TOWARDS REDUCTION OF THE TRAINING AND SEARCH RUNNING TIME COMPLEXITIES FOR NON-RIGID OBJECT SEGMENTATION

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

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

ABSTRACT

The problem of non-rigid object segmentation is formulated in a two-stage approach in Machine Learning based methodologies. In the first stage, the automatic initialization problem is solved by the estimation of a rigid shape of the object. In the second stage, the non-rigid segmentation is performed. The rational behind this strategy, is that the rigid detection can be performed at lower dimensional space than the original contour space. In this paper, we explore this idea and propose the use of manifolds to reduce even more the dimensionality of the rigid transformation space (first stage) of current state-of-the-art top-down segmentation methodologies. Also, we propose the use of deep belief networks to allow for a training process capable to produce robust appearance models. Experiments in lips segmentation from frontal face images are conducted to testify the performance of the proposed algorithm.

1. INTRODUCTION AND RELATED WORK

This article presents a method for direct object detection and segmentation using a manifold learning based approach. Given a database of objects represented by a two dimensional contour, in a first step, a reduced order parameterization is determined from the corresponding manifold that incorporates the rigid transformation of the object. The image data term is learned by a deep belief network learned in a canonical space. In the test phase, the execution of the segmentation is accomplished by performing gradient ascent iterative procedure directly on the manifold, i.e. at the low dimensional parameter space. This makes the segmentation less complex and faster; the two main goals targeted in this paper.

The proposed method contrasts with most current top-down segmentation of non-rigid visual objects based on machine learning approaches [4,7,9,10]. Basically, the above methods partition the problem into the following procedures: (i) rigid detection followed by (ii) non-rigid segmentation that is constrained by the result of the rigid-detection. In this two-stage strategy, the introduction of a rigid detection procedure is twofold: it allows to reduce the search running time complexities as well as the training complexities. In practice, this is achieved by constraining the search for the visual object borders within a small window around the output produced by the rigid detection, i.e image patch (see Fig.2 (a) for an illustration).

In this paper, we use a manifold learning strategy, recently proposed in [1] which provides a low intrinsic dimensionality for the rigid detection stage (see Fig. 1 for an illustration of the proposal). In this way, the intrinsic low dimensionality of the manifold can decrease the complexity of the rigid detection stage of the current state of the art methodologies. As an example, let us suppose that we have an input image patch $\mathrm{x} \in \mathbb{R}^p$ (here p stands for the patch dimensionality - number of pixels). Current two-stage methodologies, outputs a multi-dimensional variable $\mathrm{y}(\mathrm{x}) \in \mathbb{R}^{\mathrm{q}}$, where $q = 5$ (accounting for finding the translation, scale and rotation transformations of the sough contour) as in the majority of the proposed approaches (e.g. [3]). Here, we are able to produce $\mathrm{y}(\mathrm{x}) \in \mathbb{R}^{\mathrm{n}}$, with $n < q$, being n the intrinsic dimension of the manifold. In this way we achieve a decrease in the running time complexity of the rigid detection stage.

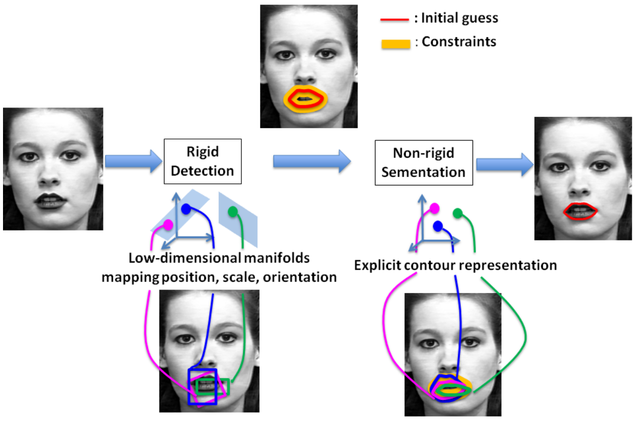
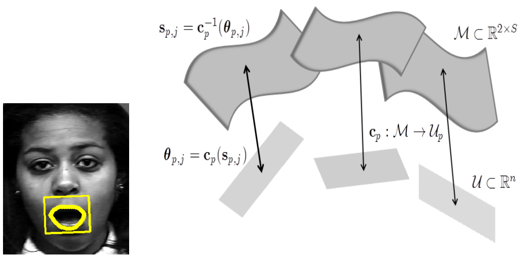

: Initial guess
: Constraints
Rigid Detection
Non-rigid Sementation
Low-dimensional manifolds mapping position, scale, orientation
Explicit contour representation

Fig. 1. Illustration of the two-stage strategy in which is clear the use of the manifold in the rigid detection step.

Another concern targeted by the introduction of the rigid detection, is that of alleviating the need of large annotated training data sets, typically encountered in the majority of current methodologies. The typical solution to circumvent this problem, is to generate a pre-defined number of positive and negative samples by randomly perturbing the rigid parameters (i.e. translation, rotation and scale), following a normal distribution. What happens is that, there is no guarantee that the actual training data distributions indeed follows such theoretical probability density functions, thus unnecessarily increasing the complexity of the training stage. With the introduction of manifold, we expect to limit the positive and negative samples, since now the samples are expected to belong to the manifold. Thus, the methodology proposed also permits to improve the second goal (i.e. training complexity) targeted by the rigid detection.

To summarize, we present the use of manifolds with low intrinsic dimension for the rigid detection (the first stage) that can be applied in pattern recognition based methods [5,7,10]. This is twofold; first, the intrinsic dimension of the the manifold can decrease the search running time complexity in the rigid detection. Second, it is possible to reduce the number for additional positive and negative samples of the classifier, during the training process.

This work was supported by the FCT project [UID/EEA/50009/2013] and Project HEARTTRACK (PTDC/EEA-CRO/103462/2008).


$\mathbf{s}_{p,j} = \mathbf{c}_p^{-1}(\boldsymbol{\theta}_{p,j})$
$\mathcal{M} \subset \mathbb{R}^{2\times S}$
$\boldsymbol{\theta}_{p,j} = \mathbf{c}_p(\mathbf{s}_{p,j})$
$\mathbf{c}_p : \mathcal{M} \to \mathcal{U}_p$
$\mathcal{U} \subset \mathbb{R}^n$

(a) (b)

Fig. 2. (a) In yellow a window enclosing the segmentation contour is illustrated. (b) The manifold and its charts [1]. It is shown three patches (top), tangent hyperplanes (bottom) and one-to-one mappings (arrows).

2. PROBLEM DEFINITION

Let us assume the availability of an image with an object. The goal is to perform the non-rigid segmentation of that object, producing the contour $\mathbf{s} \in \mathbb{R}^{2\times S}$ of 2-D points, that can be achieved by the following decision function

$$\mathbf{s} = \mathbb{E}\left[\mathbf{s}|I,\mathcal{D}\right] = \int_{\mathbf{s}} \mathbf{s}\, p(\mathbf{s}|I,\mathcal{D})d\mathbf{s} \qquad (1)$$

where $\mathcal{D} = \{(I,\mathbf{s})_j\}_{j=1}^{|\mathcal{D}|}$ is the training set, I_j denotes the training image and $\mathbf{s}_j$ denotes the corresponding manual segmentation.

Equation (1) can be expanded in order to account for the two terms of rigid and nonrigid detections, as follows

$$p(\mathbf{s}|I,\mathcal{D}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|I,\mathcal{D})\, p(\mathbf{s}|\boldsymbol{\theta},I,\mathcal{D})d\boldsymbol{\theta} \qquad (2)$$

where $\boldsymbol{\theta}$ is the variable that encodes the linear transformation of the window coordinates enclosing the segmentation contours (see Fig.2 (a)). The linear transformation can be obtained as $\mathbf{A}_{\theta} = f(\boldsymbol{\theta})$, with $\mathbf{A}_{\theta} \in \mathbb{R}^{3\times 3}$ comprising the translation, rotation and scale. The second term in (2), represents the non-rigid (regression) part of the segmentation, that finds the segmentation $\mathbf{s}$ in the image I, given the value of $\boldsymbol{\theta}$. Thus, $\boldsymbol{\theta}$ is viewed an initial guess of $\mathbf{s}$ constraining the search space of $\mathbf{s}$ to be around the mean segmentation contour transformed by $\boldsymbol{\theta}$.

3. LEARNING THE MANIFOLD

The framework presented herein uses the manifold learning algorithm recently proposed in [1]. Basically, from a training samples $\mathbf{s}_j, j = 1,..,|\mathcal{D}|$, this framework finds a manifold $\mathcal{M}$ contained in $\mathbb{R}^{2S}$, associated with a set of one-to-one mappings $\mathbf{c}_p : \mathcal{P}_p \to \mathcal{U}_p$ (i.e. the *charts*) and invertible functions $\mathbf{s}_{p,j} = \mathbf{c}_p^{-1}(\boldsymbol{\theta}_{p,j})$, (or *parameterizations*), where $\mathcal{P}_p \subset \mathcal{M}$ and $\mathcal{U}_p \subset \mathbb{R}^n$. The manifold $\mathcal{M}$ is covered by the union of the overlapped $\mathcal{P}_p$, with $p = 1,..,P$. The $\mathcal{P}_p$ are called *patches* and $\mathcal{U}$ are the *parametric domains* of $\mathcal{M}$. Locally, $\mathcal{M}$ it is at least homeomorphic to $\mathbb{R}^n$ having an *intrinsic dimension* of n. See Fig. 2 for an illustration.

4. TRAINING THE MANIFOLD THROUGH DEEP-STRUCTURED INFERENCE

In general, the training set does not contain enough contour samples to provide reliable information for a robust training process. The usual strategy to circumvent this, is to artificially generate positive and negative training samples perturbing the deformation parameters of the training data. The generation of the positive and negative samples from training data can be obtained with the following two-step strategy: (i) first, we estimate the contour in the original image space $\widehat{\mathbf{s}}_{p,j} = \mathbf{c}_p^{-1}(\boldsymbol{\theta}_{p,j})$ and (ii) find the transformation $\mathbf{A}_{\theta} \in \mathbb{R}^{3\times 3}$ of the image window that contains the segmentation contour $\widehat{\mathbf{s}}_{p,j}$ produced in the previous step (i). Recall that, the rigid classifier (the first term in (2)) is modeled by the parameter vector ϕ_{MAP}, learned with a *maximum a posteriori* criterion, which is estimated with a set of training samples taken from the patch member points $\boldsymbol{\theta}_{p,j} = \mathbf{c}_p(\mathbf{s}_{p,j})$, for $p = 1,..,P$. Thus, we take $\boldsymbol{\theta}_{p,j}$ in $\mathcal{M}$ to build the positive and negative sets, as follows

$$Dist(\mathcal{P}_p) = \mathcal{U}(R(\boldsymbol{\theta}_{p,j})) \qquad (3)$$

where $\mathcal{U}$ denote an uniform distribution over the range R of the patch-member points in $\mathcal{P}$. More specifically, for the p-th patch we define,

$$\begin{aligned} \mathcal{P}os_{(p,j)} &= \Big\{\boldsymbol{\theta}\Big|\boldsymbol{\theta} \sim \text{Dist}(\mathcal{P}_p), |\boldsymbol{\theta} - \boldsymbol{\theta}_{p,j}| \prec \mathbf{r}_p\Big\} \\ \mathcal{N}eg_{(p,j)} &= \Big\{\boldsymbol{\theta}|\boldsymbol{\theta} \sim \text{Dist}(\mathcal{P}_p), |\boldsymbol{\theta} - \boldsymbol{\theta}_{p,j}| \succ 2 \times \mathbf{r}_p, \\ &\quad \text{for all } j \in \{1,...,J\}\Big\} \end{aligned} \qquad (4)$$

where $\mathbf{r}_p$ is the margin between positive and negative samples and where $|.|$ returns the absolute value of the difference. The samples drawn in (4) are used to learn the rigid classifier by maximizing the following cost function [8]

$$\begin{aligned} \phi_{\text{MAP}} = \arg\max_{\phi} \prod_{p=1}^{P}\prod_{j=1}^{J_p} & \left(\prod_{\boldsymbol{\theta}\in\mathcal{P}os_{(p,j)}} p(\boldsymbol{\theta}|I,\phi) \right) \\ \times & \left(\prod_{\boldsymbol{\theta}\in\mathcal{N}eg_{(p,j)}} (1 - p(\boldsymbol{\theta}|I,\phi)) \right). \end{aligned} \qquad (5)$$

where J_p is the number of patch members in the p-th patch. For training the non-rigid classifier (see second term in (2)) we follow our previous work [2]

$$\psi_{\text{MAP}} = \arg\max_{\psi} \prod_{p=1}^{P}\prod_{j=1}^{J_p}\prod_{l=1}^{L} p(\mathbf{s}_{p,j}(l)|\boldsymbol{\theta}_{p,j},I,\psi) \qquad (6)$$

where ψ represents the deep neural net (DNN) weights and $\mathbf{s}_{p,j}(l) \in [0,..C]$, is the l-th orthogonal line of the contour represented by $\mathbf{s}_{p,j}$ with C-length. More specifically, $p(\mathbf{s}_{p,j}(l)|\boldsymbol{\theta}_{p,j},I)$, is a regressor that receives as the input a profile of the image gray levels taken at the orthogonal line from each contour point $\mathbf{s}_{p,j}(l)$, and outputs an image location at that l-th orthogonal line. Notice that, the training strategy follows the same strategy as in [2] with the following key difference: the inference procedure to generate the segmentation contour in the image I, takes each patch-member $\boldsymbol{\theta}_{p,j}$ from each learned patch $\mathcal{P}_p$ (with $p = 1,..,P$) as an initial guess for the gradiente procedure on the output of the rigid classifier $p(\boldsymbol{\theta}|I,\phi_{\text{MAP}})$ in the manifold $\mathcal{M}$. Whereas in [2], the initial guess of the gradient is taken at $\boldsymbol{\theta} \in \mathbb{R}^5$ that represents the parameters of an affine transformation that aligns the contour in a canonical coordinate system. This approach herein proposed has the advantage of providing $\boldsymbol{\theta}_{p,j} \in \mathbb{R}^n$, with the lower intrinsic n-dimensionality of the manifold[1].

[1] In the experiments shown in Section 5 we obtained an intrinsic dimension of $n = 2$.

For the gradient ascent, a number of iterations is used[2]. Once the gradient ascent is reached for each patch-member $\boldsymbol{\theta}_{p,j}$, the estimate $\widehat{\mathbf{s}}$ is obtained by the following Monte-Carlo approximation

$$\widehat{\mathbf{s}} \propto \sum_{p=1}^{P} \sum_{j=1}^{J_p} \mathbf{s}\, p(\widetilde{\boldsymbol{\theta}}_{p,j}|I, \phi_{\mathrm{MAP}})\, p(\mathbf{s}|\widetilde{\boldsymbol{\theta}}_{p,j}, I, \psi_{\mathrm{MAP}}) \qquad (7)$$

where $\widetilde{\boldsymbol{\theta}}_{p,j}$ is the last estimate in the gradient process. In Section 5 it will be shown the impact of (7) by progressively incorporating the results of the patches.

5. EXPERIMENTAL EVALUATION

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

5.1. Experimental setup

We present results concerning the lip segmentation problem. We use the Cohn-Kanade (CK+) database [6] of emotion sequences taken from frontal view, where the manual ground truth is available. Eight different emotions are available, among which we selected the "happy" and "surprise" since they exhibit higher variation from onset (neutral frame) to peak expression (last frame).

We used a total of 34 sequences which is split in two disjoint sets: training and testing sets. The training set contains 10 sequences consisting of five sequences of "happy" expression and five sequences of "surprise" expression. In both, the lips boundary undergo three distinct phases (i.e. closed, semi-open and open). The test set consists 24 sequences containing 12 "happy" sequences and 12 "surprise" sequences.

In the experiments conducted, the dimensionality to represent the lips boundary is $S = 40$ key points. The manifold learning produces $P = 4$ patches with 395 patch member-points; and an intrinsic dimension of $n = 2$ (see for [1] details). Recall that the current methodologies dimensionality of the rigid search space is 5.

Finally, in order to estimate the robustness of our approach to small training sets, we gradually enlarge the size of the set of positive samples as follows $|\mathcal{P}os(p,j)| \in \{1, 5, 10, 15, 20\}$, and the size of negative samples as $|\mathcal{N}eg(p,j)| \in \{10, 50, 100, 150, 200\}$. We added more additional negative samples due to the larger area occupied by the negative region. Our goal with this experiment is to study how accurate is the methodology against the variation in the small size of the training data.

For comparison purposes, we provide results with "CAR" [4] for the lip segmentation problem. In [4], the coarse-to-fine rigid detector $p(\boldsymbol{\theta}|I, \phi_{\mathrm{MAP}})$ and non-rigid classifier $p(\mathbf{s}|\boldsymbol{\theta}, I, \psi_{\mathrm{MAP}})$ are based on deep belief networks (DBN) [8]. Recall that, in [4] used $|\mathcal{P}os(p,j)| = 10$, and $|\mathcal{N}eg(p,j)| = 100$ per each image in the training set (i.e., 10 additional positive samples and 100 negative samples per training image). The extension herein proposed, consists of training and running the rigid classifier in the space defined by the sparse manifold described in Section 3.

5.2. Accuracy measurements

The segmentation performance is assessed using (i) the metrics commonly used in the literature for quantitative comparison between the generated and the ground truth segmentations; (ii) running time needed to perform the segmentation and (iii) performance of the classifier using different number of positive and negatives samples.

For evaluating the quantitative performance, we use the two following error measurements: the *Jaccard* distance and the *average* error metrics. The performance of our approach is assessed by a quantitative comparison over the test sets with CAR [4] as well as with the manual ground-truth. For both segmentation problems, we also compare the running times between our approach and CAR [4].

5.3. Experimental results

In this section we illustrate both qualitative and quantitative performances following the guideline mentioned in Section 5.2 for the problem of the lip segmentation. Fig. 3 shows the error metrics used (Jaccard and average) obtained using 12 sequences of "happy" expression. Only for the illustration purposes, it is shown the performance accuracy with the patch integration, showing the benefits of incorporating the results of the patches (see (7)), meaning that each patch has a particular response and should be integrated in the final segmentation. This figure also shows that, as we increase the number of positive and negative samples, the results are improved, being the best accuracies obtained with the sets $\{\{10, 100\}, \{\{15, 150\}, \{20, 200\}\}$.

Fig. 4 shows the results using 12 sequences of the "surprise" expression. Again, the results are shown in terms of the error metrics as in the previous example. An additional experiment is performed in this sequence that consists to provide comparison results with [4]. As above, we see the improvement in the accuracy performance agglomerating the results of the patches and stressing that the proposed methodology consistently improves for all of the positive-negative configurations (see Fig. 4 right).

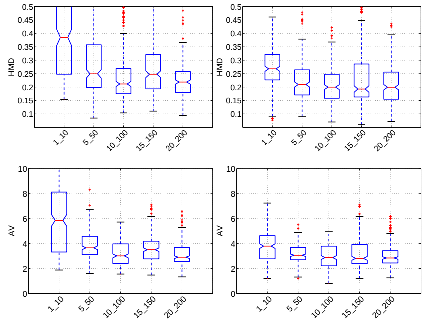


Fig. 3. Jaccard (top row) and average (bottom row) error metrics for the happy sequences. The accuracy is shown in terms of $|\mathcal{P}os(p,j)|$ and $|\mathcal{N}eg(p,j)|$ that are in the range $\{\{1, 10\}, \{5, 50\}, \{10, 100\}, \{15, 150\}, \{20, 200\}\}$. For the illustration purposes, on the left it is shown the results with one patch. On the right it is shown the integration of the 4 patches estimated in the manifold.

It is interesting to note the segmentation results of each patch in the manifold and the corresponding confidences given by the deep belief network (DBN). In each of the eight examples shown in Fig. 5, from two happy sequences, one may see the quality of the patch segmentation along with the confidence degree. In each

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

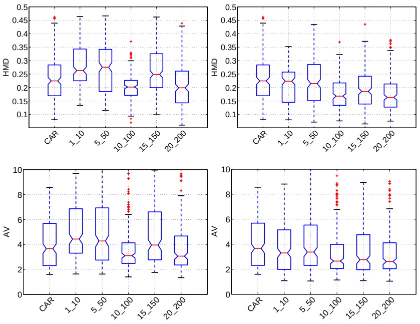


Fig. 4. Comparison with state-of-the-art (CAR) for the expression surprise. Jaccard (top row) and average (bottom row) error metrics are displayed. The accuracy is shown in terms of $|\mathcal{P}os(p,j)|$ and $|\mathcal{N}eg(p)|$ that are in the range $\{\{1,10\},\{5,50\},\{10,100\},\{15,150\},\{20,200\}\}$. For the illustration purposes, on the left it is shown the results with one patch. On the right it is shown the integration of the 4 patches estimated in the manifold.

image of this figure and for viewing convenience, we plot the estimated contour of each patch (red) and the corresponding manual ground-truth (green). Recall that the confidences of the patches are not normalized (this should be done when performing the final segmentation). In both sequences, we see that the better segmentation looks (see the similarity of the green and red contours) the higher the DBN confidence is.

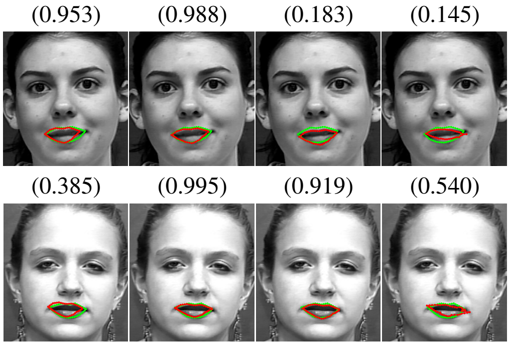


Fig. 5. Individual patch segmentations in a surprise sequence along with the confidence provided by the Deep Belief Net at he top of each image.

We also compare the running time figures of the proposed methodology with [4] (the most-left box-plot shown in Fig. 4) in each image. For the "happy" sequences the proposed approach takes 2.40 seconds for the rigid detection plus 0.2 seconds (average time per frame) for the non-rigid segmentation (total of 2.63 sec.). For the "surprise" sequences, the mean time spent is similar, 2.43 seconds plus 0.2 seconds for the rigid a non-rigid segmentations, respectively. The running time for the approach in [4] is 11.8 seconds. Recall that, these running time figures were obtained with unoptimized Matlab implementations.

Finally, Fig. 6 illustrates several images comparing the machine generated contours (magenta) to human ground-truth contours (cyan).

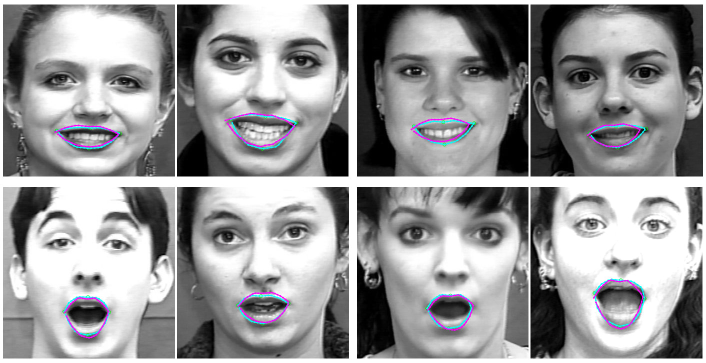


Fig. 6. Test lip sequences displaying the "happy" (top) and "surprise" (bottom) expressions. The ground truth (in cyan) is superimposed with the segmentation results (in magenta).

6. CONCLUSIONS

In this paper we presented a new method for non-rigid object segmentation. The methodology proposed deals with both deep learning inference and manifold learning. The focus of contribution is the dimensionality reduction of the segmentation contour parametrization for the rigid components. A manifold learning based approach has been proposed and allows to reduce the dimension of the rigid space. Also, it allows for a faster running time in both training and segmentation stages. This is because, the training and parameters search are both reformulated directly in terms of the manifold parametrization. Further work will include other directions, for instance, to incorporate a tracking mechanism directly in the manifold, where the object dynamics is learned directly at a low dimensional space. Concerning the manifold learning strategy, parametric based approaches should be explored in a nearly future.

7. REFERENCES

[1] J. C. Nascimento, J. G. Silva, J. M. Lemos and J. S. Marques, "Manifold learning for object tracking with multiple nonlinear models", *IEEE Trans. Image Processing*, vol. 23, no. 4, pp. 1593-1605, 2014.

[2] 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.

[3] G. Carneiro and 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", *IEEE Trans. on Image Processing*, vol. 21, no. 3, pp. 968-982, March 2012."

[4] 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", *CVPR*, pp. 2815-2822, 2010.

[5] S. Zhou, "Shape regression machine and efficient segmentation of left ventricle endocardium from 2D B-mode echocardiogram", *Medical Image Analysis*, vol. 14, pp. 563-581, 2010.

[6] Lucey, P. and Cohn, J.F. and Kanade, T. and Saragih, J. and Ambadar, Z. and Matthews, I., "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression", *IEEE Comp. Vision and Pattern Recognition Workshops*, pp. 94-101, 2010.

[7] 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.

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

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

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