Multiple Kernel Learning SVM and Statistical Validation for Facial Landmark Detection

Vincent RAPP[1], Thibaud SENECHAL[1], Kevin BAILLY[2], Lionel PREVOST[3]

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

[2]Institut Telecom - Telecom ParisTech
CNRS/LTCI, Paris
kevin.bailly@telecom-paristech.fr

[3]LAMIA
EA 4540
University of French West Indies & Guyana
lionel.prevost@univ-ag.fr

***Abstract*— In this paper we present a robust and accurate method to detect 17 facial landmarks in expressive face images. We introduce a new multi-resolution framework based on the recent multiple kernel algorithm. Low resolution patches carry the global information of the face and give a coarse but robust detection of the desired landmark. High resolution patches, using local details, refine this location. This process is combined with a bootstrap process and a statistical validation, both improving the system robustness. Combining independent point detection and prior knowledge on the point distribution, the proposed detector is robust to variable lighting conditions and facial expressions. This detector is tested on several databases and the results reported can be compared favorably with the current state of the art point detectors.**

I. Introduction

Facial landmarks detection is an important step in computer vision applications such as facial expression recognition, face identification, face alignment, face tracking or facial synthesis. Despite many works have been proposed, locating facial landmarks is still an unsolved problem for applications that need to operate under a wide range of conditions such as illumination variations, occlusions, poses, expressions, etc. One challenge for the development of such detectors is an inherent tradeoff between robustness and accuracy.

Previous method for facial landmark detection can be classified into two categories: model-based methods and independent detection points methods (without models). Model-based methods regard all facial landmarks as a shape which is learned from a set of labelled faces, and try to find the proper shape for any unknown face. The second category usually tries to find each facial landmark independently, without any model.

Typical model-based methods use two types of models: explicit or implicit. Explicit models based methods include active shape or active appearance models (ASM/AAM) [1], [2]. Other approaches using extended AAM or ASM have also been proposed as well. Milborrow and Nicolls [3] make some simple extensions to the ASM and use it to locate landmarks in frontal views of upright faces. Approaches combining texture and shape-based methods have also been proposed. Cristinacce and Cootes [4] use PCA on the grey level images combined with ASM. Implicit models based methods use unstated models. For example, [5] and [6] use pixel gray levels as input of a Neural Network to detect multiple facial landmark. This way, spacial relation between points are implicitly learned by the Neural Network. All these methods use strong relation between points but they 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.

Independent detection points methods detect each facial landmark independently. Vukadinovic and Pantic [7] detect 20 facial points using GentleBoost classifier learned on features extracted with Gabor filters. These methods, because of the absence of relation between points, can detect some outliers that introduce some robustness problems.

Recently, methods combining these two kinds of approach have been proposed. For example, the Pictorial Structure Matching (PSM) approach of Felzenszwalb and Huttenlocher [8] learns detectors for a set of manually points and a tree structure for the spatial relationships between selected pairs of features. Valstar *et al.* [9] propose a method based on Support Vector Regression to detect independently each facial points and use Markov Random Field to exploit relation between points.

This paper fits into this scheme combining independent point detection with explicit models. This approach combines advantages from the two facial landmarks detection paradigms : The strong relation between the model points avoids some outlier detections but this method does not suffer from AAM's initialization and convergence problems. Our

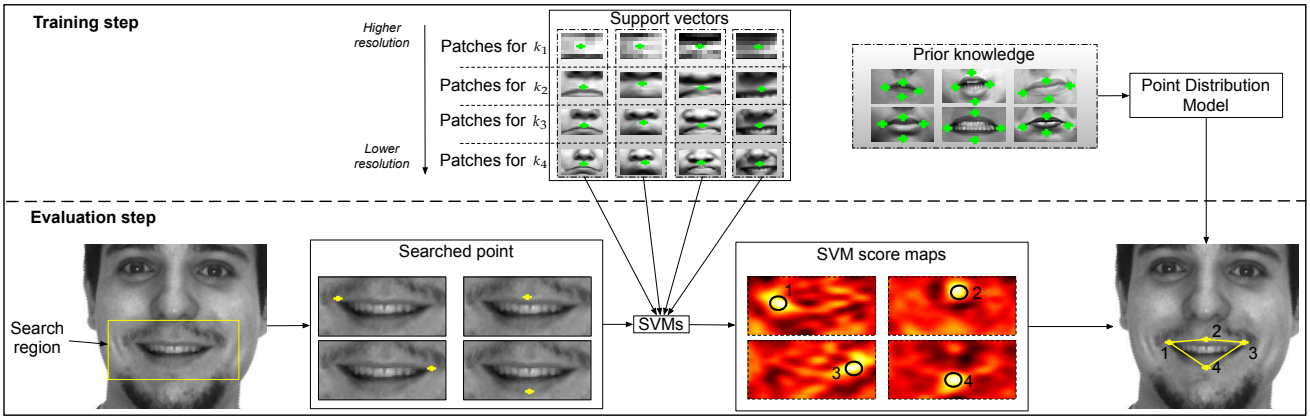

Fig. 1. Overview of the proposed method

contributions are three-folds :

1) A new facial landmark localizer combining Support Vector Machine point detection and statistical Point Distribution Models.
2) A new training methodology based on a specific bootstrap process which leads to better generalization abilities.
3) A new multi-resolution detector using the recent multiple kernel algorithms for SVM.

The remainder of this paper is organized as follow: in section II we present our method including multiple kernel learning applied to facial point detection, the pyramidal patches multi-resolution extraction process, the bootstrap process and the statistical validation. In section III, we present our experimental results on several databases including frontal, near frontal and expressive images. Finally, section IV concludes this paper.

II. Methodology

To detect facial feature points in an image, we first locate the face using the Viola-Jones face detector [10]. During the training step, we create sets of features corresponding to patches at different resolution. Weights on these sets are learned using the recent multiple kernel algorithms for Support Vector Machine combined to an original bootstrap process. During the evaluation step, the SVM gives, for each candidate pixel contained in a search region, a confidence index of being the searched facial feature. Finally, we combine information arising from the confidence index of each candidate pixel of each facial feature with a Point Distribution Model (PDM). This PDM is learned on the expressive training database using Gaussian Mixture Models (GMM).

A. Feature Extraction

The proposed facial landmark detection method uses individual feature patch templates. The size of these patches have to be relevant according to the inter-ocular distance: a small size will only encode the local information loosing global details, and on the other hand a large size will only encode coarse information. In this paper, we use multi-resolution patches extracting different level of information. For a pixel i, we take the first patch (p_i^1) large enough to encode plenty of general information. The other patches ($p_i^2, p_i^3, ..., p_i^N$) are extracted cropping a progressively smaller area giving increasingly detailed information. Then, all patches are subsampled so that all samples have the same size (fig.2). All patches are built from gray level intensities. Thus, high resolution patches encode local information and small details, such as canthus or pupil location, around the point. Low resolution patches, on the other hand, encode global information. In the case of the patch p_i^4 in fig. 2, eye landmark localization is helped by the nose and hair positions.

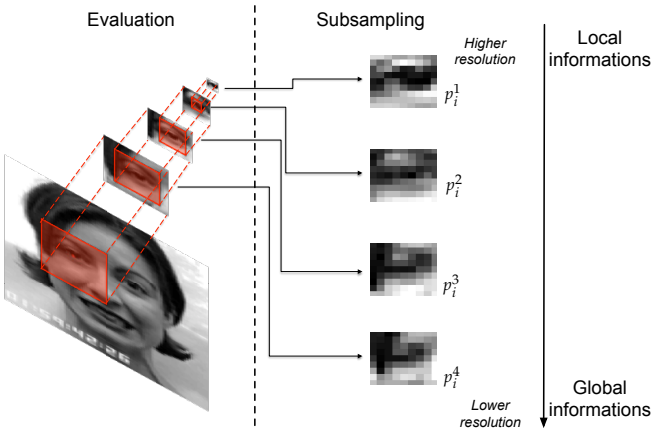

Fig. 2. Pyramidal multi-resolution patches for a pixel i. At each step, the patch area is divided by 4, emphasizing different level of information

This method uses one detector per point. For each facial landmark, the training is performed with positive and negative examples. For a given facial point, we use 9 multi-resolution patches as positive examples (target samples): the first centered on the ground truth and the 8 others on 8 positions surrounding the ground truth. As negative (non-target samples) set, we use 16 multi-resolution patches more or less distant from the true facial point (fig. 3).

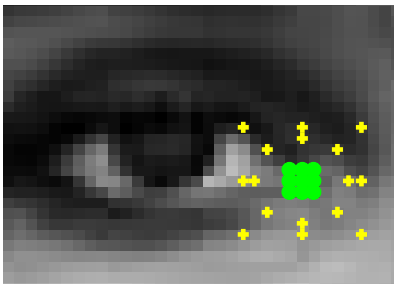

Fig. 3. The target samples are the 9 closest points to the ground truth, and the non-target samples are the other ones.

B. Multiple Kernel Learning

As classifier, we decided to use Support Vector Machines. One advantage of SVMs is that the determination of the model parameters corresponds to a convex optimization problem: any local solution is also a global optimum.

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

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

With α_i the dual representation of the hyperplane's normal vector [11]. The function k is the kernel function resulting from the dot product in a transformed high-dimensional feature space.

In multi-kernel SVM, the kernel k can be any convex combination of semi-definite functions.

$$k(x_i, x) = \sum_{j=1}^{K} \beta_j k_j \text{ with } \beta_j \geq 0, \sum_{j=1}^{K} \beta_j = 1 \tag{2}$$

In our case, we have one kernel function per set of features (each resolution)

$$k = \sum_{j=1}^{N} \beta_j k_j(p_i^j, p^j) \tag{3}$$

Weights α_i and β_j are set to have an optimum hyperplane in the feature space induced by k. This hyperplane separates the two classes samples and maximizes the margin: the minimum distance of one sample to the hyperplane. This optimization problem has proven to be jointly-convex in α_i and β_j [12], therefore there is a unique global minimum that can be found efficiently.

$\beta_1 ... \beta_N$ represent the weights given to each resolution. Thus, using a learning database the system is able to find the best combination of these types of feature in order to maximize the margin.

This is an innovative way of using multi-kernel learning. Usually, multiple kernel learning is used to combine different kind of kernel functions such as Gaussian Radial Basis functions or Polynomial functions. This paper introduces a new framework for multi-resolution point detection: each linear kernel function k_i is dedicated to a specific resolution.

Among all pixels in the search region, we need to choose the one that corresponds to the desired facial landmark. In the perfect case, we should have $s > 0$ if the candidate pixel is close to the landmark, $s < 0$ 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 make a decision. This score given by the SVM classifier can be seen as a confidence index for a pixel to be the desired facial landmark.

2) Evaluation Step: During the test phase we extract the pyramidal-patches for each pixel. The region of interest (ROI) can be the whole face detected by the Viola-Jones detector. In order to reduce the computational time of patches extraction, we use two large regions of interest: one for the eyes and one for the mouth. The position and the size of these regions have been statistically extracted during the training step. Thus, they are large enough to take into account variation such as head rotations into account. Using the classifier, we test each candidate pixel in the search ROI. This leads to a SVM score for each candidate pixel. We obtain a SVM score map depicting, for each pixel of the ROI, the confidence index belonging to positive features (Fig. 1). We want that the best SVM score corresponds to the landmark position.

C. Bootstrap & Negative Patches

To have a robust detector, non-target samples have to be relevant. As explained in II-A , this detector is applied to large ROIs (one for the eyes and one for the mouth). Because random patches taken in these regions should not be representative, we use a bootstrap process in order to add relevant false alarms in the training process. The system is iteratively retrained with two updated training sets containing false alarms produced after facial points detection has been performed.

We split the training database into three different databases without mixing subjects: A, B and C. Eventually, A and B are used for training and C is used for cross-validation. The training-bootstrapping algorithm that we implemented proceeds as follows:

1) Train on A and evaluate on B.
2) Gather false alarms and add them to the B set of negative examples.
3) Proceed to Validation on C. If detection rate does not increase anymore, go to step 5. Else go to step 4.
4) Switch A and B database and go to step 1
5) Concatenate A and B set of positive and negative examples, and compute the final training.

This original bootstrap adds false alarms at each iteration without mixing any samples of A and B. This strategy is really important to avoid overfitting. False alarms detected on A are only added in A. These false alarms force the SVM, in the next iteration, to refine the optimum hyperplane between positive and negative examples. This procedure helps to find relevant patches. In fig. 4, we want to detect the inner corner of the left eye, but some wrong detections appear: these false

alarms are added to the negative examples set of the same database. By doing so, a very large number of redundant false alarms are eventually grabbed. This high number of redundant patches assists the detector to add some weights on significant false alarms.

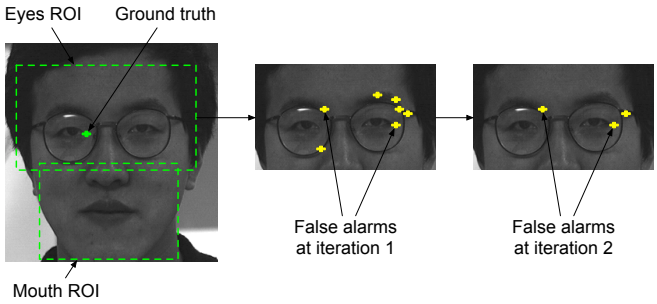

Fig. 4. Relevant false alarms added to the set of negative samples at each iteration.

D. Statistical Model & Validation

The detector learns a set of manually labelled points without spatial relationships between them. Consequently, in order to valid detection results, we have to restrict the configuration of detected points to a set of plausible shapes. We can split the point distribution model into several models: right eye, left eye and mouth. These models have to be flexible because of the possible variation of expressions, poses, morphology or identity. Therefore, we use a Gaussian Mixture for each model which can handles complex distributions [13]. The Gaussian Mixture Model can be written as a linear superposition of K Gaussian in the form :

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \tag{4}$$

where $N(x|\mu_k, \Sigma_k)$ is the probability density function of a gaussian with mean μ and covariance Σ. Such a mixture can approximate any distribution up to arbitrary accuracy, assuming that sufficient components are used. The hope is that a small number of components will give a good enough estimate. We use the EM algorithm to fit such a mixture to a set of data.

We need these models to represent expressive shapes. Thus, we use 3 gaussians for each model : 3 for the right eye, 3 for the left and 3 for the mouth. Then, we proceed to the statistical validation by computing the Mahalanobis distance d for each model M:

$$d_k^M = \min_k [(x - \mu_k)\Sigma_k^{-1}(x - \mu)] \tag{5}$$

With x the hypothesis to valid. We need d_k^M to be smaller than a threshold T_k^M selected such that 95% of the training shapes have a Mahalanobis distance to the gaussian model lower than T_k^M. During the evaluation step, for each facial landmark, we have SVM scores for their different possible localizations. Then, we want to choose the set of candidate pixels that leads to a shape validated by the model. We first test the shape having the best sum of SVM scores to see if its Mahalanobis distance d is lower than T. If outliers are detected by the SVM, d will not be lower than T, thus we try the next best combination of SVM (using a combinatorial optimization) scores and so on.

III. Experiments

A. Training Data Sets

The proposed facial landmark detection method was trained on 2 different databases:

The Cohn-Kanade database [14] which is a representative, comprehensive and robust test-bed for comparative studies of facial expression. It contains image sequences with lighting conditions and context are relatively uniform. The database contains 486 sequences starting with the neutral expression and ending with the expression apex. For our training, we used the first frame (neutral expression) and the last frame (expression apex) of 209 samples.

The CMU Pose, Illumination and Expression Database (PIE) [15] which consists of over 40,000 facial images of 68 people. Each person were imaged across 13 different poses, under 43 different illumination conditions, and with 4 different expressions. For our training, we used 108 samples randomly taken.

We train our detector on 317 faces (extracted using Viola-Jones face detector) resized to 100x100 pixels. The inter-ocular distance varies between 30 and 40 pixels. At the first bootstrap iteration, we have 9 target samples and 16 non-target samples of 4 different resolutions: the first patch is 9x9 pixel, the second 17x17, the third 25x25 and the last 37x37, all resized to 9x9 pixels. Our bootstrap process is applied for each points. Thus, the number of false alarms we add varies, but we can approximate this augmentation by an increase of 30% of negative patches, with only relevant examples.

B. Multi-Kernel Learning evaluation

We use the SimpleMKL [16] algorithm to train multi-kernels SVMs. Fig. 5 shows the impact of the multi-resolution patches on facial point detection. Decision maps obtained with the kernels $k_1 ... k_4$ show that each kernel k_i emphasize different level of information. Kernel k_1 deals with this problem using a local point of view, whereas kernel k_4 uses global information. The final decision map, given by the SVM ouput, uses these different levels of information to give a confidence index to each candidate pixel of the search region.

As we have one SVM for each facial landmark, we find one set of weights $\beta_1, \beta_2, \beta_3, \beta_4$ for each facial landmark. Mean weights learned for the points belonging to the same facial feature are reported in Table II.

We notice that the lower resolution patches (corresponding to k_4) always have the biggest β. This means that a more important weight is associated to global information. These patches help to find the coarse location of the searched point.

Point	C	m	e	Point	C	m	e
Outer corner of the right eye	99.1%	4.0%	2.5%	Bottom of the left eye	100%	4.5%	2.4%
Top of the right eye	99.2%	3.9%	2.5%	Inner corner of the left eyebrow	84.5%	6.6%	6.5%
Inner corner of the right eye	99.3%	4.2%	2.8%	Outer corner of the left eyebrow	91.1%	5.0%	3.5%
Bottom of the right eye	99.5%	3.4%	2.4%	Nose	98.5%	4.0%	5.8%
Outer corner of the right eyebrow	89.4%	6.0%	5.1%	Right mouth corner	96.2%	3.8%	4.7%
Inner corner of the right eyebrow	84.1%	7.1%	6.1%	Mouth top	94.3%	4.9%	6.6%
Inner corner of the left eye	99.6%	5.0%	2.4%	Left mouth corner	95.5%	5.3%	4.5%
Top of the left eye	100%	3.1%	2.5%	Mouth bottom	95.0%	5.2%	4.2%
Out corner of the left eye	100%	3.1%	2.4%	-	-	-	-

TABLE I

RESULTS FOR TEST ON 266 COHN-KANADE EXPRESSIVE IMAGES NEVER SEEN IN TRAINING. C REPRESENTS THE CUMULATIVE ERROR DISTRIBUTION OF POINT TO POINT AT 10% ($m_e < 0.1$). THE MEAN m AND THE STANDARD DEVIATION e OF THE ERROR ARE MEASURED IN PERCENTAGES OF d_{IOD}

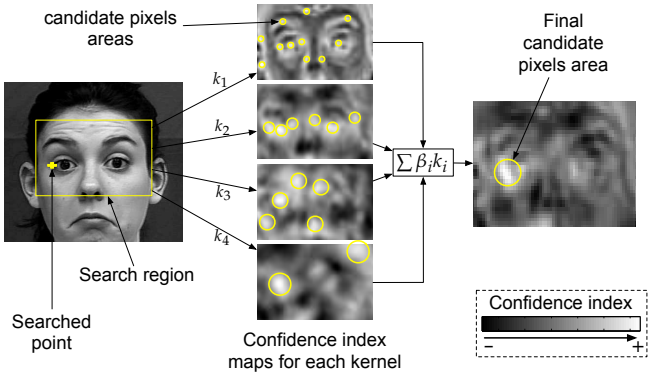

Fig. 5. Decision map of each kernel for the outer corner of the right eye detection.

Facial landmarks	β_1	β_2	β_3	β_4
Brows (6 points)	0.1863	0.2774	0.1823	0.3541
Eyes (8 points)	0.2140	0.2229	0.1654	0.3977
Mouth (4 points)	0.1496	0.2639	0.2380	0.3484

TABLE II

MEAN WEIGHTS ASSOCIATED TO EACH KERNEL.

Then, the other patches using local information refine this location.

C. Distance Error Measure

The criteria for success is the distance of the points computed using our detector compared to manually labelled ground truth. The detection error of a point i is defined as the Euclidian distance d_i between the point and its manually labelled ground truth. The average is given as

$$m_e = \frac{1}{n d_{IOD}} \sum_{i=1}^{n} d_i \qquad (6)$$

With d_{IOD} the Inter-Ocular Distance and n the total number of images. The detection is defined as success if $m_e < 0.10$ (10% of d_{IOD}).

D. Bootstrap Evaluation

To evaluate the bootstrap process impact on the overall system performance, we proceed to two trainings: in the first, non-target samples are collected with the full bootstrap strategy. In the second training, bootstrapped samples are replaced by patches randomly chosen in the whole ROI. We also evaluate the system performance with only one bootstrap iteration. As test database, we use Cohn-Kanade test set (with only new subjects). The accuracy of each training is reported in table III.

Method	Accuracy at 10% ($m_e < 0.1$)
MKL SVM + Random patches	88%
MKL SVM + Bootstrap (1 it)	92%
MKL SVM + Full bootstrap (6 it)	**97%**

TABLE III

EVALUATION OF THE BOOTSTRAP PROCESS

Training with random patches and training with the full bootstrap process have the same number of non-target samples. Thus, this study shows how much this bootstrap process increases results on detection.

E. Experimental Results & Comparaison

Several databases have been used to test this detector. We first evaluated it on the same Cohn-Kanade database as in III-D. The results of this study are shown in Table I depicting the classification rate for all points. As we can see, all points are detected with a high accuracy even if the database includes expressive images. Eyebrows points have low detections results because of the difficulty to locate these points manually

Two others database are also used to evaluate the generalization of this method: AR Face Database [17] which contains frontal view faces with different facial expressions, illumination conditions, and occlusions. We also evaluate this method on BioID database [18] which contains 1521 frontal face images that vary with respect to illumination, background, face size and slight head pose variation. Based on results from literature [3], [4], [5], [9], this database is considered more difficult. Thus, test on BioID constitutes a benchmark comparison with the existing state of the art.

Fig. 7 shows the cumulative error distribution measured on these two databases. We can see that our detector is really accurate on unseen faces from other databases. On the AR

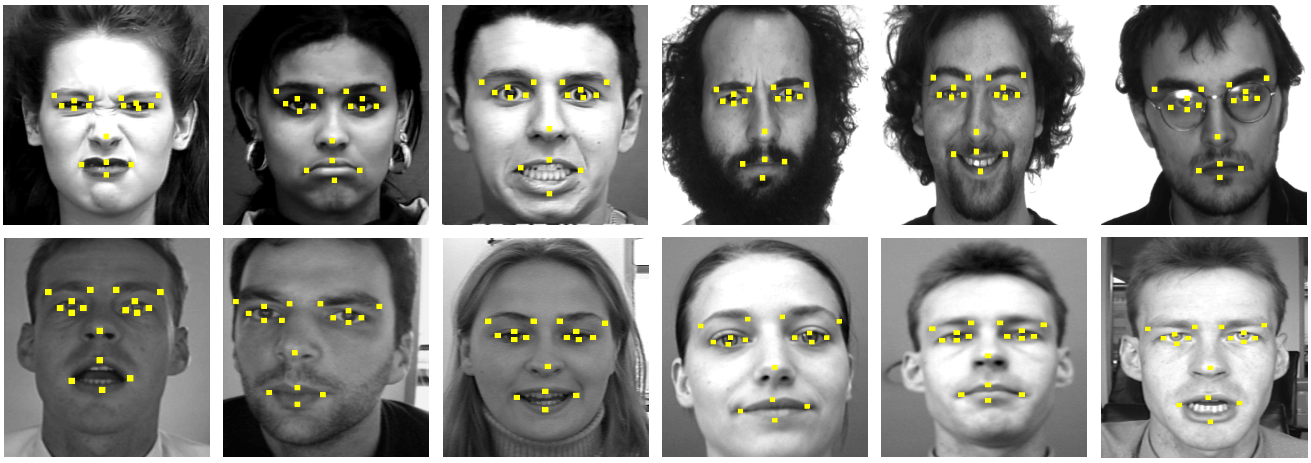

Fig. 6. Some typical results on the Cohn Kanade, AR Face and BioID databases.

Face database, which contains faces with facial expressions, we can see that 80% of the images have an average point error less than 5% of $dIOD$ which roughly correspond to 2 pixels per point.

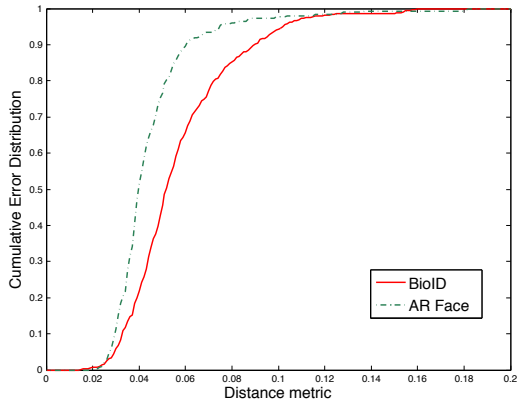

Fig. 7. Cumulative error distribution measured on BioID and AR Face database.

Then, to compare our facial point detector with those of the state of the art [2], [9], [4], [3] we have to use the BioID database. All of these detector are publicly available and they all have been tested on BioID database. Table IV shows the cumulative error distribution of the $me17$ error measure. This measure represents the mean over all internal points (all points that lie on facial landmarks instead of the edge of the face). It shows that our methods can be compared favorably with state of the art experimental results.

Method	Accuracy at 10% ($m_e < 0.1$)
AAM [2]	85%
BoRMaN [9]	95%
CLM [4]	90%
STASM [3]	95%
Our system	**95%**

TABLE IV

COMPARISON OF THE CUMULATIVE ERROR ON BIODID DATABASE.

IV. CONCLUSION

In this papers we present a robust and accurate method for fully automatic detection of facial landmarks in expressive images.

The system aims at exploiting different resolutions of the face image. Low resolution patches permit to catch the whole face structure, resulting in robust but inaccurate detection. High resolution patches extract information on a small area of the face and lead to several possible accurate detections, in which one is the desired landmark. The multi-kernel learning provides a gentle way to combine these different levels of information. The system is trained using an original bootstrap process. Evaluations have proven that false alarms added during this operation are more relevant than patches randomly chosen. Finally, we combine SVM point detections with a statistical validation step correcting outlier detections and providing more robust results.

The whole system, trained on Cohn-Kanade and PIE databases, is robust to varying lighting conditions, facial expressions and occlusions of the face caused by glasses or hair. Cross-bases validation tests on the AR Face and BioID databases show that our detector can be compared favorably with the start of the art.

The accuracy of our detector allows the understanding of subtle changes in human face. As our current works are based on emotion recognition, we plan to utilize this detector as a first extraction step of emotional relevant features

V. ACKNOWLEDGMENTS

This work has been partially supported by the French National Agency (ANR) in the frame of its Technological Research CONTINT program. (IMMEMO, project number ANR-09-CORD-012)

REFERENCES

[1] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *IEEE Conf. Comp. Vision and Pattern Recognition (CVPR'95)*, vol. 61, no. 1, p. 38, 1995.

[2] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Proc. IEEE European Conference on Computer Vision (ECCV '98)*, p. 484, 1998.

[3] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR '08)*, p. 504, 2008.

[4] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008.

[5] T. Senechal, L. Prevost, and S. Hanif, "Neural Network Cascade for Facial Feature Localization," *Fourth Int'l Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR '10)*, p. 141, 2010.

[6] S. Duffner and C. Garcia, "A Connexionist Approach for Robust and Precise Facial Feature Detection in Complex Scenes," *Image and Signal Processing and Analysis*, 2005.

[7] D. Vukadinovic and M. Pantic, "Fully automatic facial feature point detection using Gabor feature based boosted classifiers," *Proc. IEEE Conf. Systems, Man and Cybernetics (SMC'05)*, vol. 2, p. 1692, 2005.

[8] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[9] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial Point Detection using Boosted Regression and Graph Models," *IEEE Conf. Comp. Vision and Pattern Recognition (CVPR'10)*, 2010.

[10] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, p. 137, 2002.

[11] B. Scholkopf and A. Smola, *Learning with kernels*. Cambridge, MIT Press, 2002.

[12] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan, "Learning the kernel matrix with semidefinite programming," *The Journal of Machine Learning Research*, vol. 5, p. 27, 2004.

[13] T. Cootes and C. Taylor, "A mixture model for representing shape variation," *Image and Vision Computing*, vol. 17, no. 8, pp. 567–573, 1999.

[14] T. Kanade, Y. Tian, and J. Cohn, "Comprehensive database for facial expression analysis," *Proc. IEEE Conf. Face and Gesture Recognition (FG'00)*, p. 46, 2000.

[15] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, p. 1615, 2003.

[16] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, p. 2491, 2008.

[17] A. Martinez and R. Benavente, "The AR face database," tech. rep., CVC Technical report, 1998.

[18] O. Jesorsky, K. Kirchberg, and R. Frischholz, "Robust face detection using the hausdorff distance," in *Audio-and Video-Based Biometric Person Authentication*, p. 90, 2001.