary of the object given the information produced in the first stage. We call such methodologies *coarse-to-fine*, where the coarse step consists of the rigid detector and the fine stage comprises the constrained non-rigid segmentation, as explained in Section I. Recently, the non-rigid segmentation problem has avoided the coarse stage altogether by addressing the task either as a *structured learning and inference problem* [49,50] or as a *convex active contour* method [13]. In the structured inference problem the input image is represented with a graph combining multiple bottom-up and top-down functions; while in convex active contour methods, the main idea is to convexify the level set energy function, which means that it no longer depends on the initial guess provided by the coarse stage.

Classic bottom-up methods [9,10] are based on a coarse-to-fine methodology, where the coarse step is usually represented with a manual initialization, which is followed by a series of standard image processing techniques to detect the border of the sought object. In general, these image processing techniques only take into account low-level image information, such as edges, texture and colour, and use simple prior information, such as boundary smoothness and continuity. The simplicity of the techniques make these approaches attractive from a computational complexity point of view, but the lack of high-level information about the visual object and the dependence on a good initialization (from the coarse step) make these approaches too sensitive to imaging conditions and to the appearance and shape variations of the sought visual object.

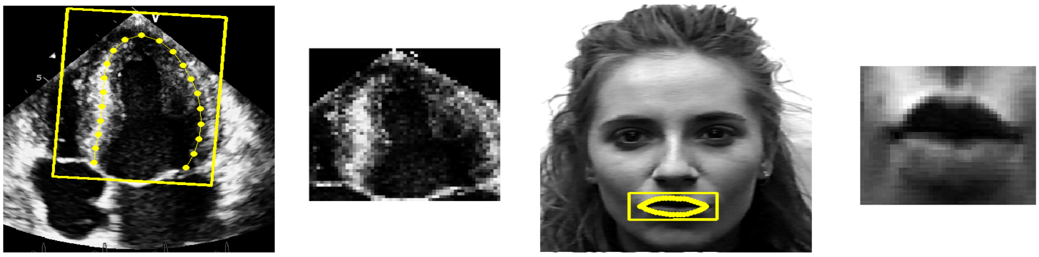
Active contour methods [17] improved the robustness of segmentation algorithms to imaging conditions and to the variations of the visual object by formulating the problem with a unified energy function that could be minimized with standard optimization methods. The development of level-set methods [20] improved the performance of active contours with respect to imaging conditions and visual object topology. We refer to such methods as coarse-to-fine non-convex active contours, since their energy function is not convex, and depends strongly on good initial conditions that are usually provided manually during the coarse step. The latest developments of these approaches have been focused on increasing the robustness of the method with the integration of region and boundary segmentation, reduction of the search dimensionality, modeling of the implicit segmentation function with a continuous parametric function, and the introduction of shape and texture priors [11,12,14]–[16,18,19,21]–[23]. The convexification of the energy function used in active contour methods has been a central topic of research in the field [13], which allows for more efficient optimization methods in addition to the lack of need of manual initialization (i.e., the coarse step is no longer needed). Nevertheless, these convex active contour methods can only avoid the coarse search step when the visual object of interest has strong priors in terms of texture, shape and rigid transform, which may not be the case for some examples (see Fig. 2-a) and the search process will not be able to extend much from these priors. Deformable templates [24]–[29] introduce the use of more specific prior models about the shape and appearance of the visual object of interest with the goal of deforming this prior model to match the test image. Similarly to the case of non-convex level sets, this approach also needs a coarse step comprising a good initialization for the optimization process. Level-sets and deformable templates are among the most successful techniques applied in non-rigid segmentation problems, but their main weakness is the strong prior knowledge defined in the optimization function, such as the definition of the object border, the prior shape, the prior distribution of the texture or gray values, or the shape variation. This prior knowledge can be either designed by hand or learned using a (usually) small training set. As a result, the effectiveness of such approaches is limited by the validity of these prior models, which are unlikely to capture all possible shape variations and nuances present in the imaging of the visual object [37].

These issues are the main motivation of data-driven binary segmentation methods, where the shape and appearance of the LV is fully learned from a manually annotated training set. Active shape and appearance models [31]–[33,39] are usually based on optimization methods of an energy functional composed of shape and appearance terms, represented by generative classifiers learned using a manually annotated training set. The use of discriminative classifiers has also been explored in data-driven binary segmentation methods [6,37]. The commonality between these two approaches is the use of a coarse-to-fine search, with the coarse stage represented by a search for the rigid transform of the mean shape of the sought visual object, which is followed by a fine stage that transforms the mean shape in a non-rigid way to match visual object in the test image. In general, the coarse stage must efficiently provide a precise rigid transformation, so there has been a large number of papers about effective coarse search strategies. Exploring the whole rigid transform space is in general intractable, so the main idea is to progressively constrain this search space. The cascade classifier [51] does that by firstly exploring the entire search space with highly robust low-complexity classifiers, and then further testing the regions that survived that previous steps with increasingly more complex classifiers. A similar approach is followed in [52], that imposes a prior distribution on the initial search space, which is used to sample the initial search locations that are refined based on a gradient-based search. Another related approach is the marginal space learning [41], which partitions the search space into sub-spaces of increasing dimension and complexity. The branch and bound approach for the coarse search step [53] is another way to progressively reducing the complexity of the search space.

The recent development of structured learning and inference methods [49,50] allowed the design of convex data-driven binary [30,38] and multi-class [42]–[48] segmentation methods. The main potential advantage of such approaches lies in their ability to avoid the coarse search step because the structured inference is designed as a convex problem independent of the initial guess. However, as explained above for convex level set methods, this advantage can be realized only if the visual object of interest can be reasonably characterized by strong priors in terms of appearance, shape, rigid transformation, etc. An alternative usually followed by state-of-the-art structured inference and learning methods is the integration of the result of coarse visual object detectors into the framework, which effectively means that most of the methods above run a coarse search step.

Another relevant point of our proposal, is the gradient based search in manifolds, which have also been studied in other works. For instance, Helmke et al. [54] have introduced a new optimization approach for the essential matrix computation with the use of Gauss-Newton iterations. Hüper et al. [55] also propose a numerical optimization of a cost function defined on a manifold. Similarly, the use of Newton's method along the geodesics and variants of projections have also been proposed by other authors [56]–[58]. Our approach represents an application of such gradient-based search methods in the problem of top-down non-rigid segmentation with the specific goals of reducing the search running time and the training complexity.

Finally, sparse manifold learning is another topic visited by our proposal. This basically involves the estimation of a low-dimensional representation of a data set using a small number of observations [59,60]. One popular technique for finding a sparse representation is the *Matching Pursuit* (MP), which is based on a suboptimal forward sequential algorithms [61]–[64]. Other techniques are based on optimization methodologies that maximize sparsity, such as the ℓ_1 norm [65,66], or the more general $\ell_{(\rho\leq 1)}$ explored by FOCal Underdetermined System Solver (FOCUSS). Techniques tailored to be applied in the context of noisy data have also been proposed, such as a robust version of the FOCUSS algorithm, called *Regularized* FOCUSS, that can also be used as an efficient representation for compression [67]. Other important variation of the sparse linear inverse problem is the multiple-measurement vector (MMV) that achieves sparse representations from single-measurement vectors (SMVs) [68]. Recent theoretical studies focus on the convex relaxation of the MMV such as the approach based on the (ℓ_2,ℓ_1) norm minimization [60,69]–[71]. A similar relaxation technique (via the ℓ_1 norm minimization) is employed in the SMV model, but efficient MMV methods for sparse representation have been proposed, in which some known results of SMV are generalized to MMV. The sparse manifold learning proposed in this paper is inspired on our previous work [72], which introduces a manifold learning method that requires a large number of samples that leads to an inference lacking efficiency because each sample would need to be used as an



(a) Left ventricle segmentation

(b) Lip segmentation

Fig. 2. Application of the transformation $\mathbf{A_t}$ to the window enclosing the mean segmentation contour for the case of (a) left ventricle segmentation, and (b) lip segmentation. Both figures depict the explicit segmentation contour with the rectangular window (left panel) and zoomed in image of the visual information within the window (right panel). Note that the images on the right panels are the ones used by the rigid classifier $p(\mathbf{t}|\mathbf{x}, \mathcal{D})$ in (2).

initial guess to a gradient-based search in the manifold. In this paper, we introduce a learning approach that requires a small number of observations, leading to our efficient search mechanism [73]. In fact, this paper represents an extension of [73], where in this submission we provide a more comprehensive literature review and more detailed explanations of the methodology. More specifically, we present a proposal, where patch members are obtained from the manifold - this forms the "baseline" version of the proposal. Then, we describe how sparsity is promoted in the manifold. Both versions (baseline and sparsity solutions) are theoretically described and a systematic comparison between them is conducted for several datasets.

III. Non-rigid Top-down Segmentation Problem Definition

We start by considering an image that contains the sought object to be segmented. The goal is to produce a non-rigid segmentation $\mathbf{y} \in \mathbb{R}^{2S}$ containing S 2-D points, that constitutes the explicit representation of the segmentation contour. Let us 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:

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

The high dimensionality of $\mathbf{y}$ makes the computation of (1) difficult, and the usual solution to alleviate the problem is the introduction of preliminary coarse search steps 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 [1,2,5,6]:

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

In practice, the variable $\mathbf{t}$ is used to transform linearly the coordinates of a window that encloses the mean segmentation contour (see Fig. 2). This linear transform is obtained from the variable $\mathbf{t}$ that forms $\mathbf{A}_{\mathbf{t}} \in \mathbb{R}^{3\times 3}$ [1,2,5,6]. For example, suppose $\mathbf{t} = [x, y, \vartheta, \nu_x, \nu_y]$ denotes a transformation comprising a translation x and y, rotation ϑ, and non-uniform scaling ν_x and ν_y; then

$$\mathbf{A}_{\mathbf{t}} = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_x & 0 & 0 \\ 0 & \nu_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Hence the term $p(\mathbf{t}|\mathbf{x}, \mathcal{D})$ in (2) represents the rigid detection classifier that outputs the probability of having the sought visual object within the boundaries of the window transformed by $\mathbf{t}$. The term $p(\mathbf{y}|\mathbf{t}, \mathbf{x}, \mathcal{D})$ in (2) is the non-rigid segmentation classifier denoted by the probability of finding the contour $\mathbf{y}$ in image $\mathbf{x}$ given the value of $\mathbf{t}$. That is, $\mathbf{t}$ 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 $r = R$, one of the objectives of this paper is the introduction of a new space for $\mathbf{t}$ with dimension $r = M < R$, based on a sparse manifold, where this rigid search will take place with gradient descent search mechanisms. Before discussing this search mechanisms, we describe the sparse manifolds developed for this paper in Section IV.

IV. Sparse Manifolds

This section describes the learning of the sparse manifold representation, and the inference used in the coarse search mechanism (i.e., the rigid detection described in (2)). The manifold learning strategy takes as input the training annotations $\{\mathbf{y}_j\}_{j=1}^{|\mathcal{D}|}$ that belong to the training set $\mathcal{D}$, and produces the manifold $\mathcal{M} \in \mathbb{R}^M$ with intrinsic dimension M (with $M < R << 2S$) divided into patches $\{\mathcal{P}_i\}_{i=1}^{|\mathcal{P}|}$ (with $\mathcal{P}_i \subset \mathcal{M}$), each one containing its respective *chart* $\zeta_i : \mathcal{P}_i \to \mathcal{U}_i$, and *parametrization* $\xi_i : \mathcal{U}_i \to \mathcal{P}_i$, where $\mathcal{U}_i \subset \mathbb{R}^M$ denotes the *parametric domain*. Our learning method also produces the tangent hyperplanes $T_{\mathcal{P}_i}$ for each patch, which is formally defined as $T(\mathcal{P}_i) = \{(\mathbf{y}, \mathbf{v}) : \mathbf{y} \in \mathcal{P}_i, \mathbf{v} \in T_{\mathbf{y}}(\mathcal{P}_i)\}$ where $T_{\mathbf{y}}(\mathcal{P}_i)$ is the tangent space of $\mathcal{P}_i$ at observation $\mathbf{y}$. According to this algorithm, each patch $\mathcal{P}_i$ is represented by

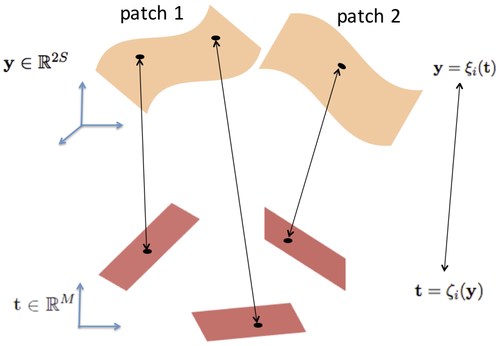

patch 1
patch 2
$\mathbf{y} \in \mathbb{R}^{2S}$
$\mathbf{y} = \xi_i(\mathbf{t})$
$\mathbf{t} \in \mathbb{R}^{M}$
$\mathbf{t} = \zeta_i(\mathbf{y})$

Fig. 3. Partition of the manifold into patches (top) and the corresponding tangent hyperplanes (bottom). The arrows illustrate the mappings back and forth between the patches and the hyperplanes. The black dots are the annotations on the manifold and their respective low dimensional representation.

$|\mathcal{P}_i|$ samples drawn from the training set $\mathcal{D}$, where the $|\mathcal{P}_i|$ points belonging to patch $\mathcal{P}_i$ are known as the *patch member* points, and in general $|\mathcal{P}_i| \neq |\mathcal{P}_j|$ for $i \neq j$.

One of the innovations of this paper is the execution of the rigid detection in (2) directly on the manifold $\mathcal{M}$. This is accomplished by performing the optimization process in each of the low dimensional patches $\mathcal{P}_i$ with initial guesses (for the segmentation process in (2)) taken from the patch member points $\mathbf{t}_{i,j} = \zeta_i(\mathbf{y}_{i,j})$, for $i \in \{1, ..., |\mathcal{P}|\}$ and $j \in \{1, ..., |\mathcal{P}_i|\}$. Consequently, the efficiency of the segmentation depends on a low number of patch member points in each patch. For completeness of the exposition, we provide details of the manifold learning algorithm in the Appendix.

A. Subset Selection - Problem Statement[1]

In order to reduce the number of patch member points in each patch $\mathcal{P}_i$, let us first arrange the training annotations in the following matrix $\mathbf{Y}_i \in \mathbb{R}^{2S \times |\mathcal{P}_i|}$ (each column containing a contour), with

$$\mathbf{Y}_i = [\mathbf{y}_{i,1}, ..., \mathbf{y}_{i,|\mathcal{P}_i|}], \tag{4}$$

where the charting process generates the matrix $\mathbf{T}_i = [\mathbf{t}_{i,1}, ..., \mathbf{t}_{i,|\mathcal{P}_i|}]$ with $\mathbf{T}_i \in \mathbb{R}^{M \times |\mathcal{P}_i|}$, i.e. the low dimensional representations of the annotations. The reduction in the number of patch member points involves the selection of a small number of columns in $\mathbf{Y}_i$ (and thus a subset of columns in $\mathbf{T}_i$) to be used as *landmarks*. These columns are selected by minimizing the amount of information lost with

[1] All the exposition formulated in this Sections IV-A, IV-B, IV-C, are in terms of the i^{th} patch, being the same strategy applied to other patches in the manifold.

respect to ζ_i, but note that preserving the chart ζ_i is equivalent to preserve its inverse mapping, i.e. the parameterization ξ_i, which is more practical to use since we can use the following generative model

$$\widehat{\mathbf{y}}_{i,j} = \xi_i(\mathbf{t}_{i,j}) + \boldsymbol{\omega}, \tag{5}$$

where $\widehat{\mathbf{y}}_{i,j}$ is an approximation of $\mathbf{y}_{i,j}$ and $\boldsymbol{\omega}$ is a Gaussian random variable representing noise. Our main goal in this context is to design a method that estimates the fewest number of landmarks so that $\mathbf{y}_{i,j}$ is reliably approximated by $\widehat{\mathbf{y}}_{i,j}$ in (5).

B. Linear regression

To accomplish the goal formulated above in Sec. IV-A, we start by building the radial basis function (RBF) kernel matrix $\mathbf{K}$, with each of its elements represented by $k(\mathbf{t}_{i,l}, \mathbf{t}_{i,q}) = \exp(-\|\mathbf{t}_{i,l} - \mathbf{t}_{i,q}\|^2/2\sigma_K)$ and reformulate (5) as a single measurement vector problem (SMV) [73,74], as follows[2]

$$\boldsymbol{\theta} = \mathbf{K}\boldsymbol{\beta} + \eta, \tag{6}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{P}_i|}$ represents the vector containing the maximum principal angle between the tangent bundles $T_{\mathbf{y}_{i,j}}$ and $T_{\mathcal{P}_i}$ [75] [3], $\boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{P}_i|}$ denotes the vector of coefficients for reconstructing the input data $\boldsymbol{\theta}$, and $\eta \in \mathbb{R}^{|\mathcal{P}_i|}$ is a random variable representing the additive Gaussian noise process.

An interpretation of the regression in (6) is that $\boldsymbol{\beta}$ preserves angular information within a patch, using a small number of points, and $\mathbf{K}$ preserves information regarding the distance between points. Thus, points with similar angular or distance information are included in the regression.

C. Sparsity with Least Angle Regression

In order to select a small subset of the patch member points of $\mathcal{P}_i$ given $\boldsymbol{\theta}$, we estimate $\boldsymbol{\beta}$ in (6), denoted by $\widehat{\boldsymbol{\beta}}$, constraining it to be sparse via a regularization term. More specifically, the estimate $\widehat{\boldsymbol{\beta}}$ can be found by minimizing the following expected generalization error

$$\mathbb{E}(\widehat{\boldsymbol{\beta}}) = \left\|\boldsymbol{\theta} - \mathbf{K}\widehat{\boldsymbol{\beta}}\right\|^2, \tag{7}$$

defining the absolute norm of $\widehat{\boldsymbol{\beta}}$ as $T(\widehat{\boldsymbol{\beta}}) = \sum_{j=1}^{|\mathcal{P}_i|} |\widehat{\boldsymbol{\beta}}_j|$, the minimization of $\mathbb{E}(\widehat{\boldsymbol{\beta}})$ subject to a bound t on $T(\widehat{\boldsymbol{\beta}})$ can be solved as follows

$$\text{minimize} \quad \mathbb{E}(\widehat{\boldsymbol{\beta}}) \quad \text{subject to} \quad T(\widehat{\boldsymbol{\beta}}) \leq t, \tag{8}$$

[2] In the following equations (6),(7), (9), and (10) we have omitted the subscript i for simplifying the notation.

[3] See the Appendix for additional details regarding the computation of principal angle. Also, note that $T_{\mathbf{y}_{i,j}}$ is the tangent subspace of $\mathbf{y}_{i,j}$ (the jth column of $\mathbf{Y}$ in (4)) and $T_{\mathcal{P}_i}$ is the tangent subspace computed in the seed point of the ith patch.

which is solved with *least angle regression* (LARS) [76]. Basically, the algorithm starts with the zero vector, $\widehat{\boldsymbol{\beta}} = \mathbf{0}$, and adds covariates (i.e., the columns of $\mathbf{K}$) to the model in accordance to their correlation with the prediction error vector, $\|\boldsymbol{\theta} - \mathbf{K}\widehat{\boldsymbol{\beta}}\|^2$ in (7), setting the corresponding jth entry, $\widehat{\boldsymbol{\beta}}_j$, to a value such that another covariate becomes equally correlated with the error and is, itself, added to the model. The LARS algorithm then proceeds in a direction equiangular to all the added $\widehat{\boldsymbol{\beta}}_j$ and the process is repeated until all covariates have been added. This strategy of adding a new $\widehat{\boldsymbol{\beta}}_j$ (making it non-zero), requires an amount of m steps (each step adding a new $\widehat{\boldsymbol{\beta}}_j$ and making it non-zero). It has been shown [76] that the risk (i.e. the structural risk equivalent to the expected error in (7)) can be estimated as

$$\mathcal{R}(\widehat{\boldsymbol{\beta}}_p) = \|\boldsymbol{\theta} - \mathbf{K}\widehat{\boldsymbol{\beta}}\|^2 / \sigma_\theta^2 - m + 2p. \tag{9}$$

where σ_θ can be computed from the unconstrained least squares solution of (6), m is the number of steps (i.e. dimension of $\boldsymbol{\theta}$) and p is the number of non-zero entries of $\widehat{\boldsymbol{\beta}}_j$.

The landmarks are the columns $\mathbf{t}_{i,j}$ of $\mathbf{T}_i$ (or equivalently of $\mathbf{Y}_i$) with the same indexes j as the non-zero elements of $\widehat{\boldsymbol{\beta}}_p$ such that

$$p^* = \arg\min_p (\mathcal{R}(\widehat{\boldsymbol{\beta}}_p)). \tag{10}$$

The estimated $L_i = p^*$ landmarks correspond to p^* non-zero elements in $\widehat{\boldsymbol{\beta}}_p$. This strategy ensures that the landmarks are the kernel centers that minimize the risk of the regression formulated in (6).

V. Training and Inference on the Sparse Manifold using Deep Belief Networks

The rigid detection classifier in (2) is modeled by the parameter vector γ_{MAP} (learned with a maximum a posteriori learning algorithm), which means that $p(\mathbf{t}|\mathbf{x}, \mathcal{D})$ is hereafter represented 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 $\mathcal{P}_i$ (for $i \in \{1, ..., |\mathcal{P}|\}$) 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 the landmark, $\widehat{\mathbf{y}}_{i,j} = \zeta_i^{-1}(\mathbf{t}_{i,j})$; 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).

For training the classifier, the sets of positives and negatives are formed by sampling a distribution of the patch members $\mathbf{t}_{i,j}$. The distribution in patch $\mathcal{P}_i$ is defined by

$$\text{Dist}(\mathcal{P}_i) = U(\mathbf{T}_i), \tag{11}$$

where $U(\mathbf{T}_i)$ denotes an uniform distribution in the interval $[\max_{row}(\mathbf{T}_i) - \min_{row}(\mathbf{T}_i)] \in \mathbb{R}^M$ with $\mathbf{T}_i$ being a matrix whose columns contain the patch members $\mathbf{t}_{i,j} \in \mathcal{P}_i$; the functions $\max_{row}(\mathbf{T}_i) \in \mathbb{R}^M$

and $\min_{row}(\mathbf{T}_i) \in \mathbb{R}^M$ representing the maximum and minimum row elements of the matrix $\mathbf{T}_i$. The positive and negative sets are generated as follows:

$$\begin{aligned} \mathcal{T}_+(i,j) &= \{\mathbf{t}|\mathbf{t} \sim \text{Dist}(\mathcal{P}_i), d(\mathbf{t}, \mathbf{t}_{i,j}) \prec \mathbf{m}_i\} \\ \mathcal{T}_-(i) &= \{\mathbf{t}|\mathbf{t} \sim \text{Dist}(\mathcal{P}_i), d(\mathbf{t}, \mathbf{t}_{i,j}) \succ 2 \times \mathbf{m}_i\,, \\ &\quad \forall j \in \{1, ..., |\mathcal{P}_i|\} \end{aligned} \tag{12}$$

where

$$\mathbf{m}_i = \left[\max_{row}(\mathbf{T}_i) - \min_{row}(\mathbf{T}_i)\right] \times \kappa \tag{13}$$

represents the margin between positive and negative cases with $\kappa \in (0,1)$ defined as a constant, $\prec$ and $\succ$ are the element wise "less than" or "greater than" operators between two vectors, and $d(\mathbf{t}, \mathbf{t}_{i,j}) = |\mathbf{t} - \mathbf{t}_{i,j}| \in \mathbb{R}^M$ is the dissimilarity function in (12), with $|.|$ denoting an operator that returns the absolute value of the vector $\mathbf{t} - \mathbf{t}_{i,j}$. Note that the randomly generated parameter $\mathbf{t}$ in (12) is projected to the patch $\mathcal{P}_i$ in order to guarantee that it belongs to the manifold. Basically, (12) generates positive samples that are relatively close to patch member points and negative samples that are sufficiently far to all patch members, and that both the positive and negative samples belong to the learned sparse manifold described in Section IV.

Finally, the discriminative learning of the rigid classifier is achieved with the maximization of the following objective function [77]:

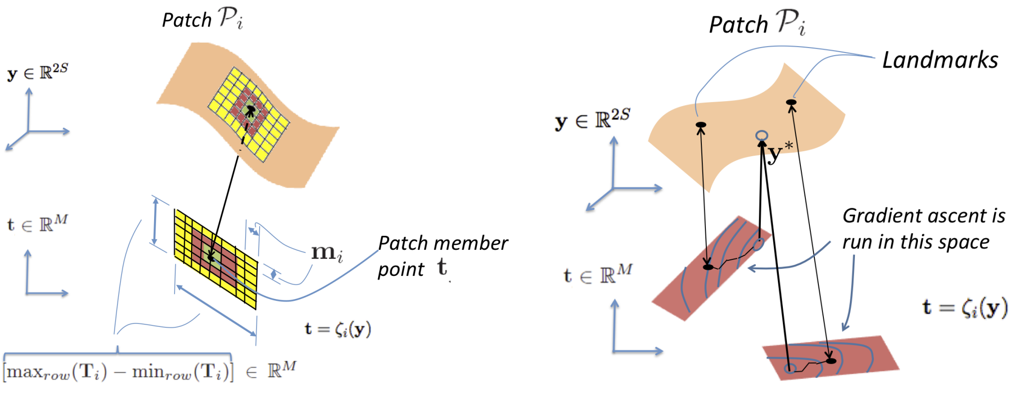
$$\gamma_{\text{MAP}} = \arg\max_{\gamma} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} \left[\prod_{\mathbf{t} \in \mathcal{T}_+(i,j)} p(\mathbf{t}|\mathbf{x}_{i,j}, \gamma) \right] \times \left[\prod_{\mathbf{t} \in \mathcal{T}_-(i)} (1 - p(\mathbf{t}|\mathbf{x}_{i,j}, \gamma)) \right], \tag{14}$$

where $p(\mathbf{t}|\mathbf{x}, \mathcal{D})$ in (2) is represented with $p(\mathbf{t}|\mathbf{x}, \gamma_{\text{MAP}})$ [52] hereafter. Fig.4(a) displays the training process explained in this section, where the positive samples are extracted from the green region in the center, and the negative samples are drawn from the yellow region. The parameters λ of the non-rigid classifier in (2) are learned in a similar way with the following optimization:

$$\lambda_{\text{MAP}} = \arg\max_{\lambda} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} p(\mathbf{y}_{i,j}|\mathbf{t}_{i,j}, \mathbf{x}_{i,j}, \lambda), \tag{15}$$

where $p(\mathbf{y}|\mathbf{t}, \mathbf{x}, \mathcal{D})$ in (2) is represented with $p(\mathbf{y}|\mathbf{t}, \mathbf{x}, \lambda_{\text{MAP}})$ [52] hereafter.

The estimation of the segmentation contour follows an inference procedure that takes a test image $\mathbf{x}$ as the input, and outputs the contour $\mathbf{y}^* \in \mathbb{R}^{2S}$ using (1). Recall that, this inference strategy uses each *landmark* $\mathbf{t}_{i,j}$ (for $j \in \{1, ..., L_i\}$) from each learned patch $\mathcal{P}_i$ as initial guesses for a gradient ascent (GA) procedure [78] on the output of the classifier $p(\mathbf{t}|\mathbf{x}, \gamma_{\text{MAP}})$ over the search parameter space


Patch $\mathcal{P}_i$
$\mathbf{y} \in \mathbb{R}^{2S}$
$\mathbf{t} \in \mathbb{R}^M$
$\mathbf{m}_i$
Patch member point $\mathbf{t}$
$\mathbf{t} = \zeta_i(\mathbf{y})$
$[\max_{row}(\mathbf{T}_i) - \min_{row}(\mathbf{T}_i)] \in \mathbb{R}^M$
Patch $\mathcal{P}_i$
Landmarks
$\mathbf{y} \in \mathbb{R}^{2S}$
$\mathbf{y}^*$
Gradient ascent is run in this space
$\mathbf{t} \in \mathbb{R}^M$
$\mathbf{t} = \zeta_i(\mathbf{y})$

(a) Training

(b) Inference

Fig. 4. The proposed training (a) and inference (b) procedures using sparse manifolds (please see text for details).

on the manifold $\mathcal{M}$. Given that the initial guesses of the GA procedure come from the landmarks, we have $\mathbf{t}_{i,j}^{(0)} = \mathbf{t}_{i,j}$, and after N GA iterations, the final value for the search parameter is $\mathbf{t}_{i,j}^{(N)}$, where the superscript (n) for $n \in \{0, ..., N\}$ represents the GA iteration index. Assuming that $p(\mathbf{t}) = p(\mathbf{t}|\mathbf{x}, \gamma_{MAP})$, the GA algorithm uses the Jacobian $\nabla p(\mathbf{t}) = \left[\begin{array}{ccc} \frac{\partial p(\mathbf{t})}{\partial \mathbf{t}(1)} & \cdots & \frac{\partial p(\mathbf{t})}{\partial \mathbf{t}(M)} \end{array} \right]^\top$, which is computed numerically using central difference, with step size $\mathbf{m}_i$ (13), as follows:

$$\frac{\partial p(\mathbf{t})}{\partial \mathbf{t}(1)} = \frac{p(\mathbf{t} + [\mathbf{m}_i(1)/2, ..., 0]^\top) - p(\mathbf{t} - [\mathbf{m}_i(1)/2, ..., 0]^\top)}{\mathbf{m}_i(1)} \tag{16}$$

where the parameter for $\mathbf{t}(.)$ stands for the dimensionality index and $\mathbf{t}(1)$ denotes the first dimension of $\mathbf{t}$, and similarly for $\mathbf{m}_i(.)$. In (16), the parameter $\mathbf{t} \pm [\mathbf{m}_i(1)/2, ..., 0]^\top$ is projected to the patch $\mathcal{P}_i$ (i.e., $\mathbf{y} = \xi_i(\mathbf{t})$) in order to guarantee that it belongs to the manifold $\mathcal{M}$. Once the GA process is over and the parameter $\mathbf{t}_{i,j}^{(N)}$ is reached for each landmark $\mathbf{t}_{i,j}$ of each patch $\mathcal{P}_i$, the contour $\mathbf{y}^*$ is estimated with a Monte-Carlo approximation of (1) as follows:

$$\mathbf{y}^* = \frac{1}{Z} \sum_{i=1}^{|\mathcal{P}|} \sum_{j=1}^{L_i} \mathbf{y} \times p(\mathbf{t}_{i,j}^{(N)}|\mathbf{x}, \gamma_{\text{MAP}}) \times p(\mathbf{y}|\mathbf{t}_{i,j}^{(N)}, \mathbf{x}, \lambda_{\text{MAP}}), \tag{17}$$

where Z is a normalization constant. Figure 4(b) shows the setting of the segmentation procedure, with the level sets representing the results of the rigid classifier $p(\mathbf{t}_{i,j}|\mathbf{x}, \gamma_{\text{MAP}})$. Notice that the rigid search procedure is performed only in the low dimensional space of $\mathbf{t}$.

VI. Search Complexity Reduction

One of the bottlenecks of current top-down non-rigid segmentation methods lies in the number of executions of the rigid classifier $p(\mathbf{t}|\mathbf{x}, \gamma_{\text{MAP}})$ that runs in the intermediate space represented by the variable $\mathbf{t} \in \mathbb{R}^r$, where $r = R$ indicates the original rigid search space and $r = M$ denotes the reduced dimensionality search space. For the complexity analysis below, assume that $K = \mathcal{O}(10^3)$ denotes the number of samples used in each dimension of this intermediate space. An exhaustive search in this r-dimensional space represents a running time complexity of $\mathcal{O}(K^r)$, which is in general intractable for relatively small values of $r = R$ (note that $R \in \{4, 5\}$ in state-of-the-art approaches). The reduction of this running time complexity has been studied by Lampert et al. [79], who proposed a branch-and-bound approach that can find a global optimum in this rigid search space in $\mathcal{O}(K^{r/2})$. Zheng et al. [5] proposed the marginal space learning that finds local optima using a coarse-to-fine approach, where the search space is recursively broken into spaces of increasing dimensionality (i.e., the search begins with one of the r dimensions, whose result is used to constrain the search in the space of two dimensions, until arriving at the space of r dimensions). Carneiro et al. [1] also proposed a local optima approach based on a coarse-to-fine derivative-based search that uses a gradient ascent approach in the space of r dimensions. In general, these last two methods provide a search complexity of $\mathcal{O}(K + \sharp\,\sigma \times K_{fine} \times r)$, where $\sharp\,\sigma$ is the number of scales (for the methods above, $\sharp\,\sigma = 3$), with $\sigma \in \{4, 8, 16\}$, and $K_{fine} << K$ (commonly, $K_{fine} = \mathcal{O}(10^1)$).

In the proposed approach, we are able to reduce the complexity of the rigid segmentation, that is, reduce r from R to M, and in this way, increase the efficiency of this segmentation stage. Therefore, in methods that only have one coarse step [1,80], represented by the rigid detector, this smaller dimensionality allows for a faster search process; and for methods that rely on multiple coarse steps [5], our approach can reduce the number of coarse steps to run (e.g., from R to M steps). Thus, if we are using the patch member points (without manifold sparsity) the complexity is given by $\mathcal{O}((\sum_i \mathcal{P}_i) \times \sharp\,\sigma \times r)$, meaning that we have to perform the segmentation in every patch of the manifold. When using sparsity, we use of L_i landmarks per patch $\mathcal{P}_i$, we avoid the expensive initial search of K points in the coarsest scale. Taking all this together, we have a final complexity of $\mathcal{O}((\sum_i L_i) \times \sharp\,\sigma \times r)$. Typically, we have $\sum_i L_i = \mathcal{O}(10^1)$, so our approach leads to a complexity of $\mathcal{O}(3 \times 10 \times r)$, which compares favorably to $\mathcal{O}(10^3 + 3 \times 10 \times r)$ [1,5] and $\mathcal{O}((10^3)^{r/2})$ [79]. One possible drawback of our proposal resides in the frequent use of the parametrization to map $\mathbf{t}$ to annotation $\mathbf{y}$, but we show in the experiments that the cost associated with that procedure is not significant compared to the running time of the rigid classifier.

VII. Experimental Setup

This section presents the experimental setup used for testing the proposed framework for object segmentation. Recall that the objectives of the proposed methodology are: 1) achieve superior efficiency with competitive accuracy, when compared to the state of the art, and 2) reach high robustness to small training sets given that training samples are constrained to lie in the learned low-dimensional manifold. It is important to emphasize that the inference efficiency depends not only on the dimensionality of the manifold (that is the tangent space), but also on the number of landmarks. Therefore, in order to test the robustness of the inference process to a limited number of landmark points, we run two experiments. In one of the experiments, we only use the landmarks during the inference process, making the whole process quite efficient. In the other experiment, we use all patch member points, which decrease significantly the search efficiency, but can potentially improve the segmentation accuracy. In order to assess the robustness of the learning process to training sets of different sizes, we train the rigid detector using augmented training sets of different sizes. The segmentation results of our methodology are then compared related approaches in terms of accuracy and running time figures.

A. Material

Two different problems are considered in order to empirically demonstrate our claims. The first problem is the segmentation of the left ventricle (LV) of the heart from ultrasound sequences [28], and the second problem is the segmentation of lips from sequences containing the faces of several people showing different types of emotions [8].

For the LV segmentation problem, 14 sequences taken from 14 different subjects are considered, where 12 sequences present some kind of cardiopathy (e.g., mild to severe dilation of the LV, hypertrophy of the LV, wall motion abnormalities, dysfunction of the LV, and valvular heart disease) and are used for training; 2 sequences are normal and used for testing (i.e., there is no overlap between subjects in training and test sets). All these sequences display the left ventricle of the heart using the apical two and four-chamber views (note that we refer to the test sets as $\mathcal{T}_1$ and $\mathcal{T}_2$). We worked with a cardiologist, who annotated 400 images in the training set (an average of 34 images per sequence) and 80 images in (average of 40 images per sequence) in the test set. It is important to mention 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.

For the lip segmentation problem, we use the Cohn-Kanade (CK+) database [8] of emotion sequences taken from frontal view, where the manual lip annotation is available. Among several emotion sequences we take the “happy” and “surprise” sequences, since they contain more challenging lip boundary deformations in comparison with the remaining emotion sequences. The training sets contain 12 sequences with 7 subjects where we use 5 “happy” sequences and 5 “surprise” sequences, with 3 subjects being used in both sequences, but exhibiting different lip motions. This training set consists of 209 frames for training , with 91 and 118 frames of the “happy” and “surprise” sequences, respectively. The test set also contains 12 sequences with 24 subjects where none of the subjects in the test sequences are present in the training sequences. This test set comprises 444 images, with 250 frames for “happy” and 194 frames for “surprise”.

B. Methods

The dimensionality of the explicit representation for the LV contour is $S = 21$ (i.e., 21 $2-$ dimensional points), and for the lip contour is $S = 40$ (i.e., 40 $2-$dimensional points). For the LV segmentation problem, the manifold learning algorithm produces: (i) $|\mathcal{P}| = 14$ patches, with a total of 1158 patch member points and 63 landmark points, and (ii) $M = 2$ for the dimensionality of the rigid search space (i.e., this represents the intrinsic dimensionality of the manifold). For the lip segmentation, the manifold learning produces: (i) $|\mathcal{P}| = 4$ patches with 395 patch-member points and 46 landmark points, and (ii) $M = 2$ for the dimensionality of the rigid search space. 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 [1,2,6]. Fig. 5 illustrates the result of our manifold learning algorithm on the LV and lip segmentation problems (see Section VIII), where each patch $\mathcal{P}_i$ contains a set of the patch-member and landmark points. In this figure, the blue dots are the annotations after PCA reduction (the first three components are shown), and the red circles indicate the estimated landmarks.

The training and inference methods used in this paper are adapted from a methodology that we have proposed recently [1], consisting of 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) [77]. The main difference lies in the use of sparse 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. Moreover, our rigid classifier is estimated using training sets of different sizes, where we show that the number of additional (artificial) 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 $|\mathcal{T}_{+}(i, j)| \in \{1, 5, 10, 15, 20\}$, and the size of negative samples as $|\mathcal{T}_{-}(i)| \in$

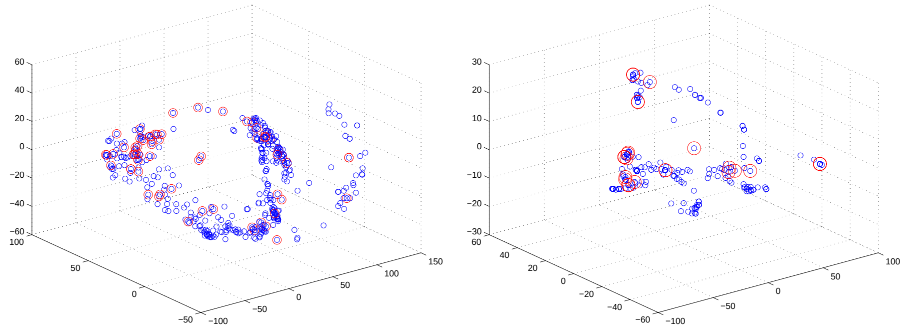


Fig. 5. Manifold learning algorithm for the LV (left) and lip (right) segmentation problems. The graphs show the annotation points in blue and landmarks in red after a PCA reduction. From our experiments, a total of 1158 patch member points (blue dots) and 63 landmark points (circle red) are estimated for the LV case. The right graph (lip case) depicts the manifold estimation with 395 patch-member points (in blue dots) and with the 46 landmarks (in red circles). Notice that larger number of patch member points (and landmarks) are obtained for the LV case, which is due to larger LV shape deformation obtained across different patients.

$\{10, 50, 100, 150, 200\}$, as explained in (12)[4]. We added more additional negative samples due to the larger area occupied by the negative region.

The performance of our approach is assessed with a quantitative comparison over the test sets using the following state-of-the-art methods based on machine learning techniques proposed in the literature for the LV segmentation problem: COM [2,6], CAR [1]. For the lip segmentation, we compare the performance of our approach only with CAR [1] because that was the only one available for comparison in this problem. For both segmentation problems, we also compare the running times between our approach and CAR [1].

C. *Accuracy Measurements*

The performance is evaluated in terms of contour accuracy using several metrics commonly adopted in the literature and running time spent to perform the object segmentation. The segmentation accuracy is assessed using the following error measures proposed in the literature: (i) *Hausdorff* (MAX) [81], (ii) *Mean Sum of Squared Distance* (MSSD) [6], (iii) *Jaccard distance* (JCD), (iv) *average distance* (AV) [28], (v) *F-measure*, and (vi) *the Intersection over Union* (IoU), which are commonly used metrics for contour evaluation.

[4] Note that for both databases, the training of the original algorithm in [1] used $|\mathcal{T}_{+}| = 10$, and $|\mathcal{T}_{-}| = 100$ per image in the training set.

VIII. Experimental Results

This section presents segmentation results of the proposed approach for the LV segmentation in ultrasound (US) and in magnetic resonance imaging (MRI) sequences, and also for the lip segmentation in video sequences. For the LV in US problem we conduct two distinct experiments. In Sec. VIII-A.1 we evaluate the accuracy of rigid detection separately, as this is the focus of the paper. In Sec. VIII-A.2 we compare the proposed framework with other related approaches tailored for the same problem and mentioning the run time figures obtained, as well as the the quantitative contour assessment. Similarly, for the lip segmentation (see Sec. VIII-B) we also perform comparisons concerning both quantitative and run time figures experiments. Finally, we provide a comparison between the proposed method and the semantic segmentation model based on the Convolutional Neural Network (CNN) [82] for the segmentation of the endocardium of the LV in MRI in short axis (see Section VIII-D).

A. LV Segmentation in US

This section is divided into two parts. First, we evaluate the accuracy of rigid detection, which is accomplished by presenting the results of the LV segmentation using an 14-fold cross validation (leave one sequence out). Then, we perform a comparison with the state-of-the-art methodologies applied in the same context (i.e., LV segmentation).

1) Comparison between rigid and improvement obtained with the non-rigid procedure in the problem of the LV Segmentation: To obtain the results of the rigid segmentations we performed a 14-fold cross validation, where the final result produced by the rigid detector is assessed based on the mean shape placed at the center of the detected window. The 14-fold cross validation is accomplished as follows:

(a) Generate 14 versions of the manifold as described in Section IV. Each version of the manifold is obtained using 13 sequences for training, leaving one sequence out for testing. This allows us to obtain the set of landmark points (Section IV-C) for each of the 14 manifolds.

(b) For each manifold, several DBN classifiers are trained, as follows: for a given configuration of data augmentation (recall from Sec. VII-B that there are five possible configurations), three classifiers are learned (one for each scale $\sigma \in \{4, 8, 16\}$). This amounts a total of 15 classifiers.

(c) The two above steps are repeated 14 times, and produce a total of 210 DBN classifiers.

The testing stage comprises the following main steps:

(a) For each frame of each held-out test sequence, 5 segmentations are produced (from the five data augmentation versions).

(b) Each sequence is tested in 17 images comprising the systolic and diastolic phase in the cardiac cycle (note that these 17 images represent a subset of the annotated images per sequence). This means that $14 \times 17 = 238$ segmentations are produced for all 14 sequences for a given data augmentation configuration.

(c) Considering the five data augmentation possibilities, this amounts a total of 1190 segmentations.

Figure 6 (top) shows the quantitative performance of the shape produced by the rigid detector (i.e., using Jaccard, average distance and Hausdorff metrics). Figure 6 (bottom) shows the improvement brought by the non-rigid segmentation compared to the rigid detection. From Fig. 6, we can observe that the non-rigid segmentation always improves the result from the rigid detector, which already produces a reasonably competitive result.

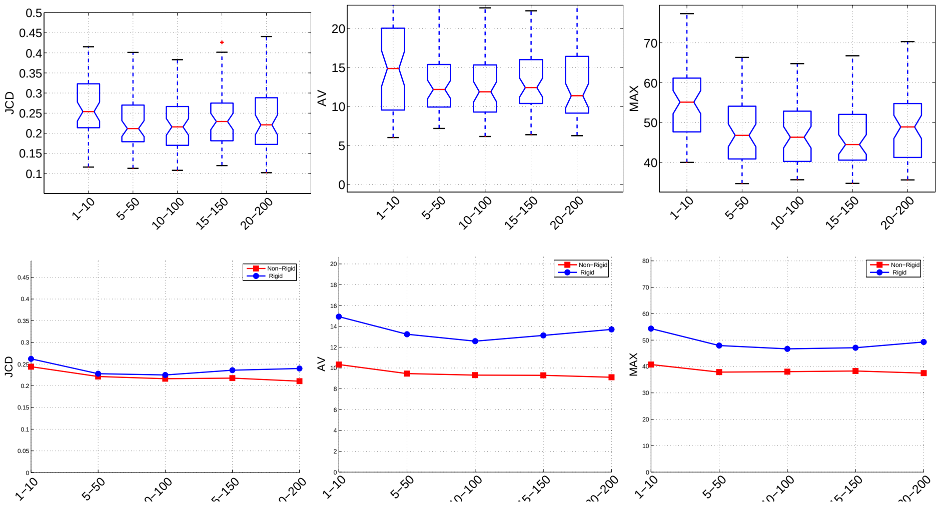


Fig. 6. The performance of the rigid detector in the 14-fold cross validation on the LV data is shown on the top graphs, while the bottom shows the improvements produced by the non-rigid detector, compared to the initial segmentation by the rigid detector, represented by the mean shape placed at the center of the detection window. From left to right, the graphs show the Jaccard, average and Hausdorff measures. Furthermore, all measures are shown with respect to varying sizes of positive and negative additional training samples in the ranges $\{(1, 10), (5, 50), (10, 100), (15, 150), (20, 200)\}$.

We also provide another experiment concerning the scatter-plots of the rigid and non-rigid stages using the 14-fold validation for all of the positive-negative configurations. This is illustrated in Fig. 7, where each dot represents one of the images in the left-out sequence. In this experiment we compare the gold standard LV volume using manual annotations and the computer-generated LV volume. To estimate the LV volume from 2D contour annotations we use the area-length equation [83] with $V = \frac{8A^2}{3\pi L}$, where A denotes the projected surface area and L is the distance from upper aortic valve point to apex,

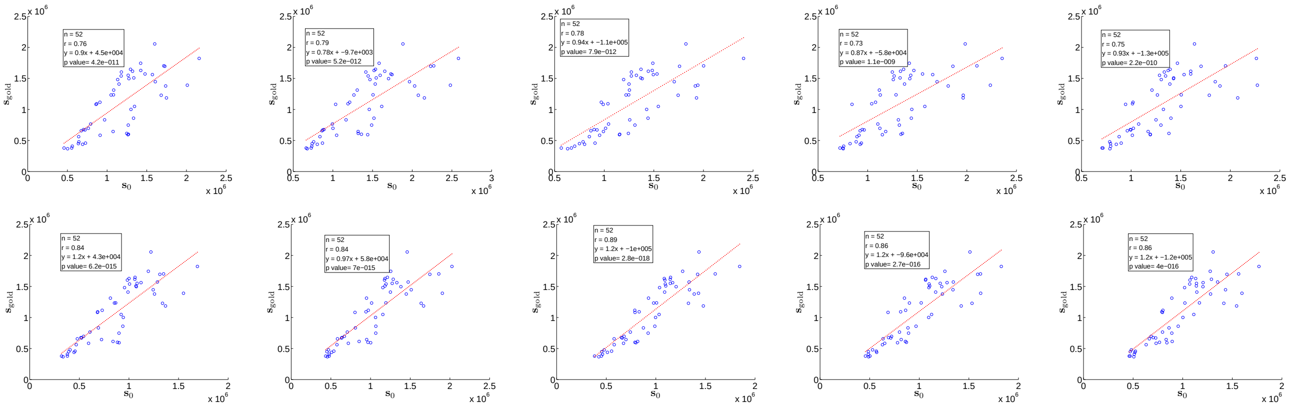


Fig. 7. Scatter plots with linear regression (top) and Bland-Altman bias plots (bottom). Rigid detection (top) non-rigid segmentation (bottom) for five data augmentation configurations $\{\{1,10\},\{5,50\},\{10,100\},\{15,150\},\{20,200\}\}$.

V is the resulting volume. In this scatter plot, we see that the non-rigid segmentation provides better results compared with the rigid segmentation. More specifically, the following correlation coefficients r are obtained: (i) rigid detector: $r = \{0.76, 0.79, 0.78, 0.73, 0.75\}$ (see Fig. 7 top) and (ii) non-rigid segmentation: $r = \{0.84, 0.84, 0.89, 0.86, 0.86\}$ (see Fig. 7 bottom).

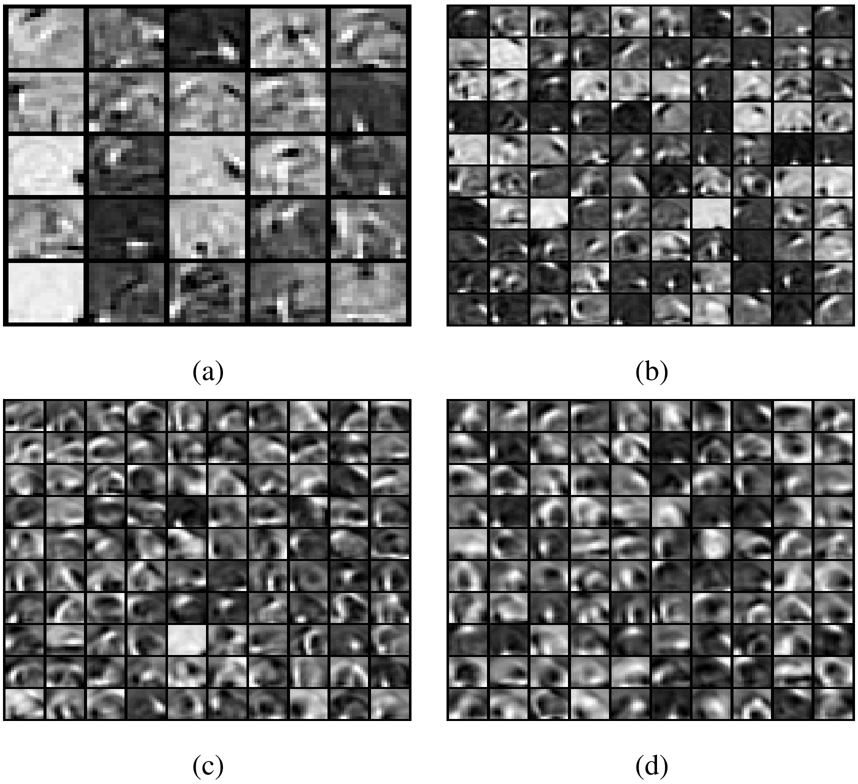
We also show the type of learned features for the rigid detector in the deep belief network. Figures 8 and 9 show the features learned for the configuration of 20-200 (i.e. positive-negative) for the patch member and landmark points, respectively. Denoting $\mathbf{W}_i$ with $i = 1, ..., n_L$ (n_L the number of layers), as the matrices of weights of the DBN learned for $\sigma = 4$, where the number of nodes of the learned architectures are the following:

- Patch member points: (196×100), (100×100), (100×200), (200×200), nodes, for layers 1, 2, 3 and 4, respectively;
- Landmark points: (196×100), (100×100), (100×200), nodes, for layers 1, 2 and 3, respectively.

Hence, we have for the patch member points, $\mathbf{W}_1 \in \mathbb{R}^{196\times100}$, $\mathbf{W}_2 \in \mathbb{R}^{100\times100}$, $\mathbf{W}_3 \in \mathbb{R}^{100\times200}$, and $\mathbf{W}_4 \in \mathbb{R}^{200\times200}$. For the landmark points the complexity of the architecture is lower providing the following weights matrices, $\mathbf{W}_1 \in \mathbb{R}^{196\times100}$, $\mathbf{W}_2 \in \mathbb{R}^{100\times100}$, and $\mathbf{W}_3 \in \mathbb{R}^{100\times200}$.

The features shown in Figs. 8, 9 depict the first 25 columns[5] (a) and the first 100 columns ((b), (c) and (d)). Each cell is a 14×14 matrix, that corresponds to a reshaped 196-dimensional vector. In Fig. 8 we have the matrices : $\mathbf{W}_1$ (a), $\mathbf{W}_1\mathbf{W}_2$ (b), $\mathbf{W}_1\mathbf{W}_2\mathbf{W}_3$ (c), $\mathbf{W}_1\mathbf{W}_2\mathbf{W}_3\mathbf{W}_4$ (d). In Fig. 9 we have

[5] 25 features are shown for the sake of better visualization



(a) (b)

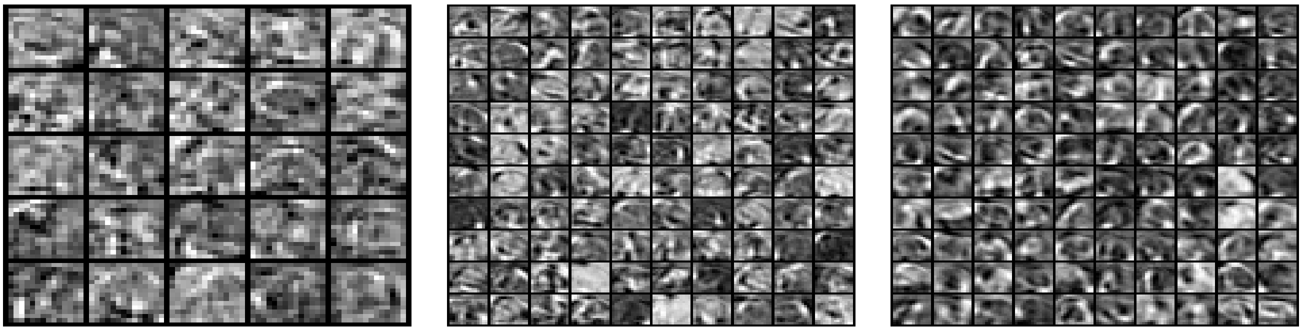
(c) (d)

Fig. 8. First 25 features (a) and 100 first features ((b),(c),(d)) for each layer of the rigid classifier at $\sigma = 4$ for the patch member points. Layers (a) 1, (b) 2, (c) 3, and (d) 4.

the matrices: $\mathbf{W}_1$ (a), $\mathbf{W}_1\mathbf{W}_2$ (b) and $\mathbf{W}_1\mathbf{W}_2\mathbf{W}_3$ (c). Note that the features learned in higher layers tend to be more global than features in lower layers. This fact demonstrates the abstraction capabilities of the DBN, that was already noticed in previous studies (see [84] for other type of experiments).

Also note that the features defined by the landmark points (Fig. 9) are a bit less ”well formed” than the the ones from the patch member points (Fig. 8),but they share some of the same high-level structures. In addition, the model learned for the landmark points is less complex.

2) Comparison with other related approaches: Fig. 10 shows the segmentation accuracy results for the LV test sequences $\mathcal{T}_1$ and $\mathcal{T}_2$ using the JCD and AV measures, which vary as a function of whether the inference used patch member or landmark points and also of the training set size. Fig. 11 and Fig. 12 show a comparison with COM [2,6] and CAR [1] approaches using all measures described in Sec. VII-C for the test sequences of the LV. For this comparison, we used the landmarks in the manifold and varying training set sizes. Fig. 13 shows a qualitative comparison displaying the segmentation result from our method and also from COM [2,6] and CAR [1] illustrating some snapshots of the LV test sequences.



(a) (b) (c)

Fig. 9. First 25 features (a) and 100 first features ((b),(c)) for each layer of the rigid classifier at $\sigma = 4$ for the landmark points. Layers (a) 1, (b) 2, and (c) 3.

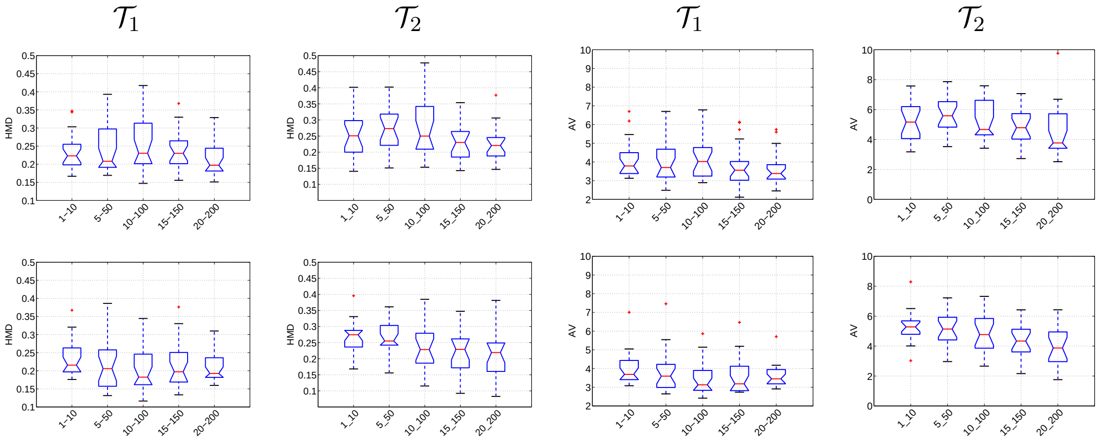


Fig. 10. Jaccard (columns 1 and 2) and average distance (columns 3 and 4) metrics for the test sequences $\mathcal{T}_1$ and $\mathcal{T}_2$. The accuracy is shown by varying the sizes of positive and negatives additional training samples in the ranges $\{\{1, 10\}, \{5, 50\}, \{10, 100\}, \{15, 150\}, \{20, 200\}\}$. Results are shown using the patch member points (top row) and using landmark points (bottom row) in the manifold.

We compare the running time figures of our approach with CAR [1]. The obtained results are shown in Table III. These running time figures were obtained on a computer with the following configuration: Intel Core i5, with $4GB$ of RAM.

In order to measure the statistical significance of the results shown in this section, we perform a t-test in Table I, where we compare two-variable measures and compute their probabilities of being drawn from independent samples (using the p-value). Our main objective is to show that our results are competitive, which means that the difference between our and the state-of-the-art results are not statistically significant,

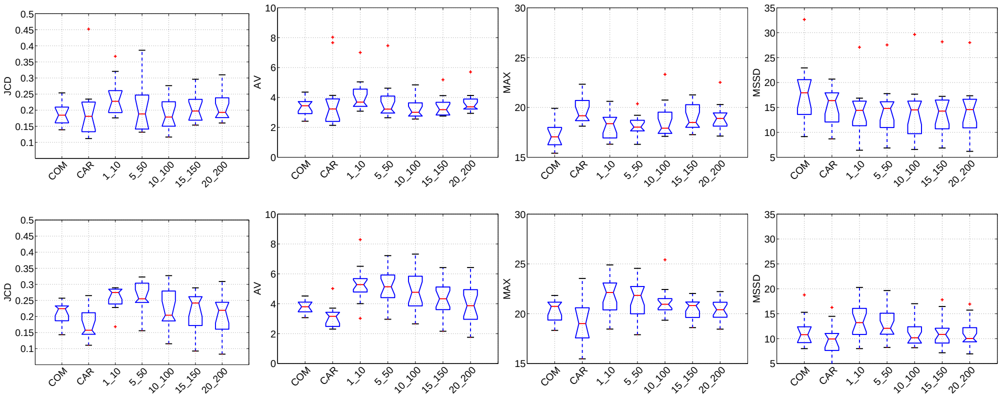


Fig. 11. Comparison of the state of the art (CAR [1] and COM [2,6]) against our approach using the sparse manifold with landmark selection and varying training set sizes on test sequences $\mathcal{T}_1$ (top row) and $\mathcal{T}_2$ (bottom row).

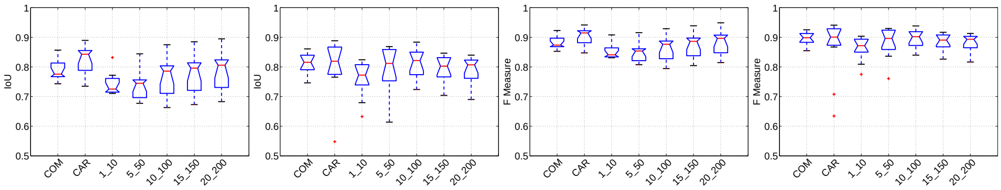


Fig. 12. Comparison of the state of the art (CAR [1] and COM [2,6]) against our approach using the sparse manifold with landmark selection and varying training set sizes on test sequences. It is shown the Intersection over union (two left images) and F-measure (two right images) obtained for the two test sequences $\mathcal{T}_1$ (1st and 3rd images) and $\mathcal{T}_2$ (2nd and 4th images).

which is represented by a p-value of $p > 0.01$. Also note that the p-value is obtained using the LV volumes computed from the 2D contours [83].

B. Lip Segmentation

Figures 14, 15 and 16 show a quantitative comparison between the proposed framework (using landmark points and different training set sizes, using the surprise and happy sequences, respectively) and the CAR [1] approach.

We also compare the running times of our approach with CAR [1]. See the obtained results for the happy and surprise sequences in Table III.[6]

Figs. 17, 18 show examples of the final lip segmentation produced by our methodology on the happy and surprise test sequences, along with the manual annotation.

[6] Notice that we compute the overall mean of the 12 happy sequences and 12 surprise sequences.

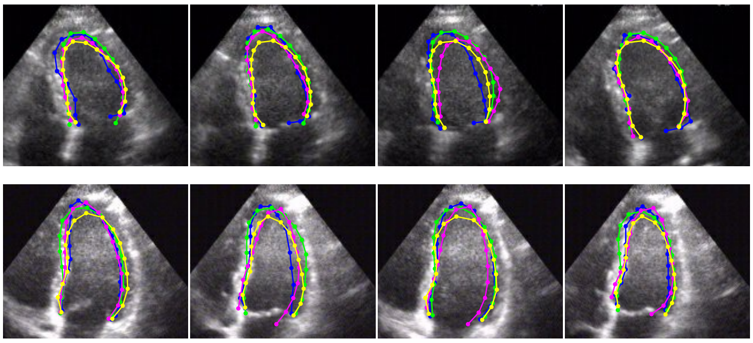


Fig. 13. Qualitative comparison between the expert annotation (GT in blue) and the results of our approach (green), COM (yellow), and CAR (purple). The results show the segmentations for the teste sequence $\mathcal{T}_1$ (top row) and for $\mathcal{T}_2$ (bottom row).

TABLE I

T-TEST BETWEEN THE VOLUMES ESTIMATED WITH THE PROPOSED APPROACH AND WITH THE CAR [1] AND COM [2,6] APPROACHES ON THE LV TEST SEQUENCES.

		Training set sizes (positive-negative)				
		1-10	5-50	10-100	15-150	20-200
COM [2,6]	p-value	0.316	0.079	0.139	0.249	0.138
CAR [1]	p-value	0.153	0.028	0.066	0.135	0.067

Finally, as in the previous LV sequences, we also perform a statistical significance of the results on both test sequences. Table II shows the comparison of the two-variable measures and the computation of their probabilities of being drawn from independent samples (using the p-value). The p-value is obtained using the lip area computed from the 2D contours.

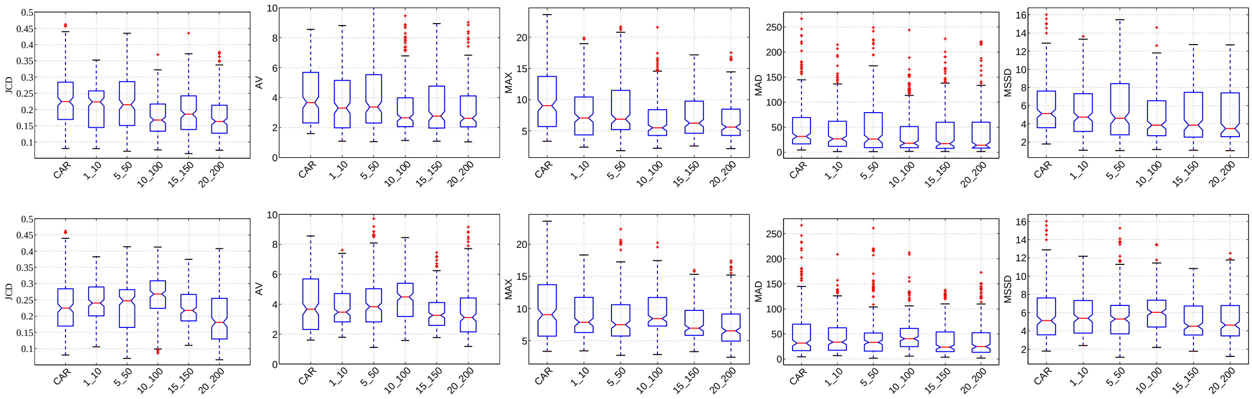


Fig. 14. Comparison with (CAR) method for the surprise sequences. Error metrics (from left to right: HMD, AV, MAX, MAD and MSSD) for the surprise sequences. The accuracy is shown varying the sizes of positive and negatives examples in the range $\{\{1, 10\}, \{5, 50\}, \{10, 100\}, \{15, 150\}, \{20, 200\}\}$. Results are shown using the patch member points (top row) and using landmark points (bottom row) in the manifold.

TABLE II

T-TEST BETWEEN THE AREAS ESTIMATED WITH THE PROPOSED APPROACH AND WITH THE CAR [1] ON THE SURPRISE AND HAPPY SEQUENCES.

		Training set sizes (positive-negative)				
		1-10	5-50	10-100	15-150	20-200
surprise	p-value	0.140	0.085	0.163	0.067	0.271
happy	p-value	0.047	0.063	0.039	0.030	0.060

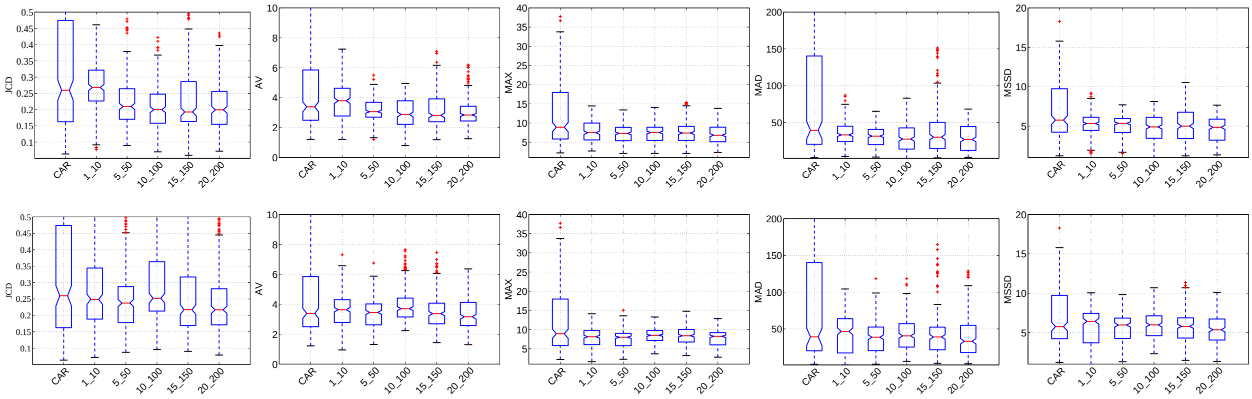


Fig. 15. Comparison with (CAR) method for the happy sequences. Error metrics (from left to right: HMD, AV, MAX, MAD and MSSD) for the surprise sequences. The accuracy is shown varying the sizes of positive and negatives examples in the range $\{\{1, 10\}, \{5, 50\}, \{10, 100\}, \{15, 150\}, \{20, 200\}\}$. Results are shown using the patch member points (top row) and using landmark points (bottom row) in the manifold.

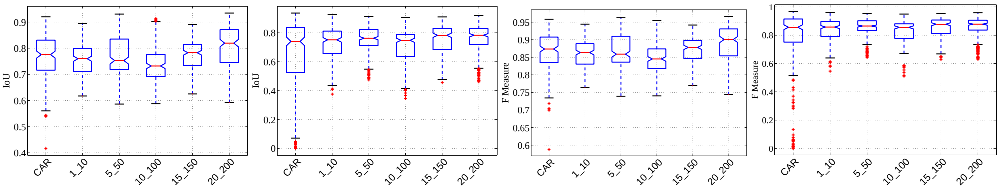


Fig. 16. Comparison with (CAR) method for the surprise sequences (1st and 3rd images) and for happy sequences (2nd and 4th images) for IoU (two left images) and F measure (two right images). Results are shown using the landmark points in the manifold.

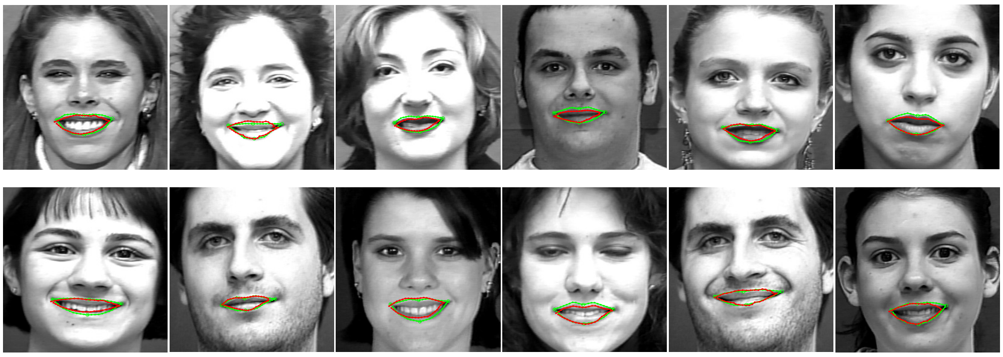


Fig. 17. Test lip sequences displaying the “happy” expression. The ground truth (in green) is superimposed with the segmentation results (in red).

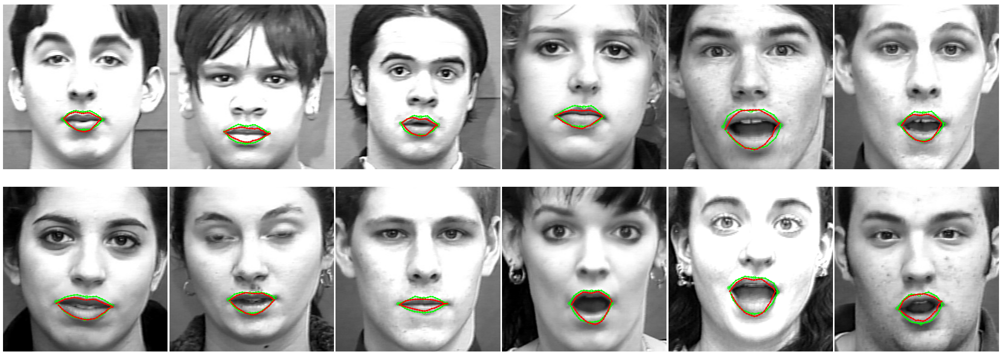


Fig. 18. Test lip sequences displaying the “surprise” expression. The ground truth (in green) is superimposed with the segmentation results (in red).

TABLE III

RUNNING TIME FIGURES FOR THE LV AND LIPS EXPERIMENTS. THE RESULTS ARE SHOWN IN SEC. PER FRAME. IN THE RIGID DETECTION STAGE THE TIME SPENT FOR THE PARAMETERIZATION IS SHOWN IN PARENTHESIS.

		CAR	Proposed		
		Total	Rigid	Non-Rigid	**Total**
LV	$(\mathcal{T}_1, \mathcal{T}_2)$	7.4	2.20 (1.26)	0.17	2.37
Lips	Happy	7.4	2.41 (1.29)	0.19	2.60
	Surprise	7.4	2.44 (1.30)	0.19	2.63

C. Comparison with Other Classification Methodologies

In this section we perform a comparison between the proposed approach and other shallow classification methods, such as SVM and Random Forests (RF). The SVM and Random Forest (we use 100 decision trees) training is based on the configuration of $\{20, 200\}$ positives and negatives to generate the input patches. This stage allows the estimation of soft confidences for the two-class (binary) classification task, *i.e.* object segmentation. For the testing phase, we have to plug in the two methods in the framework. This is done as described above, *i.e.* given the learned manifold, the landmarks are used as the initial guesses for the gradient ascent procedure. The only difference is that, we replace the learned DBN classifiers, by the confidences of the two methods in the segmentation procedure.

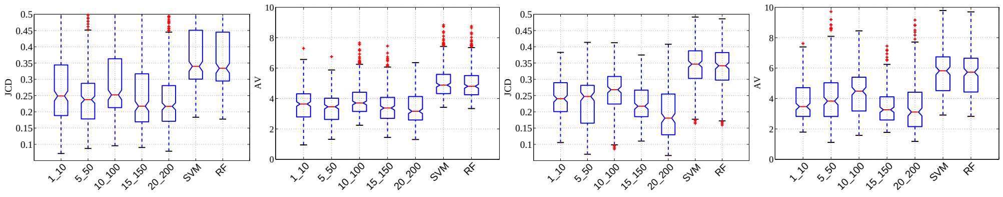


Fig. 19. Performance of the SVM and RF for the happy (two left images) and surprise (two right images) sequences. Jaccard distance (JCD) and average distance (AV) are used in this study.

In this experiment, we use the surprise and happy sequences for comparison purposes. Fig. 19 shows the performance of the proposed method for all configurations of the positive and negative examples, and the performance of the SVM and RF. It is clear that the performance of the shallow methods are less accurate that the proposed methodology. This is somehow expected since it is well known that RF does not train well on small datasets (similar performance is obtained for SVM). On the other hand, the better accuracy presented by DBN works well with small training sets.

We also performed an additional study, that explains the gradient ascent procedure during segmentation. Fig. 20 shows the evolution of the gradient magnitude of the SVM (left) and the DBN (right) in the 12 surprise sequences. In this experiment we use five iterations and we plot the evolution of the gradient agnitude in one patch of the manifold using the configuration $\{20, 200\}$ for positives and negatives, where each line corresponds to the gradient magnitude evolution for each frame in the sequence. We see that, for the SVM the gradient magnitude is more unstable, which can limit the accuracy of the segmentation. For the DBN, we can observe that this procedure is more stable, where the classifier results are able to provide a better guidance during the segmentation task. This happens since the classifier has an additional information about the features learned in the hidden layers and thus, they can provide more reliable confidence concerning the object position.

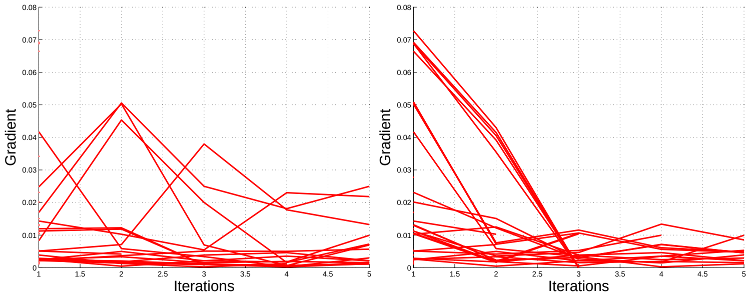


Fig. 20. Evolution of the gradient in the segmentation process (a) SVM and (b) DBN.

D. LV Segmentation in MRI

This section provides a comparison of the proposed methodology with the recently proposed semantic segmentation model based on Convolutional Neural Networks (CNNs) [82]. For this comparison, we use the publicly available dataset [85] containing 33 sequences acquired from 33 subjects (both healthy and diseased), where each sequence comprises 20 volumes, covering one cardiac cycle. As in the LV US and lip datasets, the object undergoes a rigid plus non-rigid deformation throughout time. In this dataset, the number of slices in each volume ranged from 5 to 10, with a spacing of 6 - 13 mm, where each slice is a 256×256 image, with a resolution in the range of $[0.93 - 1.64]$ mm per pixel. The ground truth of the LV segmentation in each slice is also provided.

The CNN architecture of the semantic segmentation model has 14 layers defined as follows (the size of the input channels are represented in parenthesis):

- Layer 1: 50 input filters of size 5×5, with stride=1, in the 'conv' layer (97×97)
- Layer 2: activation with 'ReLU', (97×97)
- Layer 3: max-pooling with size of 2×2, and stride=2, (48×48)
- Layer 4: 50-250 input/output filters of size 5×5 in the 'conv' layer, (44×44)
- Layer 5: activation with 'ReLU', (44×44)
- Layer 6: max-pooling with size of 2×2 and stride=2, (22×22)
- Layer 7: 250-500 input/output filters of size 5×5 in the 'conv' layer, (18×18)
- Layer 8: activation with 'ReLU', (18×18)
- Layer 9: 500-500 input/output filters of size 5×5 in the 'conv' layer, (14×14)
- Layer 10: activation with 'ReLU', (14×14)

- Layer 11: 500-500 input/output filters of size 5×5 in the 'conv' layer, (10×10)
- Layer 12: activation with 'ReLU', (10×10)
- Layer 13: 2-500 input/output filters of size 10×10 in the 'deconv' layer, (28×28)
- Layer 14: 'loss', (28×28)

The hyper paremeters of the network are as follows: (i) batchSize = 10, (ii) numEpochs = 100, (iii) learningRate = 0.0001, (iv) weightDecay = 0.0005, (v) momentum = 0.9, and (vi) random Gaussian initialisation with weightInit = 1/100.

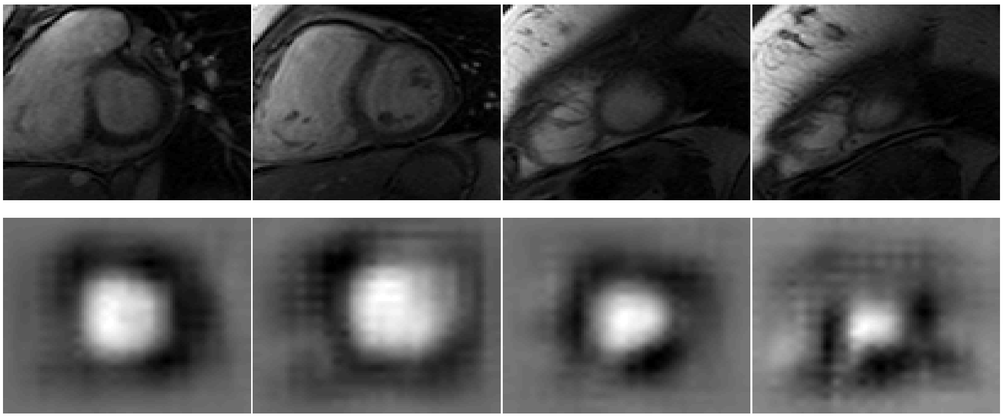
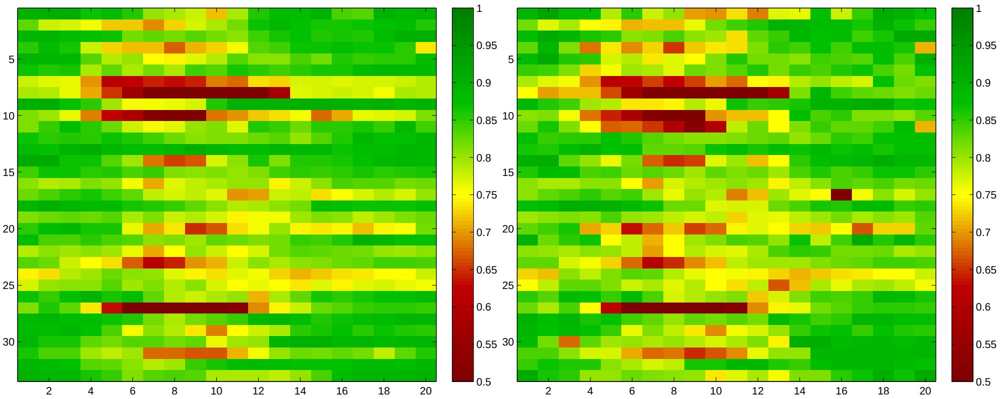


Fig. 21. MRI slices during a cardiac cycle (top) and the corresponding CNN output (bottom).

The evaluation process of the CNN semantic segmentation model and the proposed framework is the same as described in Sec. VIII-A, that is, performing a leave one sequence out (i.e. 33-fold cross validation). Fig. 21 shows the original MRI images of the LV (top) and the semantic segmentation produced by the CNN (bottom).

For comparison purposes, we compute the mean of IoU value per volume for the semantic segmentation and the proposed model. Fig. 22 shows the volumetric IoU coefficient obtained with the two methodologies for each of the 33×20 volumes in the dataset. It is possible to see that the proposed method is able to achieve comparable results with the CNN semantic segmentation model. Also note that most of the volumes are well segmented. The poorer segmentations can be identified in the regions of the red pixels in the maps. This figure shows that both methods perform better in the diastolic phase (roughly at frames 1-5 and 11-20) than in the systolic phase (frames 6-10). This is somehow expected where the structure to be segmented is small (see the rightmost image in Fig. 21, where the high probability map seems less defined).

Fig. 23 shows the quantitative comparison comprising using the metrics described in Sec. VII-C. The quantitative performance of the FCN is comparable, it is shown that the proposed methodology exhibits competitive results.



(a)

(b)

Fig. 22. Discriminated evaluation of the segmentation of each volume in the dataset for the CNN (a) and the proposed approach (b). Each map is a 33 (patients) × 20 (volumes) matrix. The colormap indicates the IoU, in which green correspond to good segmentation and red to poor segmentation.

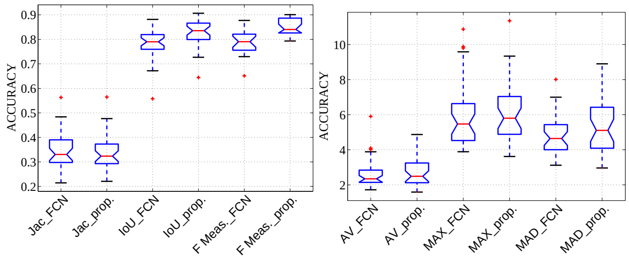


Fig. 23. Comparison between the FCN and with the proposed method (".prop" in the legends)using Jaccard distance, IoU, F measure (left) and AV, MAX and MAD metrics (right).

IX. Discussion

In this section, we first discuss the LV segmentation and then the lip segmentation results. We conclude the section with a presentation of the limitations of our method.

A. LV segmentation

Concerning the LV segmentation results shown in Fig. 10, we see that the inference process achieves similar accuracy with patch member and landmark points. This is relevant because it allows an improvement of 2 orders of magnitude in the inference process. Moreover, the number of additional positive and negative samples in the training set also shows insignificant impact on the accuracy of the methodology, as shown in Figures 10 and 11, which demonstrates that our methodology is robust to small training sets, allowing a more efficient training process. The comparison with the state of the art in Fig. 11 and Table II shows that our methodology produces competitive segmentation results that are comparable to the current state of the art in this database. It is interesting to see that the training process with 1 additional positive and 10 additional negatives achieves results that are comparable to COM [2,6] and CAR [1] (notice the large p-value indicating statistically insignificant differences in the results). In terms of running time, our method is about 3 times faster than CAR [1], but notice that the fact that the landmarks initialize independent search processes could have been exploited to improve even more this running time.

B. Lip segmentation

For the lip segmentation results shown in Figures 15 and 14 we also notice similar accuracy with patch member and landmark points. Similarly to the LV segmentation, the number of additional positive and negative samples in the training set also shows insignificant impact on the accuracy of the methodology. The comparison with CAR [1] shows that our proposed approach is mostly comparable (but actually slightly better in the sequence ”happy”), as demonstrated by the results in Table II. As observed in the LV segmentation results, the training process for the lip segmentation problem also produces competitive results with only 1 additional positive and 10 additional negatives. Finally, the running time is also also about 3 times faster than CAR [1].

C. Limitations of the Method

One of the main issues of the methodology is with respect to the number of estimated patches during the manifold learning process. In practice, during the segmentation procedure, we observe that only a small subset of the patches are important for the contour estimation in (17). The large number of patches is important for the robustness of the methodology, but a more efficient search process could have been designed to avoid the search in patches where the DBN produces a segmentation result with low confidence. Another issue with the methodology is with respect to the distribution of landmarks with respect to rigid deformation. In general, the training set must contain a good representation of this distribution in order for our approach to work robustly, because the rigid detection search is limited to the initial landmark locations.

X. Conclusion

In this paper, we presented a novel methodology for non-rigid object segmentation. The methodology proposed combines the deep learning architecture with the use of manifold learning. The main contribution and focus of the article is the dimensionality reduction of the segmentation contour parametrization for the rigid components. A manifold learning based approach has been proposed and allows to reduce the dimension of the rigid space. Thus, the framework allows for a 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 sparse manifold parametrization.

Further work will be focused on other directions. For instance, we plan to incorporate a dynamical model using 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. In fact, a fully parallel implementation can make the whole process 10-times more efficient.

Appendix

In this appendix, we briefly review the main steps of the manifold learning algorithm [75], summarized in Sec. IV (see [86] for supplementary information). Briefly, the algorithm takes the annotations $\{\mathbf{y}_j\}_{j=1}^{|\mathcal{D}|}$ and produces: (i) the intrinsic manifold dimensionality, (ii) the partitioning of the manifold into patches, and (iii) the charts and parameterizations.

1) Intrinsic Manifold Dimensionality: The estimation of the intrinsic dimensionality relies on a selection process over the significant eigenvalues of the following covariance matrix for each $\mathbf{y}_j$ [87,88]:

$$\mathbf{S}_{\mathbf{y}_j} = \frac{1}{|\mathcal{B}_{\mathbf{y}_j,\epsilon}| - 1} \sum_{\mathbf{y}_k \in \mathcal{B}_{\mathbf{y}_j,\epsilon}} (\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}})(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}})^\top, \tag{18}$$

where $\mathcal{B}_{\mathbf{y}_j,\epsilon}$ represents a set containing the annotations in the neighborhood of $\mathbf{y}_j$ within ϵ-radius, $\boldsymbol{\mu}_{\mathcal{B}_{\mathbf{y}_j,\epsilon}}$ denotes the mean of all annotations in the set $\mathcal{B}_{\mathbf{y}_j,\epsilon}$, and $|.|$ is the set cardinality operator. The intrinsic dimension is found by first computing the eigendecomposition of (18) for all elements of $\{\mathbf{y}_j\}_{j=1}^{|\mathcal{D}|}$. For each neighborhood, the eigenvalue immediately before the greatest drop in value should correspond to intrinsic dimension estimated by

$$\widehat{M_j} \equiv \arg\max_i |\lambda_{i+1} - \lambda_i|, \tag{19}$$

where λ_i are the sorted eigenvalues of (18). The global estimate of the intrinsic dimensionality M is the median over the estimates $\widehat{M_j}$, with $j \in \{1, ..., |\mathcal{D}|\}$ [75].

2) Partitioning of the Manifold into Patches: The partioning of the manifold $\mathcal{M}$ into p patches $\mathcal{P}_1, ..., \mathcal{P}_p$ is based on a clustering method that uses the concept of principal angles (e.g. [89,90]) and point distance as clustering criteria. The q-principal angles between subspaces spanned by the columns $\mathbf{a}$ and $\mathbf{b}$ of two matrices $\mathbf{A}$, $\mathbf{B}$ are defined as ([89,90]):

$$\cos\theta_k = \frac{|\mathbf{a}_k \mathbf{A}^\top \mathbf{B} \mathbf{b}_k|}{||\mathbf{A}\mathbf{a}_k||\,||\mathbf{B}\mathbf{b}_k||} \tag{20}$$

with $k \in \{1, ..., q\}$, where $q = \dim(\mathbf{A}) = \dim(\mathbf{B})$. Assuming that the matrices $\mathbf{A}$ and $\mathbf{B}$ are in fact the matrices $\mathbf{V}_i$ and $\mathbf{V}_j$ of column eigenvectors found by the eidendecomposition of (18) on elements i and j of $\{\mathbf{y}_j\}_{j=1}^{|\mathcal{D}|}$, the partitioning method clusters $\mathbf{V}_i$ and $\mathbf{V}_j$ that have a maximum principal angle smaller than a threshold (in this work, this threshold is $\pi/2$) and the distance between respective $\mathbf{y}_i$ and $\mathbf{y}_j$ is also smaller than a pre-specified threshold. This process produces a set of $|\mathcal{P}|$ patches that covers the manifold $\mathcal{M}$.

Given the $|\mathcal{P}|$ patches above, the next step of the learning process involves the estimation of the hyperplane for each one of those patches, which form a local coordinate system for each patch. This process involves the computation of the covariance matrix for each patch $\mathcal{P}_i$, as follows:

$$\mathbf{S}_{\mathcal{P}_i} = \frac{1}{|\mathcal{P}_i| - 1} \sum_{\mathbf{y}_k \in \mathcal{P}_i} (\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{P}_i})(\mathbf{y}_k - \boldsymbol{\mu}_{\mathcal{P}_i})^\top, \tag{21}$$

where $\boldsymbol{\mu}_{\mathcal{P}_i}$ represents the average of the annotations $\mathbf{y}_k \in \mathcal{P}_i$. Then, the eigendecomposition

$$\mathbf{S}_{\mathcal{P}_i} = \mathbf{V}_{\mathcal{P}_i} \mathbf{D}_{\mathcal{P}_i} \mathbf{V}_{\mathcal{P}_i}^\top \tag{22}$$

produces the matrix $\mathbf{V}_{\mathcal{P}_i}$ containing an orthonormal basis that forms the local coordinate system for $\mathcal{P}_i$.

A description of the clustering algorithm, that allows for the parch formation is given in Algorithm 1.

3) Charts and Parameterizations: Given the partition of $\mathcal{M}$ into $|\mathcal{P}|$ patches and the local coordinate system $\mathbf{V}_{\mathcal{P}_i}$ of each patch $\mathcal{P}_i$, the chart is obtained by projecting the patch point $\mathbf{y}_{i,j}$ according to

$$\mathbf{t}_{i,j} = \zeta_i(\mathbf{y}_{i,j}) \tag{23}$$

where $\zeta_i(\mathbf{y}_{i,j}) = [\mathbf{V}_{\mathcal{P}_i}^\top(\mathbf{y}_{i,j} - \boldsymbol{\mu}_{\mathcal{P}_i})]_M$, where the operator $[.]_M$ truncates the input vector at its first M components. The inverse mappings (i.e., the parameterizations) is given by

$$\mathbf{y}_{i,j} = \xi_i(\mathbf{t}_{i,j}) = \mathbf{V}_{\mathcal{P}_i} \left[\mathbf{t}_{i,j}^{(1)}, ..., \mathbf{t}_{i,j}^{(M)}, \widetilde{\xi}_i(\mathbf{t}_{i,j})\right]^\top + \boldsymbol{\mu}_{\mathcal{P}_i} \tag{24}$$

where the remaining $2S - M$ components of $\widetilde{\xi}_i$ are obtained through Gaussian process [75].

Algorithm 1 Soft clustering for patch formation

```
$i \leftarrow 0$
while $\mathcal{M}$ not covered do
    $i \leftarrow i + 1$
    Start new patch $\mathcal{P}_i$
    $\mathbf{y}_0 \leftarrow$ random seed chosen among points not attributed to any patch
    $\mathbf{V}_0 \leftarrow$ tangent subspace basis at $\mathbf{y}_0$ found by PCA in $\mathcal{B}_{\mathbf{y},\epsilon}$
    while not all $N$ points visited do
        $\mathbf{d} \leftarrow$ distances between all unattributed points and all points in $\mathcal{P}_i$
        $\mathbf{y}_* \leftarrow$ choose unattributed point with minimum $\mathbf{D}$
        $\mathbf{V}_* \leftarrow$ tangent subspace at $\mathbf{y}_*$
        $\theta_1, \ldots, \theta_n \leftarrow$ principal angles between $\mathbf{V}_0$ and $\mathbf{V}_*$
        if $\max_{k=1,\ldots,n} \theta_k < \tau$ and $\min \mathbf{d} < \epsilon$ then
            append $\mathbf{y}_*$ to $\mathcal{P}_i$
        end if
    end while
end while
```

References

[1] 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.

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

[3] Y. Zhan, X. S. Zhou, Z. Peng, and A. Krishnan, "Active scheduling of organ detection and segmentation in whole-body medical images," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*. Springer, 2008, pp. 313–321.

[4] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. N. Metaxas, and X. S. Zhou, "Towards robust and effective shape modeling: Sparse shape composition," *Medical image analysis*, vol. 16, no. 1, pp. 265–277, 2012.

[5] 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.

[6] 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.

[7] 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.

[8] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010, pp. 94–101.

[9] L. Zhang and E. Geiser, "An effective algorithm for extracting serial endocardial borders from 2-d echocardiograms," *IEEE Trans. Biomed. Eng.*, vol. BME-31, pp. 441–447, 1984.

[10] M. Sonka, X. Zhang, M. Siebes, M. Bissing, S. Dejong, S. Collins, and C. Mckay, “Segmentation of intravascular ultrasound images: A knowledge-based approach,” *IEEE Trans. Med. Imag.*, vol. 14, pp. 719–732, 1995.

[11] 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.

[12] 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.

[13] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, “Fast global minimization of the active contour/snake model,” *Journal of Mathematical Imaging and vision*, vol. 28, no. 2, pp. 151–167, 2007.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] M. Lynch, O. Ghita, and P. F. 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.

[20] 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.

[21] 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.

[22] N. Paragios, “A level set approach for shape-driven segmentation and tracking of the left ventricle,” *IEEE Trans. Med. Imag.*, vol. 22, no. 6, pp. 773–776, 2003.

[23] A. Sarti, C. Corsi, E. Mazzini, and C. Lamberti, “Maximum likelihood segmentation of ultrasound images with rayleigh distribution,” *IEEE T. on Ult., Fer. and F.C.,*, vol. 52, no. 6, pp. 947–960, 2005.

[24] 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.

[25] 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.

[26] 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.

[27] M. Mignotte, J. Meunier, and J. Tardif, “Endocardial boundary e timation and tracking in echocardiographic images using deformable template and markov random fields,” *Pattern Analysis and Applications*, vol. 4, no. 4, pp. 256–271, 2001.

[28] 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.

[29] 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.

[30] L. Bertelli, T. Yu, D. Vu, and B. Gokturk, "Kernelized structural svm learning for supervised object segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2153–2160.

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

[32] 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.

[33] 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.

[34] G. Carneiro, J. C. Nascimento, and A. Freitas, "Robust left ventricule segmentation from ultrasound data using deep neural networks and efficient search methods," in *Int. Symp. Biomedical Imaging: from nano to macro (ISBI)*, 2010.

[35] 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.

[36] 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.

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

[38] M. Maire, S. X. Yu, and P. Perona, "Object detection and segmentation from joint embedding of parts and pixels," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2142–2149.

[39] 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.

[40] 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.

[41] 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.

[42] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3378–3385.

[43] T. Brox, L. Bourdev, S. Maji, and J. Malik, "Object segmentation by alignment of poselet activations to image contours," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2225–2232.

[44] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 430–443.

[45] J. M. Gonfaus, X. Boix, J. Van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez, "Harmony potentials for joint classification and segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3280–3287.

[46] P. S. L'ubor Ladickỳ, K. Alahari, C. Russell, and P. H. Torr, "What, where and how many? combining object detectors and crfs," in *Proceedings of the 11th European conference on Computer vision: Part IV*, 2010, pp. 424–437.

[47] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, “Graph cut based inference with co-occurrence statistics,” in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 239–253.

[48] J. Yao, S. Fidler, and R. Urtasun, “Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 702–709.

[49] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, “Learning structured prediction models: A large margin approach,” in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 896–903.

[50] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer, “Large margin methods for structured and interdependent output variables.” *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.

[51] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[52] 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.

[53] C. H. Lampert, M. B. Blaschko, and T. Hofmann, “Efficient subwindow search: A branch and bound framework for object localization,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 12, pp. 2129–2142, 2009.

[54] U. Helmke, K. Hüper, P. Y. Lee, and J. Moore, “Essential matrix estimation using Gauss-Newton iterations on a manifold,” *Int. Journal of Comp. Vision*, vol. 74, no. 2, pp. 117–136, 2007.

[55] K. Huper and J. Trumpf, “Newton-like methods for numerical optimization on manifolds,” in *In Proc. of the 38th Asilomar Conference on Signals, Systems and Computers*, vol. 1, 2004, pp. 136–139.

[56] P.-A. Absil, R. Mahony, and R. Sepulchre, “Riemmanian geometry of Grassman manifolds with a view on algorithmic computation,” in *Acta Applicandae Mathematicae*, vol. 80, 2004, pp. 199–220.

[57] A. Edelman, T. Arias, and S. Smith, “The geometry of algorithms with orthogonality consraints,” in *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, 1998, pp. 303–353.

[58] S. Smith, “Optimization techniques on Riemmanian manifolds,” in *In Hamiltonian and gradient flows, algorithms and control*, A. Bloch, Ed. American Mathematical Society, 2004, pp. 113–136.

[59] I. F. Gorodnitsky, J. S. George, and B. D. Rao, “Neuromagnetic source imaging with focuss: A recursive weighted minimum norm algorithm,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 95, no. 4, pp. 231–251, 1995.

[60] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, “Sparse solutions to linear inverse problems with multiple measurement vectors,” *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[61] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[62] E. S. Cheng, S. Chen, and B. Mulgrew, “Efficient computational schemes for the orthogonal least squares learning algorithm,” *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 373–376, 1995.

[63] S. Chen and J.Wigger, “Fast orthogonal least squares algorithm for efficient subset model selection,” *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp. 1713–1715, 1995.

[64] S. F. Cotter, J. Adler, B. D. Rao, and K. Kreutz-Delgado, “Forward sequential algorithms for best basis selection,” *Proc. Inst. Elect. Eng. Vision, Image, Signal Process.*, vol. 146, no. 5, pp. 235–244, 1999.

[65] S. Chen and D. Donoho, “Basis pursuit,” in *Asilomar Conf. Signals, Syst., Comput.*, vol. I, 1994, pp. 41–44.

[66] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Sci. Comput.*, vol. 21, no. 1, pp. 33–61, 1998.

[67] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, “Subset selection in noise based on diversity measure minimization,” *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 760–770, 2003.

[68] J. A. Tropp, “Algorithms for simultaneous sparse approximation part ii: Convex relaxation,” *Signal Process. (Special Issue on Sparse Approximations in Signal and Image Processing)*, vol. 86, pp. 589–602, 2006.

[69] J. Chen and X. Huo, “Theoretical results on sparse representations of multiple-measurement vectors,” *IEEE Trans. Signal Processing*, vol. 54, no. 12, pp. 4634–4643, 2006.

[70] Y. Eldar and M. Mishali, “Robust recovery of signals from a structured union of subspaces,” *IEEE Trans. Inform. Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[71] L. Sun, J. Liu, J. Chen, and J. Ye, “Efficient recovery of jointly sparse vectors,” in *NIPS*, 2009.

[72] J. C. Nascimento, J. G. Silva, J. S. Marques, and J. Lemos, “Manifold learning for object tracking with multiple nonlinear models,” *IEEE Trans. on Imag. Proc.*, vol. 23, no. 4, pp. 1593–1605, 2014.

[73] J. C. Nascimento and G. Carneiro, “Top-down segmentation of non-rigid visual objects using derivative-based search on sparse manifolds,” in *CVPR*, 2013, pp. 1963–1970.

[74] T. Poggio and S. Smale, “The mathematics of learning: Dealing with data,” *Notices of the American Mathematical Society*, vol. 50, no. 5, pp. 537–544, 2003.

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

[76] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

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

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

[79] C. H. Lampert, M. B. Blaschko, and T. Hofmann, “Beyond sliding windows: Object localization by efficient subwindow search,” in *CVPR*, 2008.

[80] S. Zhou, J. Zhou, and D. Comaniciu, “A boosting regression approach to medical anatomy detection,” in *Proc. of the IEEE Conf. on Comp. Vision Pattern Recognition*, pp. 1–8.

[81] D. Huttenlocher, G. Klanderman, and W. Rucklidge, “Comparing images using hausdorff distance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[82] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015.

[83] 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.

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

[85] A. Andrepoulos and J. K. Tsotos, “Efficient and generalizable statistical models of shape and appearance for analysis of cardiac mri,” *Med. Image Anal.*, vol. 12, no. 3, pp. 335–357, 2008.

[86] J. G. Silva, J. S. Marques, and J. M. Lemos, “Selecting landmark points for sparse manofold learning,” in *NIPS*, 2005.

[87] J. Bruske and G. Sommer, “Intrinsic dimensionality estimation with optimally topology preserving maps,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 572–575, 1997.

[88] K. Fukunaga and D. R. Olsen, “An algorithm for finding intrinsic dimensionality of data,” *IEEE Trans. on Computers*, no. 2, pp. 176–183, 1971.

[89] A. Bjorck and G. H. Golub, “Numerical methods for computing angles between linear subspaces,” no. 123, pp. 579–594, 1973.

[90] P. A. Wedin, “On angles between subspaces of a finite dimensional inner product space,” in *Matrix Pencils*, 1983, pp. 263–285.