This full text paper was peer-reviewed at the direction of IEEE Instrumentation and Measurement Society prior to the acceptance and publication.

Incremental Multi-Model Dictionary Learning for Face Tracking

1st Aasim Khurshid
Institute of Informatics, UFRGS
Porto Alegre, Brazil
akhurshid@inf.ufrgs.br

2nd Jacob Scharcanski
Institute of Informatics, UFRGS
Porto Alegre, Brazil
jacobs@inf.ufrgs.br

***Abstract*—In this work, a new method based on a multi-model dictionary is proposed for face tracking. A reconstruction and a classification dictionary are combined, and each dictionary is learned from positive and negative examples. This scheme tends to enhance the discrimination between a tracked target face and the background. Also, an efficient scheme that collects data during face tracking is proposed to update the dictionaries in an incremental learning scheme, allowing to track faces even when the face appearance changes (e.g. under different face expressions). The preliminary experimental results suggest that the proposed method tends to perform better than comparative methods, which are representative of the state-of-the-art.**

***Index Terms*—Face Detection, Face Tracking, Dictionary Learning, Incremental Learning, Motion Modeling.**

I. Introduction

Different aspects must be handled by visual tracking processes [1], such as appearance change, object deformation, temporary occlusion and/or different illumination conditions. For example, tracked non-rigid objects (e.g. faces) are likely to undergo appearance or shape changes. Similarly, noise and different lighting conditions during the day may affect the local illumination in various ways [2]. Therefore, numerous algorithms have been proposed in the literature for tracking objects in video sequences [3]. However, these methods often fail in long video sequences because of background or appearance changes. Also, numerous geometrical descriptors have been proposed to represent spatial information in object recognition (e.g., Constrained Local Models (CLM) [4]). These geometrical methods try to track visual objects by using an optimization process, which slows down the overall computation. Consequently, Zheng et al. [5] proposed an approximation for CLM, which unfortunately tends to fail in longer video sequences.

Recently, the linear decomposition of data using a few atoms of a learned dictionary, instead of using a pre-defined set of bases, have been investigated in different areas of machine learning and image processing [6], including object recognition [7] and texture analysis [8]. Although sparse representation approaches like K-SVD [9] have a promising performance in various applications, it is difficult to employ K-SVD for large datasets due to the intense memory usage arising when the K-SVD is computed for large datasets. For this reason, attention has changed towards incremental update of dictionary atoms, since online dictionary learning is promising for large datasets [6]. Dictionary learning has been explored in object tracking, usually with static dictionaries that are not updated during object tracking [10]. Most of these proposed methods use dictionaries for the target object representation [11], or for the target-background discrimination [12].

In this work, a new approach called Multi-Model Dictionary Learning (MMDL) is proposed for face tracking that builds two dictionaries in parallel. The dictionaries are based on the k-SVD and the two dictionaries - a classification dictionary and a reconstruction dictionary - are combined into a single multi-model. The proposed MMDL scheme can reconstruct the face, in addition to discriminating the face from the background. The proposed method learns the face appearance using dictionary atoms constructed from patches, that are taken from positive and negative samples of the training data. Moreover, a smart approach is proposed to update the dictionaries incrementally and efficiently, making the application of our method to realistic tracking scenarios feasible. Furthermore, the proposed method collects training samples to update the two dictionaries during face tracking using a proposed scheme (see details in Section II-E). The quality of the samples (i.e., reconstruction error) is assessed before utilizing them to update the dictionaries, which is an aspect that other methods that implement incremental learning seem to miss [3]. Both the dictionaries are initialized using the Singular Value Decomposition (SVD), which is more efficient than initializing the process by combining random training samples as proposed elsewhere [13]. As both dictionaries are learned incrementally, the number of atoms can increase until a limit is reached. Additionally, the weights of the atoms are updated in an adaptive manner.

The remaining of the paper is organized as follow. The proposed face tracking approach is described in Section II, which details the proposed incremental learning scheme and the multi-model dictionary. Section III discusses our preliminary experimental results. Finally, Section IV concludes the paper and also presents the future prospects of this work.

II. Methodology

The proposed approach is illustrated in the block diagram shown in Fig. 1, which is explained below :

The research is funded by CAPES, Brazil.

978-1-5386-2222-3/18/$31.00 ©2018 IEEE

- Block 1: Initializes the face tracking process for the first frame, the initial target face, the affine parameters ($\chi(t)$), while the face landmarks are provided by a face landmark localization method [4]. The tracked target face is assumed to be contained in a window of fixed size (u $\times$ u). This window and its contents are warped according to the affine parameters to obtain the candidate target face samples. These samples are used in a template matching process to detect the tracked target face in the frame at time $t + \delta t$. The initial target face serves as the mean face ($\mu(t)$) until two dictionaries are created. The two dictionaries are build and updated after a number (τ) of new tracked target face samples have been gathered during face tracking;
- Block 2: In the subsequent frames, a finite number (η) of affine parameters are drawn around the affine parameters of the initial/tracked target face using a Gaussian distribution (see Eq. 12);
- Block 3: To locate the tracked target face in the frame at time t, the candidate target face samples (u $\times$ u) are warped according to the computed affine parameters to be compared with the tracked target face (see the example in Fig. 1 at the left of Block 3: in red, the tracked target face from previous frame; in green, candidate target face samples). See details in Section II-G;
- Block 4: Next, a test is performed to check if the dictionaries already exist, or if enough tracked target face samples (τ) have been collected to create the two dictionaries;
- Block 5: When τ tracked target face samples are available, the samples are decomposed into fixed size patches (v $\times$ v, and v $\leq$ u), and the dictionaries are created using these patches;
- Block 6: Afterwards, the probability of each candidate target face sample being the tracked target face is computed using the learned dictionaries (see Eq. 18);
- Block 7: On the contrary, if the condition in Block 4 is not satisfied, the probability of the candidate target faces being the tracked target face is calculated using the distance from the mean face $\mu(t)$ (see Eq. 14);
- Blocks 8-9: Next, the candidate target face with the highest probability is selected to be the current tracked target face, and is used as training data to update the two dictionaries depending on its quality (i.e., reconstruction error);
- Blocks 10-11: The two dictionaries are created, or updated depending on the number of tracked target face samples accumulated (see details in Section II-D and II-E). The training samples are gathered during the face tracking process based on their reconstruction errors (see Section II-C);
- Block 12: Finally, if there are more frames to process the affine parameters of the current tracked target face are used in the next frames, and the process re-starts from Block 2.

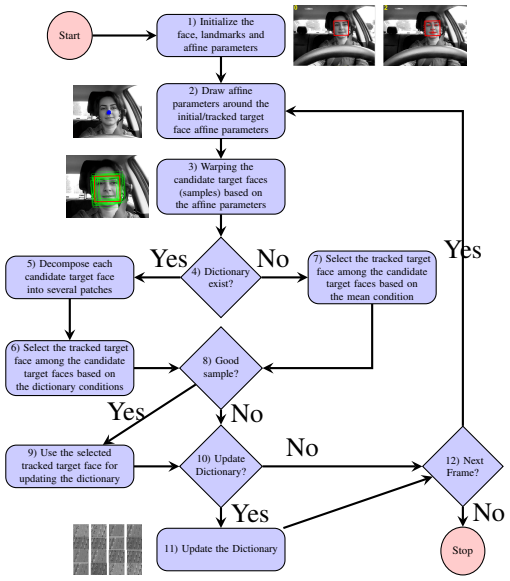


Fig. 1. Block Diagram of the Proposed Face Tracking Method.

The dictionary learning is detailed in Sections II-A, II-B. The face appearance is represented based on the multi-model dictionary learning as explained in Sections II-C,II-D,II-E. The current appearance of the mean target face is updated as explained in Section II-F. Finally, Section II-G details how to locate the tracked target face in consecutive frames using a comprehensive appearance model based on multi-model dictionary learning.

A. Dictionary Learning

Given a set of data items $X = x_1, x_2, ..., x_n \in \mathbb{R}^{e \times n}$, and its sparse code α on a dictionary $D \in \mathbb{R}^{e \times k}$ with k atoms/columns, there are two stages in the dictionary learning algorithm. The sparse coding stage, that is given for a fixed dictionary D using L_0 regularization as follows:

$$\arg\min_{\alpha} ||X - D\alpha||_2^2 + \lambda||\alpha||_0, \tag{1}$$

where λ is an adjustment parameter controlling the sparsity. Afterwards, the dictionary and its atoms are updated to represent the data sparsely. The dictionary atoms are obtained as follows:

$$\arg\min_{D} ||X - D\alpha||_2^2. \tag{2}$$

B. Incremental Dictionary Update

The discussion below applies to both the dictionaries. Consider a dataset $X_n = x_1, x_2, ..., x_n$ that can be represented using a dictionary D with a sparse code α_n. When new data $X_m = x_{n+1}, x_{n+2}, ..., x_{n+m}$ is added to X_n, the dictionary D is updated generating $D^{'}$. As D already describes X_n sparsely in terms of α_n, it is useful to test if the dictionary D represents the new data sparsely as well, and in this case the dictionary update is not necessary. Otherwise, the dictionary D must be updated and the coefficients α_{m+n} need to be calculated. Since

the dictionary is applied in an object tracking application, it should be updated to adapt to the current appearance of the tracked target (e.g face). Furthermore, the dictionary is obtained using the K-SVD [9], which tries to represent the data as sparsely as possible. For this reason in order to update the dictionary, the first step is to apply the incremental SVD [3] to embed new data. While updating the dictionary using the SVD, old atoms are down-weighted with a forgetting factor so the dictionary have an updated appearance of the tracked target. To make the dictionary adaptive to new changes in the appearance of the tracked object (e.g. face), the objective function in Eq. 2 is modified as follows [14]:

$$\arg\min_{\alpha} ||X_{n+m} - D^{'}\alpha_{n+m}||_2^2 + \lambda||\alpha_{n+m}||_0, \quad (3)$$

where the parameter λ is used for regularization. Eq. 3 is minimized until the maximum required sparsity is achieved.

C. Multi-Model Dictionary Learning

The proposed Multi-Model Dictionary Learning (MMDL) scheme, combines a reconstruction dictionary (D_p) that represent strictly the appearance of the target, and a classification dictionary (D_c) that discriminates the tracked target from the background. The dictionary D_c is primarily helpful in target tracking in changing background conditions. The dictionary update uses positive and negative samples obtained during target tracking to learn the dictionary incrementally, making it adaptable to contextual changes. The positive samples are the tracked target faces, and also scaled, rotated versions of these faces from previous frames. The negative samples correspond to the background and are taken from the previous frame with an overlap ratio ov (ov=0.05 in our experiments) with the tracked target face. Each positive/negative sample is decomposed into $v \times v$ sized patches, which are combined to make a patch matrix used to learn the dictionaries.

D. Reconstruction Dictionary

Firstly, the reconstruction dictionary (D_p) is learned using patches of only positive samples as described in Section II-A. The reconstruction error is given by:

$$\varepsilon_r = ||\mathbf{I}_c - D_p\alpha_j||_2^2, \quad (4)$$

where, $\mathbf{I}_c$ is the patch matrix of the candidate target face samples, and α_j are the D_p sparse coefficients.

E. Classification Dictionary

Secondly, the classification dictionary (D_c) is built by Unsupervised Information-Theoretic Dictionary Learning (UITDL) proposed by Flores et al [15]. However, the proposed MMDL method is based on K-SVD dictionary, while Non-Negative Matrix Factorization (NMF) is used in the case of UITDL. Given a dictionary $(D_c^{(0)})$ obtained by K-SVD (see Section II-A), and the initial sparse representation α of the patch matrix Y (with positive and negative samples), and the maximum number of atoms to be selected, MMDL tries to learn a dictionary D_c by maximizing the following criteria:

$$f(.) = \sigma_1 MI(D_c; D_c^{(0)} - D_c) + \sigma_2 MI(Y; D_c), \quad (5)$$

where, the parameters $\sigma_1 \in [0\ 1]$ and $\sigma_1 \in [0\ 1]$, $\sigma_1 + \sigma_1 = 1$, and are used to balance the dictionary representation and compactness; $MI(A;B)$ represents the mutual information between two matrices A and B as below:

$$MI(A;B) = \sum_{A_{i,j}}\sum_{B_{i,j}} p(A_{i,j}, B_{i,j}) \times log\frac{(p(A_{i,j}, B_{i,j})}{p(A_{i,j}), p(B_{i,j})}. \quad (6)$$

To find an atom d_i that maximizes Eq. 5 is equivalent to optimize [15]:

$$\arg\max_{d_i \in D_c^{(0)} - D_c} \{\frac{[\Sigma]_{(i,i)} - \sigma_{D_i}^T \Sigma_D^{-1} \sigma_{D_i}}{[\Sigma_{\bar{D}}]_{(i,i)} - \sigma_{\bar{D}_i}^T \Sigma_{\bar{D}}^{-1} \sigma_{\bar{D}_i}}\}, \quad (7)$$

where, $\bar{D_c} = D_c^0 - (D \cup d_i)$, Σ_D is the covariance matrix of $D_c = [\sigma_{D_1 c}, \sigma_{D_2 c}, ..., \sigma_{D_n c}]$ and $\Sigma_{\bar{D}c}$ denotes the covariance matrix of $\bar{D_c}$. For classification, each candidate target face is represented in terms of the compact and representative dictionary D_c. The classification error is essentially a regression loss given by:

$$\varepsilon_c = ||H_i - W\alpha_i||^2, \quad (8)$$

where $H_i \in [0\ 1]$ is the label indicator, and $W \in R_{b \times k}$ is the linear classification parameters learned with a labeled dictionary computed using:

$$W = (\alpha \times \alpha^T)^{-1} \times \alpha \times H', \quad (9)$$

where H is the label vector of the training data represented in D_c by the sparse matrix α.

These two dictionaries D_p and D_c (reconstruction and classification) are combined in a single model, as shown in Eq. 18, to create a multi-model dictionary which tend to improve the tracking robustness. Both the dictionaries are updated separately using the technique mentioned in Section II-B, after τ new tracked target face samples have been gathered. The collection of the samples is based on the proposed MMDL Face Tracker with Update test (MMDL-FTU), which collects only the samples with a reconstruction error smaller than ε (ε=0.05 in our experiments).

F. Mean Update

In face tracking, the mean face plays an important role, and must adapt to the current appearance of the tracked target face. Therefore, more weight is given to more recent observations, by employing a forgetting factor, and the mean face $\mu(t)$ at time t is updated as follows:

$$\mu(t) = \frac{f \cdot n \cdot \mu_n + m \cdot \mu_m}{m + f \cdot n}, \quad (10)$$

where μ is the updated mean, μ_n represents the mean of the older data (X_n), μ_m is the mean of the newly added observations (X_m) and $t = m + n$, whereas f is the forgetting factor. An important advantage of the forgetting factor is that the mean face can still change in response to the new observations, irrespective of the total number of observations.

G. MMDL-based Face Tracker

Visual tracking can be designed using a Markov model with hidden state variables. In the proposed approach, the affine parameters at time t of the tracked target face are represented by variable $\chi(t)$. Furthermore, the affine parameters of the tracked target face are used to estimate the face landmarks and calculate the tracking error (see Eq. 19). For a set of tracked target face samples at time t, $\mathcal{I}(t)$=$\{\mathbf{I}(1),\mathbf{I}(2)..., \mathbf{I}(T)\ \}$, the face tracker estimates the hidden state variable $\chi(t)$ using:

$$p(\chi(t)|\mathcal{I}(t)) \propto p(\mathbf{I}(t)|\chi(t))\times \int p(\chi(t)|\chi(t-1))p(\chi(t-1)|\mathcal{I}(t-1))d\chi(t-1). \quad (11)$$

The candidate target faces that may contain the tracked target face are sampled following the motion model between two states $p(\chi(t) \mid \chi(t-1))$, assuming a Gaussian distribution around the tracked target face location in the previous frame. At time t, the state of the target face in a video sequence is described by the affine parameters $\chi(t)$=($x(t)$, $y(t)$, $s(t)$, $\theta(t)$, $\beta(t)$, $\phi(t)$), where, $(x(t), y(t))$ is the translation of the target face w.r.t the origin of the image, $(s(t))$ is the scale of the target face w.r.t the image size, $\theta(t)$ is the rotation angle , $\beta(t)$ is the aspect ratio and the skew direction w.r.t the horizontal axis $\phi(t)$, respectively. The dynamics of each parameter in $\chi(t)$ is modeled independently by a Gaussian distribution centered at $\chi(t-1)$, and going from $\chi(t-1)$ to $\chi(t)$ is given by:

$$p(\chi(t)|\chi(t-1)) = \mathcal{N}(\chi(t); \chi(t-1), \psi(t)), \quad (12)$$

where $\psi(t)$ is a diagonal matrix with each element representing the variance of its corresponding affine parameters element, and $\mathcal{N}$ represents a Gaussian distribution. These affine parameters are used to warp the candidate target faces that may contain a face in the current frame.

The probability $p(\mathbf{I}(t) \mid \chi(t))$ of each warped candidate target face being the tracked target face is estimated using the probabilistic interpretation of MMDL (see Eq. 18). The probability of the candidate target face being the tracked target face is based on the joint probability of reconstruction and classification errors. For the reconstruction dictionary D_p, given a candidate target face $\mathbf{I}(t)$ predicted by $\chi(t)$, it is assumed that $\mathbf{I}(t)$ is represented in terms of the dictionary (D_p). The probability of a candidate target face being well represented by D_p is inversely proportional to the distances d_t and d_w of the candidate target face to the reference (mean face $\mu(t)$) represented by D_p. The term d_t is the distance of the candidate target face to D_p and d_w is the distance of the candidate target face to the reference face $(\mu(t))$ represented by D_p. The probability p_{dt} of the candidate target face being the tracked target face based on the current dictionary D_p is:

$$p_d(\mathbf{I}(t)|\chi(t)) = \mathcal{N}(\mathbf{I}(t); \mu(t), D_p^{\dagger} + \epsilon I) = exp(-d_t), \quad (13)$$

which is the negative exponential value of d_t, where, $d_t = ||(\mathbf{I}(t)-\mu(t))-D_p^{\dagger}(\mathbf{I}(t)-\mu(t))||^2$, $D_p^{\dagger} = (D_p^T D_p)^{-1} D_p^T$ is the pseudo-inverse matrix of D_p, the ϵI term is the additive noise, and D_p is the re-constructive dictionary, which is constructed with positive samples only [15]. It is worth mentioning that when a dictionary is not available, $D_p^{\dagger}$ is set to 0 in Eq. 13, and only the distance from the mean face $\mu(t)$ is used to find the tracked target face as:

$$p_d(\mathbf{I}(t)|\chi(t)) = exp(-||(\mathbf{I}(t) - \mu(t))||^2). \quad (14)$$

Furthermore, the probability $p_{wt}(\mathbf{I}(t)|\chi(t))$ of the candidate target face being the tracked target face represented by the dictionary D_p can be modeled by the Mahalanobis distance from $\mu(t)$ represented by D_p:

$$p_w(\mathbf{I}(t)|\chi(t)) = \mathcal{N}(\mathbf{I}(t); \mu(t), D_p {C_p}^{-2} D_p^T) = exp(-d_w), \quad (15)$$

where, $d_w = ||(\mathbf{I}(t)-\mu(t))^T D_p {C_p}^{-2} D_p^T (\mathbf{I}(t)-\mu(t))||^2$, C_p is a diagonal matrix containing the coefficients of the dictionary D_p atoms. The probability of the candidate target face being the tracked target face is given by (similar to $\mu(t)$ represented by D_p):

$$p_r(\mathbf{I}(t)|\chi(t)) = p_d(\mathbf{I}(t)|\chi(t))p_w(\mathbf{I}(t)|\chi(t)). \quad (16)$$

Furthermore, the probability of the candidate target face being well represented in terms of the classification dictionary D_c is given by the negative exponential value of the classification error, in other words, the candidate target face with the small classification error receives higher value, and vice versa:

$$p_c(\mathbf{I}(t)|\chi(t)) = exp(-\varepsilon_c), \quad (17)$$

where, $\varepsilon_c = ||Y_i - W\alpha_i||^2$ is given by Eq. 8. To obtain the combined probability of the candidate target face to be the tracked target face, the reconstruction and classification probabilities are combined as follows :

$$p(\mathbf{I}(t)|X(t)) = \Omega p_r(\mathbf{I}(t)|\chi(t)) + (1-\Omega)p_c(\mathbf{I}(t)|\chi(t), \quad (18)$$

where, Ω is a weight associated to the classification and reconstruction dictionaries. The candidate face sample that has higher combined probability is selected to be the tracked target face, and the associated affine parameters $\chi(t)$ are used to estimate landmarks on the tracked target face:

$$\Lambda_T(t) = \chi(t) \times [\Lambda(1); \vec{1}], \quad (19)$$

where, $\Lambda(1)$ are the landmark locations in the initial target face and $\vec{1}$ is an unitary vector of length Z (total number of landmarks). This tracked target face is used to update the two dictionaries depending on the reconstruction error obtained with Eq. 4. The pseudo code of the proposed procedure is shown in Algorithm 1.

III. Experimental Results and Discussion

The proposed method was implemented in Matlab 2015a on an IBM PC compatible with 3.40GHz i7-6700 CPU with 16GB internal memory. Furthermore, to increase computational efficiency, the initial/tracked target face is resized to $32 \times 32, i.e., u = 32$ for learning the dictionaries. To keep both the dictionaries representing the current tracked face

Algorithm 1 Dictionary Update and Face Tracking.

procedure DUT($I(t)$, $\Lambda_T(t-1)$, $\chi(t-1)$, D_n, α_n, μ_o, $flag, \Upsilon$)

▷ $I(t)$ is current frame, $\Lambda_T(t-1)$, $\chi(t-1)$ are landmarks and affine parameters of previous frame respectively, Υ is 1 if there is at least one more frame to process, otherwise Υ is 0.

  while ($\Upsilon \equiv 1$) **do**
    Draw affine parameters against $\chi(t-1)$ using Eq.12.
    Warp candidate target face samples from $I(t)$ using these affine parameters.
    Find the probability of each candidate target face sample being the target face using Eq. 18.
    Select the tracked candidate target face ($\mathbf{I}(t)$) using Eq. 18.
    Estimate the landmarks of the tracked target face using Eq. 19.
    calculate reconstruction error using Eq. 4.
    if ($\varepsilon_r \leq 0.05$) **then**
      $flag \leftarrow flag + 1$. ▷ Use this tracked target face sample for training.
    end if
    if ($flag \geq \tau$) **then**
      $flag \leftarrow 0$.
      Update the dictionaries D_p and D_c using Eq. 3.
      Update the mean $\mu(t)$ using Eq. 10.
    end if
  end while
  return $\Lambda_T(t), \chi(t), D, \mu(t), \mathbf{I}(t), \alpha$.
end procedure

appearance, the dictionaries update is performed after each set of three frames ($\tau = 3$) with a forgetting factor (f) (f=0.95). Various values of Ω, in the range [0 1] have been tested, and $\Omega = 0.8$ was used in the experiments. The proposed method runs 6 frames per second with patch size of $8 \times 8, i.e., v = 8$ and the configuration above. In addition, 400 affine parameters sample values are drawn to obtain and test the candidate target face samples (Eq. 12).

The YawDD dataset [16] was used in the experiments, which includes videos of drivers with various facial expressions and head poses, in varied illumination conditions in real driving scenarios, such as neutral expression, talking, laughing, singing, yawning, and so on. The camera is installed on the dash or under the front mirror. The face tracker is evaluated using Center Location Error (CLE), that estimates the difference between center locations of the tracked target face and the ground truth. Five videos were chosen for a detailed evaluation, which contain background and varied illumination. Additionally, person-specific characteristics, such as face changes, head motion, and glasses are also included. These five videos have been annotated manually, including the target face and landmarks ($Z = 68$) on the face, nose and the eyes. The proposed method was tested to verify if it can track consistently these face landmarks on these videos. Hence, the error was evaluated by the root mean squared error (RMSE) between the estimated landmark locations (Λ_T) and the manually-labeled ground truth (Λ_G) locations of the landmarks as follows:

$$\varepsilon(t) = \frac{1}{Z}\sum_{i=1}^{Z}||\Lambda_G^{(i)}(t) - \Lambda_T^{(i)}(t)||_2, \tag{20}$$

where $\varepsilon(t)$ represents the tracking error of the current frame at time t, whereas i is the i^{th} landmark and $\Lambda_G^{(i)}$, and $\Lambda_T^{(i)}$ represent the ground truth and estimated location in (x, y) of the i^{th} landmark. Fig. 2 shows some examples from the experimental results obtained with the proposed method. It can be seen that the proposed method performs well in different scenarios, including tilted face (see Fig. 2a), face expression changes (see Fig. 2b), change in the face size compared to the other drivers (see Fig. 2c), visual angle (see Fig. 2d) and illumination changes (see Fig. 2e). In the MMDL-Face Tracker (MMDL-FT), the dictionaries are updated using the tracked target face samples collected without checking their quality, as proposed by Ross et al [3].

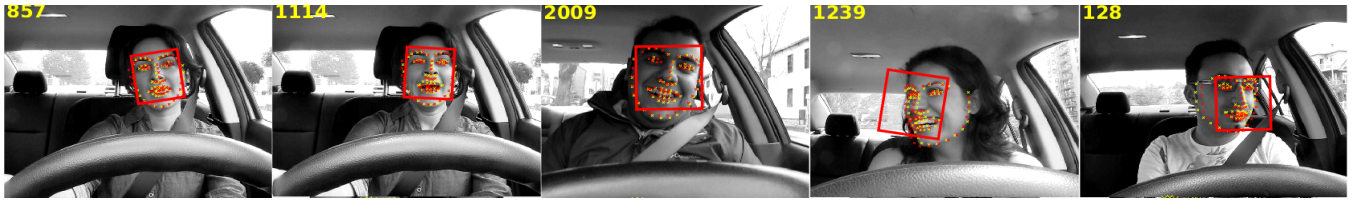
Table I compares the CLE of the proposed MMDL-FTU with the state-of-the-art methods based on all the videos of YawDD dataset with the camera installed on dash. Similarly, Table II provides the RMSE for the 68 landmarks for the five selected videos. The comparative methods include Incremental Learning for Robust Visual Tracking (ILRVT) [3] and Approximate Structured Output Learning for Constrained Local Models (CLM) [5]. The smallest error for each video is shown in bold. Fig. 3 provides RMSE plots of the proposed and the comparative methods. The experimental results are shown in Tables I and II, and in Fig. 3 indicate that the proposed MMDL-FT and MMDL-FTU methods tend to perform better than the comparative methods, and MMDL-FTU shows a slightly better performance. Also the face tracking methods perform better on female videos, because of the smooth texture compared to male videos, that may have different styles of beard, mustache, etc. The experimental results obtained for face tracking suggest the potential of dictionary learning for a non-rigid object (e.g. face) tracking algorithms, and provide insights for further improvements.

TABLE I
CENTER LOCATION ERROR (CLE) COMPARISON.

Video	[5]	[3]	MMDL-FT [3]	MMDL-FTU
Male videos	13.02	14.74	10.61	**10.36**
Female videos	10.14	11.33	8.70	**8.68**
Average	11.58	13.03	9.65	**9.52**

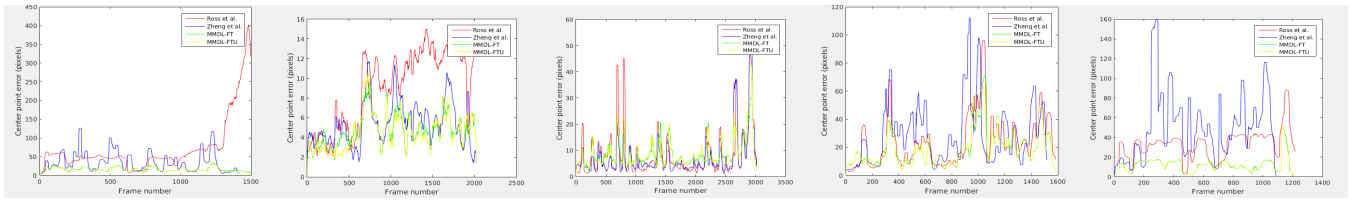
IV. CONCLUSION

This work proposed a new method for tracking a target face, which is adaptive to the face appearance changes (e.g. changes in facial expressions and pose). The proposed method relies on online learning of a multi-model dictionary, that



(a) Tilted face; (b) Expression change; (c) Face size change; (d) Visual angle; (e) Illumination change.

Fig. 2. Some examples of tracking face conditions evaluated in the tests.



(a) (b) (c) (d) (e)

Fig. 3. RMSE across frames of 68 landmarks.

TABLE II
RMSE COMPARISON IN TERMS OF FACIAL LANDMARKS TRACKING.

Video	[5]	[3]	MMDL-FT [3]	MMDL-FTU
1	11.46	10.56	10.12	**9.73**
2	12.26	**6.23**	7.19	6.50
3	14.02	12.17	**7.63**	7.76
4	33.93	21.43	22.02	**16.62**
5	18.73	15.43	8.06	**7.76**
Average	17.92	12.61	10.45	**9.67**

helps detecting the target face against a complex background. Furthermore, the proposed method takes advantage of data collected during face tracking to learn the multi-model dictionary incrementally (i.e. to adaptively learn a dictionary for face detection/reconstruction and another dictionary for face/background discrimination), updating the dictionary when the face appearance changes. The preliminary experimental results suggest that the proposed method can provide competitive face tracking results in comparison to methods that are representative of the state-of-the-art. In the continuation of this work, we plan to investigate the applicability of the proposed scheme to other types of non-rigid objects.

ACKNOWLEDGMENT

The authors would like to thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brazil) for financial support.

REFERENCES

[1] M L Siqueira, J Scharcanski, and POA Navaux, "Echocardiographic image sequence segmentation and analysis using self-organizing maps," *The Journal of VLSI Signal Processing*, vol. 32, no. 1, pp. 135–145, 2002.

[2] CR Jung and J Scharcanski, "Wavelet transform approach to adaptive image denoising and enhancement," *Journal of Electronic Imaging*, vol. 13, no. 2, pp. 278–285, 2004.

[3] D A Ross, J Lim, R S Lin, and M H Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[4] S Lucey, Y Wang, M Cox, et al., "Efficient constrained local model fitting for non-rigid face alignment," *Image and vision computing*, vol. 27, no. 12, pp. 1804–1813, 2009.

[5] S Zheng, P Sturgess, and P Torr, "Approximate structured output learning for constrained local models with application to real-time facial feature detection and tracking on low-power devices," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE, 2013, pp. 1–8.

[6] J Mairal, F Bach, J Ponce, and G Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.

[7] S R Fanello, N Noceti, G Metta, and F Odone, "Multi-class image classification-sparsity does it better.," in *VISAPP (1)*, 2013, pp. 800–807.

[8] G Peyré, "Sparse modeling of textures," *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 17–31, 2009.

[9] A Michal, E Michael, and B Alfred, "K-svd: Design of dictionaries for sparse representation," *SPARS*, vol. 5, pp. 9–12, 2005.

[10] H Liu, S Li, and L Fang, "Robust object tracking based on principal component analysis and local sparse representation," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 11, pp. 2863–2875, 2015.

[11] X Cheng, N Li, T Zhou, L Zhou, and Z Wu, "Visual tracking via sparse representation and online dictionary learning," in *International Workshop on Activity Monitoring by Multiple Distributed Sensing*. Springer, 2014, pp. 87–103.

[12] Y Xie, W Zhang, C Li, S Lin, Yanyun Qu, and Y Zhang, "Discriminative object tracking via sparse representation and online dictionary learning," *IEEE Transactions on Cybernetics*, vol. 44, no. 4, pp. 539–553, 2014.

[13] M Elad and M Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[14] L Wang, Ke Lu, P Liu, Rajiv Ranjan, and L Chen, "Ik-svd: dictionary learning for spatial big data via incremental atom update," *Computing in Science & Engineering*, vol. 16, no. 4, pp. 41–52, 2014.

[15] E Flores and J Scharcanski, "Segmentation of melanocytic skin lesions using feature learning and dictionaries," *Expert Systems with Applications*, vol. 56, pp. 300–309, 2016.

[16] S Abtahi, M Omidyeganeh, S Shirmohammadi, and B Hariri, "Yawdd: a yawning detection dataset," in *Proceedings of the 5th ACM Multimedia Systems Conference*. ACM, 2014, pp. 24–28.