Non-rigid Segmentation using Sparse Low Dimensional Manifolds and Deep Belief Networks [*]

Jacinto C. Nascimento
Instituto de Sistemas e Robótica
Instituto Superior Técnico, Portugal

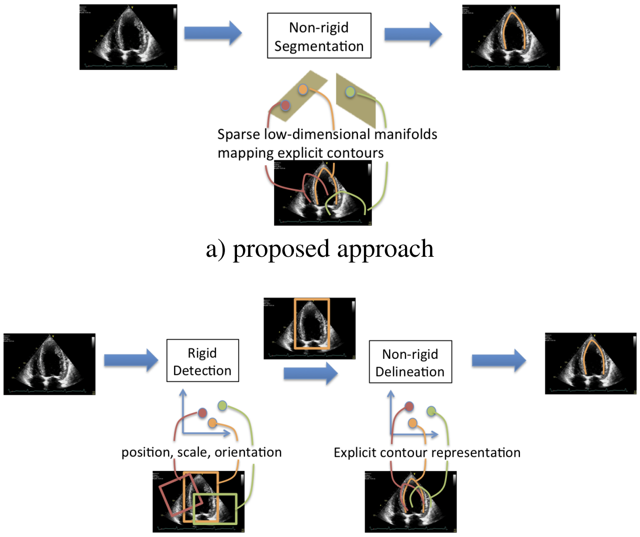
Gustavo Carneiro
Australian Centre for Visual Technologies
The University of Adelaide, Australia

Abstract

In this paper, we propose a new 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. Our main contribution is the fact that we do not rely on the typical sub-division of segmentation tasks into rigid detection and non-rigid delineation. Instead, the non-rigid segmentation is performed directly, where points in the sparse low-dimensional can be mapped to an explicit contour representation in image space. Our proposal shows significantly smaller search and training complexities given that the dimensionality of the manifold is much smaller than the dimensionality of the search spaces for rigid detection and non-rigid delineation aforementioned, and that we no longer require a two-stage segmentation process. We focus 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. Our experiments show that the use of sparse low dimensional manifolds reduces the search and training complexities of current segmentation approaches without a significant impact on the segmentation accuracy shown by state-of-the-art approaches.

1. Introduction

Current methodologies for top-down segmentation of deformable objects using machine learning techniques usually divide the problem into two stages [1, 2, 3, 4, 5]: (i) rigid detection followed by (ii) non-rigid segmentation. The fundamental reason for having this first stage is to reduce the complexity of the training and search mechanisms. For instance, if we have a contour represented by S 2-D points, a naive exhaustive search (e.g., by quantizing each of the $2 \times S$ dimensions into K samples) leads 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 $\mathbf{t} \in \mathbb{R}^R$ (with $R << 2S$) that estimates the translation, scale and rotation transformations of a mean contour (note that R represents the dimensionality of the search space). Furthermore, the resulting transformed mean contour is used to initialize and constrain the non-rigid segmentation, which decreases even more the search and training complexities of the methodology. The final search and training complexities of such approaches are usually dominated by the rigid detection, which is a function of the rigid transformation space dimensionality R.


Non-rigid Segmentation
Sparse low-dimensional manifolds mapping explicit contours

a) proposed approach

Rigid Detection
Non-rigid Delineation
position, scale, orientation
Explicit contour representation

b) typical non-rigid segmentation approach [1, 2, 3, 4, 5]

Figure 1. Proposed approach compared to the typical 2-step non-rigid segmentation.

In this paper, we propose a new 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. Our main contribution is the fact that we do not rely on the typical sub-division of seg-

[*] This work was supported by the FCT project [PEst-OE/EEI/LA0009/2013] and by the Australian Research Council's Discovery Projects funding scheme (project DP140102794).

mentation tasks into rigid detection and non-rigid delineation (Fig. 1 shows a comparison between our proposal and the typical non-rigid segmentation approach found in the literature). The objectives of this paper are the following: 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, 2, 3, 4, 5], which are based on the typical sub-division of the segmentation problem into rigid detection and non-rigid delineation (Fig. 1-(b))[1]. Moreover, the use of manifold in non-rigid segmentation problems has been explored in a slightly different fashion by Yang et al. [6], who use a manifold to learn a motion model instead of a shape model. Finally, it can be argued that this work lies in the realm of statistical shape model [7], but note that the ideas presented here can in principle be extended to other shape models that reduce the dimensionality of the shape representation (however, this extension is out of the scope of this paper).

In order to test the efficacy of our approach, we apply it to *two non-rigid segmentation problems*. The *first problem* is the segmentation of the left ventricle (LV) of the heart from ultrasound images and the *second problem* is the segmentation of lip boundary from video sequences. We show that our approach obtains a significant reduction in terms of search and training complexities without affecting the segmentation accuracy produced by state-of-the-art methodologies.

2. State-of-the-art Non-rigid Segmentation

In this section, we briefly describe how the problem of non-rigid segmentation using machine learning techniques is addressed by current state-of-the-art methodologies. Consider that an grey-scale image $I_j : \Omega \rightarrow [0, 255]$ (Ω denotes the image space) has a region containing the visual object of interest, which is explicitly represented by a list of S 2-D points (this is also known as the annotation), forming a matrix $\mathbf{S} \in \mathbb{R}^{2\times S}$ (see for example Fig. 5 that shows several annotations of two different types of visual objects - left ventricle and lips). Assume that a training set containing several images and their respective annotations is available and represented by $\mathcal{D} = \{(I, \mathbf{S})_j\}_{j=1}^{|\mathcal{D}|}$. The optimal segmentation is found using the following optimization problem:

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

where this function denotes the probability of finding a non-rigid segmentation $\mathbf{S}$ in image I, assuming a model is learned from the training set $\mathcal{D}$. In general, the high dimensionality of $\mathbf{S}$ makes the direct optimization of (1) highly complex, and a common solution adopted to reduce this complexity is a divide-and-conquer type of algorithm, where preliminary lower dimensional problems are introduced and summed out. For instance, several approaches [1, 2, 3, 4, 5] introduce one preliminary problem represented by a hidden variable $\mathbf{t} \in \mathbb{R}^R$, with $R << (2 \times S)$, leading to the following new problem formulation:

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

In general, the variable $\mathbf{t}$ in (2) represents a rigid transform that is applied to the coordinates of a canonical contour $\mathbf{C} \in \mathbb{R}^{2\times S}$ in order to move them to the image space Ω. Then, the search for the segmentation contour $\mathbf{S}$ is performed around the points of this transformed contour. The canonical contour $\mathbf{C}$ is usually represented by the mean shape of the sought segmentation shape represented in a grid space $\mathbf{G_C} \in \mathbb{R}^{2\times G}$, forming a rectangular 2-D region. The canonical contour is transformed to a region of the image space via a linear transformation matrix $\mathbf{A} \in \mathbb{R}^{3\times 3}$, which is obtained from the variable $\mathbf{t}$ as follows [1, 2, 3, 4, 5]: $\mathbf{A_t} = h(\mathbf{t})$ [2].

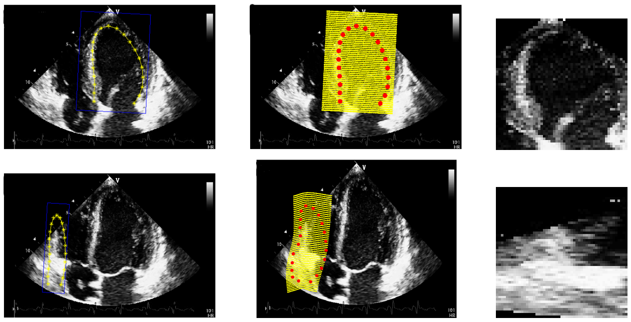
The rigid detection represented by the term $p(\mathbf{t}|I, \mathcal{D})$ in (2) computes the probability that the visual object underwent a transform represented by $\mathbf{t}$ in image I. In practice, the rigid classifier $p(\mathbf{t}|I, \mathcal{D})$ receives an image patch $I(g(\mathbf{t}))$, with $g(\mathbf{t}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{A_t}[\mathbf{G}_{\mathbf{C}}^{\top}, \mathbf{1}_G]^{\top}$ and returns the probability that the input sub-window contains the structure of interest.

The non-rigid delineation represented by term $p(\mathbf{S}|\mathbf{t}, I, \mathcal{D})$ in (2) computes the probability of the segmentation $\mathbf{S}$ in image I given the value of $\mathbf{t}$ (*i.e.*, $\mathbf{t}$ constrains and initializes the search for $\mathbf{S}$ to be around the image patch $I(g(\mathbf{t}))$.

It is important to 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 $p(\mathbf{t}|I, \mathcal{D})$ in (2) becomes more complex, requiring larger amounts of data to avoid over-fitting.

[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.

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



a) contour b) grid c) patch

Figure 2. Generation of samples $I(g(\mathbf{m}))$ that is used as an input for the proposed classifier $p(\mathbf{m}|I, \mathcal{D})$ in (3). The 1st row shows a positive sample and the 2nd row shows a negative sample. The first column displays the contour, the second column shows the non-rigidly deformed grid $\widetilde{\mathbf{G}}_{\mathbf{C}}$, and the last column displays the non-rigidly deformed input patch $I(g(\mathbf{m}))$.

3. Proposed Non-rigid Segmentation using Sparse Low Dimensional Manifolds

We propose a re-formulation of the optimization problem in (1) as follows:

$$\mathbf{m}^* = \arg\max_{\mathbf{m}} p(\mathbf{m}|I, \mathcal{D}), \tag{3}$$

where $\mathbf{m} \in \mathbb{R}^M$ is a point in a low dimensional manifold, which is used to produce $\mathbf{S}^*$, as described below in Sec. 3.2. Notice that in our proposal, we no longer require an intermediate rigid detection because we estimate directly a non-rigid contour segmentation via $\mathbf{m}$, with dimension $M < R << S$. In order to use the same types of classifiers as the ones described in Sec. 2, 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.

3.1. Thin-Plate Splines

The thin-plate splines TPS is a tool for modeling coordinate mappings [8]. In our case, the TPS allows the mapping from the grid $\mathbf{G}_{\mathbf{C}}$, used to represent the canonical contour $\mathbf{C}$, to the non-rigidly deformed grid $\widetilde{\mathbf{G}}_{\mathbf{C}}$, as follows:

$$[\widetilde{\mathbf{G}}_{\mathbf{C}}^\top, \mathbf{1}_G] = [\mathbf{G}_{\mathbf{C}}^\top, \mathbf{1}_G]\widetilde{\mathbf{A}} + [{\mathbf{K}_{\mathbf{G}}}^\top \mathbf{w}_x, {\mathbf{K}_{\mathbf{G}}}^\top \mathbf{w}_y, \mathbf{0}_G] \tag{4}$$

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 $\mathbf{K}_{\mathbf{G}} \in \mathbb{R}^{S\times G}$, with $\mathbf{K}_{\mathbf{G}}(i,q) = U((\sum_{j=1}^{3}(c_{ij} - g_{qj})^2)^{1/2})$, $U(r) = r^2\log(r)$, c_{ij} being 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 linear system for the TPS coefficients

$$\begin{bmatrix} \mathbf{w}_x & \mathbf{w}_y \\ \widetilde{\mathbf{a}}_1 & \widetilde{\mathbf{a}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{\mathbf{C}} & [\mathbf{1}_S, \mathbf{C}^\top] \\ [\mathbf{1}_S, \mathbf{C}^\top]^\top & \mathbf{0}_{3\times 3} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{S}^\top \\ \mathbf{0}_{3\times 2} \end{bmatrix} \tag{5}$$

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 $\mathbf{K}_{\mathbf{C}} \in \mathbb{R}^{S\times S}$ 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.

As we did for $g(\mathbf{t})$, we can now write $g(\mathbf{m}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} [\widetilde{\mathbf{G}}_{\mathbf{C}}^\top, \mathbf{1}_G]^\top$. The difference of using $I(g(\mathbf{m}))$ instead of $I(g(\mathbf{t}))$ is that it allows to obtain a patch that underwent a non-rigid deformation (see Fig. 2).

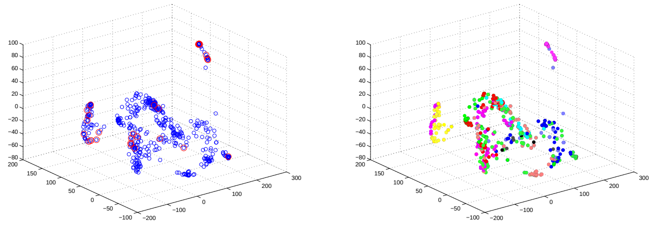
3.2. Sparse Low Dimensional Manifold

This section describes how to map contours $\mathbf{S}$ to and from a manifold $\mathcal{M}$ using a lower dimensional variable $\mathbf{m}$, which is used in the optimization (3). We follow the manifold learning of [9] that is based on the *tangent bundle* concept of an M-dimensional manifold $\mathcal{M}$ and works by building and assembling multiple local models in an agglomerative fashion. More specifically, given a set of contours $\mathbf{S}$, it finds the intrinsic dimension M, builds a partition into $|\mathcal{P}|$ patches, and estimates the forward-backward mappings between contours $\mathbf{S}$ and respective lower representations $\mathbf{m} \in \mathbb{R}^M$, that is, the *charts* $\mathbf{m} = \zeta_i(\mathbf{S})$ and the *parameterizations* $\mathbf{S} = \xi_i(\mathbf{m})$, respectively.

The search process for the optimization in (3) takes place in each of the low dimensional patches $\mathcal{P}_i$ with initial guesses denoted by the patch member points $\mathbf{m}_{i,j} = \zeta_i(\mathbf{S}_{i,j})$, for $i = 1, ..., |\mathcal{P}|$ (i.e., index of patches), and $j = 1, ..., |\mathcal{P}_i|$ (index of patch member points). 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*.

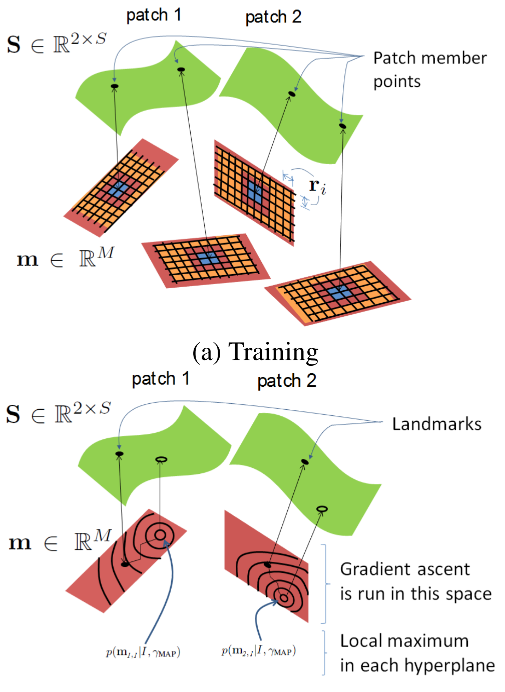
The selection of the *landmarks* is based on the solution of a regression problem that minimizes a regularized cost function [10]. For instance, if we have the patch member points represented by the set $\mathcal{Q}_i = \{\mathbf{m}_1, ..., \mathbf{m}_{|\mathcal{P}_i|}\}$, after the landmark selection we are able to obtain a subset $\mathcal{L}_i \subseteq \mathcal{Q}_i$ of size $|\mathcal{L}_i|$, which corresponds to the number of landmarks in the patch i. 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 $i \neq j$.

Fig. 3 illustrates the input and output of the manifold learning algorithm run on the experimental setup of the LV segmentation problem that will be described in Sec. 6.



(a) Annotations and landmarks (b) Patches learned

Figure 3. The graph in (a) shows the input to the sparse low dimensional manifold learning (Sec. 3.2) for the LV segmentation problem, with 496 annotations after a PCA reduction in blue circles (the first three components are shown). The manifold learning algorithm estimates 1270 patch member points distributed in 13 patches. In red it is shown the 47 landmarks estimated [11]. The graph in (b) shows the 13 patches found using the training algorithm, where each separate patch is represented with a different color.


patch 1
patch 2
$\mathbf{S} \in \mathbb{R}^{2\times S}$
Patch member points
$\mathbf{r}_i$
$\mathbf{m} \in \mathbb{R}^M$
patch 1
patch 2
$\mathbf{S} \in \mathbb{R}^{2\times S}$
Landmarks
$\mathbf{m} \in \mathbb{R}^M$
Gradient ascent is run in this space
Local maximum in each hyperplane
$p(\mathbf{m}_{1,1}|I,\gamma_{\text{MAP}})$
$p(\mathbf{m}_{2,1}|I,\gamma_{\text{MAP}})$

(a) Training

(b) Segmentation

Figure 4. Training (top) and segmentation (bottom) procedures (please see text for details).

4. Training and Segmentation using the Sparse Low Dimensional Manifold

Usually, the training set available for the most common segmentation problems does not provide enough information for a robust training process, so the usual remedy consists of generating artificial positive and negative training samples by randomly perturbing the deformation parameters of the annotated data [12]. These perturbations usually follow simple noise distributions (e.g., Gaussian) in the deformation parameter space, which form artificial training samples that might never happen in practice. In addition, the large dimensionality of this parameter space means that a robust training can only be achieved with a reasonably large training set. We also augment our training sets by perturbing the training set annotations, but differently from current approaches, the artificial training samples are re-projected onto the learned manifold. There are two advantages associated with our approach: 1) given the small dimensionality of the manifold, it is no longer necessary to generate a large artificial training set in order to produce a robust classifier; and 2) because of the re-projection onto the manifold, the generated artificial samples may be more likely to happen in practice.

The generation of positive and negative samples from training data is obtained with the following two simple steps: 1) estimate the contour in the original image space from the landmark, $\hat{\mathbf{S}}_{i,j} = \zeta_i^{-1}(\mathbf{m}_{i,j})$; and 2) apply (5) to obtain the grid with the non-rigid deformation $\widetilde{\mathbf{G}}_{\mathbf{C}}$ (Fig.2). In order to generate these positive and negative samples we use the following distribution based on the patch members $\mathbf{m}_{i,j} \in \mathcal{Q}_i$ of each patch $\mathcal{P}_i$:

$$\text{Dist}(\mathcal{P}_i) = U(range(\mathcal{Q}_i)), \tag{6}$$

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

$$\begin{aligned} \mathcal{T}_+(i,j) &= \Big\{\mathbf{m}|\mathbf{m} \sim \text{Dist}(\mathcal{P}_i), |\mathbf{m} - \mathbf{m}_{i,j}| \prec \mathbf{r}_i\Big\} \\ \mathcal{T}_-(i) &= \Big\{\mathbf{m}|\mathbf{m} \sim \text{Dist}(\mathcal{P}_i), |\mathbf{m} - \mathbf{m}_{i,j}| \succ 2 \times \mathbf{r}_i, \\ &\quad \text{for all } j \in \{1, ..., |\mathcal{P}_i|\}\Big\} \end{aligned} \tag{7}$$

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. Fig.4(a) shows how the artificial training samples are generated, where the positive samples are drawn from the blue region, and the negative samples are extracted from the orange region. These samples are then used to train the parameters of our discriminative classifiers, as follows [13]:

$$\begin{aligned} \gamma_{\text{MAP}} = \arg\max_{\gamma} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} & \left[\prod_{\mathbf{m} \in \mathcal{T}_+(i,j)} p(\mathbf{m}|I,\gamma) \right] \\ \times & \left[\prod_{\mathbf{m} \in \mathcal{T}_-(i)} (1 - p(\mathbf{m}|I,\gamma)) \right], \end{aligned} \tag{8}$$

where γ represents the model parameters of $p(\mathbf{m}|I,\gamma)$, which denotes the classifier $p(\mathbf{m}|I,\mathcal{D})$ in (3).

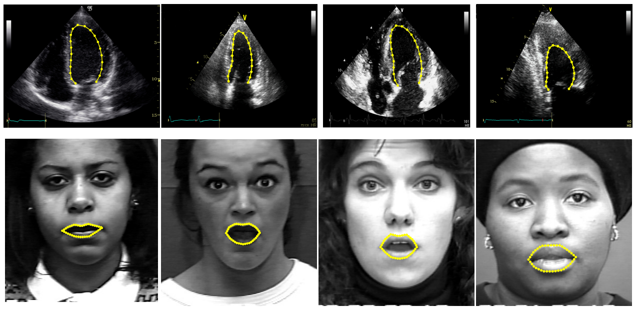


Figure 5. Examples of training samples for the LV (1st row) and lip (2nd row) sequences.

The segmentation procedure takes an input test image I and performs a gradient ascent procedure [14] on the output of $p(\mathbf{m}|I, \gamma_{\text{MAP}})$ that is computed in the sparse low dimensional manifold described in Sec. 3.2. This process will generate the final contour $\mathbf{S}^\star$ (3). Similar gradient-based search methods on manifolds have been recently studied by Helmke et al. [15], 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[16, 17, 18]. Our approach represents an application of such gradient-based search methods in the problem of top-down non-rigid segmentation. The contour $\mathbf{S}^*$ is found by:

$$\mathbf{S}^* = \xi_i(\mathbf{m}^*_{i,j}), \tag{9}$$

where

$$\mathbf{m}^*_{i,j} = \arg \max_{\mathbf{m} \in \mathcal{L}_i, i \in \{1,\dots,|\mathcal{P}|\}} p(\mathbf{m}|I, \gamma_{\text{MAP}}). \tag{10}$$

More specifically, each landmark point $\mathbf{m}_{i,j} \in \mathcal{L}_i$ (for $i \in \{1, ..., |\mathcal{P}|\}$) is used as an initial guess for a gradient ascent (GA) procedure [14] on the output of the classifier $p(\mathbf{m}|I, \gamma_{\text{MAP}})$ over the search parameter space on the manifold $\mathcal{M}$. Fig. 4(b) displays the segmentation process, where the level sets denote the results of the classifier $p(\mathbf{m}|I, \gamma)$ that is used in the GA process. Note that the search is performed on the low dimensional space of $\mathbf{m}$, and each patch has its own local maximum.

5. Running-time Complexity Comparisons

This section details the complexity reduction achieved by the proposed method. The complexity of current non-rigid segmentation methods is based on the number of executions of the rigid and non-rigid classifiers in (2). The rigid classifier runs in the intermediate space represented by the variable $\mathbf{t} \in \mathbb{R}^R$ (Sec. 2), where in the current methodologies [1, 2, 3, 4, 5] $R \in \{4, 5\}$ (accounting for two translation, one rotation and one or two scale parameters). The non-rigid classifier runs in the space of $\mathbf{S} \in \mathbb{R}^{2 \times S}$, and we will assume it has linear complexity $O(S)$. An exhaustive search leads to a complexity of $O(K^R + S)$, with K denoting the number of samples used in each of the dimensions in the intermediate space (typically $K = O(10^3)$). Notice that this naive approach leads to a highly complex method. This complexity has been reduced in several ways, such as with the *branch and bound* framework [19], which allows a reduction to $O(K^{R/2} + S)$ or the *marginal space learning* (MSL) [4] which partitions the search space into subspaces of increasing complexity, achieving a complexity reduction of $O(K + (R - 1) \times \sharp\text{scales} \times K_{\text{fine}} + S)$, where $\sharp$scales accounts for the number of scales and K_{fine} represents a reduced number of promising samples (note that in [1, 3, 4, 20], $\sharp\text{scales} = 3$ and $K_{\text{fine}} = \mathcal{O}(10^1)$). Coarse-to-fine based derivative search has also been proposed in [1, 3, 20], which uses GA approach in the space of R dimensions [1, 20] achieving a complexity of $O(K + \sharp\text{scales} \times K_{\text{fine}} \times R + S)$. The use of sparse manifolds in the rigid detection has been implemented in [3] achieving a complexity of $O((\sum_i |\mathcal{L}_i|) \times \sharp\text{scales} \times M + S)$, with $M < R$ and $\sum_i |\mathcal{L}_i|$ representing the total number of landmarks. Finally, the complexity of the methodology proposed in this paper is $O((\sum_i |\mathcal{L}_i|) \times \sharp\text{scales} \times M)$, which is in general smaller than the ones above because we are able to reduce the dimensionality of the search space from R to $M < R$, and because we run the non-rigid segmentation directly (thus removing the term $O(S)$).

In practice, assuming the figures above, the branch and bound approach [19] has complexity $O(1000^{5/2} + S)$, MSL [4] has complexity $O(1000 + 4 \times 3 \times 10 + S)$, the coarse-to-fine derivative method of [1, 20] has complexity $O(1000 + 3 \times 10 \times 5 + S)$ (where we assume that $R = 5$), and the sparse manifold in rigid detection of [3] has complexity $O(10 \times 3 \times 2 + S)$ (where we assume $M = 2$ and $\sum_i |\mathcal{L}_i| = O(10^1)$). Our approach leads to a complexity of $O(10 \times 3 \times 2)$. Another important point to consider in practice is that the running time complexity of the detectors in the coarse space are smaller than in the fine space, which means that the figures for the coarse classifier complexity should be multiplied by a factor in $(0, 1]$. In the experiments, we provide an estimate for this factor.

It is important to mention 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.

6. Experimental Setup and Results

In the section we show empirical evidence that the use of the proposed sparse low-dimensional manifold leads to less complex classifiers and to segmentation methods that are more efficient than the state of the art, without a negative impact on the segmentation accuracy. Additionally, we show evidence that the gradient ascent described in Sec. 4 is convergent.

Table 1. Complexity of the trained DBN's at all scales for CAR1 [1], CAR2 [20], and our proposal (for the classifier $p(\mathbf{m}|I, \mathcal{D})$ in (3)) in the LV dataset. Note that $\sigma = 4$ denotes the finest scale and $\sigma = 16$ represents the coarsest scale.

Methodologies	Number of Nodes					
	Input Layer	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Hidden Layer 4	Output Layer
CAR$\{1, 2\}$ (rigid,$\sigma = 4$)	196	100	100	200	200	2
CAR$\{1, 2\}$ (rigid,$\sigma = 8$)	49	50	100	-	-	2
CAR$\{1, 2\}$ (rigid,$\sigma = 16$)	16	100	50	-	-	2
CAR$\{1, 2\}$ (non-rigid,$\sigma = 4$)	41	50	50	-	-	1
Proposal ($\sigma = 4$)	196	50	100	-	-	2
Proposal ($\sigma = 8$)	49	100	50	-	-	2
Proposal ($\sigma = 16$)	16	100	50	-	-	2

Table 2. Error measures in the test sequences for COM [2, 5], CAR1 [1], CAR2 [20], NAS [3], and our approach. Each cell shows the mean and standard deviation of each error measure and the best value among the four methods is highlighted.

Test set A					
measures	COM [2, 5]	CAR1 [1]	CAR2 [20]	NAS [3]	Proposal
HDF	20.5(1.2)	**19.2(2.3)**	20.6(2.6)	19.9(1.6)	20.4(2.1)
MAD	11.4(3.2)	9.9(3.3)	**9.4(2.1)**	11.1(3.3)	12.3(3.3)
HMD	.21(.04)	**.17(.05)**	.18(.06)	.23(.06)	.23(.04)
AV	3.9(0.6)	3.3(0.9)	**3.3(0.8)**	4.9(1.5)	4.6(0.9)
Test set B					
HDF	**17.2(1.4)**	19.4(1.5)	19.9(1.9)	18.9(2.2)	18.3(1.5)
MAD	18.2(6.1)	15.7(5.7)	17.7(5.5)	15.5(6.2)	**13.4(5.5)**
HMD	.19(.03)	**.16(.04)**	.17(.02)	.20(.03)	.18(.03)
AV	3.4(0.6)	**2.9(0.5)**	3.1(0.6)	3.8(0.6)	3.6(0.6)

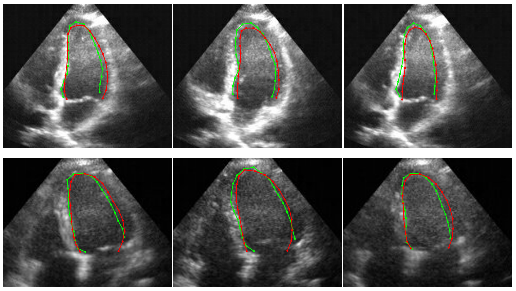


Figure 6. Examples of the LV segmentation results produced by our proposal (red) in comparison with the manual annotations (green).

Two 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 [21] (see Fig. 5), where the goal is to segment the LV endocardial border. This LV database has a training set with 496 manual annotations from 18 sequences, and a test set containing 80 annotations from two sequences. These 20 sequences have been manually annotated by a cardiologist, and 16 of them contain some abnormality while 4 of them represent normal cases. The second database is a sub-set of the Cohn-Kanade (CK+) database [22], 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* emotion, which contains large shape deformation. We use 4 sequences for training (with 103 annotations) and 10 sequences for testing (with 171 annotations). The dimensionality of the object shape is $S = 21$ for the LV and $S = 40$ for the lips.

For the experiments below, we extend the method in [1, 20] (referred to as CAR1 and CAR2, respectively), where a coarse-to-fine approach based on deep belief networks (DBN) [13] is used for the segmentation procedure. The automatically learned sparse manifold (Sec. 4) is different depending on the dataset used. For the LV we obtain an intrinsic dimensionality of $M = 3$, with $|\mathcal{P}| = 13$ patches, 1270 patch member points and 47 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 $|\mathcal{P}| = 1$ with 103 patch members points (corresponding to frame used for training) and four landmark points.

In the training stage of the DBN, we used $|\mathcal{T}_+(i, j)| = 10$ positive and $|\mathcal{T}_-(i)| = 100$ negative samples (for both datasets), which represents the same number of training samples used in [20]. We follow the same learning procedure described in [20], which divides the initial training set into training and validation sets containing 80% and 20% of the original set, respectively. This validation set is used to determine the following parameters: a) number of nodes per hidden layer, and b) number of hidden layers. The number of nodes per hidden layer varies from 50 to 500 in intervals of 50, and the number of hidden layers varies from 1 to 4. The learned configurations of the DBN for our proposed methodology, CAR1 [1], and CAR2 [20] are shown in Table 1. Notice that for the LV segmentation problem, compared to CAR$\{1, 2\}$, our approach selects network models that are of similar complexity for coarser scales ($\sigma \in \{8, 16\}$), but of considerable smaller complexity for the finest scale ($\sigma = 4$). Furthermore, we no longer require the learning of a non-rigid classifier.

A quantitative performance is conducted using the following error measures proposed in the literature for contour comparison: (i) *Hausdorff* (HDF) [23], (ii) *mean absolute distance* (MAD) [5], (iii) *Hammoude* (HMD) [24], and (iv) *average* (AV) [21]. A comparison with the following state-of-the-art approaches for LV segmentation is presented: COM [2, 5], CAR1 [1], CAR2 [20], and NAS [3]. Table 2 shows this quantitative comparison in the test sequences. Notice that, despite the ambitious dimensionality reduction achieved, we obtain competitive results. Fig. 6 illustrates the segmentations obtained with our proposal in both test sequences.

For the lip segmentation, we also provide a quantitative comparison between our approach and the state-of-the-art methods in [1, 3] in Table 3 [3]. Notice that the results presented in this table shows that the proposed approach is comparable or better than the other approaches in terms of segmentation accuracy.

We also present the running time figures of the proposed method and that of CAR1 [1], CAR2 [20], and NAS [3]. For the LV sequences, the methods CAR$\{1, 2\}$ [1, 20] provide the following running time: rigid detector on coarse scale (2.48s) + rigid detector on finer scales (4.25s) + non-rigid detector (0.67s) = 7.4s. NAS [3] achieves the following results: rigid detector on manifold (1.7s) + non-rigid detector (0.67s) = 2.37s. Our proposal reaches the following running time: non-rigid detection on manifold (1.68s) + TPS image warp (0.017s) = 1.7s. This means that the rigid classifier (used on CAR1,2) in the 5-dimensional space runs in approximately 0.0025s for the coarse scale, and for the finer scales it runs in 0.028s. Using the manifold and in our method, the classifiers run in approximately 0.0036s in the coarse scales and 0.0136 in the finer scales, where the time added by the TPS image warp is negligible ($<$0.0001 s). As a result, we note that the complexity figures in Sec. 5 can have a factor of around (1/10) for the coarse classifiers.

For the lip sequences, the methods CAR1 [1] runs on average (mean) in 11.8 seconds for the 10 test lip sequences, the method NAS [3] has mean of 2.62 seconds, while the mean running time of our approach is 2.2 seconds. In all methodologies the running times were obtained from unoptimized Matlab implementations.

Finally, Fig. 8 illustrates that the gradient ascent (GA) of Sec. 4 is convergent. Specifically, we show the evolution of the gradient magnitude (recall that this magnitude is computed from the values produced by the classifier $p(\mathbf{m}|I, \gamma_{\text{MAP}})$ before and after the GA step) as a function of the iteration index of the GA step (a maximum of 5 steps is used) using one of the lip test sequences (results are similar for other sequences). As expected, the magnitude is bigger for the first iterations and reduces for the last iterations.

7. Discussion 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 of non-rigid deformations. 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 is more efficient than other state-of-the-art approaches [1, 20, 3], 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, 20].

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. We also 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 [6].

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

References

[1] G. Carneiro and J. C. Nascimento, “Multiple dynamic models for tracking the left ventricle of the heart from ultrasound data using particle filters and deep learning architectures,” in *CVPR*, 2010, pp. 2815–2822. 1, 2, 5, 6, 7, 8

[2] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta, “Databased-guided segmentation of anatomical structures with complex appearance,” in *CVPR*, 2005. 1, 2, 5, 6, 7

[3] J. C. Nascimento and G. Carneiro, “Top-down segmentation of non-rigid visual objects using derivative-based search on sparse manifolds,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1963–1970. 1, 2, 5, 6, 7, 8

[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. 1, 2, 5

[5] X. S. Zhou, D. Comaniciu, and A. Gupta, “An information fusion framework for robust shape tracking,” vol. 27, no. 1, pp. 115–129, 2005. 1, 2, 5, 6, 7

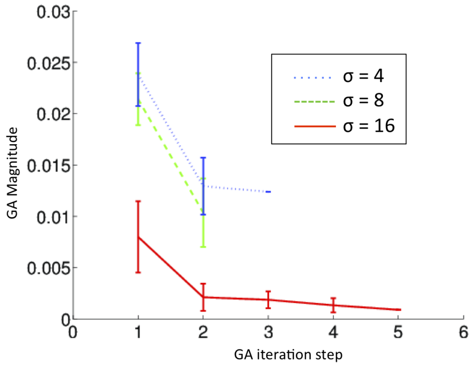


Figure 8. Evolution of the gradient magnitude (using mean and standard deviation) as a function of the iteration index of the GA step in one of the test lip sequences.

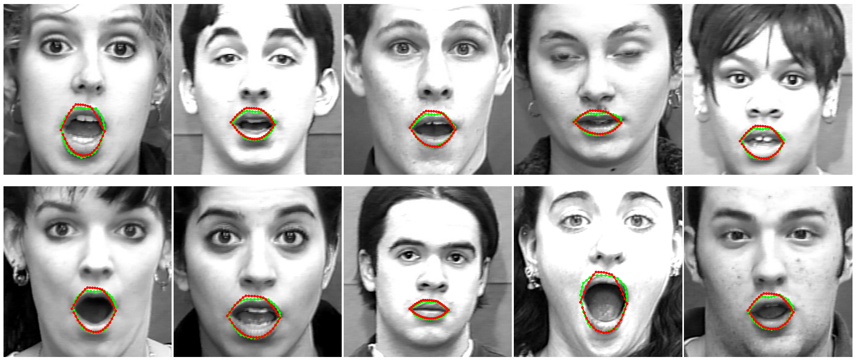


Figure 7. Examples of the lip segmentation results produced by our proposal (red) in comparison with the manual annotations (green).

Table 3. Comparison in the test sequences of the lip segmentation problem. Each column in this table corresponds to one image in Fig. 7, as denoted in the table title.

		Lip Sequences in Fig.7 (top row, left to right)					Lip Sequences Metrics in Fig.7 (bottom row, left to right)				
	Method	♯ 1	♯ 2	♯ 3	♯ 4	♯ 5	♯ 6	♯ 7	♯ 8	♯ 9	♯ 10
HDF	CAR1 [1]	12.76(3.98)	11.27(2.51)	7.29(4.67)	5.80(1.65)	5.35(1.27)	8.76(4.35)	8.95(2.45)	7.20(1.80)	11.54(3.66)	10.33(2.50)
	NAS [3]	8.04(3.76)	5.05(1.23)	**5.07(1.18)**	**5.14(0.67)**	**4.56(0.65)**	**3.49(0.62)**	8.25(1.62)	5.53(1.08)	**6.72(1.90)**	**4.74(0.40)**
	OurAppr.	**6.13(1.19)**	**4.95(0.68)**	5.21(0.55)	6.15(0.21)	5.72(0.42)	5.60(0.31)	**7.57(1.47)**	**4.45(0.38)**	6.83(1.56)	5.66(0.39)
MAD	CAR1 [1]	8.33(1.77)	6.98(2.89)	6.43(3.45)	5.05(1.00)	3.97(0.81)	7.22(2.75)	7.33(2.79)	6.01(1.71)	9.38(3.12)	9.43(1.91)
	NAS [3]	6.26(2.00)	3.44(0.40)	3.77(0.62)	3.39(0.76)	3.38(0.45)	**2.61(1.01)**	**6.07(1.22)**	3.88(0.51)	**5.14(1.20)**	**3.42(0.41)**
	OurAppr.	**5.88(2.85)**	**2.91(1.65)**	**2.41(0.37)**	**3.16(0.39)**	**2.97(1.36)**	2.86(0.81)	6.10(2.69)	**3.69(1.24)**	8.03(4.26)	3.78(0.32)
HMD	CAR1 [1]	0.21(0.03)	0.21(0.05)	0.20(0.08)	0.18(0.07)	0.13(0.02)	0.21(0.06)	0.19(0.09)	0.22(0.02)	0.22(0.03)	0.25(0.03)
	NAS [3]	0.17(0.05)	0.13(0.02)	0.16(0.02)	**0.15(0.02)**	**0.12(0.01)**	**0.12(0.04)**	0.16(0.01)	0.16(0.01)	0.14(0.02)	**0.12(0.03)**
	OurAppr.	**0.13(0.03)**	**0.12(0.03)**	**0.14(0.03)**	0.17(0.04)	0.14(0.02)	0.14(0.04)	**0.12(0.01)**	**0.15(0.03)**	**0.13(0.02)**	0.18(0.05)
AV	CAR1 [1]	4.20(0.81)	3.67(1.48)	3.32(1.87)	**2.49(0.47)**	**2.01(0.39)**	3.74(1.51)	3.65(1.37)	3.00(0.82)	4.69(1.47)	4.86(1.09)
	NAS [3]	3.76(1.36)	2.14(0.53)	2.46(0.29)	2.63(0.47)	2.01(0.50)	**1.91(0.31)**	4.04(0.87)	2.44(0.33)	3.29(0.72)	**2.30(0.14)**
	OurAppr.	**2.88(0.59)**	**2.06(0.64)**	**2.26(0.24)**	2.63(0.12)	2.20(0.37)	2.39(0.24)	**3.09(0.73)**	**2.18(0.30)**	**2.81(0.77)**	3.00(0.38)

[6] L. Yang, B. Georgescifi, 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. 2, 7

[7] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3d medical image segmentation: A review," *Medical image analysis*, vol. 13, no. 4, pp. 543–563, 2009. 2

[8] G. Donato and S. Belongie, "Approximate thin plate splines mappings," in *ECCV*, 2002, pp. 21–31. 3

[9] J. C. Nascimento and J. G. Silva, "Manifold learning for object tracking with multiple motion dynamics," in *ECCV*, 2010. 3

[10] 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. 3

[11] J. G. Silva, J. S. Marques, and J. M. Lemos, "Selecting landmark points for sparse manofold learning," in *NIPS*, 2005. 4

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2001, pp. 511–518. 4

[13] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. 4, 6

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004. 5

[15] 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. 5

[16] 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. 5

[17] 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. 5

[18] 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. 5

[19] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *CVPR*, 2008. 5

[20] 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. 5, 6, 7

[21] 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. 6

[22] 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. 6

[23] 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. 6

[24] A. Hammoude, "Computer-assited endocardial border identification from a sequence of two-dimensional echocardiographic images," Ph.D. dissertation, University Washington, 1988. 6