One shot segmentation: unifying rigid detection and non-rigid segmentation using elastic regularization

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

***Abstract*— This paper proposes a novel approach for the *non-rigid* segmentation of deformable objects in image sequences, which is based on one-shot segmentation that unifies rigid detection and non-rigid segmentation using elastic regularization. The domain of application is the segmentation of a visual object that temporally undergoes a *rigid transformation* (*e.g.*, affine transformation) and a *non-rigid transformation* (*i.e.*, contour deformation). The majority of segmentation approaches to solve this problem are generally based on two steps that run in sequence: a rigid detection, followed by a non-rigid segmentation. In this paper, we propose a new approach, where both the rigid and non-rigid segmentation are performed in a single shot using a sparse low-dimensional manifold that represents the visual object deformations. Given the multi-modality of these deformations, the manifold partitions the training data into several patches, where each patch provides a segmentation proposal during the inference process. These multiple segmentation proposals are merged using the classification results produced by deep belief networks (DBN) that compute the confidence on each segmentation proposal. Thus, an ensemble of DBN classifiers is used for estimating the final segmentation. Compared to current methods proposed in the field, our proposed approach is advantageous in four aspects: (i) it is a unified framework to produce rigid and non-rigid segmentations; (ii) it uses an ensemble classification process, which can help the segmentation robustness; (iii) it provides a significant reduction in terms of the number of dimensions of the rigid and non-rigid segmentations search spaces, compared to current approaches that divide these two problems; and (iv) this lower dimensionality of the search space can also reduce the need for large annotated training sets to be used for estimating the DBN models. Experiments on the problem of left ventricle endocardial segmentation from ultrasound images, and lip segmentation from frontal facial images using the extended Cohn-Kanade (CK+) database, demonstrate the potential of the methodology through qualitative and quantitative evaluations, and the ability to reduce the search and training complexities without a significant impact on the segmentation accuracy**

Keywords: Deep Learning, Data augmentation, Manifold learning, Object Segmentation,

I. Introduction

Object segmentation is one of the most studied topics in machine learning and computer vision. This task relies on the ability to partition the image into sets of pixels that correspond either to foreground or background. In this paper, object segmentation is defined as the process of obtaining S 2D points in the image that are located at the object contour (*i.e.*, that separates the foreground from the background). This process is however difficult to be robustly accomplished due not only to the image conditions (*e.g.* poor resolution or contrast), but also to the (non)-affine transformations plus local deformations that the foreground object can suffer, which are difficult to be modeled.

This segmentation problem has motivated the use of top-down based methodologies, which decouple the above transformations to simpler ones that are easier to be modeled. In general, two sequencial steps characterize the methodologies for top-down segmentation of deformable objects using machine learning techniques [1]–[6]. The first step aims to perform a *rigid detection*, while the second step is related to the *non-rigid segmentation*. The main motivation behind this strategy is the reduction of the training and inference complexities. For instance, for an object contour having S 2-D points, and quantizing each of the $2 \times S$ dimensions into K samples, a naive approach would lead to a complexity of $O(K^{2S})$. The introduction of an intermediate rigid detection allows for a drastic reduction of the search and training complexities with the use of a low dimensional rigid space with dimensionality $\mathbf{t} \in \mathbb{R}^R$ (with $R << 2S$ and R representing the dimensionality of the search space) that estimates the translation, scale and rotation transformations of a mean contour. The transformed mean contour obtained from the rigid detection is used to initialize and constrain the non-rigid segmentation, which decreases the original inference and training complexities of the methodology. The inference complexity decrease is achieved from the reduced dimensionality of the rigid space, which allows for a faster search process, and from the non-rigid segmentation that is constrained by the mean contour. Furthermore, the training complexity reduction is obtained from the need of smaller training sets in the small dimensional rigid problem and constrained non-rigid segmentation.

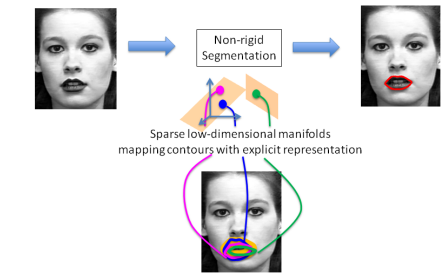
In this paper we argue that the segmentation of deformable visual objects can be done in a single shot using a methodology that is comparably accurate and more efficient than current approaches based on the aforementioned two-step process (*rigid and non-rigid segmentation*). In addition, because we focus on the reduction of the dimensionality of the underlying search process, we also argue that our proposed approach relies on relatively small training sets. In order to support our argument, we propose a methodology, where the search procedure is conducted on a sparse low-dimensional manifold (learned with the training set annotations of the visual objects contours), guided by the classification results computed from deep belief networks. Also, we no longer sub-divide the segmentation procedure into the *rigid detection* and *non-rigid delineation* mentioned above. Fig. 1 illustrates the difference between our proposal and the typical non-rigid segmentation approaches found in the literature. The development of our approach aims at the following goals: 1) increase the efficiency of the search process given the small dimensionality of the manifold (where the search takes place) and the fact that we solve the segmentation problem directly (without sub-dividing it into rigid and non-rigid detection); and 2) decrease the training complexity by constraining the shape distribution on the manifold (thus reducing the complexity of the trained models). Notice that this paper represents a significant extension of the segmentation approaches proposed in [1]–[5,7], where the *rigid detection* and *non-rigid delineation* (Fig. 1-(b))[1] are performed in a unified framework. Since we are proposing a single stage segmentation process, the non-rigid delineation is directly introduced in the learning phase.

The usefulness of the proposed approach is demonstrated in the *segmentation of non-rigid objects*. To accomplish this, we apply our

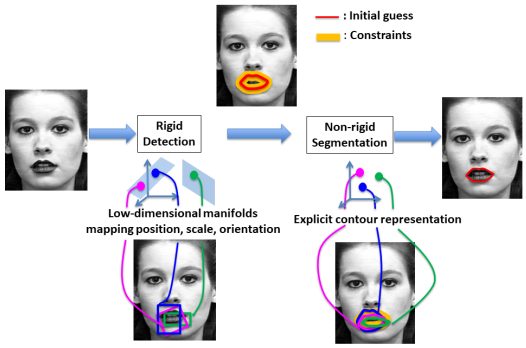
This work was supported by the FCT [UID/EEA/50009/2019] and by Australian Research Council grants (DP180103232 and CE140100016).

J. 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:** +351-218418196, **Fax:** +351-218418291. G. Carneiro is with the Australian Centre for Visual Technologies, University of Adelaide, Australia

[1]Note that [3] also uses a sparse low-dimensional manifold, but only for the rigid detection and still sub-divides the problem into rigid detection and non-rigid delineation.



a) proposed approach



b) typical non-rigid segmentation approach [1]–[6]

Fig. 1: Proposed segmentation approach for deformable visual objects (e.g., lips) that merges the rigid and non-rigid segmentation tasks (a), compared to the more common 2-step segmentation (b).

approach on the segmentation of the left ventricle (LV) of the heart from ultrasound images and on the segmentation of lip boundary from video sequences. Both datasets contain object contours that undergo a non-rigid deformation through time. It is experimentally shown that the proposed approach has a significant smaller search and training complexities exhibiting competitive segmentation accuracy results with respect to other related approaches proposed in the literature.

II. Related work

Image segmentation is an important and challenging stage to be accomplished in image analysis systems, where the obtained segmentation is usually used as an intermediate result to support an ensuing classification process. Basically, the segmentation can be viewed as a partition process in which the *regions of interest* in the image are separated from the *background*, where these two entities collectively cover the entire image. The great interest in this segmentation problem has allowed the development of a wide range of works over the years. In this section we provide an overview of existing methodologies, classifying them as active contours, deformable templates, database-guided (DB-guided), saliency object segmentation, multi-atlas and hybrid models. The first segmentation methods were rooted in the seminal work proposed in [8] called *snakes* or deformable models. This technique gave rise to a large number of works known in the literature as *active contours* based approaches [8]–[19]. Although some approaches introduce some knowledge based constraints, [18] the main shortcoming of deformable models is that the priors (e.g., strong edges, smooth contours) used in the segmentation process are usually not enough to reliably represent the range of appearance and geometric deformations suffered by the visual object of interest. The development of level-set methods [16] (another variant of the active contours) improved the performance of active contours with respect to imaging conditions and visual object topology. Nevertheless, active contours are based on non-convex energy formulations that strongly depend on good initial conditions, which are generally provided manually.

In an attempt to circumvent the above difficulties, template based approaches have been proposed. Deformable templates [20]–[25] introduce the use of more specific prior models regarding the shape and appearance of the object. This strategy has the goal of deforming this prior model to match the test image. Similarly to the problem of non-convex optimisation in level sets, this approach also needs a good initialization for the segmentation process. The literature testifies that the level-sets and deformable template models are among the most successful techniques applied in non-rigid segmentation problems, but the strong prior knowledge (often designed by hand or learned using a small training set) defined in the optimization function remains a challenge that is not entirely fulfilled. As a result, the effectiveness of such approaches is limited by the validity of those prior models, which are unlikely to capture all possible shape and appearance variations present in the imaging of the visual object [26].

The above methodologies gave room to the development of more sophisticated database-guided or machine learning methods for object segmentation. These techniques explore the use of large and rich annotated data sets that allows to improve the performance of the active contour and template based methods, producing the current state of the art segmentation accuracy results in the field. A typical example of the above class of approaches is the active shape model (ASM) introduced in [27,28]. ASM is an example of a machine learning algorithm that estimates a statistical model of the object shape using an annotated training set. Active appearance models (AAM) [27,29,30] improves over ASM, by using both shape (as in ASM) and texture models. The optimization of the above models is based on a cost energy function containing two terms, representing the shape and appearance, also learned using a manually annotated training set. These models are generative, since it is possible to generate synthetic images and shapes with ASM and AAM. Discriminative classifiers also constitute a valuable methodology for object segmentation [31]–[37]. Other class of machine learning algorithms are based on graphical models, such as Markov random fields (MRF) [38,39] and Conditional random fields (CRF) [40]–[45], which rely on structured output classification models.

Salient object segmentation constitutes another class of methods, where the segmentation is done by minimizing a cost associated with a graph. The main motivation for graph-based methods is the reduction of the dimensionality of the segmentation, and the use of a max-flow [46], [47] or minimum cuts algorithms [48]. In [46], an interactive segmentation is proposed where the user has to indicate certain pixels as part of the object or background (*i.e.*, hard constraints). The segmentation problem is achieved by combining these hard constraints with soft constraints, which are defined based on boundary and region properties. In [47] the saliency and objectness are obtained from the input image using two methods: the objectness computed from bounding boxes [49] and the saliency obtained with a saliency tree [50]. The input image is segmented into a set of superpixels using SLIC [51], which builds the graph. In [48] a region-based algorithm is proposed where the object saliency detection is viewed as a regression problem. The goal is to learn a regressor (with random forest) that maps the regional feature vector (*i.e.* color and texture) to a saliency score. All these works [46], [48], [47] rely on handcrafted features, *e.g.*, edge weights graph are based on

the color histograms differences between superpixels [47], or image edges [48], and node weights rely on intensity histograms [46]. A comprehensive survey of salient object detection can be found in [52]. All the works above, however, do not take in consideration the parametric description of the contour as we do, in particular, the non-rigid deformation that the object may undergo through time. Another issue that is not addressed by the works above is how to design a deep learning segmentation method that require small training datasets and have low training complexity, an important aspect that we emphasise in this paper.

Atlas-guided segmentation is a different way of performing object segmentation. The reasoning behind this class of approaches is that the visual object can be assumed to be consistent regarding its shape, appearance and localization. This assumption allows the use of a reference object model as a template. The segmentation process proceeds by registering the template using a non-rigid transformation model [53,54]. As suggested in [55], four classes of methods haven been developed in this context: (i) individual atlas images (IND), [56]; (ii) most similar atlas image from a database (SIM), where the registration process can be either performed by mutual information [57,58] or normalized mutual information [59]; (iii) average shape atlas (AVG) based on the computation of the average model of a set atlases, which is then registered to the test image; and (iv) multiple atlases (MUL), where the test image is registered to each member of a set of atlas templates and the final segmentation is achieved with a fusion strategy [60].

Regarding the class of approaches that use several atlases (AVG and MUL), it has been demonstrated that the fusion of multiple segmentations [61] can improve the segmentation accuracy. This improvement has been shown in [55], where different techniques for atlas selection and fusion are discussed. In an attempt to improve the reliance on a single segmentation, [62] proposed STAPLE, in which the classifiers are weighted using the expectation-maximization (EM) algorithm. It has also been shown that MUL based approaches are useful in the context of brain segmentation in MRI. The study presented in [63] replaces the use of a single atlas by a family of templates. In [64] the LEAP framework for multi-atlas is proposed, where the initial atlases may represent a subset of target images. The goal is then to propagate a small number of atlases through a large set of MRI brain containing a significant amount of variability among the anatomical structures. Another contribution that uses manifold learning as a tool for atlas selection has been presented in [65], where three different manifold learning techniques are assessed to select the best atlases and to combine in the multi-atlas segmentation context.

Other class of methodologies are the hybrid models, where machine learning and deformable contours are combined in some fashion: combining deformable contour models with MRF (*e.g.* [38,66]), or CRF with level sets [42]. The combination of SVM and CRF has also been explored in [37,44]. More recently, deep learning has been combined with levels sets [67], and deep convolution and deep belief networks have been shown to be useful as potential functions in structured output prediction [68].

More recently, deep learning has been intensively explored for the problem of image segmentation with the development of the region based convolutional neural network (R-CNN) [69], and its extensions: fast-CNN [70] and faster-CNN [71]. Basically, the goal of R-CNN is to take in an input image, and correctly localise objects in the image with bounding boxes. This is achieved following a three-stage process: (i) generate of a set of proposals using bounding boxes, (ii) run the images in the bounding boxes through a pre-trained AlexNet with a SVM classifier and (iii) perform a linear regression model to output tighter coordinates for the bounding boxes. Even though the R-CNN was successful, it could be quite slow given the large number of proposals per image. To alleviate this shortcoming, alternatives have rapidly become available in the literature. For instance, Fast R-CNN uses RoIPool (Region of Interest Pooling) in order to share the computation across proposals. Although clearly advantageous from a computational viewpoint, the proposals are still created using selective search, which is a fairly slow process. To address the above mentioned bottleneck, Faster-CNN has been proposed, where the region proposal step is almost cost free. The insight is that region proposals depended on features of the image that were already calculated with the forward pass of the CNN. This is achieved by simply adding a fully convolutional network on top of the features of the CNN creating what is known in the literature as the "Region Proposal Network". Contrary to the previous approaches where the segmentation is achieved via bounding boxes (in a two stage process), the Fully Convolutional Network (FCN), based on CNNs [72] is trained end-to-end, and is able to provide a pixel wise semantic segmentation which is the task addressed in this paper.

Our approach shares some common principles with [73]. Concretely, they also use the two segmentation stages, *i.e.* rigid detection and non-rigid segmentation. The former is accomplished using the affine transformation, the later (non-rigid deformations) is modelled using thin plate-splines (TPS). However, the use of TPS for modelling the non-rigid deformations was first proposed in [7]. The main differences between [73] and our method is that [73] does not address a reduction of training complexity, and the training with small training sets. These issues are addressed by this paper with a manifold learning strategy as part of the segmentation process.

III. Proposed approach

In this paper, we propose a novel methodology for segmenting non-rigid visual objects, where the search procedure is conducted directly on a sparse low-dimensional manifold, guided by the classification results computed from a deep belief network. The main novelty of our proposed approach is that we do not rely on the typical sub-division of segmentation tasks into rigid detection and non-rigid delineation. Instead, the rigid and non-rigid segmentation are solved in an unified framework (see Fig. 1 (a)). Also, an elastic regularization is proposed as a way to learn the deformations of the object contour. The manifold is divided into several patches, with each patch providing a *segmentation proposal* during the inference process. Since several patches are obtained, a fusion strategy is required, which is accomplished with the use of the confidence measure of each segmentation patch produced by the DBN. We show that our proposal reduces the *search complexity* and the *amount of training*, since the dimensionality of the manifold is much smaller than the dimensionality of the search spaces for rigid detection and non-rigid delineation aforementioned. The proposed multiple atlas segmentation exhibits the following advantages: (i) reduction of the rigid detection space dimensionality, using manifold with low intrinsic dimensionality, which allows for a faster inference process; (ii) reduction of the training data sets, since now the positives and negatives samples lie on the learned low-dimensional manifold; and (iii) multiple segmentations fusion that improves the segmentation robustness.

Although some of the contributions have been proposed in [7], in this paper, we provide a more comprehensive literature review, explanations, and experimental results, which not only compares the segmentation results with the state of the art, also provides statistical evaluation of the framework. Moreover, a new method for data augmentation (DA) is provided that allows to incorporate local deformation in the objects boundary.

Some of the contributions have been proposed in [6,7]. In [6] a "two-shot" method is presented where the detection step uses sparse

manifolds, but the segmentation step is based on an independent method. Regarding [7], this paper provides a more comprehensive literature review, explanations, and experimental results, which not only compares the segmentation results with the state of the art, also provides statistical evaluation of the framework. Moreover, a new method for data augmentation (DA) is provided that allows to incorporate local deformation in the objects boundary.

IV. Classic Non-rigid Segmentation Approaches

This section describes how the non-rigid object segmentation is usually tackled by machine learning based methodologies. Let us consider the image $\mathbf{I} : \Omega \rightarrow [0, 255]$, ($\Omega$ denotes the image lattice) that contains the visual object of interest. We will assume that the contour of the object is explicitly represented by a list of S 2-D points, *i.e.* the annotation is represented by $\mathbf{S} \in \mathbb{R}^{2\times S}$. The training data set $\mathcal{D}$ contains images and their respective annotations: $\mathcal{D} = \{(\mathbf{I}, \mathbf{S})_j\}_{j=1}^{|\mathcal{D}|}$. The optimal segmentation is found by solving the following optimization problem:

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} p(\mathbf{S}|\mathbf{I}, \mathcal{D}), \tag{1}$$

where $p(\mathbf{S}|\mathbf{I}, \mathcal{D})$ represents the probability of finding a non-rigid segmentation $\mathbf{S}$ in image $\mathbf{I}$, computed by a model that is learned from the training set $\mathcal{D}$. This is a hard problem to be solved due to the high dimensionality of $\mathbf{S}$, which makes the direct optimization of (1) highly complex. One way to circumvent the problem above is to reduce this complexity by a divide-and-conquer algorithm, where preliminary lower dimensional problems are introduced and summed out. As an example, several approaches [1]–[5] introduce one preliminary problem represented by a hidden variable $\mathbf{t} \in \mathbb{R}^R$, with $R << (2 \times S)$. This sort of solution leads to the following formulation:

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

In general, the variable $\mathbf{t}$ in (2) represents a rigid transform that is applied to the coordinates of a canonical contour $\mathbf{C} \in \mathbb{R}^{2\times S}$ (built from the mean shape of the annotations $\mathbf{S}$ from $\mathcal{D}$), where the search for the segmentation contour $\mathbf{S}$ is then performed in the neighborhood of the points of this transformed version of $\mathbf{C}$. The grid of points around $\mathbf{C}$ in the image $\mathbf{I}$ is represented by $\mathbf{G_C} \in \mathbb{R}^{2\times G}$, forming a rectangular 2-D region. The transformation of $\mathbf{G_C}$ to a region of the image space is achieved via a linear transformation matrix $\mathbf{A} \in \mathbb{R}^{3\times 3}$, which is obtained from the variable $\mathbf{t}$ as follows [1]–[5]: $\mathbf{A_t} = h(\mathbf{t})$ [2]. The term $p(\mathbf{t}|\mathbf{I}, \mathcal{D})$ in (2) represents the rigid detection and computes the probability that the visual object underwent a transform represented by $\mathbf{t}$ in image $\mathbf{I}$. In practice, the rigid classifier $p(\mathbf{t}|\mathbf{I}, \mathcal{D})$ receives an image patch $\mathbf{I}(g(\mathbf{t}))$, with $g(\mathbf{t}) \in \mathbb{R}^{2\times G}$ defined by

$$g(\mathbf{t}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{A_t} [\mathbf{G_C}^\top, \mathbf{1}_G]^\top \tag{3}$$

where $\mathbf{1}_G \in \mathbb{R}^G$ is a vector of ones. Eq. 3 is used to acquire the image region to be used in the computation of $p(\mathbf{t}|\mathbf{I}, \mathcal{D})$, which ultimately estimates the probability that the input sub-window $\mathbf{I}(g(\mathbf{t}))$ contains the structure of interest.

The term $p(\mathbf{S}|\mathbf{t}, \mathbf{I}, \mathcal{D})$ in (2) computes probability of the segmentation $\mathbf{S}$ in image $\mathbf{I}$ given the value of $\mathbf{t}$. Notice the importance of $\mathbf{t}$ in this procedure: it is responsible for constraining and initializing the search for the contour $\mathbf{S}$ to be around the image patch $\mathbf{I}(g(\mathbf{t}))$. Also, notice that the rigid search space, represented by the variable $\mathbf{t}$ has dimension R. We shall demonstrate later that the search complexity in these state-of-the-art approaches is dominated by this rigid detection, which is in turn a function of R. Moreover, as the dimensionality of $\mathbf{t}$ increases, the training process for the classifier $p(\mathbf{t}|\mathbf{I}, \mathcal{D})$ in (2) becomes more complex, requiring larger amounts of data to avoid over-fitting.

[2] Recall that current methodologies use $\mathbf{t} = [x, y, \vartheta, \nu_x, \nu_y]$ ($\mathbf{t} \in \mathbb{R}^5$) that denotes a transformation comprising a translation x and y, rotation ϑ, and non-uniform scaling ν_x and ν_y; then $h(\mathbf{t}) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_x & 0 & 0 \\ 0 & \nu_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

V. Reformulating the non-rigid segmentation using sparse low dimensional manifolds

The main goal of this paper is to reformulate the optimization problem in (1). More specifically, the methodology is based on the following maximization problem:

$$\mathbf{m}^* = \arg\max_{\mathbf{m}} p(\mathbf{m}|\mathbf{I}, \mathcal{D}), \tag{4}$$

where, $\mathbf{m} \in \mathbb{R}^M$ is a point in a low dimensional manifold $\mathcal{M}$, which is directly used in the estimation of $\mathbf{S}^*$. That means that we no longer require an intermediate rigid detection because we estimate directly a non-rigid contour segmentation via $\mathbf{m}$, with dimension $M < R << S$. The implication of such procedure, is that the non-rigid part must be accounted in the manifold. In order to use the same types of classifiers as the ones described in Sec. IV, which require an input consisting of a rectangular window, we resort to the use of thin-plate splines (TPS) deformation. With the TPS deformation, we can represent a non-rigid deformation from the test image to a rectangular image patch to be used as an input to the classifier.

VI. Thin-Plate Splines

The thin-plate spline (TPS) is a useful non-rigid model for estimating image and shape alignment, which has been applied in different contexts, ranging from biological form [74,75] to computer vision applications and image analysis problems [76]–[78]. A short overview of TPS warp is now formalized. Let us denote $\hat{v}_s \in \mathbb{R}$ as the target function values at locations $(\hat{x}_s, \hat{y}_s) \in \mathbb{R}^2$, with $s = 1, ..., S$. Herein, we set $\hat{v}_s$ equal to the target coordinates $(\hat{x}_s, \hat{y}_s)$. The TPS interpolation function $f(x, y)$ defines the mapping $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ that minimizes the following nonnegative quantity

$$\mathbb{E}_f = \int\int_{\mathbb{R}^2} \Big((\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2 \Big) dx\, dy, \tag{5}$$

which is called "integral quadratic variation" or "'integral bending norm", having the form

$$f(x, y) = a_1 + a_2\, x + a_3\, y + \sum_{s=1}^{S} w_s\, U(\|(\hat{x}_s, \hat{y}_s) - (x, y)\|), \tag{6}$$

with $U = r^2 \log r$, where $r = (x^2 + y^2)^{1/2}$ in Cartesian origin. Notice that the interpolant function $f(x, y)$ contains two terms: the affine transformation parameter parameterized by $\mathbf{a} = (a_1, a_2, a_3)$, and the non-affine warping given by $\mathbf{w} = (w_1, ..., w_S)$. Equation (6) can be re-written to a matrix formulation that allows the estimation of parameters $\{\mathbf{w}, \mathbf{a}\}$ by solving the following linear equation:

$$\begin{bmatrix} K_{S\times S} & \mathbf{P}_{S\times 3} \\ \mathbf{P}^\top & \mathbf{0}_{3\times 3} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{S\times 1} \\ \mathbf{a}_{3\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix}, \tag{7}$$

where the subscripts indicate the dimension of each variable. The TPS in (7) allows the mapping from the grid $\mathbf{G_C}$, used to represent the canonical contour $\mathbf{C}$, to the non-rigidly deformed grid, that is denoted by $\widetilde{\mathbf{G}}_\mathbf{C}$. The deformed grid is computed as follows:

$$[\widetilde{\mathbf{G}}_\mathbf{C}^\top, \mathbf{1}_G] = [\mathbf{G}_\mathbf{C}^\top, \mathbf{1}_G] \widetilde{\mathbf{A}} + [\mathbf{K_G}^\top \mathbf{w}_x, \mathbf{K_G}^\top \mathbf{w}_y, \mathbf{0}_G] \tag{8}$$

where $\widetilde{\mathbf{G}}_{\mathbf{C}} \in \mathbb{R}^{2\times G}$, $\mathbf{w}_x, \mathbf{w}_y \in \mathbb{R}^{S\times 1}$, $\mathbf{1}_G, \mathbf{0}_G \in \mathbb{R}^{G\times 1}$ and $\mathbf{K}_{\mathbf{G}} \in \mathbb{R}^{S\times G}$. The elements of $\mathbf{K}_{\mathbf{G}}$ are given by

$$\mathbf{K}_{\mathbf{G}}(i,q) = U\Big((\sum_{j=1}^{3}(c_{ij}-g_{qj})^2)^{1/2}\Big) \tag{9}$$

and

$$U(r) = r^2\log(r) \tag{10}$$

where c_{ij} is the (i,j)-th element of $[\mathbf{C}^\top, \mathbf{1}_S] \in \mathbb{R}^{S\times 3}$ and g_{qj} the (q,j)-th element of $[\mathbf{G}_{\mathbf{C}}{}^\top, \mathbf{1}_G] \in \mathbb{R}^{G\times 3}$. The affine transformation matrix $\widetilde{\mathbf{A}} \in \mathbb{R}^{3\times 3}$ and $\mathbf{w}_i$ (for $i \in \{x,y\}$) are found from the following linear system for estimating the TPS coefficients

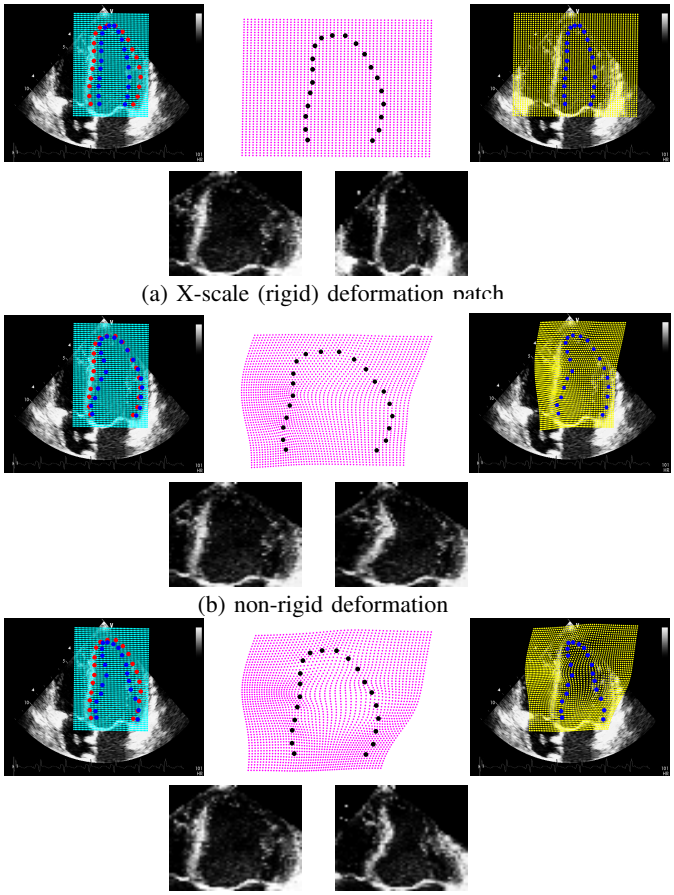
$$\begin{bmatrix}\mathbf{w}_x & \mathbf{w}_y \\ \widetilde{\mathbf{a}}_1 & \widetilde{\mathbf{a}}_2\end{bmatrix} = \begin{bmatrix}\mathbf{K}_{\mathbf{G}} & [\mathbf{1}_S, \mathbf{C}^\top] \\ [\mathbf{1}_S, \mathbf{C}^\top]^\top & \mathbf{0}_{3\times 3}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{S}^\top \\ \mathbf{0}_{3\times 2}\end{bmatrix} \tag{11}$$

where $\widetilde{\mathbf{a}}_1$, $\widetilde{\mathbf{a}}_2 \in \mathbb{R}^{3\times 1}$ represent the vectors that are used to build the matrix $\widetilde{\mathbf{A}}$, $\mathbf{w}_x$, $\mathbf{w}_y$ are $S\times 1$ vectors with the constraint that $[\pi_1\mathbf{S}\mathbf{w}_x, \pi_2\mathbf{S}\mathbf{w}_y]^\top$ is a 2×1 null vector (with $\pi_1 = [1,0]$, $\pi_2 = [0,1]$), and $\mathbf{K}_{\mathbf{C}} \in \mathbb{R}^{S\times S}$ is a matrix whose (i,q)-th entries are computed as $\mathbf{K}_{\mathbf{C}}(i,q) = U((\sum_{j=1}^{3}(c_{ij}-c_{qj})^2)^{1/2})$ with c_{ij}, c_{qj} the (i,j)-th and (q,j)-th elements of $[\mathbf{C}^\top, \mathbf{1}_S] \in \mathbb{R}^{S\times 3}$, respectively. Notice that (11) corresponds to the solution of (7).

As we did for $g(\mathbf{t})$ (in (3)), we can now write

$$g(\mathbf{m}) = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}[\widetilde{\mathbf{G}}_{\mathbf{C}}^\top, \mathbf{1}_G]^\top \tag{12}$$

The difference of using $\mathbf{I}(g(\mathbf{m}))$ instead of $\mathbf{I}(g(\mathbf{t}))$ is that it allows to obtain a patch that underwent a non-rigid deformation (see Fig. 2, in which this procedure is illustrated for some images). Fig. 2 (a) top, shows a horizontal contraction example. The red dots show the initial canonical shape of the heart, and the blue dots depict the horizontally scaled version of the initial shape. At the bottom of the Fig. 2 (a) it can be seen the generated patches from the canonical and deformed shapes. Fig. 2 (b,c) show examples of non-rigid deformations.



(a) X-scale (rigid) deformation patch

(b) non-rigid deformation

(c) non-rigid deformation

Fig. 2: Generation of samples $\mathbf{I}(g(\mathbf{m}))$ that are used as an input for the proposed classifier $p(\mathbf{m}|\mathbf{I},\mathcal{D})$ in (4). For each example ((a),(b) and (c)) the top-left image shows the initial grid containing the initial shape (red) and the X-scale deformed shape (blue). The middle and top-right images show the grid obtained according to the imposed deformation (magenta and yellow). At the bottom ((a),(b) and (c)) it is shown the initial patch (bottom-left) and the resulting deformation patch obtained with the TPS (bottom-right).

VII. Sparse Low Dimensional Manifold

In this section we describe how the mappings between the contour $\mathbf{S} \in \mathbb{R}^{2\times S}$ (high dimensional) and the manifold $\mathcal{M}$ with dimensionality M are accomplished, using the lower dimensional variable $\mathbf{m} \in \mathbb{R}^M$ introduced in (4). We follow the strategy proposed in [3,79] that is based on the *tangent bundle* concept of an M-dimensional manifold $\mathcal{M}$. Basically, this works by building and assembling multiple local models or representations (*i.e.* the patches) in an agglomerative fashion that are valid in distinct regions (please see supplementary for more details). More specifically, the soft partitioning is accomplished by using local PCA, which is based on a on maximum principal angle between neighboring tangent subspaces. This procedure allows to partition the manifold into several patches, where each patch contains data points that are neighbours not only in terms of proximity but also concerning its tangent space angles. The manifold learning algorithm follows the main stages: (i) *data partitioning* and *subsequent estimation of each local coordinate system*, and (ii) *charts estimation*. Basically, given a set of contours $\mathcal{S}$, it finds the intrinsic dimension M, partitions the data into $|\mathcal{P}|$ patches, and estimates the forward-backward mappings between contours $\mathbf{S} \in \mathcal{S}$ and respective lower representations $\mathbf{m} \in \mathbb{R}^M$. That is, the *charts* are estimated as

$$\mathbf{m} = \zeta(\mathbf{S}) \tag{13}$$

and the corresponding *parameterizations* as

$$\mathbf{S} = \xi(\mathbf{m}) \tag{14}$$

The search process for the optimization in (4) takes place in each of the low dimensional patches $\{\mathcal{P}_i\}_{i=1}^{|\mathcal{P}|}$ with initial guesses denoted by the patch member points $\mathbf{m} = \zeta(\mathbf{S})$. Since the manifold learning may provide a large number of patch members, this may result in an inefficient search process. Thus, we resort to a patch member point selection procedure, where the goal is to pick a subset of representatives in each patch that preserves enough information about the chart ζ_i. This subset is referred to as the *landmarks*.

The set of *landmarks* is obtained using the *least square angle regression* (LARS) [80], which is based on the solution of a regression problem that minimizes a regularized cost function [81]. If the set of patch-member points is represented by $\mathcal{Q}_i = \{\mathbf{m}_1 ..., \mathbf{m}_{|\mathcal{P}_i|}\}$, then after the application of the above procedure, we obtain the subset $\mathcal{L}_i \subseteq \mathcal{P}_i$ of size $|\mathcal{L}_i|$, which corresponds to the number of landmarks in the ith patch. These landmarks will be the points used for the initial guesses in the segmentation procedure, where in general, $|\mathcal{L}_i| \neq |\mathcal{L}_j|$ for $\mathcal{P}_i \neq \mathcal{P}_j$. The sparsity is thus achieved by finding the above set of landmarks. Table I describes the main steps for building the manifold as described in this section (for more details, please refer to [3]).

Fig. 3 depicts the obtained manifold for the problem of left ventricle segmentation (detailed in Sec. XIII-A), where the blue circles are the training LV contours after the PCA procedure (only the first three components are shown), and the red stars indicate the

TABLE I: Obtaining the manifold $\mathcal{M}$ and sparsity.

- **Input:** training data $\mathcal{D}_{\mathbf{S}}$.
- **Output:** patches $\mathcal{P}$, low dimensional contours $\mathbf{m} \in \mathbb{R}^M$, mappings between $\mathbf{S} \leftrightarrow \mathbf{m}$ and landmarks $\mathcal{L}$.

1) **Data collection**: Collect training object contour samples $\mathcal{D}_{\mathbf{S}} = \{\mathbf{S}_1, ..., \mathbf{S}_{|\mathcal{D}|}\}$.
2) **Intrinsic dimension**: From the above dataset, estimate the *intrinsic manifold dimension* M (see Appendix in the supplementary material).
3) **Partition**: the manifold $\mathcal{M}$ is partitioned into patches $\mathcal{P}_1, ..., \mathcal{P}_p$ using a soft-clustering method, based on two criteria: principal angle and distance between points (see Appendix in the supplementary material). See also Fig. 3 where the blue dots are the patch-members found.
4) **Chart and parameterizations**: Given the above patch-partition in $\mathcal{M}$, compute the charts, *i.e.* the mappings between the patches and tangent planes (see black arrows in Fig. 5) by computing $\mathbf{m} = \zeta(\mathbf{S})$, (see (13)). The parameterizations are simply $\mathbf{S} = \xi(\mathbf{m})$, (see Eq. (14)) (see also [7,79] for more details).
5) **Sparsity**: Finally, from the above set of patches member points $\mathcal{P}_1, ..., \mathcal{P}_p$, obtain a subset of its representatives, *i.e.* landmarks $\mathcal{L}_1, ..., \mathcal{L}_p$, with $\mathcal{L}_i \in \mathcal{P}_i$ and $|\mathcal{L}_i| \ll |\mathcal{P}_i|$. See the landmarks in red dots in Fig. 3, that are a subset of the patch-members (also see Section VII)

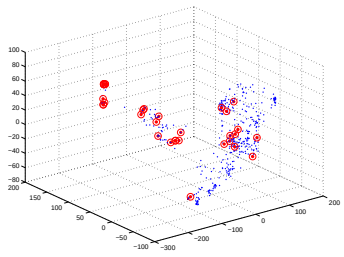


Fig. 3: The graph shows the input LV contours (described in Sec. XIII-A, *i.e.* $\mathbf{S}$ after the PCA reduction, the first three components are shown for each contour (see blue dots). A total of 496 annotations are given. This is the input to the sparse low dimensional manifold learning (Sec. VII) The manifold learning algorithm estimates patch member points distributed in 13 patches. In red it is shown the landmarks estimated for these 13 patches.

landmarks obtained.

VIII. Training procedure in the Sparse Low Dimensional Manifold

In this section, we first explain the main aspects of the training process. The training of the DBN models relies on data augmentation, but an interesting aspect of our work is that we can augment the data by the application of geometric and appearance changes, and then we can project the newly generated data to the learned manifold.

Data augmentation plays a regularization role, in which new training data samples (*i.e.* positives and negatives training samples) are artificially generated by sampling a noise vector from a Gaussian distribution [82] using the available training data. However, the artificial training samples are re-projected onto the learned manifold. There are two advantages associated with this approach. First, since the dimension of the manifold is small, there is no need to generate

TABLE II: Training.

- **Input:** low dimensional contours $\mathbf{m}$.
- **Output:** positive $\mathcal{T}_+$ and negative $\mathcal{T}_+$ samples built from $\mathbf{m}$

1) Train a multi-scale classifier, (represented by $\sigma = \{4, 8, 16\}$) using data augmentation (see (16)). Fig. 2 shows examples of positive samples used for training the classifier.

a large number of artificial training data samples. Second, given that the artificial samples are reprojected onto the manifold learned from the training data, they resemble well the training samples. Also, data augmentation has to be performed taking into account the deformation of the object contour. To accomplish this, we resort to the use of TPS (see Sec. VI), where the deformation of each contour point can be estimated from the training data set. More specifically, we collect the training set to obtain all the localizations for each contour point, and run the Expectation Maximization (EM) algorithm to obtain the statistical distribution for all contour points. Fig.4 shows the statistical distribution of the contour points.[3]

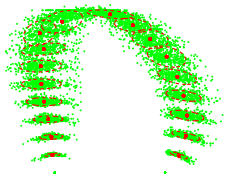


Fig. 4: Statistical distribution for the LV contour points obtained with the EM algorithm.

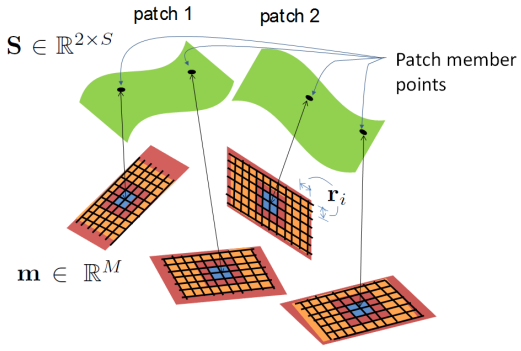


Fig. 5: Training procedure in the manifold. It is displayed the patches (in green), at the bottom, the positive (blue) and negative (orange) regions are displayed in the tangent hyper-planes.

For the classifier, we use the multiscale approach (as in [83]), where an image scale space $L(\mathbf{I}, \sigma)$ is produced by convolving the Gaussian kernel with the input image $\mathbf{I}$, where the scales are $\sigma \in \{4, 8, 16\}$ (see [83]). In order to train each classifier, we generate a set of positive and negative samples, which are obtained as follows. First, the estimation of the object contour in the image space from the set is performed *i.e.* $\hat{\mathbf{S}} = \underset{\sim}{\zeta}^{-1}(\mathbf{m})$. Second, use (11) to obtain the non-rigid deformation grid $\mathbf{G}_{\mathbf{C}}$ (see Fig. 5). In order to generate

[3]In this case 21 initializations are provided for the EM algorithm, one for each contour point.

the positive and negative samples, we use the following distribution based on the patch members $\mathbf{m} \in \mathcal{Q}_i$ of each patch $\mathcal{P}_i$:

$$\text{Dist}(\mathcal{P}_i) = U(range(\mathcal{Q}_i)), \tag{15}$$

where $U(range(\mathcal{Q}_i))$ denotes the uniform distribution over the set $\mathcal{Q}_i$ of patch member points. The positive and negative generated according to:

$$\begin{aligned}\mathcal{T}_+(i,j) &= \Big\{\mathbf{m}|\mathbf{m} \sim \text{Dist}(\mathcal{P}_i), |\mathbf{m} - \mathbf{m}_{i,j}| \prec \mathbf{r}_i\Big\} \\ \mathcal{T}_-(i) &= \Big\{\mathbf{m}|\mathbf{m} \sim \text{Dist}(\mathcal{P}_i), |\mathbf{m} - \mathbf{m}_{i,j}| \succ 2 \times \mathbf{r}_i, \\ &\qquad \text{for all } j \in \{1, ..., |\mathcal{P}_i|\}\Big\}\end{aligned} \tag{16}$$

where the margin between positive and negative samples is represented by $\mathbf{r}_i = range(\mathcal{Q}_i) \times \kappa$, with $\kappa \in (0, 1)$, $\prec$ and $\succ$ denote the element-wise operators "less than" and "greater than" between vectors. The margin defined in this way facilitates the training process, avoiding the existence of similar samples with different labels that can overtrain classifiers. Fig. 5 illustrates graphically the data augmentation described above and how the artificial training samples are generated. It is seen that the positive samples are drawn from the blues area, while the negative samples are drawn from the orange area. See Fig. 2 for an illustration of some of patches generated. These samples are then used to train the discriminative classifier parameters.

The DBN classifier (at a given scale σ) is trained by stacking several hidden layers to reconstruct the input patches in $\mathcal{T}_+$ and $\mathcal{T}_-$ (see (16)). After this process, two nodes are added to the top layer of the DBN, indicating $p(\mathbf{m}_+|\mathbf{I}, \mathcal{D})$ and $p(\mathbf{m}_-|\mathbf{I}, \mathcal{D})$. The classifier in (4) is modeled by γ_{MAP}, meaning that $p(\mathbf{m}|\mathbf{I}, \mathcal{D})$ can be represented by $p(\mathbf{m}|\mathbf{I}, \gamma_{\text{MAP}})$. The classifier parameter γ_{MAP} is learned from the patch-member points $\mathbf{m}_i = \zeta_i(\mathbf{S}_i)$ belonging to the patch $\mathcal{P}_i$, or from the landmark points, *i.e.* $\mathcal{L}_i \subseteq \mathcal{P}_i$. This is accomplished as follows [84]:

$$\begin{aligned}\gamma_{\text{MAP}} = \arg\max_{\gamma} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} \left[\prod_{\mathbf{m}_+ \in \mathcal{T}_+(i,j)} p(\mathbf{m}_+|\mathbf{I}, \gamma)\right] \\ \times \left[\prod_{\mathbf{m}_- \in \mathcal{T}_-(i)} (1 - p(\mathbf{m}_-|\mathbf{I}, \gamma))\right],\end{aligned} \tag{17}$$

where γ represents the model parameters of $p(\mathbf{m}|\mathbf{I}, \gamma)$, which denotes the classifier $p(\mathbf{m}|\mathbf{I}, \mathcal{D})$ in (4) and

$$p(\mathbf{m}|\mathbf{I}, \gamma_{\text{MAP}}) = f_{\text{softmax}} \circ f_Q \circ g_{Q-1} \circ f_{Q-1} ... \circ g_1 \circ f_1(\mathbf{I}^{(t)}), \tag{18}$$

where $\circ$ represents the function composition operator, $f_{\text{softmax}}(.)$ denotes the softmax activation function indicating the probability that the input image $\mathbf{I}^{(t)}$ contains the visual class of interest, $f_q(\mathbf{I}_{q-1}) = \mathbf{W}_q^{(\gamma)}\mathbf{I}_{q-1}$ represents a linear transform from layer $q-1$ to q (note that there are Q layers in total), $g_{q-1}(.)$ denotes a logistic activation function that takes as input the result from $f_q(\mathbf{I}_{q-1})$, $\mathbf{I}^{(t)}$ (which is equal to $\mathbf{I}_0$, *i.e.*, the input to $f_1(.)$) denotes the image region extracted with the transformation parameters in $\mathbf{A_t}$ (see 3), and γ represents the weight matrices for the Q layers, *i.e.* $\{\mathbf{W}_q^{(\gamma)}\}_{q=1}^{Q}$. Table II summarizes the main steps for the training process.

IX. Inference procedure for the segmentation

In this section, we provide the details of the optimisation used in the inference after the training process described in Sec. VIII. The inference procedure to produce the segmentation takes an input test image $\mathbf{I}$ and performs a gradient ascent procedure [85] on the output of $p(\mathbf{m}|\mathbf{I}, \gamma_{\text{MAP}})$ that is computed in the low dimensional manifold described in Sec. VII. This process will generate the final contour $\mathbf{S}^\star$ (see (1)) that we detail next.

The inference process can use either the landmark points in $\mathcal{L}$ or patch-member points in $\mathcal{P}$, depending on wether sparsity is used. These points represent the initial guess for the gradient ascent (GA) on the output of the classifier $p(\mathbf{m}|\mathbf{I}, \gamma_{\text{MAP}})$. Assuming the presence of the object shape, *i.e.* $p(\mathbf{m}) = p(\mathbf{m}_+|\mathbf{I}, \gamma_{\text{MAP}})$, the GA algorithm uses the Jacobian

$$\nabla p(\mathbf{m}) = \Big[\frac{\partial p(\mathbf{m})}{\partial \mathbf{m}_1}, ..., \frac{\partial p(\mathbf{m})}{\partial \mathbf{m}_M}\Big]^\top \tag{19}$$

where each element of $\nabla p(\mathbf{m})$ is defined by

$$\frac{\partial p(\mathbf{m})}{\partial \mathbf{m}(1)} = \frac{p(\mathbf{m} + v_1) - p(\mathbf{m} - v_1)}{\mathbf{r}_i(1)} \tag{20}$$

where $\mathbf{r}_i$ is the step size (see (16)), $\mathbf{m}(1)$ denotes the first dimension of $\mathbf{m}$ and $v_1 = [\mathbf{r}_i(1)/2, 0, ..., 0]^\top$. Recall that in (20) the parameter $\mathbf{m} \pm v_1$ is projected to the corresponding patch, *i.e.*, $\mathbf{S} = \xi(\mathbf{m})$ (see eq.(14)) in order to guarantee that this new quantity still belongs to the manifold $\mathcal{M}$.

Notice that the approach herein proposed resembles other gradient-based search methods on manifolds, for instance, the method studied by Helmke et al. [86], who propose a new optimization approach for the essential matrix computation with the use of Gauss-Newton iterations on a manifold in order to reduce the computational effort. Another similar example is the use of Newton's method on a manifold structure [87]–[89]. Our approach represents an application of such gradient-based search methods in the problem of top-down non-rigid segmentation.

Once the GA search reaches the solution at the Nth iteration, a DBN segmentation and respective confidence are obtained. Notice, however, that the GA process is applied to every patch in the manifold $\mathcal{M}$. The following section describes how to blend the proposal segmentations of the patches to produce the final segmentation $\mathbf{S}^\star$ (see (1)) for image $\mathbf{I}$.

X. Ensemble of DBN classifiers

To achieve the final segmentation $\mathbf{S}^\star$, we propose a combination of the segmentations from the patches (*i.e.*, segmentation proposals) using the DBN confidence. More specifically, the ensemble strategy of the DBN classifiers comprises the following threes steps: (i) for each patch compute the classification confidence, (ii) compute the corresponding contour of the above confidence, and (iii) compute the final segmentation combining the segmentation proposals of all patches.

In the first step, and taking into account that each patch $\mathcal{P}_i$ has a number of patch-member (or landmark) points, the GA procedure provides several confidences for each patch as follows

$$\mathcal{F}_{\mathcal{P}_i} = \{p(\mathbf{m}_{ij}|I\gamma_{\text{MAP}})\}_{\mathbf{m}_{ij} \in \mathcal{P}_i} \tag{21}$$

where $\mathbf{m}_{ij}$ is the estimate from the jth member point in the ith patch denoted by $\mathcal{P}_i$.

The segmentation proposal for each patch is obtained by

$$\mathbf{m}_i^\star = \arg\max_{\mathbf{m}_{ij} \in \mathcal{P}_i} p(\mathbf{m}_{ij}|\mathbf{I}, \gamma_{\text{MAP}}). \tag{22}$$

In the second step, the corresponding (high dimensional) contour is found by computing $\mathbf{S}_i^\star = \xi_i(\mathbf{m}_i^\star)$ (see (14)). Finally, the third step blends the DBN as follows:

$$\mathbf{S}^\star = \frac{1}{Z} \sum_{i=1}^{|\mathcal{P}|} \mathbf{S}_i^\star \times p(\mathbf{m}_i^\star|\mathbf{I}, \gamma_{\text{MAP}}) \tag{23}$$

Fig. 6 displays the segmentation process, where the level sets denote the results of the classifier $p(\mathbf{m}|\mathbf{I}, \gamma)$ that is used in the GA process.

TABLE III: Segmentation.

- **Input:** image $\mathbf{I}$;
- **Output:** segmentation $\mathbf{S}^\star \in \mathbb{R}^{2S}$

For each scale σ_i, and for each patch $\mathcal{P}_1, ..., \mathcal{P}_{|\mathcal{P}|}$ perform the segmentation as follows:

1) Take the patch-member $\mathbf{m} \in \mathcal{P}$ or landmark point $\mathbf{m} \in \mathcal{L}$ as an initial guess, *i.e.* $\mathbf{m}^{(0)} = \mathbf{m}$ to perform gradient ascent (GA) on the output of the classifier $p(\mathbf{m}|\mathbf{I}, \gamma_{\text{MAP}})$ (see Eqs. (19,20))
2) Perform the ensemble DBN classification, as follows
   a) Find the segmentation proposal $\mathbf{m}_i^\star$ for each patch (see Eq. (22))
   b) Find the corresponding (high dimensional) contour $\mathbf{S}_i^\star$
   c) Compute the final segmentation by performing the following blending rule (see Eq. (23))

$$\mathbf{S}^\star = \frac{1}{Z}\sum_i \mathbf{S}_i^\star \times p(\mathbf{m}_i^\star|\mathbf{I}, \gamma_{\text{MAP}})$$

Note that the search is performed on the low dimensional space of $\mathbf{m}$, and each patch has its own local maximum. Figure 11 illustrates the segmentation process with four segmentation proposals (green dots) along with the DBN confidences (top of each figure). On bottom image (Fig. 11), the final segmentation is estimated (23).

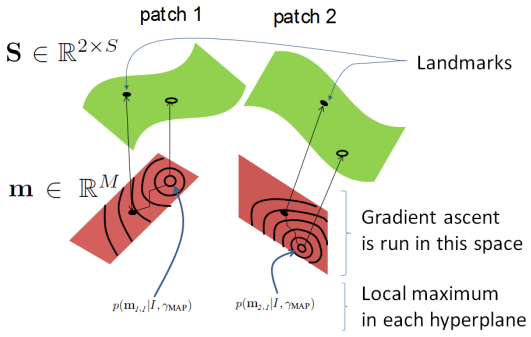


Fig. 6: Segmentation procedure in the manifold. The patches are also shown (top) and the gradient ascent (GA) is performed a low dimensional space (bottom) with intrinsic dimension M.

When considering the sparsity on the manifold, the ensemble strategy works in the same way, where the only difference is that the set of landmark points *i.e.* $\mathcal{L}$ replaces $\mathcal{P}$. The advantage is that the sparsity leads to a $|\mathcal{L}| \ll |\mathcal{P}|$, resulting in fewer computations in the inference. Table III shows the steps of the inference process to perform the segmentation using the ensemble of classifiers.

XI. Running-time Complexity Comparisons

This section provides a comparison concerning the running time complexity between the proposed approach and other approaches available in the literature. The complexity of the non-rigid segmentation approaches can be measured by the number of executions of the rigid and non-rigid classifiers.

Table IV shows the different complexities concerning several methodologies. A naive exhaustive search approach requires a complexity shown in the first row of the Table IV, if we quantize each of the $2 \times S$ dimensions into K samples, where $K = \mathcal{O}(10^3)$. The complexity of this naive approach has motivated the development of more efficient methods.

TABLE IV: Comparison of object segmentation complexity in different approaches.

Methodologies	Complexity
Naive approach	$\mathcal{O}(K^{2S})$
Exhaustive search (rigid + non-rigid))	$\mathcal{O}(K^R + S)$
Branch and bound [90]	$\mathcal{O}(K^{R/2} + S)$
Marginal space learning (MSL) [4]	$\mathcal{O}(K + (R-1) \times \sharp\text{scales} \times K_{\text{fine}} + S)$
Coarse-to-fine based methods [1,3,91]	$\mathcal{O}(K + \sharp\text{scales} \times K_{\text{fine}} \times R + S)$
Sparse manifolds in the rigid detection [3,6]	$\mathcal{O}((\sum_i \lvert\mathcal{L}_i\rvert) \times \sharp\text{scales} \times M + S)$
Proposed approach	$\mathcal{O}((\sum_i \lvert\mathcal{L}_i\rvert) \times \sharp\text{scales} \times M)$

The methodologies proposed in [1]–[5] try to reduce this complexity by using a rigid classifier in the intermediate space $\mathbf{t} \in \mathbb{R}^R$, where $R \in \{4, 5\}$, and a non-rigid classifier that runs in the space of $\mathbf{S} \in \mathbb{R}^{2\times S}$. (second row of the Table IV).

Other strategies have been proposed to reduce the segmentation complexity. This includes the *branch and bound* framework [90] (third row of Table IV) and *marginal space learning* (MSL) [4] (fourth row). In the latter approach, $\sharp$scales stands for the number of scales, and K_{fine} represents the number of hypotheses to be tested by the more complex classifiers in the model. Methodologies using $\sharp$scales are characterized as a *coarse-to-fine* based approaches. For instance, in [1,3,4,91], the number of scales is $\sharp\text{scales} = 3$ and the complexity of promising samples is assumed to be $\mathcal{O}(10^1)$. The above methods use gradient ascent (GA) method in the space of R dimensions (see fifth row of the Table IV).

Recent techniques using sparse manifolds [3] can reduce even more the complexity as shown in the sixth row of Table IV. In this case, the rigid detection is performed only in the set of landmarks (*i.e.* $\sum_i \mathcal{L}_i$) of the estimated manifold, with the dimension $M < R$. The non-rigid detection, however, takes place (as in the previous coarse-to-fine methodologies) at the image domain. As such, the reduction is achieved only in the rigid detection stage. In the present formulation, we are able to integrate both rigid and non-rigid stages in the manifold, thanks to the introduction of TPS deformations. This means that the complexity no longer depends on S (thus removing $\mathcal{O}(S)$) (see seventh row of Table IV).

As a final remark, we should mention that that our approach is *orthogonal* to all methods presented above, in the sense that any of these methods can use our approach to achieve even higher efficiency gains.

XII. Experimental Setup

In this section we show empirical evidence that the use of the proposed sparse low-dimensional manifold leads to less complex classifiers and to segmentation methods without a negative impact on the segmentation accuracy.

Three different databases are used to demonstrate the effectiveness of the proposed approach. The first database contains ultrasound (US) sequences of the LV of the heart [24], where the goal is to segment the LV endocardial border. The second database has a sub-set of the Cohn-Kanade (CK+) database [92], containing lip expressions in video sequences. The third data set comprises a publicly available data set from cardiac MRI sequences [93].

Note that the above datasets are chosen, since they share the conditions where the object of interest undergoes a rigid transformation (e.g. affine transform) followed by a non-rigid deformation. The main goal is to have a deformable object that changes its shape through time (i.e. sequences of images containing the same object that

suffered the two above transformations), and we wish to segment the object using an explicit representation, where neighboring keypoints in the segmentation are strongly correlated.

Concerning the problem of the LV segmentation, the data set used for the experiments comprises 12 sequences for training and testing (12 sequences from 12 subjects with no overlap), from which eight present some kind of cardiopathy. According to the cardiologist's report[4], the following cardiopathies/abnormalities are considered:

A total of 204 manual annotations were collected and the total number of frames acquired in the 12 sequences is 3993. The expert provided an average of 17 annotations per sequence with two annotations in the systole phase plus 2 annotations in the diastole phase. The annotation process was repeated during eight cardiac cycles in each LV sequence.

The second database is a sub-set of the Cohn-Kanade (CK+) database [92], where the objective is to segment the lips from video sequences of people demonstrating different types of emotions. The manual annotation of the lips has been provided. In order to test our proposed approach, we select the *surprise*, *happy* and *fear* emotions, which contains large shape deformation. We use 9 sequences for training (roughly 189 annotations), where we used three sequences for each of the emotions and each sequence has about 20-30 frames. For testing, we used: 12 "surprise" sequences (194 frames), 12 "happy" sequences (250 frames) and 15 "fear" sequences (261 frames). The dimensionality of the object shape is $S = 21$ for the LV and $S = 40$ for the lips.

The third dataset contains 33 sequences acquired from different subjects, where each sequence comprises 20 volumes, covering one cardiac cycle, and the number of slices in each volume ranges from 5 to 10, with a spacing of 6 to 13 mm. In this dataset both healthy and disease cases are present. The ground truth (GT) of the LV segmentation in each slice is publicity available.

For the experiments below, we extend the method in [1,91] (referred to as CAR1 and CAR2, respectively), where a coarse-to-fine approach based on deep belief networks (DBN) [84] is used for the segmentation procedure. The automatically learned sparse manifold (Sec. VII) is different depending on the dataset used. For the LV we obtain an intrinsic dimensionality of $M = 2$, with $|\mathcal{P}| = 13$ patches, 1270 patch member points and 27 landmark points, where the majority of the patches contains only one landmark (see Fig. 3). For the lip case, the obtained intrinsic dimension is $M = 2$, with $|\mathcal{P}| = 4$ with 103 patch members points (corresponding to frame used for training) and four landmark points.

In the training stage of the DBN, we used $|\mathcal{T}_+(i,j)| = \{10, 15, 20, 50\}$ positive and $|\mathcal{T}_-(i)| = \{100, 150, 200, 500\}$ negative samples (for both datasets)[5]. We follow the same learning procedure described in [91], which divides the initial training set into training and validation sets containing 80% and 20% of the original training set, respectively. This validation set is used to determine the following DBN parameters: a) number of nodes per hidden layer, and b) number of hidden layers.

A quantitative performance is conducted using the following error measures proposed in the literature for contour comparison: (i) *Hausdorff* (MAX) [94], (ii) *mean absolute distance* (MAD) [5], (iii) *Jaccard index* (JCD) [95], (iv) *mean sum of squared distance* (MSSD) and (v) *average* (AV) [24]. A comparison with the following baseline approaches for LV segmentation is presented: COM [2,5], CAR1 [1,91,96], CAR2 [3], and MMDA [24]. For the lip segmentation, we also provide a quantitative comparison between our approach and the baseline methods in [1,3] [6].

We also present the running time figures of the proposed method and that of CAR1 [1,91,96], CAR2 [3].

XIII. Results

This section provides an extensive evaluation and comparisons in terms of qualitative, quantitative and run time figures using the two datasets mentioned above. Section XIII-A addresses the results for several LV echocardiography sequences, Section XIII-B describes the results achieved for lip segmentation in human face expressions and Section XIII-C describes the results obtained in 3D LV Magnetic Resonance Imaging.

A. Segmentation of the LV Endocardium from 2D B-mode Echocardiogram

For testing the proposed methodology in the segmentation of the LV, we performed a leave-one-out cross validation (*i.e.*, given that we have 12 sequences, this implies a 12-fold cross validation). For the training stage the following steps are performed:

1) 12 different versions of the manifold are generated (see Section VII), where each version of the manifold is obtained using 11 sequences for training, leaving one sequence out for testing. This provides a set of landmark points for each of the 12 manifolds.
2) Given that we work with the configurations sets of $\{\{10, 100\}, \{15, 150\}, \{20, 200\}, \{50, 500\}\}$ for positive and negatives (*i.e.* data augmentation), and since a coarse-to-fine approach is used (*i.e.* $\sigma \in \{4, 8, 16\}$), a total of $11 \times 4 \times 3 = 132$ DBN classifiers are trained.
3) The obtained classifiers have a small number of hidden layers as follows: two layers for $\sigma =\in \{8, 16\}$ and four layers for $\sigma = 4$

For the segmentation stage, we have:

1) Each sequence has on average 17 manually annotated frames. Since four data augmentation configurations are used for every version of the manifold, we have $17 \times 12 \times 4 = 816$ LV segmentations.

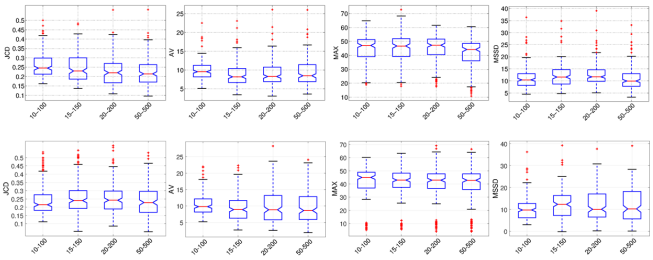


Fig. 7: LV segmentation using 12-fold cross validation. From left to right: Jaccard index (JCD), average (AV) mean absolute distance (MAD) and *mean sum of squared distance* (MSSD) as a function of the number of positives and negatives used during training. The top row shows the results using the proposed method with the landmarks and the bottom row is the baseline approach based on the patch member points.

In this experiment we compute the measures described in Sec. XII. The measures are shown in box-plots for several data augmentation

[4]This was done in collaboration with cardiologist from Hospital Fernando Fonseca who detailed each of the sequences.

[5]In [91] the adopted data augmentation was $|\mathcal{T}_-(i,j)| = \{10\}$, $|\mathcal{T}_+(i,j)| = \{100\}$.

[6]We have not provided the results of COM and CAR2 because they are not available for this database

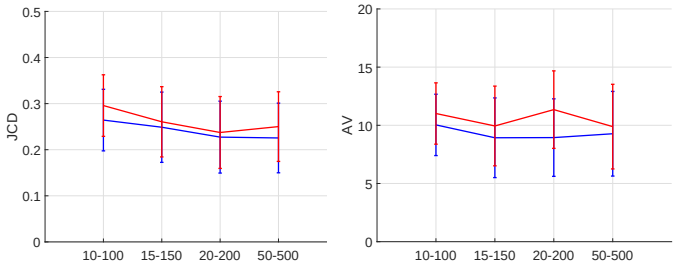


Fig. 8: Several positive-negative data augmentation configuration, using Jaccard index (a) and Average metric (b). The proposed data augmentation (blue) using elastic deformation is shown in blue, while the traditional data augmentation (with rigid deformations) is shown in red.

configurations. The results presented herein are calculated from 194 segmentations ($\approx$17 annotations $\times$ 12 sequences). In this experiment, we compare the baseline version of the framework without sparsity and the version based on sparse manifold. In the first version, several patch member points are obtained for each patch, whilst for the second, only sparse landmark points are considered. The main difference is that the use of patch member points results in a more computationally intensive GA (see (21)). The goal of this experiment is to demonstrate that the use of landmarks allows for a smaller computational cost while maintaining the accuracy of the segmentation when compared with the baseline version.

Fig. 7 compares the results obtained with the landmark points (top) against the baseline version of the proposal (bottom). It can be seen that, generally, the accuracy results does not change significantly with the variation of the number of positive and negatives (see JCD, MAX and MSSD measures) for both versions. Both versions show competitive results, where the Jaccard index is always below 0.2 and the average metric is lower than 10 pixels. Concerning the the MAX and MSSD measures, both versions provide a distance between 40-50 pixels and around 10-12 pixels, respectively.

1) Comparison with Classic Data Augmentation Strategy: In this section, we provide a comparison between the proposed data-augmentation (see Sec XIII-A) against a more traditional data augmentation. Basically, we aim at studying the gains of using the *elastic* deformations through TPS proposed compared with a more traditional data augmentation procedure based on affine transformation to the original training images, in order to produce new artificial training samples. More specifically, we generate new samples per training image following the previous configuration of positive and negatives, *i.e.* {(10-100), (15-150), (20-200),(50-500)}, where the transformation consists of randomly cropping the original training image from the top-left and bottom-right corners within a range of $[1, 10]$ pixels from the original corners and a small rotation within 15 degrees of the LV shape. Figure 8, shows the performance of the two DA strategies, using the Jaccard index (JCD) and average (AV) measures. The results follows the same 12-fold cross validation strategy as described above. From the results we conclude that the proposed data augmentation seems to be more effective at producing more realistic artificial data for training the models.

2) Comparison with Previously Proposed LV Segmentation Approaches: We also compare the proposed framework with other related approaches, which have been demonstrated to be successful at segmenting the LV (see Fig. 9). A particle filtering using deep learning classifiers proposed in [1,96] (CAR) that yields state-of-the-art performance in this dataset. An information fusion tracker proposed in [2,5] (COM) that decouples the uncertainty in terms of dynamics and statistical shape constraints introducing a unified framework that fuses both the subspace shape model, system dynamics, as well as the uncertainty measurements. Also, we perform a comparison with the Multiple Model Data Association tracker (MMDA) [24]. This tracker is based on a deformable active contour that switches in the prediction step between two pre-trained dynamical models (one contraction and expansion model to cope with systole and diastole phases, respectively). All the above trackers were run in $\mathcal{T}_1$ and $\mathcal{T}_2$ test sequences.

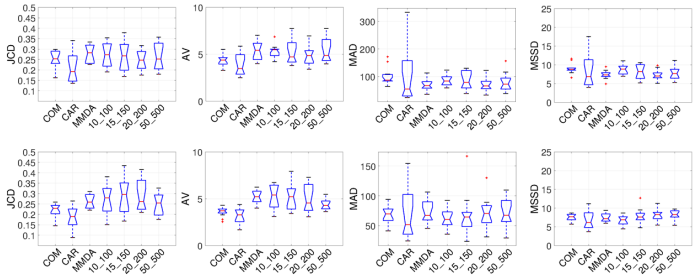


Fig. 9: Quantitative results using the test sequences $\mathcal{T}_1$ (top) and $\mathcal{T}_2$ (bottom). Quantitative comparison using (from left to right) JCD, AV, MAD and MSSD, measures. In each graph (from left to right in the box-plots) we show the performance of the algorithms COM, CAR, MMDA and the proposed framework for the $\{(10, 100), (15, 150), (20, 200), (50, 500)\}$ positive-negative configurations.

Fig. 9 shows a quantitative comparison of the LV segmentation accuracy using the measures described in Sec. XII for the first test sequence $\mathcal{T}_1$ (top row) and for the second test sequence $\mathcal{T}_2$ (bottom row). Fig. 10 shows some examples of the segmentations obtained with the proposed approach and with the other related techniques. It can be seen that the best results achieved belongs to CAR method. Also COM exhibits remarkable performance. The methodology herein proposed achieves a competitive accuracy performance compared to the other methods. We may argue that both CAR and COM are tracking procedures, meaning that temporal information is incorporated, while our approach does not explore the temporal information.

Fig. 11 shows partial the patch segmentations along with the corresponding (un-normalize) weights given by the DBN (left). On the right, we can see the linear combination as a final segmentation result (green) superimposed with the ground-truth (red). It is interesting to notice that the deep belief network gives more weight to the correct segmentations (see two top images).

Table VII shows a comparison with [6] in which the segmentation is partitioned in two stages: *rigid* and *non-rigid* (RNR). We see that [6] spends 2.37s for the segmentation from which 1.7s is for the rigid-detection and 0.67 for the non-rigid detection. The proposed framework spends 2.07s for the segmentation that corresponds to the non-rigid segmentation. Notice that the TPS parameterization (*i.e.* image warp) is negligible, taking only 0.017 sec. The accuracy performance of the proposed approach is similar to [6] regarding all metrics used. For both approaches the obtained best scores are JCD ≈ 0.25, AV < 6 pixel, and $5 <$ MSSD < 10 pixel.[7]

In order to measure the statistical significance of the results presented, we also perform the Wilcoxon signed-rank tests, assuming

[7]For comparison purposes the same configurations of positive-negative samples are used.

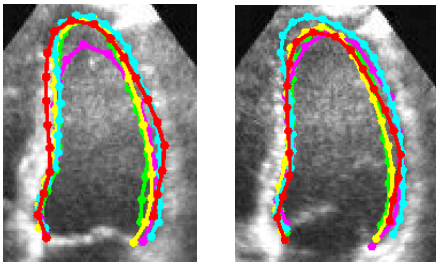
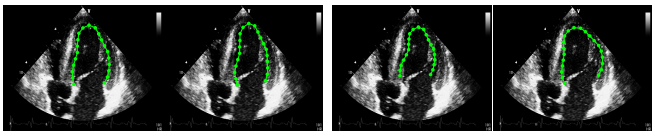
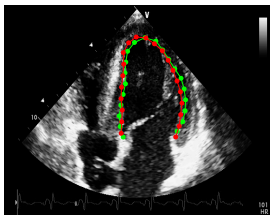


Fig. 10: Qualitative comparison for the test sequence. The segmentation of each tracker (and the expert annotation) is shown in different colors as follows: green (medical ground-truth), pink (COM tracker), yellow (CAR tracker), cyan (MMDA tracker) and red (proposed approach).



0.046 0.039 0.017 0.012

(a) Partial Segmentations



(b) Final segmentation

Fig. 11: Illustration of the LV segmentation in a test sequence. (Top) Segmentation of the most relevant patches with the corresponding (non-normalized) DBN weights (a) and the final segmentation (b). (Bottom) The estimated contours are in green and the ground truth in red.

that null hypothesis is the hypothesis that there is no difference between the results of the proposed method and the other methods. This is accomplished by computing the LV volumes of the ground truth (provided by an expert annotations) and the volumes provided by the contour estimates of the proposed framework. The objective of this study is to show that the results obtained with the proposed approach are competitive with the state of the art related methodologies. Assuming a significance level of 1%, (see Table V), we can conclude that we cannot reject the null hypothesis stated above.

B. Lip Segmentation

Fig. 12 illustrates the quantitative evaluation concerning the metrics described in Sec. XII. We see that competitive results are achieved

TABLE V: Wilcoxon signed-rank test (WSR test) between the volumes estimated with the proposed approach and with the CAR [1], COM [2,5] and MMDA approaches on the LV test sequences.

	Data augmentation training sizes			
	$10-100$	$15-150$	$20-200$	$50-500$
CAR	0.2368	0.0347	0.0879	0.0898
COM	0.2358	0.0597	0.1027	0.1495
MMDA	0.2366	0.2457	0.1671	0.2108

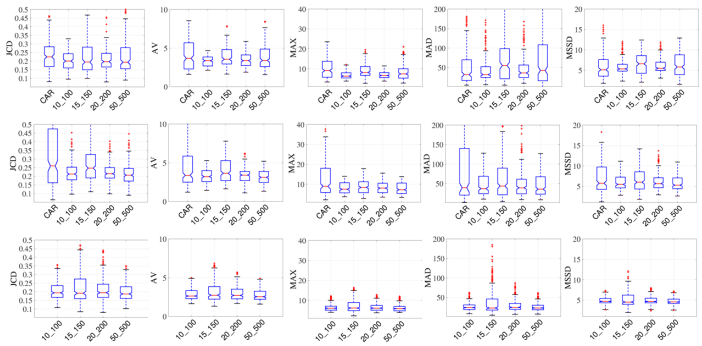


Fig. 12: Quantitative performance for: 12 "surprise" sequences (top), 12 "happy" sequences (middle) and 15 "fear" sequences (bottom) using the metrics mentioned in Sec. XII.

TABLE VI: Wilcoxon signed-rank test (WSR test) between the areas estimated with the proposed approach and with the CAR [1] on the Surprise and Happy test sequences.

	Data augmentation training sizes			
	$\{10-100\}$	$\{15-150\}$	$\{20-200\}$	$\{50-500\}$
Surprise	0.2893	0.2222	0.3113	0.2910
Happy	0.2425	0.2366	0.2600	0.1723

in all configuration of the data augmentation. The best results are achieved for the "fear" sequences (see bottom row of the table). This is perhaps due to the smaller lip motion when comparing with the other two sequences.

Fig. 13, show three snapshots of "surprise", (top row) "happy" (middle row) and "fear" (boom row) sequences, respectively. Each image corresponds to a given phase of the expression in a different subject. See supplementary material for more results in these sequences. As in the LV case, we also perform a statistical significance of the results on both sequences, we again assume that null hypothesis is the hypothesis that there is no difference between the results of the proposed method and the other methods. The results of the Wilcoxon signed-rank test are shown in Table VI, where we again cannot reject the null hypothesis. A comparison between the proposed methodology and with CAR is also conducted (see the bottom of the table where a graphical illustration is shown). To achieve these results we have to compute first the lip area computed from the 2D contours.

Table VII shows a comparison with [6] for the lip experiments. It shows that the two-stage approach [6] takes between 2.60 and 2.63 sec. for the segmentation ("surprise", "happy" and "fear" sequences), from which 2.41 to 2.44 sec. are spent on the *rigid detection* and 0.19 sec. on the *non-rigid segmentation*. The proposed framework provides an improvement over [6] where the time spent for the segmentation varies between 1.66 and 2.22 sec. The accuracy performance of the proposed approach is similar to [6] in the metrics used. The obtained best scores are JCD ≈ 0.23, AV ≈ 3 pixel, MAX < 10 pixel, MAD < 50 pixel, and $5 <$ MSSD < 10 pixel.[8]

C. Comparison with FCN - LV Segmentation in MRI

Due to the success of Convolutional Neural Networks (CNN) in several domains of application (*e.g.* [97]), we also compare the proposed framework with CNN in the problem of the LV segmentation in magnetic resonance imaging (MRI). As already detailed, the Fully

[8]These values are obtained for same the configurations of positive and negative samples, *i.e.* using $\{10-100\}, \{20-200\}$.

Fig. 13: Lip segmentation results produced by our proposal (red) jointly with the CAR estimates (blue) and the manual annotations (green) for "surprise" (top row) and "happy" (middle row). At the bottom it is shown the results produced by our proposal (red), superimposed with the ground truth (green) for fear sequences.

TABLE VII: Running time figures for the LV and Lips sequences, comparison with [6]. The results are shown in sec. per frame.

Sequences		RNR [6]	Proposal
LV	$(\mathcal{T}_1, \mathcal{T}_2)$	2.37 = 1.7 (R) + 0.67 (NR)	**1.7** = 1.68 (NR) + 0.017 (TPS)
Lip	Surprise	2.63 = 2.44 (R) + 0.19 (NR)	**2.07** = 2.07 (NR) + 0.002 (TPS)
	Happy	2.60 = 2.41 (R) + 0.19 (NR)	**1.66** = 1.66 (NR) + 0.002 (TPS)
	Fear	2.60 = 2.41 (R) + 0.19 (NR)	**2.22** = 2.22 (NR) + 0.002 (TPS)

Convolutional Network (FCN), based on CNNs [72] is trained to produce a pixel wise semantic segmentation, which is the problem being addressed in this paper.

For this purpose, we use a publicly available data set from cardiac MRI sequences [93] that contains 33 sequences acquired from different subjects, where each sequence comprises 20 volumes.

In our experimental setup, the FCN receives an input image of size 101×101, and the architecture of the network (see Fig. 14) is as follows: the first convolutional stage has 50 5×5 filters followed by a ReLU layer and a max-pooling that sub-samples the input by 2 (48×48); the second convolutional stage has 250 5×5 filters followed by a ReLU layer (44×44) and a max-pooling that sub-samples the input by 2 (22×22); the third (18×18), fourth (14×14) and fifth (10×10) convolutional stages have 500 5×5 filters, each followed by a ReLU layer with no subsampling; and finally one deconvolutional (23×23) stage with 500 5×5. Thus, in total we have $L = 14$ layers. In the training process, the learning rate is fixed at 0.0001 and momentum is equal to 0.9; the batchSize = 10; the number of epochs is 100; the weight decay is 0.0005.

We follow a leave-one-subject-out cross validation, *i.e.* 20 volumes for testing and 20×32 volumes for training. Also, for the proposed methodology, we follow the strategy described in Sec. XIII.

For the FCN segmentation we have to estimate the threshold from the training data. Notice that the FCN provides a probability map (see Fig. 15 middle column) which have to be binarized to obtain the final segmentation (see Fig. 15 right column). For each fold (*i.e.* test subject p), this process is based on the following steps:

1) Obtain the FCN segmentation map from the training set represented as $\mathrm{FCN}^{\mathrm{trn}}_{\mathrm{map}}$ and obtain threshold $T(p)$ from that training set,
2) Obtain the FCN segmentation map for the test subject p, which is denoted by, $\mathrm{FCN}^{\mathrm{tst}}_{\mathrm{map}}(p)$,
3) Obtain the final segmentation for the test subject p, with the threshold $T(p)$ producing $\mathrm{FCN}^{\star}_{\mathrm{seg}}(p)$

The steps 1), 2) are related to the training that are detailed in the supplementary material (see Algorithm 1). The step 3) is related with the evaluation of the model and it is detailed in the supplementary material (see Algorithm 2).

Figure 15 shows several LV segmentation examples in MRI taken in basal slice (top row) and apical slice (bottom row). More specifically, it is shown the input MRI image of the LV (left column), the segmentation map $\mathrm{FCN}^{tst}_{\mathrm{MAP}}(p)$ (middle column), and the segmentation $\mathrm{FCN}^{\star}_{seg}(p)$ (right column) obtained using the the Algorithms 1,2 (see supplementary material), where, it is shown the ground truth $\mathrm{GT}(p)$ delineated by a red line, (see more qualitative results in the supplementary material).

Figure 16 shows a quantitative comparison between the FCN (left) and the proposed method (right) using the Intersection over Union (IoU) coefficient. Each boxplot refers to the segmentations obtained for the 20 volumes of each subject. It can be seen that the performance of the FCN is similar, where the proposed framework achieves competitive segmentation accuracy.

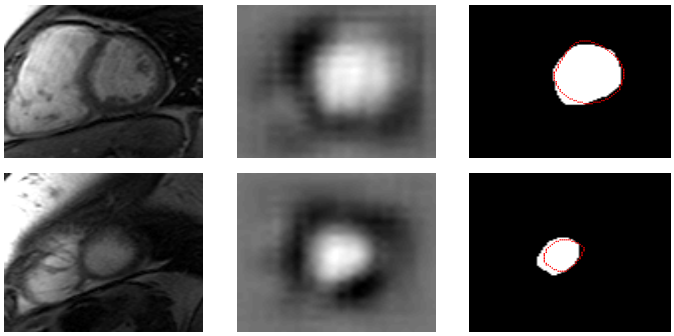


Fig. 15: Segmentation of the LV Endocardium using the FCN. Input image (left), probability map of the FCN (middle), segmentation after the threshold operation in red color (right). The white part is the segmentation while the red line is the ground truth.

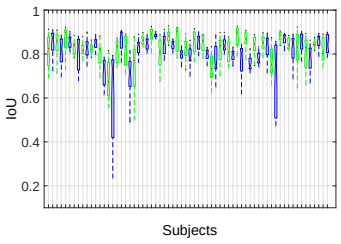


Fig. 16: Accuracy in the FCN semantic segmentation (green), and the proposed segmentation (blue) using the IoU metric. The mean(std) values are: 0.801(0.05) for the FCN and 0.785(0.05) for the proposed approach.

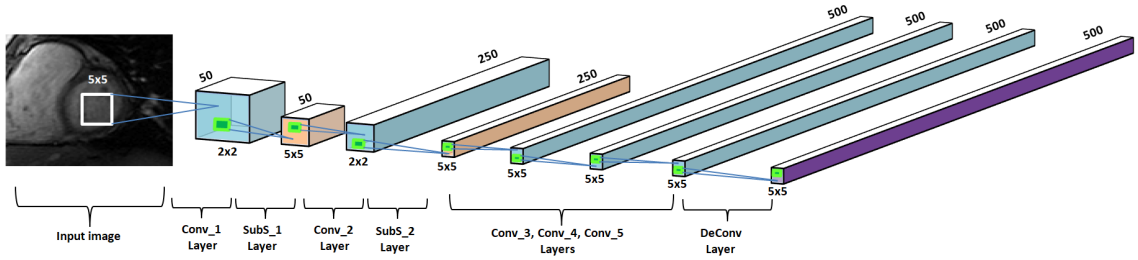


Fig. 14: Architecture of the FCN used for comparison (see text).

XIV. Concluding remarks

This paper proposes a novel deep learning method for object segmentation with low training complexity that needs small training sets. These two advantages speed up the training time when compared to recent state-of-the-art approaches. We are able to address the above challenges with a manifold learning strategy for the optimisation that results in the segmentation. The combination of deep and manifold learning is still scarce in the literature, but we anticipate that the use of these techniques will grow in the near future. An extensive study in Sec. XIII provides evidence of the robustness of the methodology when small training datasets are provided (see the positive-negative configurations in the experiments). The training times are indeed small – we obtained training times under four hours to train the three-scale classifiers with 50-500 (positive-negative) samples and under two hours for the remaining training configurations. This is possible given the low complexity of the classifier structures (see Sec. XIII-A). Naturally some issues are raised using low dimensional data for segmenting contours. One of the issues is the *patch selection* used in the initial guess for the optimisation process. We handle this issue by proposing a novel and effective strategy, where all the patches in the manifold are taken into consideration for the segmentation. Since the reliability of each patch varies, we use the deep belief networks confidence to produce a final segmentation.

XV. Discussion limitations and Future Work

In this paper, we show that it is possible to have a machine learning based segmentation system that operates directly on the space combining the rigid and non-rigid deformations. We should highlight that the non-rigid deformation is incorporated via thin plate splines, allowing the net to learn object deformations. This constitutes new way to perform the regularization of the network. Also, we show evidence that this space can be represented with manifolds of low dimensionality and by associating points in this manifold to segmentation probability values (given a test image), it is possible to run a gradient ascent algorithm that quickly finds the correct segmentation. Moreover, the reduced dimensionality of this manifold also constrains the complexity of the trained model, which further reduces the search complexity. In our experiments, we show that our approach efficient when compared to other state-of-the-art approaches [1,3,91], while producing competitive results in terms of accuracy. We also show that the models trained are less complex than the ones used by other approaches [1,91].

One of the difficulties of our approach that we plan to address in the future is with respect to its generalization capability. More specifically, if a test sample presents rigid and non-rigid transform parameters that are substantially different from the ones in the training set, our approach may fail to converge. Another difficulty is that the proposed regularization technique is tailored to objects that suffer non-rigid deformations. For other types of deformations, our approach may not be the most appropriate. As a final remark, we should stress that we plan to extend this approach to tracking problems (*i.e.*, segmentation in space and time) with the introduction of a motion model that works directly in this manifold of low dimensionality [98].

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.
[2] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta, "Databased-guided segmentation of anatomical structures with complex appearance," in *CVPR*, 2005.
[3] J. C. Nascimento and G. Carneiro, "Top-down segmentation of non-rigid visual objects using derivative-based search on sparse manifolds," in *CVPR*, 2013.
[4] 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.
[5] 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.
[6] J. C. Nascimento and G. Carneiro, "Deep learning on sparse manifolds for faster object segmentation," *IEEE Trans. Image Processing*, vol. 26, no. 10, pp. 4978–4990, 2017.
[7] ——, "Non-rigid segmentation using sparse low dimensional manifolds and deep belief networks," in *CVPR*, 2014.
[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 321–331, 1987.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] G. Jacob, J. A. Noblef, 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," in *ECCV*, 1998.

[30] ——, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[31] J. A. Noble, N. Navab, and H. Becher, "Ultrasonic image analysis and image-guided interventions," *Interface focus*, vol. 1, no. 4, pp. 673–685, 2011.

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

[33] 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.

[34] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[35] C. Kotropoulos and I. Pitas, "Segmentation of ultrasonic images using support vector machines," *Pattern Recognition Letters*, vol. 24, no. 4, pp. 715–727, 2003.

[36] Z. Yu, H.-S. Wong, and G. Wen, "A modified support vector machine and its application to image segmentation," *Image and Vision Computing*, vol. 29, no. 1, pp. 29–40, 2011.

[37] S. Bauer, L.-P. Nolte, and M. Reyes, "Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization," in *MICCAI*, 2011, pp. 354–361.

[38] R. Huang, V. Pavlovic, and D. N. Metaxas, "A graphical model framework for coupling mrfs and deformable models," in *CVPR*, 2004.

[39] Z. Kato and J. Zerubia, "Markov random fields in image segmentation, now," *Foundations and Trends in Signal Processing*, vol. 5, pp. 1–155, 2012.

[40] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001, pp. 282–289.

[41] G. Tsechpenakis and D. N. Metaxas, "Crf-driven implicit deformable model," in *CVPR*, 2007.

[42] D. Cobzas and M. Schmidt, "Increased discrimination in level set methods with embedded conditional random fields," in *CVPR*, 2009.

[43] S. Kumar and M. Hebert, "Discriminative fields for modeling spatial dependencies in natural images," in *NIPS*, 2003.

[44] C.-H. Lee, M. Schmidt, A. Murtha, A. Bistritz, J. Sander, and R. Greiner., "Segmenting brain tumors with conditional random fields and support vector machines," in *Computer Vision for Biomedical Image Applications*, 2005, vol. 3765, pp. 469–478.

[45] C. Sutton and A. McCallum, *Introduction to Conditional Random Fields for Relational Learning*. MIT Press, 2006.

[46] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," 2001.

[47] L. Ye, Z. Liu, L. Li, L. Shen, C. Bai, and Y. Wang, "Salient object segmentation via effective integration of saliency and objectness," *IEEE Trans. on Multimedia*, vol. 19, no. 8, pp. 1742–1756, 2017.

[48] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," 2013.

[49] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, 2012.

[50] Z. Liu, W. Zou, and O. LeMeur, "Saliency tree: A novel saliency detection framework," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1937–1952, 2014.

[51] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2281, 2012.

[52] A. Borji, H. J. M. Cheng, and J.Li, "Salient object detection: A benchmark," *IEEE Trans. on Image Processing*, vol. 24, no. 12, p. 2015, 5706-5722.

[53] M. Cabezas, A. Oliver, X. Lladó, J. Freixenet, and M. B. Cuadra, "A review of atlas-based segmentation for magnetic resonance brain images," *Computer methods and programs in biomedicine*, vol. 104, no. 3, pp. e158–e177, 2011.

[54] T. Rohlfing, R. Brandt, R. Menzel, D. B. Russakoff, and C. R. M. Jr., "Quo vadis, atlas-based segmentation?" in *Handbook of Biomedical Image Analysis*, 2005, pp. 435–486.

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

[56] R. Kikinis, M. E. Shenton, D. V. Iosifescu, R. W. McCarley, P. Saiviroonporn, H. H. Hokama, A. Robatino, D. Metcalf, C. G. Wible, and C. M. Portas, et al., "A digital brain atlas for surgical planning, model-driven segmentation, and teaching," *IEEE Trans. Med. Imag.*, vol. 2, no. 3, pp. 232–241, 1996.

[57] P. Viola and W. M. Wells, "Alignment by maximization of mutual information," *Int. journal of comp. vision*, vol. 24, no. 2, pp. 137–154, 1997.

[58] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 187–198, 1997.

[59] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3d medical image alignment," *Pattern recognition*, vol. 32, no. 1, pp. 71–86, 1999.

[60] J. Kittler and F. M. Alkoot, "Sum versus vote fusion in multiple classifier systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 1, pp. 110–115, 2003.

[61] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

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

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

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

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

[66] T. Chen and D. N. Metaxas, "Image segmentation based on the integration of markov random fields and deformable models," in *MICCAI*, 2000, pp. 256–265.

[67] T. Ngo and G. Carneiro, "Left ventricle segmentation from cardiac mri combining level set methods with deep belief networks," in *CVPR*, 2014.

[68] N. Dhungel, G. Carneiro, and A. P. Bradley, "Deep learning and structure prediction for the segmentation of mass in mammograms," in *MICCAI*, 2015.
[69] T. D. R. B. Girshick, J. Donahue and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.
[70] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.
[71] R. B. G. Shaoqing Ren, K. He and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.
[72] E. S. J. Long and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015.
[73] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. S. Hornung1, "Learning video object segmentation from static images," 2017.
[74] F. L. Bookstein, "Principal warps: thin-plate splines and decomposition of deformations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 6, pp. 567–585, 1989.
[75] M. H. Davis, A. Khotanzad, D. Flamig, and S. Harms, "A physics-based coordinate transformation for 3-d image matching," *IEEE Trans. Med. Imag.*, vol. 16, no. 3, pp. 317–328, 1997.
[76] G. Donato and S. Belongie, "Approximate thin plate splines mappings," in *ECCV*, 2002.
[77] J. Lim and M.-H. Yang, "A direct method for modeling non-rigid motion with thin plate spline," in *CVPR*, 2005.
[78] A. Zandifar, S. Lim, R. Duraiswami, N. Gumerov, and L. S. Davis, "Multi-level fast multipole method for thin plate spline evaluation," in *ICIP*, vol. 3, 2004.
[79] J. C. Nascimento and J. G. Silva, "Manifold learning for object tracking with multiple motion dynamics," in *ECCV*, 2010.
[80] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[81] 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.
[82] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001.
[83] 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.
[84] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[85] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.
[86] U. Helmke, K. Huper, 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.
[87] 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.
[88] 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.
[89] 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.
[90] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *CVPR*, 2008.
[91] 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.
[92] 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.
[93] A. Andreopoulos and J. K. Tsotsos, "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac mri," *Medical Image Analysis*, vol. 12, no. 3, pp. 335–357, 2008.
[94] 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.
[95] A. Hammoude, "Computer-assited endocardial border identification from a sequence of two-dimensional echocardiographic images," Ph.D. dissertation, University Washington, 1988.
[96] G. Carneiro and J. C. Nascimento, "Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, no. 11, pp. 2592–2607, 2013.
[97] L. Oakden-Rayner, G. Carneiro, T. Bessen, J. C. Nascimento, A. Bradley, and L. Palmer, "Precision radiology: Predicting longevity using feature engineering and deep learning methods in a radiomics framework," *Nature Scientific Reports*, vol. 7:1648, 2017.
[98] L. Yang, B. Georgescu, Y. Zheng, D. J. Foran, and D. Comaniciu, "A fast and accurate tracking algorithm of left ventricles in 3d echocardiography," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*. IEEE, 2008, pp. 221–224.

Jacinto C. Nascimento (S'00 - M'06- S'19) received the E.E. degree from the Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Instituto Superior Tecnico(IST), Technical University of Lisbon, Lisbon, in 1995, 1998, and 2003, respectively. He has been with the Faculty of the Department of Informatics and Computer Engineering, IST, since 2013, where he is currently a Assistant Professor. He is the recipient of several research grants and fellowships from Science and Technology Foundation (FCT - Portugal) His research interests include statistical image processing, pattern recognition, machine learning, medical imaging analysis, video surveillance, general visual object classification. He has coauthored more than 50 publications in international journals (many of which are sponsored by the IEEE), has served on program committees of many international conferences, and has been a reviewer for several international journals.

Gustavo Carneiro is a professor of the School of Computer Science at the University of Adelaide and the Director of Medical Machine Learning at the Australian Institute of Machine Learning. He is the recipient of numerous research grants and fellowships from the Australian Research Council (ARC), Humboldt Foundation, Marie Curie Actions, Science and Technology Foundation (FCT - Portugal) and Natural Sciences and Engineering Research Council (NSERC - Canada). In the past, he worked at Siemens Corporate Research, University of British Columbia, and University of California San Diego. Prof. Carneiro received his Ph.D. in computer science from the University of Toronto in 2004 and his main research interests are in the fields of computer vision, medical image analysis and machine learning.