Facial feature tracking for Emotional Dynamic Analysis

Thibaud Senechal[1], Vincent Rapp[1], and Lionel Prevost[2]

[1]ISIR, CNRS UMR 7222
Univ. Pierre et Marie Curie, Paris
{rapp, senechal}@isir.upmc.fr

[2]LAMIA, EA 4540
Univ. of Fr. West Indies & Guyana
lionel.prevost@univ-ag.fr

Abstract. This article presents a feature-based framework to automatically track 18 facial landmarks for emotion recognition and emotional dynamic analysis. With a new way of using multi-kernel learning, we combine two methods: the first matches facial feature points between consecutive images and the second uses an offline learning of the facial landmark appearance. Matching points results in a jitter-free tracking and the offline learning prevents the tracking framework from drifting. We train the tracking system on the Cohn-Kanade database and analyze the dynamic of emotions and Action Units on the MMI database sequences. We perform accurate detection of facial expressions temporal segment and report experimental results.

Keywords: Facial feature tracking, Emotion recognition, emotional dynamic, Multi-Kernel SVM

1 Introduction

A current challenge in designing computerized environments is to place the user at the core of the system. To be able to fully interact with human beings, robots or human-centered interfaces have to recognize user's affective state and interpret gestures, voice and facial movements.

While several works have been made to recognize emotions, only few extract the emotional dynamic. In particular, the emotion temporal segment, which is the limit of the emotion display, is crucial for a system waiting for a specific reaction from its user. But it is even more important for complex facial expression detectors which need to know when and how an expression appears before actually recognizing it. We propose here a facial feature tracking system dedicated to emotion recognition and emotional dynamic analysis.

There is much prior work on detecting and tracking landmarks. Appearance-based methods use generative linear models of face appearance such as Active Appearance Models [1] used in [2] and [3], 3D Morphable Models [4] or Constrained Local Models [5]. Although the appearance-based methods utilize much knowledge on face to realize an effective tracking, these models are limited to some common assumptions, e.g. a nearly frontal view face and moderate facial expression changes, and tend to fail under large pose variations or facial deformations in real-world applications. These models introduce too strong constraint between points. Features-based tracking methods [6, 7] usually track each landmark point by performing a local search for the best matching position, around which the appearance is most similar to the one in the initial frame. Tian et al [8, 9] use multiple-state templates to track the landmarks. Landmark point tracking together with masked edge filtering is used to track the upper landmarks. Over short sequences, features-based tracking methods may be accurate and jitter-free. Over long ones, these methods often suffer from error accumulation, which produces drift, and cannot deal with severe aspect changes.

As we want to track landmarks during display of facial expressions, we have to take into account the high deformation of facial shapes like eyes, brows and mouth. Hence, we try to localize 18 landmarks independently (4 per eye, 3 per brow and 4 for the mouth). In a sequence, we use the detection in previous images to detect landmarks in the current one by matching patches from precedent images with patches of the current image. But the main problem is that if a given detection is not really accurate, the following matching will lead to poorer detection resulting in a drift like other features-based tracking method. To solve this problem, we propose to incorporate prior knowledge on the appearance of the searched landmark. In this goal, we use multi-kernel algorithms in an original way to combine the temporal matching of patches between consecutive images and the hypothesis given by a static facial feature detector.

To check performances of the tracking system, two tasks are achieved. We recognize emotion on the Cohn-Kanade database [10] and we detect temporal segment of emotions and Action Units (AUs) on the MMI database [11]. Comparison with state-of-art method for AUs temporal segmentation are provided.

The paper is organized as follows. Section 2 describes the tracking method. Section 3 details the setup to train and test the system. Section 4 reports experimental results for the emotion recognition task. Section 5 deals with emotion and AUs temporal segmentation. Finally, section 6 concludes the paper.

2 Facial features tracking method

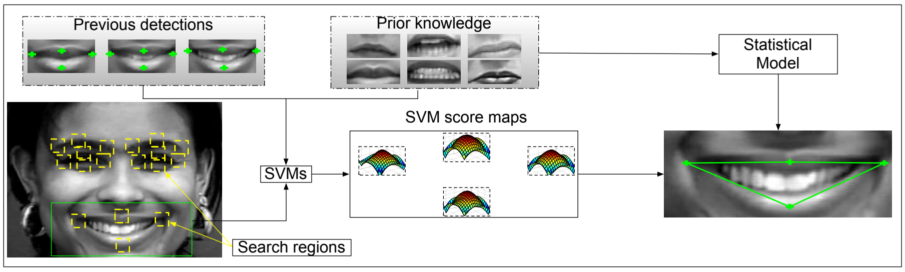

Fig. 1. Overview of the proposed method.

2.1 Overview

Fig. 1 gives an overview of the proposed system. To detect landmarks in a sequence image, we first use the previous detection to define a region of interest (ROI) containing candidate pixels that may belong to a given landmark. Then, we create two sets of features for each candidate pixel. The first set of features is called static as it just considers patch (region surrounding the candidate pixel) extracted from the current image. The second one consists of dynamic features. It measures how much this given patch matches with those extracted in the previous images. These two sets fed a Multi-Kernel Support Vector Machine (SVM). Hence, for each candidate pixel, the SVM output gives a confidence index of being the searched landmark or not. Finally, we check information arising from the confidence index of each candidate pixel for each landmark with statistical models. We use five point distribution models: two four-points models for the eyes, two three-points models for the brows and one four-points model for the mouth. Model parameters are estimated on expressive faces.

2.2 Features

Two information are extracted on each candidate pixel i as shown in fig. 2:

- Static features are the intensities g_i of the neighboring pixels. We extract an 11x11 patch centered on the candidate pixel from facial images that have an inter-ocular distance of 50 pixels. Patch intensity is normalized after mean and standard deviation estimation.
- Dynamic features are the correlations between a patch of the current image I_t with patches extracted in each previous image $(I_{t-1}, I_{t-2}, I_{t-3}, ...I_{t-N})$ and centered on the landmark detected. In this way, we compute N correlation maps. Features are the (X, Y) coordinates $(p_i^{t-1}, p_i^{t-2}, p_i^{t-3}, ...p_i^{t-N})$ of each candidate pixel in relation to the maximum of each correlation map. Thus, the best matching point has the position (0,0).

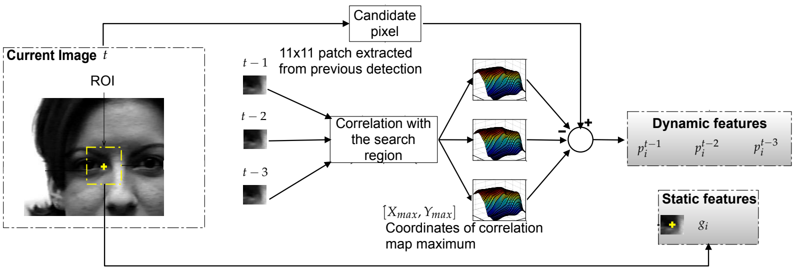

Fig. 2. Static and dynamic features extraction.

2.3 Multi-kernel learning

The system has to discriminate target samples (candidate pixel that belong to the searched landmark) from non-target samples (candidate pixel which does not belong to the searched landmark).

Given $x_i = (g_i, p_i^{t-1}, ...p_i^{t-N})$ samples associated with labels $y_i \in \{-1, 1\}$ (target or non-target), the classification function of the SVM associates a score s to the new sample (or candidate pixel) $x = (g, p_i^{t-1}, ...p_i^{t-N})$:

$$s = \left(\sum_{i=1}^{m} \alpha_i k(x_i, x) + b \right) \tag{1}$$

With α_i the dual representation of the hyperplane's normal vector [12]. k is the kernel function resulting from the dot product in a transformed high-dimensional feature space.

In case of multi-kernel SVM, the kernel k can be any convex combination of semi-definite functions. In our case, we have one kernel function per features.

$$k = \beta_g k_g(g_i, g) + \sum_{j=1}^{N} \beta_j k_j(p_i^{t-j}, p^{t-j}) \text{ with } \beta_j \geq 0, \sum_{j=1}^{K} \beta_j = 1 \tag{2}$$

Weights α_i and β_j are set to have an optimum hyperplane in the feature space induced by k. This optimization problem has proven to be jointly-convex in α_i and β_j [13]. Therefore, there is a unique global minimum than can be found efficiently.

β_g represents the weight accorded to the static feature and $\beta_1 ... \beta_N$ are the weights for the dynamic ones. Thus, by using a learning database, the system is able to find the best combination of these two types of feature that maximize the margin.

This is a new way of using multi-kernel learning. Instead of combining different kinds of kernel functions (for example, radial basis with polynomial), we combine different features corresponding to different kinds of information. The first one, represented by the function k_g, corresponds to a local landmark detector which is able to localize these points without drift but can sometimes leads to inaccurate detections. The second one, represented by the functions $k_1 ... k_N$, tries to match hypotheses between consecutive images. It is much more stable and will rarely results in bad detections but a drift can appear along the sequence. Combining both information leads to accurate detections with no drift.

Among all candidate pixels, we need to choose one representing the searched landmark. In the perfect case, we should have a positive SVM output s if the candidate pixel is close to the landmark and negative otherwise. In the general case, when we have zero or more than one candidate pixel with a positive score, we use the value of s to take a decision. This score can be seen as a confidence index of the candidate pixel belonging to the landmark.

2.4 Statistical validation

We estimate five Gaussian Point Distribution Models (PDM) representing eyes, brows and the mouth by using EM algorithm on the training dataset. For each of these models, we tune a threshold T such as 95% of the training shapes have a distance to the model lower than T. During tracking, the SVM scores each candidate pixel. Then, we have to choose among these candidates, for each landmark, the one that leads to a valid facial shape. The first hypothesis is the shape having the highest SVM scores. It is considered as valid if its distance to the model is lower than T. Otherwise, another hypothesis is built considering the next best combination of SVM scores and so on.

3 Experimental setup

The tracking system has been trained using the Cohn-Kanade database [10]. This is a representative, comprehensive and robust test-bed for comparative studies of facial expression. It contains 486 sequences (during from 10 to 50 frames) starting with the neutral expression and ending with the expression apex.

3.1 Training database

We have manually labeled landmark positions for the first and last images of each sequence. Yet, we need the facial feature position for all the images of the sequence to compute correlation maps. Instead of manually labeling it, we trained a special detector using as prior knowledge the first and last images. This has the advantage of being less time consuming. Moreover, it leads to a more robust tracker because it is trained with correlation maps computed using noisy detections. To build the training database, for each landmark, we proceed as follows:

- We resize all images to have an interocular distance of 50 pixels.
- For each sequence, we use the last image and the position of the given landmark (ground truth) to create training samples.
- We compute correlation maps between the ROI extracted in the last image and patches surrounding the ground truth in previous images.
- We choose as target samples the 9 closest points to the ground truth and as non-target samples 8 other ones distanced from 5 pixels to the ground truth.
- We repeat this process with the first image of each sequence, using the next images to compute correlation maps (as though the sequence was reversed). This way we train the tracking system with sequences in which the expression disappears from the face.

This results in 18 target samples and 16 non-target samples per sequence.

3.2 Multi-kernel learning

We use a SimpleMKL [14] algorithm to train multi-kernels SVMs. For the gray level patches, we use a linear kernel. For the position of candidate pixels in relation to correlation map maxima, we use a second order polynomial kernel. We choose this kernel because of the good samples closeness with the maximum of the correlation maps so the border between good and wrong samples looks like a circle.

The algorithm found that matching with previous images $I_{t-4}, I_{t-5}, ...$ is not useful for the detection and set $\beta_4, \beta_5 ...$ to zero. So we train the SVMs using only k_g, k_1, k_2, k_3 and we find one set of weights $\beta_g, \beta_1, \beta_2, \beta_3$ for each facial landmark. Mean weights learned for the points belonging to the same landmark set are reported table 1.

Facial features	β_g	β_1	β_2	β_3
Brows (6 points)	0.1300	0.6314	0.1774	0.0612
Eyes (8 points)	0.3142	0.4625	0.1477	0.0756
Mouth (4 points)	0.6918	0.1730	0.0822	0.0529

Table 1. Mean weights of points belonging to the same facial feature.

We first notice that we always have $\beta_1 > \beta_2 > \beta_3$. This means a more important weight is associated to the matching with most recent images. Moreover the points that are difficult to detect on static images have the weakest coefficient k_g, meaning the system does not overly use the static features. The brows are the most difficult to detect because the ground truth is not well-defined. The eyes, particularly in the Cohn-Kanade database, have some illumination problems and the eye contour is not always very clear for the tracking system. On the contrary, the mouth has the most salient points. Therefore, weight values tined by the SimpleMKL algorithm are in agreement with our intuition.

3.3 Testing phase

To test the tracking system, we use a 2-fold validation setup in which half of the sequences is used for training and cross-validation, and the other half is used as an independent set. We take care not to have sequences of the same subject in the training and testing set. This way, we have experimental results, i.e landmarks coordinates given by the tracking system for all the sequences.

During the test phase, we start by detecting landmarks on the first image with a facial landmark detector [15]. The following image is resized using the interocular distance computed by considering the previous detections. For each landmark, we perform as follows. First, we select a ROI of 15x15 pixels (30% of the interocular distance) surrounding the last detection. This allows the landmark point to move quickly from one image to another. Then, we test each candidate pixel of the ROI with the SVM classifier, leading to one score per candidate. Finally, we combine candidate pixels to create shape hypotheses and use the PDM to validate them, as described in section 2.4.

4 Experiments on the Cohn-Kanade database

4.1 Performance measures

As a first performance measure, we compute the localization error on the last image of each sequence. This is the mean Euclidian distance between the 18 detected points and the true (labelled) landmark points, normalized by the interocular distance. The Mean Localization Error is then computed over the whole test dataset. But the main objective of our system is to track an emotion. Some detection inaccuracies can reduce the emotion intensity and be harmful to emotion recognition. On the contrary, they will be harmless if they amplify the emotion intensity.

Hence, the second performance measure is the Emotion Recognition Accuracy. 400 sequences of the Cohn-Kanade database are labeled by one of the five basic emotions (anger, fear, joy, relief, sadness). We want to verify that the tracking system can accurately recognize these emotions. To do so, we compute the difference between the 18 detected landmarks at the beginning and the end of each sequence. These feature vectors are used to train five one-versus-all binary classifiers, each one dedicated to an emotion, using as positive all the samples corresponding to this emotion and all the other samples as negatives. During testing, at the end of the sequence, the feature vector is computed and fed the five classifiers. The emotion belonging to the classifier with the highest decision function value output is assigned to the sequence. To test the generalization to new subjects, we use a leave-one-subject-out cross-validation setup in which all the data from one subject are excluded from the training database and used for test.

4.2 Experimental results

Table 2 details the accuracy when of the tracking system. If we only use the static features (function k_g), the Mean Localization Error (MLE) does not overtake 11% of the interocular distance. Using only the dynamic features (functions k_1, k_2, k_3), error decreases to 5.7%. Combining both features achieves better result with an error of 5.3%. Finally, the PDM mainly corrects the outliers and reduces the error standard deviation (SD). These local models do not unduly constraint points allowing expressive shapes and do not change the emotion recognition performance.

We can notice that even if the function k_g does not achieve good results, it is still useful combined with other functions using matching with previous images. It provides information about the searched landmark and prevents the tracking system from drifting. Moreover the Cohn-Kanade sequences are relatively short and the kernel k_g would be even more useful on longer sequences.

Emotion recognition accuracy (ERA) increases in the same sense, from 60.2% with the sole static information to 78.0% when using the full tracking system. Finally, let us notice that the detections reached by the tracking system at the end of the sequences lead to an emotion recognition score (78%) close to the one

	MLE	SD	ERA
Using k_g only	11.4%	2.81%	60.2%
Using k_1, k_2, k_3 only	5.7%	1.91%	76.2%
Without stat. validation	5.3%	1.86%	78.0%
Full tracking system	5.1%	1.81%	78.0%
Ground truth	0%	0%	80.0%

Table 2. Experimental results achieved on the Cohn-Kanade database.

reached when using the ground truth (80%). This shows that the system is well suited to track emotions through sequences.

4.3 Sensitivity to initialization

In fig. 3, we investigate the robustness of the tracking system to initialization (detection on the first image). In this experiment, we do not use a facial landmark detector but the ground truth of the first image to initialize the tracker. We add artificially to this ground truth a uniform noise. As some shapes cannot be statistically possible, we choose the closest shape within the noisy detections validated by the PDM. The mean localization error of this shape normalized by the interocular distance is reported as the initialization error.

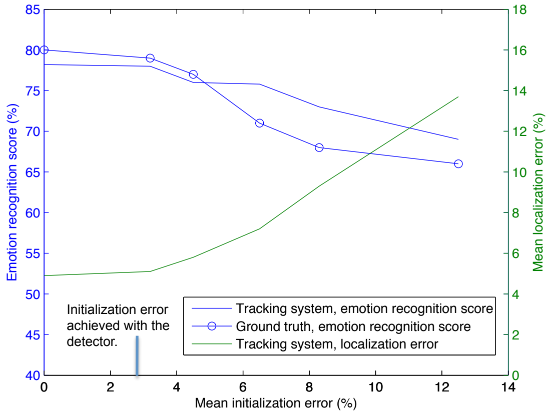

Fig. 3. Sensitivity to the initialization error: localization error and emotion recognition accuracy.

We notice that for an initialization error greater than 4% of the interocular distance, the localization error increases. It means that landmarks are tracked with less accuracy. But even with these inaccurate detections, the emotion is still correctly detected. With an initialization error of 8% the emotion recognition score only decreases from 78% to 74%. To measure the influence of poor detections in the first image, we perform another experiment. We use the labeled landmarks (ground truth) of the last image. We notice that the tracking system leads to landmark localizations on the apex expression image more useful for the emotion recognition task than the ground truth.

5 Temporal analysis of expressions

Since the first image og Cohn-Kanade sequences represents the neutral face and the last, the expression apex, dynamic analysis of these sequences can be biased. In this section, we will evaluate the ability of the proposed system to detect an emotion temporal segment. We call onset the temporal beginning of the facial movement and offset its ending. To evaluate the trackers ability to (1) follow subtle facial movements and (2) generalize on other sequences, we decide to test the tracking system on another challenging database.

5.1 MMI database

The MMI Facial Expression database [11] holds videos of about 50 subjects displaying various facial expressions on command. We apply our tracking system on the 197 sequences labeled as one of the six basic expressions and, to compare with other works, 256 AUs labeled sequences. In these sequences, AU can appear alone or in combination. We have chosen the AU-sequences in which the onset and offset are already labeled. We also manually labeled the onset and offset of the basic expression sequences. Contrary to the Cohn-Kanade database, the subjects can wear glasses and beard. Sequences last several seconds recorded at a rate of 25 frames per second.

5.2 Temporal segment detection

Our goal is to detect the temporal segment using only landmark detections along the sequence. In this way, we can check if the system is accurate enough to track subtle movements. To detect the onset and the offset in each sequence, we proceed as follow (Fig. 4):

- For each frame, we express the landmark coordinates in a barycentric basis. To detect emotions, we use the coordinates X_i^p, Y_i^p of all the landmarks p and frames i. To detect upper Aus, we only use the brow and eye landmarks (14 points). To detect lower Aus, we use only the mouth landmarks (4 points).
- We first try to separate each sequence in half, each half containing either the onset or the offset. We compute the euclidian distance $D(i)$ between the coordinates in the frame i and the coordinates in the first frame and its derivative $d(i)$.

$$D(i) = \sum_p \sqrt{(X_i^p - X_1^p)^2 + (Y_i^p - Y_1^p)^2} \tag{3}$$

$$d(i) = D(i) - D(i-1) \tag{4}$$

We cut the sequence in the frame i_c such that we maximize:

$$\max_{i_c} \sum_{i=1}^{i_c} d(i) - \sum_{i=i_c}^{end} d(i) \tag{5}$$

This way the onset of the expression is likely to be before i_c and the offset after i_c.

- We compute $G(i)$, the sum of the gradient magnitudes over 6 frames. This represents the global movement of the facial landmark.

$$G(i) = \sum_{p} \sqrt{(\sum_{k=0}^{2} X^p_{i+k} - \sum_{k=1}^{3} X^p_{i-k})^2 + (\sum_{k=0}^{2} Y^p_{i+k} - \sum_{k=1}^{3} Y^p_{i-k})^2} \quad (6)$$

- The expression onset corresponds to the maximum of $G(i)$ for $i < i_c$ and the expression offset corresponds to the maximum of $G(i)$ for $i > i_c$.

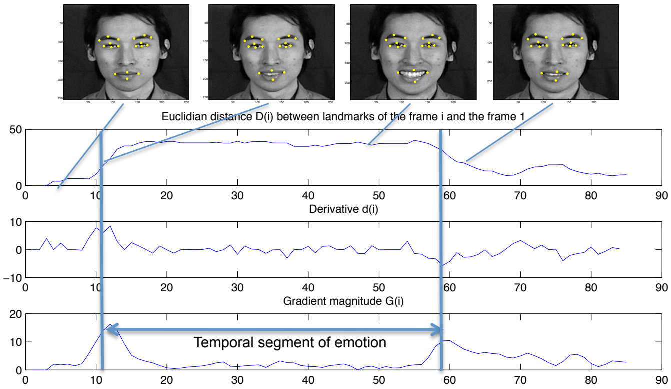

Fig. 4. Detection of the start and the end of the emotion.

5.3 Segmentation of basic emotions

We report the mean difference between the true label and the detected label in table 3. We detect the temporal segment of the emotion with an accuracy of 5 frames or 0.2 second. As the emotion in these sequences lasts between 40 and 120 frames, we can say that the tracking system leads to a good segmentation of the emotion.

	Mean error	Standard deviation
Onset	4.5	5.1
Offset	5.5	4.9

Table 3. Detection of the emotion temporal segment: mean error and standard deviation in number of frames (record speed: 25 frames/second).

5.4 Actions Units segmentation: comparison with other works.

To the best of our knowledge, there is no tracking system specifically addressing the problem of emotion temporal segmentation. We decide to compare our work with appearance-based system to check if the tracking of facial landmarks is accurate enough to lead to accurate temporal segmentation. The only results are reported by Valstar & Pantic [16] and Koelstra et al. [17]. In the last one, they detect the AU segment on 264 sequences of the MMI database. They report temporal error (in frames) for onset and offset for AU detection. In the same way, we report results in fig. 5 for each AU to perform a fair comparison. But as we do not know the sequences they used for their experiments, we are not able to straightly compare.

We can notice that the proposed tracker reaches the same overall accuracy as an appearance-based system. Such results can be obtained only if we can track very subtle facial movements. The worst results are for upper AUs, particularly AU5 (upper lid raiser), AU6 (cheek raiser) and AU7 (lid tightener) coding the eye movements. These AUs are more visible with appearance features in the higher cheek region (like wrinkles) than the eyelids motion. So, it is not surprising that the tracker on these AUs is less accurate. Good detections are reached for the lower AUs (AUs 10, 12, 15, 20, 23, 24, 25, 27). Using only the mouth points, we can detect temporal segments more accurately than state-of-art.

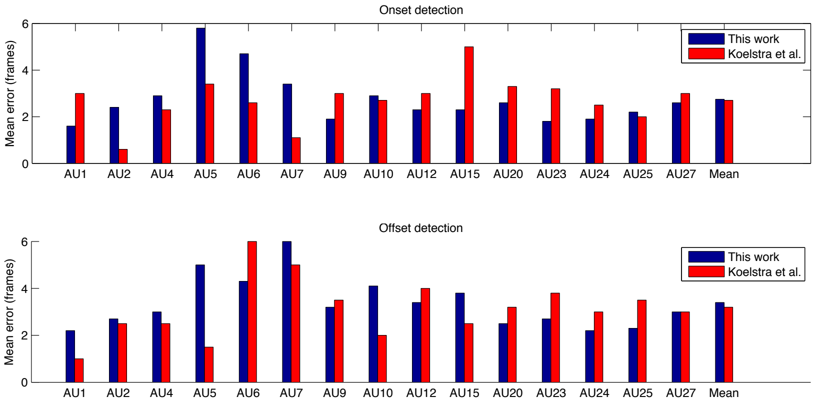

Fig. 5. Mean error in number of frames for the detection of the AU temporal segment.

6 Conclusion

We present in this paper a fully automatic tracking system of 18 facial landmarks. Advanced Multi-kernels algorithms are applied in an original way to combine point matching between consecutive images with a prior knowledge of facial landmarks. The system is suited for real-time application as we are able to treat five frames per second with a non-optimal Matlab code.

This system tested on the Cohn-Kanade database has been able to track facial emotions even with inaccurate facial feature localizations. Its localizations lead to an emotion recognition performance almost as good as the one achieved with ground truth. This confirms that our tracker is well-suited for facial feature tracking during emotional display.

Successful temporal segmentation of emotion and AUs on the MMI database has been realized. Experiments show lower AU temporal segments are as well as by state-of-art methods. Results for the upper AUs are promising too, but seem to need more than eyelid movements to be detected accurately. In future works, we will combine the landmark coordinates with texture around these points to increase results.

References

1. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. ECCV '98 (1998)
2. Al Haj, M., Orozco, J., Gonzalez, J., Villanueva, J.: Automatic face and facial features initialization for robust and accurate tracking. In: ICPR'08. (2008) 1–4
3. Zhou, M., Liang, L., Sun, J., Wang, Y., Beijing, C.: Aam based face tracking with temporal matching and face segmentation. In: CVPR '10. (2010) 701–708
4. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. PAMI (2003)
5. Cristinacce, D., Cootes, T.: Feature detection and tracking with constrained local models. In: BMVC '06. (2006) 929–938
6. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI'81. Volume 3. (1981) 674–679
7. Zhu, Z., Ji, Q., Fujimura, K., Lee, K.: Combining kalman filtering and mean shift for eye tracking under active ir illumination. Pattern Recognition **4** (2002)
8. Tian, Y., Kanade, T., Cohn, J.: Dual-state parametric eye track. In: FG '00. (00)
9. Tian, Y., Kanade, T., Cohn, J.: Recognizing upper face action units for facial expression analysis. In: CVPR '00. (2000) 294–301
10. Kanade, T., Tian, Y., Cohn, J.: Comprehensive database for facial expression analysis. In: FG '00. (2000) 46
11. Pantic, M., Valstar, M., Rademaker, R., Maat, L.: Web-based database for facial expression analysis. In: ICME '05. (2005) 5
12. Scholkopf, B., Smola, A.: Learning with kernels. Cambridge, MIT Press (2002)
13. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. JMLR **5** (2004) 27
14. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: Simplemkl. JMLR (2008)
15. Rapp, V., Senechal, T., Bailly, K., Prevost, L.: Multiple kernel learning svm and statistical validation for facial landmark detection. FG'11 (2011 (to appear))
16. Valstar, M., Pantic, M.: Fully automatic facial action unit detection and temporal analysis. In: CVPRW'06, IEEE (2006) 149
17. Koelstra, S., Pantic, M., Patras, I.: A dynamic texture based approach to recognition of facial actions and their temporal models. PAMI (2010)