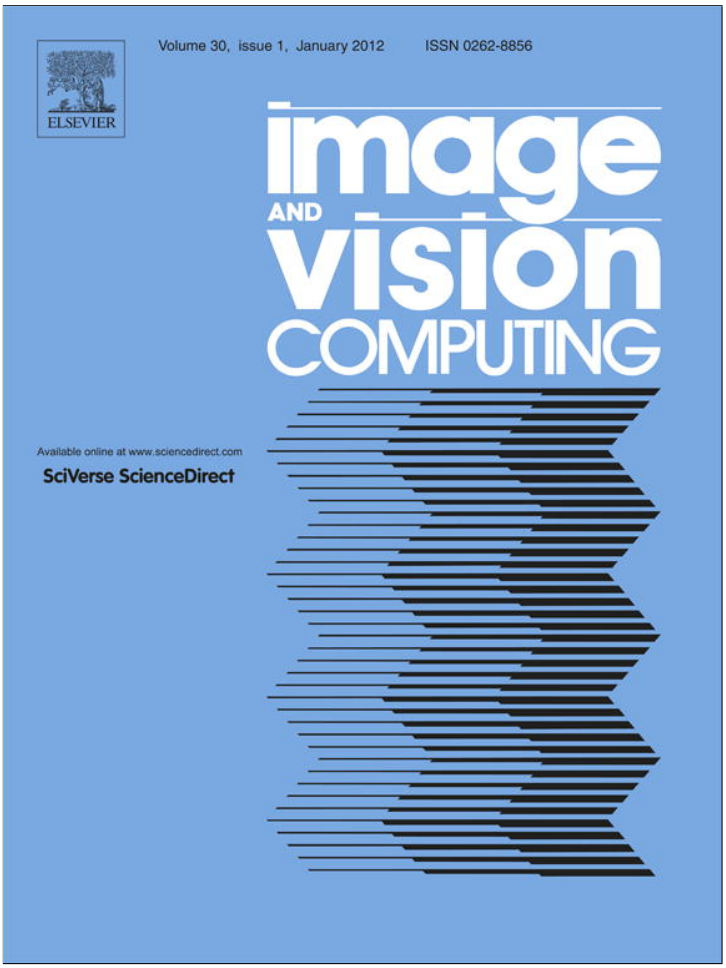

Image and Vision Computing 31 (2013) 542–554

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Multi-Kernel Appearance Model ☆

Vincent Rapp [a,*], Kevin Bailly [a], Thibaud Senechal [b], Lionel Prevost [c]

[a] UPMC — Univ. Pierre & Marie Curie, CNRS UMR 7222, ISIR, F-75005, Paris, France
[b] Affectiva Inc., Waltham, MA, USA
[c] UAG — Univ. of French West Indies & Guiana, EA 4540, LAMIA, Guadeloupe, France

ARTICLE INFO

Article history:
Received 23 May 2012
Received in revised form 2 April 2013
Accepted 25 April 2013
Available online 11 May 2013

Keywords:
Facial feature localization
Multiple-kernel learning
Two-stage classifiers
SIFT descriptor
Deformable model alignment
Gauss–Newton optimization

ABSTRACT

Automatic facial landmarking is a crucial prerequisite of many applications dedicated to face analysis. In this paper we describe a two-step method. In a first step, each landmark position in the image is predicted independently. To achieve fast and accurate localizations, we implement detectors based on a two-stage classifier and we use multiple kernel learning algorithms to combine multi-scale features. In a second step, to increase the robustness of the system, we introduce spatial constraints between landmarks. To this end, parameters of a deformable shape model are optimized using the first step outputs through a Gauss–Newton algorithm. Extensive experiments have been carried out on different databases (PIE, LFPW, Cohn-Kanade, Face Pix and BioID), assessing the accuracy and the robustness of the proposed approach. They show that the proposed algorithm is not significantly affected by small rotations, facial expressions or natural occlusions and can be favorably compared with the current state of the art landmarking systems.

1. Introduction

Automatic salient point localization is a fundamental building block in many computer vision-based applications. In facial analysis, these salient points (e.g., eye and mouth corners, nose and chin tips) are called fiducial points or landmarks. Their appearance and spatial arrangement are important for most facial analysis tasks, including face recognition, expression recognition or face animation. In human spontaneous behavior analysis, facial landmark configurations are indicative of deformations caused by expressions. Whatever the application, automatic facial landmarking is the second step (after face detection) of a larger face processing task.

Many methods have been introduced over the last decade to solve this problem. Locating facial landmarks remains challenging for applications that must operate under a wide range of conditions, including illumination variations, occlusions, poses or expressions. In general, related methods use two types of information, both essential. One is facial landmark appearance, also called texture information. The other is the spatial relationship among different facial landmarks, also called shape information. By looking at the latter and the degree of spatial constraint it introduces during the search, we can consider three main streams.

First, we consider methods that independently detect each facial feature point using any shape information. [50] detects 20 facial points using the GentleBoost classifier trained on Gabor magnitude images. The lack of constraint in such frameworks may often lead to senseless localizations and badly damage the system robustness.

The second category proceeds to a joint detection of all points. Here, the spatial relation between points is not directly formulated: the classifier implicitly learns the constraints [12,20,40]. These methods suffer from overly strong relations between points and 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.

Finally, we consider methods that explicitly formulate constraints between points using different processes. Some methods independently detect each point and apply spatial restrictions on their locations using graph-based methods [15,46,27,54]. In a recent work, [37] first detects each landmark using multi-scale gray-level features and multi-kernel Support Vector Machines. A time-consuming combinatory process is tasked with finding the optimal point constellation according to its likelihood. This category also includes fitting-based methods, usually called Parameterized Appearance Models (PAMs), widely used in face analysis. They align a previously learned model within a new face image. The main component of these approaches is the statistical model. It is characterized by parameters that describe the possible model configurations in position, pose, shape and texture. During the fitting process, PAMs try to find correspondences between the model and an image containing the face by minimizing a cost function w.r.t these parameters. Among these approaches, we find Active Appearance Models (AAM) and Active Shape Models (ASM), introduced by [4,7] and more recently the

☆ This paper has been recommended for acceptance by Stefanos Zafeiriou.

* Corresponding author. Tel.: +33 622432752.
E-mail addresses: vincent.rapp@isir.upmc.fr (V. Rapp), kevin.bailly@isir.upmc.fr (K. Bailly), thibaud.senechal@affectiva.com (T. Senechal), lionel.prevost@univ-ag.fr (L. Prevost).

0262-8856/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.imavis.2013.04.006

Constrained Local Models (CLM) introduced by [9]. For computational time issues, these methods are generally based on simple features, derived from gray levels, and use some simple similarity measures during the fitting process. This strategy can lead to ambiguities: several image areas may be similar to the searched point, making the optimization function non-convex. In consequence, these systems may be prone to local minima and strongly dependent on the initialization step.

In this paper, we propose the Multi-Kernel Appearance Model (MuKAM), a novel and efficient approach that tries to overcome PAM weaknesses without damaging the computational time. MuKAM employs a two-stage classifier designed to achieve fast and efficient detections. The first stage is based on low-level features and a fast linear kernel. It aims to select a subset of candidates without excluding the right location. The second stage employs higher level features and a non-linear kernel to estimate the candidate likelihoods. These two-stage present two main advantages: during the training step, the bootstrap process provides relevant samples to train the second stage, and the detector can quickly estimate the likelihood of each candidate pixel during the evaluation step. Moreover, we improve system robustness by introducing constraints between points. To introduce these constraints, we propose an alignment process step relying on a deformable model fitting: according to the sparse response maps obtained at the end of the second stage, we want to find the set of parameters that best fit the model on the face. This parameter optimization is performed using an iterative Gauss Newton process applied to a cost function especially designed for our problem. As the optimization employs sparse response maps, the optimal solution is found after a few iterations.

The proposed approach is based on three main contributions: (i) an accurate point detector based on multiple scale features and multiple kernel learning algorithms, (ii) a fast and well suited two-stage classifier involving increasingly complex kernels and (iii) a robust system relying on a deformable model fitting strategy.

The remainder of this paper is organized as follows. In Section 2, we briefly review the literature. In Section 3, we provide an overview of the proposed method. In Section 4, we describe the detector and detail the two-stage training process. In Section 5, we present the objective function and optimization process. In Section 6, we present an extensive evaluation of different key aspects of our method. Finally, Section 7 presents our closing remarks.

2. Related work

Our approach relies on two main aspects: detecting each facial landmark and the modeling spatial restrictions. Both have resulted in many systems. In this section, we compare relevant systems to our own work.

2.1. Landmark detection

Localizing facial features can be addressed following different approaches. One can state this problem as a regression or classification task. The first category of approaches seeks to answer the question: "Where is the searched point?" The problem is established as a regression task and can be treated in different ways. One can try to measure the resemblance between each pixel in the region of interest and the point we are seeking. [52] proposes a tree-based regression algorithm over patches characterized by haar-like features. Other methods prefer to estimate the displacement vector $\mathbf{v}$ that continuously relates a patch location to the target point. [46] decomposes this problem into two separate regression problems using Support Vector Regressors (SVR). One regressor is tasked with finding the angle of $\mathbf{v}$, and another is to predict the vector length. For [5], the displacement vector is learned using Random Forest regressor. [10] proposes using a GentleBoost algorithm to predict the horizontal and vertical displacement towards the good location. The work proposed by [14] uses an iterative regression-based approach using a sequence of regressors entirely defined during a training step. These approaches are based on the assumption that the desired point is visible, and they can get stuck if the sought point is missing or occluded.

Finding facial landmark localization can also be seen as a classification problem: "Which locations correspond to the searched point?" This process is performed by classifying each pixels in a portion of the image as a target or non-target. At each pixel in this area, features are extracted and the classifier assigns a label and, potentially, a confidence index associated with this label. Several methods use the GentleBoost classifier [50,25,13] which sequentially chooses weak classifiers and combines them to minimize an error function. A popular and successful classifier is the Support Vector Machine (SVM), widely used for facial landmark detection [32,51,38].

These approaches suffer from the local appearance representation, and the features are generally based on small local support, resulting in ambiguous detection. In previous work [37], we propose solving this problem with multi-scale features, encoding information at different levels. This system uses gray levels as features, but other ways to represent the appearance are available. Classifiers can take the response of a filter bank as input, including as Gaussian Derivative filters [17], Gabor filters [50] or Haar-like features [9,1].

Another issue is the time required to detect the points: more complex classifiers and descriptors make the system more expensive in terms of time processing. Several approaches try to overcome this problem using a cascade process, first introduced by [49]. These methods successively combine increasingly complex classifiers in a cascade structure to dramatically increase detector speed by focusing on promising image regions. The bootstrap process gives the system the ability to classify difficult examples (i.e., those for which the classification was wrong in the previous stage). These examples are then used to train the next stage. The system thus only focuses on relevant examples, adding robustness to the detections. In the Viola–Jones face detector, the authors apply a cascade of boosted classifiers on image patches to classify them as face or non-face. [8] proposes using the same cascade as [49] to detect 4 facial landmarks with Haar wavelets on local patches. [47] also successfully applied this architecture in object detection, proposing a three-stage classifier that combines linear and non-linear kernel SVMs. The authors show that increasing kernel non-linearity increases the discriminative power.

2.2. Spatial restrictions

In face landmarking, most systems generally suffer from the high incidence of false detections because the appearance may be similar under some imaging conditions. To better handle these variations, constraints between points are usually introduced. Some methods directly model the positions of each facial point based on the others [43]. Other methods model these relations using graph-based methods [46,15,27,54]. The spatial restriction can also be included in a transparent way by always allowing the point configuration to be a linear combination of the learned shape models [3].

In PAM approaches, this consistency is added with a statistical Point Distribution Model (PDM). These methods rely on a model characterized by the parameter vector, which encodes facial shape variations. This model is typically built from training shape samples first aligned into a normalized co-ordinate frame using Procrustes Analysis [16]. The remaining variations (with respect to the mean shape) are modeled as a Gaussian distribution using Principal Component Analysis (PCA). By retaining only the most significant variation modes (i.e., the largest eigenvalues of the covariance matrix), the resulting PDM approximates the initial shape $\mathbf{s}$ by $\bar{\mathbf{s}} + \Phi\mathbf{p}$, where $\bar{\mathbf{s}}$ is the mean shape, Φ is the retained eigenvector [6] and $\mathbf{p}$ is the model parameter. After estimating the variation modes, we must determine the best fit of this model to a face. This step requires a cost function that uses a fitting algorithm to estimate how well the model parametrized with $\mathbf{p}$ fits for an image containing a

face. We want the global minimum of this function to match to the best model alignment, parametrized by the optimal parameters.

Numerous approaches are based on this non-rigid model. Active Shape Models (ASMs) [7] build this model from shape variation. For the fitting step, an ASM iteratively looks along profiles normal to the model boundary through each model point for its most likely location. The model parameters are then updated to best match to these new positions. Active Appearance Models (AAMs), introduced by [4], model both shape and texture and update the model parameters by minimizing a reconstruction error characterized by the residual between a model instance and the face image. This minimization may be performed using different strategies. One can use an analytic approach by minimizing the objective function [28,45]. Other approaches solve this problem by learning a regression function directly from a set of known displacement and their corresponding residuals [44]. Despite their ability to handle facial deformations, these approaches are prone to local minima because they can mismanage illumination or occlusion.

To overcome this problem, [9] has introduced the Constrained Local Models (CLM), combining the advantages of feature detection based approaches, the model flexibility of PAM and the consistency of a full shape model. CLM are relatively close to ASM since they both employ local searches around each PDM landmark. However, CLM proceed to a 2D local search around each estimate capturing more information [30]. In CLM, the fitting step is achieved by applying an ensemble of patch experts, previously trained and applied on local region around each point of the model. CLM fitting is then posed as estimating the PDM parameters $\mathbf{p}^*$ that jointly minimize the misalignment error regularized by a term penalizing complex deformations.

In the original CLM, Cristinacce and Cootes propose an iterative process during which a PCA models the appearance of each local region around a landmark. The shape-constrained local model search is then performed by optimizing a function based on the local responses of the statistical appearance model. In this method, the patch experts adaptively change during the fitting process until convergence. Joint optimization relies on the computationally expensive Nelder–Mead simplex. The convergence of this system may be really slow, especially for a complex PDM parameters. Rather than using patch experts that adaptively change during the fitting process, other CLMs estimate the local responses with static classifiers such as SVM which assigns a score for each pixel in a region of interest [38,51]. The fitting process is then performed over these response maps. Due to their small local support, ambiguities plague these responses, resulting in several local minima on the global cost function. Several methods solve this issue by decreasing the impact of these ambiguities using different techniques. [51] suggests enforcing convexity by fitting a convex quadratic curve to the local responses. [38] smooths these local responses by applying mean-shift algorithm over all landmarks simultaneously, while [19] uses a Gaussian Mixture Model (GMM) with an expectation–maximization (EM) algorithm to find the maximum likelihood solution.

3. MuKAM overview

The approach explored here is based on two main ideas. We introduce efficient multi-scales landmark detectors, which combine multi-kernel classifiers into a two-stage framework. To introduce explicit consistency between points, we then estimate the parameters of a shape model by optimizing a cost function based on the likelihood of the facial landmark detectors. This constitutes the two steps of our framework as presented in Fig. 1: the independent facial feature localization and the model alignment.

During the first step, it is really difficult to make a firm and final decision on the position of each landmark because of the possible occlusion of the point or ambiguous detections. A more flexible and robust approach is to (i) select a set of likely locations among all the pixels in a region of interest (ROI) defined in the face image and (ii) estimate the candidate likelihood. To achieve robust detections and handle the computational time issues, we implement a two-stage classifier. The first stage $\mathcal{S}_1$ gives a quick local response for each pixel in a region of interest (Section 4.3.1). On the dense resulting response maps, we extract the most promising candidates, which correspond to the local maxima of these responses. The second stage $\mathcal{S}_2$ uses a more powerful classifier to estimate the candidate likelihoods. This process is applied for each landmark, resulting in a set of sparse response maps.

In the second step, we proceed to the deformable model alignment using the sparse maps (Section 5). The problem is posed as searching for the PDM parameters that jointly minimize a cost function involving two types of information: a data term measuring the misalignment error between the shape and detections and a regularization term penalizing non-plausible shapes. This cost function is iteratively minimized during a Gauss–Newton optimization process.

4. Independent facial feature localization

The following section explains the independent facial feature localization. First, we investigate different ways to represent the appearance of each candidate pixel to classify. We then explain the multiple-kernel classifier used. Finally, the two-stage implementation is presented: for each stage, the configuration features/kernels and bootstrap process are detailed.

4.1. Multiple-scale features

For each landmark, our framework is based on classifying pixels in a region of interest. The pixel appearance can be described using numerous features, all presenting different pros or cons. Among these features, we distinguish two groups: spatial- and histogram-based features. Regardless of the feature, the appearance is characterized using different scales (Fig. 2).

4.1.1. Gray level

The simplest and most intuitive way to describe the appearance of pixels is to only consider the gray level intensities. Our previous work [37] has shown that despite its simplicity, gray level intensities used at different scales lead to reliable results. We extract high-resolution patches, which encode the local texture and highlight small facial details, including the canthus or pupil location. Despite their precision, these patches are local and do not carry the spatial information on their locations in the facial image. To enrich this information, we progressively crop larger patches, encoding global information and giving, in different scales, the spatial information we need. As we require this stage to be as fast as possible, we resize all patches to a given size. This dimension reduction presents the advantage of decreasing the processing time while smoothing local details.

4.1.2. Scale-invariant Feature Transform descriptor (SIFT)

[26] first introduced SIFT. The SIFT descriptor aims to characterize the local appearance at a particular location. Given a landmark, this descriptor computes the gradient vector for each pixel included in a region around the landmark and builds a normalized histogram of gradient directions. SIFT descriptors typically use a set of 16 histograms, aligned in a 4×4 grid; each histogram has 8 orientation bins, one for each of the main compass directions. This results in a feature vector containing 128 elements. To include the multi-scale information, we compute the SIFT descriptor on 4 different scales calculated according to the interocular distance.

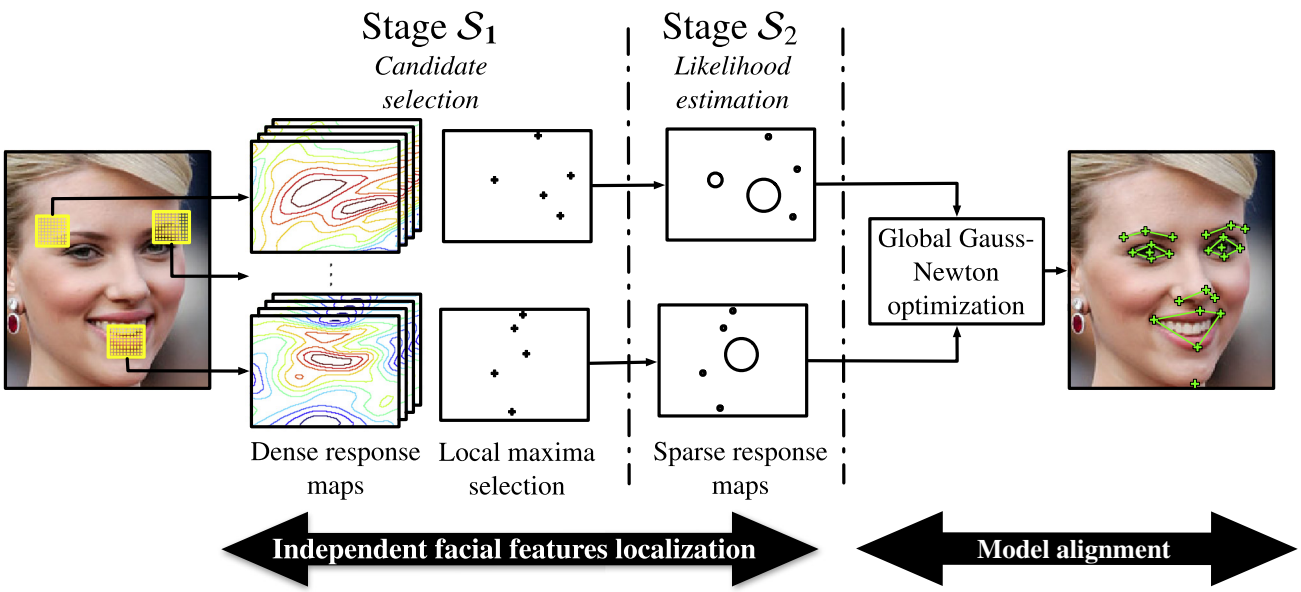

Fig. 1. System overview. $\mathcal{S}_1$ proceeds to the candidate selection by extracting local maxima on the dense response maps (Section 4.3.1). $\mathcal{S}_2$ re-classifies these candidates by using a much more powerful classifier. The radius of these sparse responses depends on the candidate likelihood $p(l_i = 1|C_2(\mathbf{x}^\mathbf{p}))$ (Section 4.3.2). Finally, the alignment process aligns the model with the Gauss–Newton algorithm by using these maps and a deformable model (Section 5).

4.1.3. Gabor magnitude features

The Gabor magnitude pictures are obtained by convolving facial images with Gabor filters:

$$G_{\mathbf{k}}(\mathbf{z}) = \frac{\mathbf{k}^2}{\sigma^2} e^{\left(-\frac{k^2 z^2}{2\sigma^2}\right)} \left(e^{i\mathbf{k}\mathbf{z}} - e^{-\frac{\sigma^2}{2}}\right) \quad (1)$$

where $\mathbf{k} = k_v e^{i\phi_u}$ is the characteristic wave vector. We use three spatial frequencies $k_v = \left(\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}\right)$ and six orientations $\phi_u = \left(\frac{k\pi}{6}, k \in \{0 \ldots 5\}\right)$ for 18 total Gabor filters. As the phase taken from image point only a few pixels apart has a very different value [31], only the magnitude is generally kept, resulting in 18 Gabor magnitude pictures. The multi-scale information is here carried through the spatial frequencies used: a low frequency only extracts local information around the pixel, whereas a high frequency captures more global information.

4.1.4. Local Gabor Binary Pattern (LGBP)

LGBP is obtained by associating the Local Binary Pattern operator (LBP) and the Gabor magnitude images. [33] first introduced the LBP. It codes each pixel in an image by thresholding its 3×3 neighborhood according to its value and considering the result as a binary number. The LBP of a pixel $\mathbf{p}$ (value $f_\mathbf{p}$) with a neighborhood $\{f_k, k = 0 \ldots 7\}$ is given by:

$$LBP(\mathbf{p_c}) = \sum_{k=0}^{7} \delta\left(f_k - f_p\right) 2^k \quad (2)$$

where

$$\delta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}. \quad (3)$$

We apply the basic LBP operator on the 18 Gabor magnitude pictures, resulting in 18 LGBP-maps per facial image. Combining the Local Binary Pattern operator with Gabor wavelets exploits multi-resolution and multi-orientation links between pixels. This has been proven to be robust to illumination changes [53]. As for Gabor magnitude features, the multi-scale information is carried through each frequency.

4.2. Multiple-Kernel classifier

Each stage of our classifier uses Multiple-Kernel Support Vector Machines (MK-SVM). The SVM classifier defines a discriminant function $C(\mathbf{x}^p)$ to classify candidate pixels p. Here, $\mathrm{x}^p = \{x_s^p\}_{s=1}^S$ represents the collection of features associated with a candidate pixel for S scales s. The discriminant function $C(\mathbf{x}^p)$ of the SVM is expressed as:

$$C\left(\mathbf{x}^p\right) = \sum_{i=1}^{M} y_i \alpha_i K\left(\mathbf{x}^p, , \mathbf{x}^i\right) \quad (4)$$

where $\mathbf{x}^i$, for $i = 1, \ldots, M$, denoting the descriptors of m training samples selected as representative by the SVM and associated with labels $y_i \in \{-1, 1\}$ (target or non-target). Coefficient α_i is the dual representation of the hyperplane's normal vector [39], and function K is the kernel function resulting from the dot product in a transformed high-dimensional feature space.

Several types of kernels can be used, whether designed for histogram-based features (typically a histogram intersection kernel or χ^2 kernel) or spatial-based features (i.e., RBF kernel).

- Polynomial kernel:

$$K\left(\mathbf{x}^p, , \mathbf{x}^\mathbf{i}\right) = \left(1 + \left\langle \mathbf{x}^i, , \mathbf{x}^p \right\rangle\right)^n \quad (5)$$

- Histogram intersection kernel:

$$K\left(\mathbf{x}^p, , \mathbf{x}^i\right) = \sum_j \min\left(\mathbf{x}_j^p . \mathbf{x}_j^\mathbf{i}\right) \quad (6)$$

- RBF kernel:

$$K\left(\mathbf{x}^p, , \mathbf{x}^i\right) = exp\left(\frac{-||\mathbf{x}^p . \mathbf{x}^i||}{2\sigma^2}\right). \quad (7)$$

More complex kernel functions have also been investigated, including the generalized radial-basis function kernels (RBF) [48] that extend RBF kernels to use a not necessarily Euclidean metric. A typical example is the exponential-χ^2 kernel, which is designed to compare probability distributions:

$$K\left(\mathbf{x}^p,,\mathbf{x}^i\right) = exp\left(\frac{\chi^2\left(\mathbf{x}^p,,\mathbf{x}^i\right)}{2\sigma^2}\right) \tag{8}$$

with

$$\chi^2\left(\mathbf{x}^p,,\mathbf{x}^i\right) = \frac{1}{2}\sum_{i=1}^{M}\frac{\left(\mathbf{x}^p-\mathbf{x}^i\right)^2}{\mathbf{x}^p+\mathbf{x}^i}\mathrm{e} \tag{9}$$

This expensive kernel combines the benefits of the homogenous additive kernels (designed to compare histograms-based features) and the RBF kernels [42].

In multi-kernel learning, kernel K can be any convex combination of semi-definite functions:

$$K\left(\mathbf{x}^p,,\mathbf{x}^\mathbf{i}\right) = \sum_s \beta_s k\left(x_s^p, x_s^i\right) \text{with} \beta_j \geq 0, \sum_s \beta_s = 1. \tag{10}$$

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 class samples and maximizes the margin. This optimization problem has proven jointly-convex in α_i and β_j [23]. A unique global minimum can therefore efficiently be found.

β_s represents the weights given to each resolution r. Using a training database, the system can thus find the best linear combination of these scales.

4.3. Two-stages classifier

In the classification task, there is a permanent trade-off between computational time and discriminative power. We try here to override this trade-off using a two-stage classifier. The first stage ($\mathcal{S}_1$) quickly gives a conservative selection of candidates from large number of pixels: a false negative rate close to zero, and a false positive rate as low as possible. To this end, $\mathcal{S}_1$ uses fast feature extractor (gray level patches) and a low complexity kernel (linear kernel). We use $\mathcal{S}_1$ outputs to drastically reduce the number of candidate pixels. The second stage ($\mathcal{S}_2$) can thus be based on more complex features (e.g., Gabor filters or SIFT descriptors) and more complex kernels (e.g., polynomial or Gaussian).

4.3.1. First stage classifier (S_1)

As explained above, the feature extraction process in this stage must be as fast as possible. We thus directly choose to use gray-level intensities and a classic linear SVM with the classification function C_1:

$$C_1\left(\mathbf{x}^p\right) = \sum_{i=1}^{M} y_i\alpha_i\left\langle\mathbf{x}^i,,\mathbf{x}^p\right\rangle. \tag{11}$$

The complexity of this classification problem is $O(NMD)$: applying the classification function $C(\mathbf{x}^p)$ to each pixels in a ROI requires a number of operations proportional to the number N of candidate pixels in the ROI, M is the number of support vectors chosen during the SVM training and D is dimension of our descriptors. One objective is to reduce this complexity using the primal SVM formulation.

This linear kernel allows us to precompute $\mathbf{w}$, the sum over support vectors and rewrite Eq. (11):

$$C_1\left(\mathbf{x}^p\right) = \left\langle\mathbf{w},\mathbf{x}^p\right\rangle \text{with}\, \mathbf{w} = \sum_{i=1}^{M} y_i\alpha_i\mathbf{x}^i. \tag{12}$$

The classification function is now independent of M (the number of support vectors), which gives an evaluation cost in $O(ND)$. Moreover, an advantage of using the primal formulation is that the response map can be computed using efficient convolution operations on each pixel p contained in region R.

At this stage, we have a coarse decision for each pixel we want to classify. Theoretically, an ideal first stage in this framework should reject all pixels that absolutely do not correspond to the sought point, while keeping all pixels likely to be the correct location. We must therefore find an optimal way to reject as few true detections as possible (high recall) while rejecting as many false detections as possible (high precision). Starting from this requirement, we decided to select as potential positions those detections corresponding to the local maxima of the dense response maps, directly built using the score given by the classification function. For each map score, we take its 3×3 neighborhood: if all pixels have a lower value than the central response, the latter is considered a local maximum. All local maxima detected then constitute the set of final candidate pixels.

4.3.2. Second stage classifier ($\mathcal{S}_2$)

The previous stage drastically reduces the number of potential locations for the sought point. Due to its conception, this stage can induce many false positive locations. The second stage thus aims to refine this selection. To achieve this, we decided to train a classifier to recognize the correct landmarks among all candidates. However, this stage is not designed to make a final decision but instead to estimate the candidate likelihood. More flexibility is thus provided to the system during alignment (Section 5).

As the previous stage has selected an example set, the classification and representation of each candidate can be more expensive. Unlike the first stage, here we can here use different features for a more suitable representation, whether based on histograms or spatial information. Section 6.4.1 will discuss the evaluation of each feature and show that SIFT descriptors lead to the best results.

Because kernel functions can also be more expensive, several types of kernels can be used, differing in their linearity and computational cost, as well as their discriminative power. To evaluate the different kernels, we proceed to a benchmark study presented in Section 6.4.2. This evaluation shows that the exponential-χ^2 kernel leads to the best results. A major drawback of these kernels is their computational cost, typically in $O(ND^2)$ or $O(ND^3)$, which is non-impeding since this kernel is applied to few candidates.

After applying the second stage classification function $C_2(\mathbf{x}^p)$ to the candidate pixels, we proceed to a calibration of the SVM outputs by fitting a logistic regression function [35] on every candidate pixel:

$$p(l_i = 1|C_2(\mathbf{x}^p)) = \frac{1}{1+exp\{aC_2(\mathbf{x}^p)+b\}} \tag{13}$$

where l_i {not aligned(-1), aligned($+1$)} denoting whether the landmark is aligned with its corresponding location in the image. Parameters a and b are estimated using a cross-validation process.

This calibration gives, for each candidate pixel, an output varying between 0 and 1. Since after normalization, the sum over the candidates is 1, this score can be interpreted as a likelihood estimation. This is a critical step for the following optimization process which is based on these response maps.

5. Deformable model alignment

The two-stage detector learns a classifier for each landmark. Moreover detections are not necessarily perfect, and the correct location does not always match with the highest detector score. This can especially happen when some points are occluded because of head pose or visual obstruction (i.e., hair or glasses). In this section, we describe how we fit a deformable model using detector outputs to better handle cases where the local detectors are likely to fail.

5.1. Problem formulation

5.1.1. Deformable model

We parametrize the 2D relative position of the landmarks using a Point Distribution Model (PDM). The position of the ith landmark $\mathbf{s_i}$ in this image is given by:

$$\mathbf{s}_i = s\mathbf{R}(\bar{\mathbf{s}}_i + \Phi_i\mathbf{q}) + \mathbf{t} \tag{14}$$

where $\bar{\mathbf{s}}_i$ denotes the mean location of the ith landmark and Φ_i the principal subspace matrix computed from training shape samples using PCA. Here, $\mathbf{p} = \{s,\mathbf{R},\mathbf{t},\mathbf{q}\}$ denotes the PDM parameters, which consist of global scaling s, rotation $\mathbf{R}$ and translation $\mathbf{t}$. Vector $\mathbf{q}$ represents the deformation parameters that describe the deformation of $\mathbf{s}_i$ along each principal direction.

If we assume that $\mathbf{q}$ is distributed as a diagonal Gaussian and place a non-informative prior information with zero mean on the rigid parameters s, $\mathbf{R}$ and $\mathbf{t}$ to place the model in the image frame, we have:

$$p(\mathbf{p}) \propto \mathcal{N}(\mathbf{p}; \mathbf{0}, \Lambda) \quad \text{with} \quad \Lambda = \text{diag}\,[\mathbf{0}, \lambda_1, \lambda_2, \ldots, \lambda_m] \tag{15}$$

where λ_i is the eigenvalue of the ith mode of the non-rigid deformation. These models have the advantage of being both simple and efficient while showing a good ability to model face deformation [7].

5.1.2. Fitting

We are here interested in combining this statistical shape model and the independent local detections presented in Section 4. Given a deformable face model, we use the second stage sparse response maps within a joint optimization process to seek the optimal parameters $\mathbf{p}$. In CLM, finding these parameters is a process involving two different types of information: a term taking considering the confidence in the model and penalizing non-relevant deformations (regularization term) and a term characterizing the observation using classifier measures (data term). From the work of [38], the probabilistic interpretation of this fitting can be formulated as finding:

$$p(\mathbf{p}|\{l_i = 1\}_{i=1}^n, \mathcal{I}) \propto p(\mathbf{p}).p(\{l_i = 1\}_{i=1}^n|\mathbf{p}, \mathcal{I}). \tag{16}$$

By assuming conditional independence between landmark detections, we can formulate

$$p(\mathbf{p}|\{l_i = 1\}_{i=1}^n, \mathcal{I}) \propto p(\mathbf{p}) \prod_{i=1}^{n} p(l_i = 1|\mathbf{s}_i, \mathcal{I}) \tag{17}$$

where the regularization term takes the form $p(\mathbf{p})$, and $p(l_i = 1|\mathbf{s}_i, \mathcal{I})$, the observation given by $\mathcal{S}_2$ denotes the probability of the point to be well aligned, given the landmark position $\mathbf{s_i}$ in the image $\mathcal{I}$.

However, classifier outputs $p(l_i = 1|\mathbf{s}_i, \mathcal{I})$ cannot be directly evaluated over the entire region of interest as S_2 outputs are sparse. By applying the Bayes theorem over the observation term, we have:

$$p(l_i = 1|\mathbf{s}_i, \mathcal{I}) \propto p(\mathbf{s}_i|l_i = 1, \mathcal{I}) p(l_i = 1|\mathcal{I}). \tag{18}$$

Considering that $p(l_i = 1|\mathcal{I})$ is uninformative, because all rigid transformations are equally likely, we have $p(l_i = 1|\mathbf{s}_i, \mathcal{I}) \propto p(\mathbf{s}_i|l_i = 1, \mathcal{I})$. We then approximate the detection responses at $\mathbf{s}_i$ with isotropic Gaussian estimators, using the pixel candidates as the Gaussian centers. Knowing that $\mathcal{S}_2$ gives for each landmark a probability for a set of K candidates, we can formulate:

$$p(\mathbf{s}_i|l_i = 1, \mathcal{I}) \approx \sum_{k=1}^{K} \pi_{ik} \mathcal{N}\left(\mathbf{s}_i; \mathbf{s}_{ik}^*, \sigma_{ik}^2 \mathbf{I}\right). \tag{19}$$

If we assume that no prior information is placed on the landmark location in the image, we can pose $\pi_{ik} = \frac{1}{K}$. Here, $\mathbf{s}_{ik}^*$ is the location of the kth maximum for the ith landmark and σ_{ik}^2 its corresponding variance.

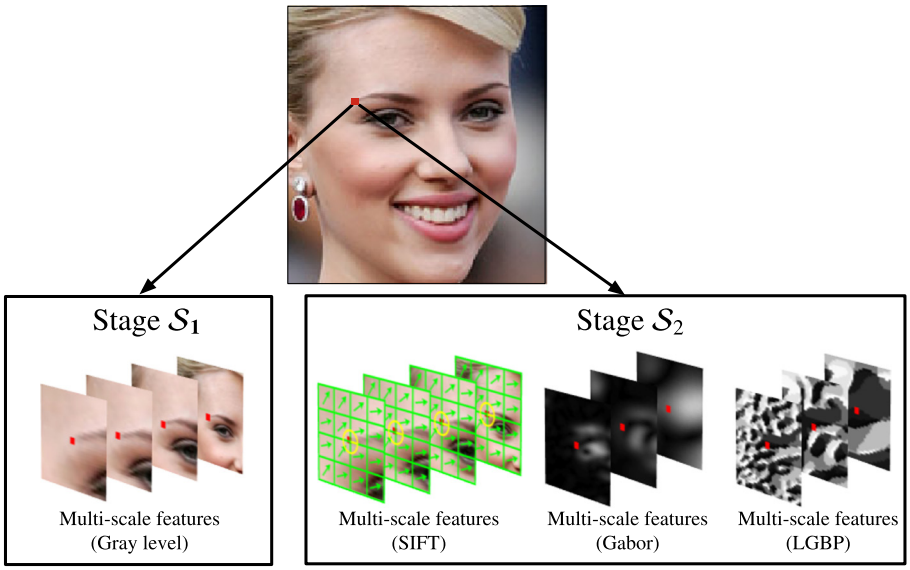

Fig. 2. Multi-scale features for a candidate pixel i (red square) for stages $\mathcal{S}_1$ and $\mathcal{S}_2$. Patch sizes increase at each level, emphasizing different levels of information. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We define the variance as inversely proportional to classifier outputs. A candidate pixel far from a highly likely landmark location is thus strongly penalized.

Using Eq. (17), taking the negative log of Eq. (15) for the regularization-term and Eq. (19) for the data-term, and introducing the likelihood of the kth maximum for the ith landmark $\rho(\mathbf{s}^*_{\mathbf{ik}})$, we obtain the optimization function:

$$\mathcal{F}(\mathbf{p}) = \|\mathbf{p}\|^2_{\Lambda^{-1}} + \sum_{i=1}^{n}\sum_{k=1}^{K}\rho(\mathbf{s}^*_{ik})\|\mathbf{s}_i(\mathbf{p}) - \mathbf{s}^*_{ik}\|^2. \quad (20)$$

From the previous assumption on σ^2_{ik}, we can easily verify that $\rho(\mathbf{s}^*_{ik}) = \frac{\sigma^{-2}_{ik}}{\log K}$.

5.2. Optimization

This optimization aims to minimize the weighted least squares difference between the PDM and the coordinates of the maxima detected by the local classifiers. Eq. (20) is iteratively minimized by using the Gauss–Newton algorithm. To proceed, we must first take the first-order Taylor expansion of the PDM landmark:

$$\mathbf{s}_i(\mathbf{p} + \Delta\mathbf{p}) \approx \mathbf{s}_i(\mathbf{p}) + \mathbf{J}_i\Delta\mathbf{p}. \quad (21)$$

$\mathbf{s}_i$ is the current shape estimate of the ith landmark and $\mathbf{J}_i$ is the Jacobian matrix calculated using Eq. (14).

Following the optimization function (20), we can formulate the parameter update:

$$\Delta\mathbf{p} = -\mathbf{H}^{-1}\left(\Lambda^{-1}\mathbf{p} + \sum_{i=1}^{n}\sum_{k=1}^{K}\rho(\mathbf{s}^*_{ik})\mathbf{J}_i^T(\mathbf{s}_i(\mathbf{p}) - \mathbf{s}^*_{ik})\right) \quad (22)$$

where $\mathbf{H}$ is the Gauss–Newton approximation to the Hessian matrix:

$$\mathbf{H} = \Lambda^{-1} + \sum_{i=1}^{n}\sum_{k=1}^{K}\rho(\mathbf{s}^*_{ik})\mathbf{J}_i^T\mathbf{J}_i. \quad (23)$$

Finally, Eq. (22) is applied additively to the current parameters: $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$.

Fig. 3 presents an overview of our optimization. First, shape parameters $\mathbf{q}$ are set to $\mathbf{0}$ and the model $\mathbf{s}(\mathbf{p})$ is initialized in the image by finding a first set of rigid parameters using the maxima of each response map. The optimization function seeks to minimize the weighted least square difference between the PDM and the coordinates of each candidates, each weighted by their confidence given by $\mathcal{S}_2$, to find the optimal variation parameter vector $\Delta\mathbf{p}$.

Even if the two-stage classifier is robust and accurate, estimating the response map can present some errors. The response maximum may not always coincide with the correct landmark localization, and some detection ambiguities can appear. The optimization function (20) used is especially designed to handle this type of problem. It uses all the potential point locations, weighted by the probability that the point is the desired landmark. By introducing these information, this optimization function thus presents two main advantages: avoiding outliers and removing the ambiguity of certain responses. Algorithm 23 outlines the complete fitting procedure.

6. Experiments

6.1. Implementation details

As face detector, we apply the [49] OpenCV implementation. Some post-processing techniques are then applied to the detected face: it is enlarged by 20% so every chin is included in the face box. Each face is then resized to 150 × 150 pixels. For $\mathcal{S}_1$, facial images are normalized using the CLAHE algorithm [34], while $\mathcal{S}_2$ does not use any gray-level normalization.

Algorithm 1. Model Alignment using two-stage SVMs

```
for each ROI do
    Compute dense response maps (Eq. (12)).
    Select local maxima.
    Measure local maxima likelihood (Eq. (13)).
end for
Set q = 0 and initialize p using the global maxima of S2.
while p not converged do
    Evaluate Jacobian at p.
    Estimate Δp (Eq. (22)).
    Update PDM (Eq. (21)).
end while
return the PDM
```

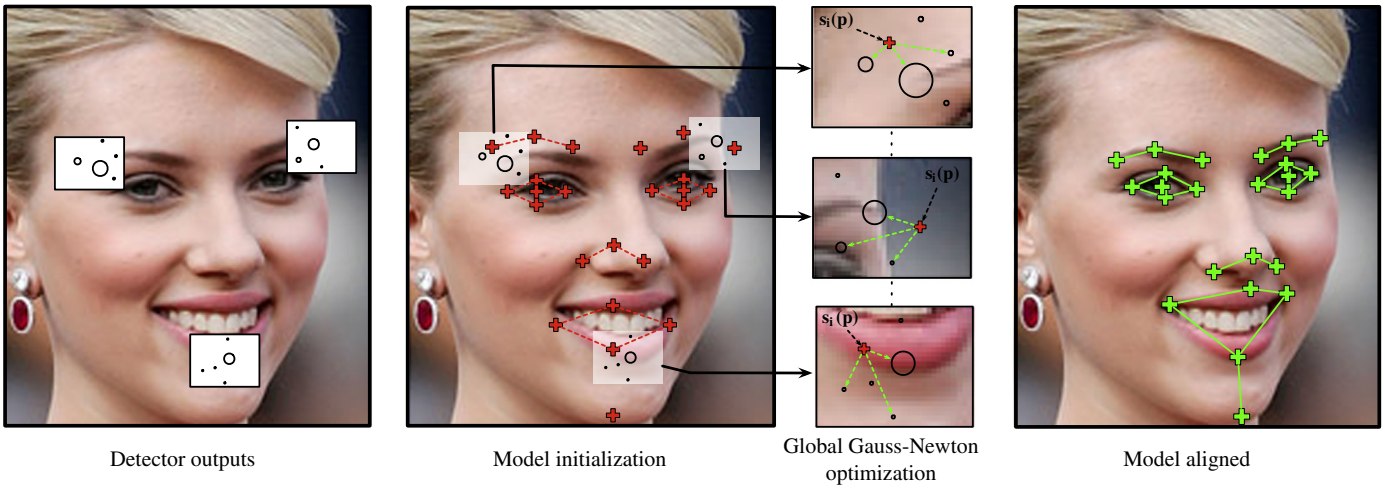

Fig. 3. Optimization overview. Using the sparse maps given by $\mathcal{S}_2$, this process tries to fit a deformable by seeking the optimal shape parameter $\mathbf{p}$ according to the distance with each map maxima.

For each facial landmark, $\mathcal{S}_1$ is trained with positive and negative examples. For a given landmark, we use 9 multi-resolution patches as positive examples (target samples): the first centered on the ground truth and the others on 8 positions surrounding the ground truth. As negative (non-target samples) set, we use 16 multi-resolution patches 4 to 6 pixels distant from the true facial point. To add relevant examples during the training, rather than using random samples, we apply the same bootstrap process as in [37]. Finally, both stages are trained with the SimpleMKL algorithm [36].

In the evaluation step, $\mathcal{S}_1$ evaluates the response of each pixels p contained in the region of interest (ROI). We use one ROI per landmark, and their sizes are calculated by searching a box encompassing 99% of the training sample point locations.

6.2. Experimental setup and data

Our work focuses on localizing landmarks in challenging face images taken under many poses and lighting conditions or in the presence of facial expressions and occluding objects. Several databases have thus been used for training and evaluation.

- MUCT [29]: The MUCT database contains 3755 faces with 76 manual landmarks. The database was created to provide more diversity in lighting, age, and ethnicity than currently available landmarked 2D face databases.
- PIE [41]: The CMU Pose Illumination and Expression database contains more than 40,000 facial images of 68 people. Each person was imaged across 13 different poses, under 43 different illumination conditions, and with 4 different expressions.
- LFPW [2]: The Label Face Part in The Wild database has 1432 faces from images downloaded from the web using simple text queries on sites like google, flickr, and yahoo. Three different annotators labeled each image. Due to copyright issues, the author must make available a list of image URLs. As links have disappeared over time, we were able to only collect 594 images. This database, in contrast to others, is not acquired in a laboratory but with samples extracted from the web. These "real life" images can thus present difficult lighting, sunglasses, partial occlusion or arbitrary facial expression.
- Cohn-Kanade [22]: This is a representative, comprehensive and robust test-bed for comparative facial expression studies. It contains image sequences with lighting conditions, and the contexts are relatively uniform. The database contains 486 sequences starting with neutral expressions and ending with the expression apex. For our experiments, we used the first (neutral expression) and last frames (expression apex) of each sequence.
- FacePix [24]: This database has been developed to precisely measure the robustness of face analysis algorithms using incremental variations in pose angles. The entire FacePix database contains 16290 images (30 people × 3 sets × 181 images per set). For our experiments, we have randomly selected 230 images with pose angle variations varying across a range from −40° to +40°.

Table 1
Summary of datasets for Training (Tr), Validation (V) and Test (Te). It depicts information (Info.) about the number of images (Img.) and subjects (Sub.) for each database. Variation sources (Var.) in each database are also listed: Pose (P), Lighting (L), Real-world conditions (R), Expressions (E), and Complex Background (B).

	Set	MUCT	LFPW	PIE	CK	FP	BioID
Info.	Img.	200	594	321	962	570	1482
	Sub.	199	–	130	100	30	23
	Var.	P, L	R	P, L, E	E	P	B, L
Tr.	Set1	500	–	–	–	–	–
	Set2	1500	200	321	–	–	–
V.	Set3	–	394	–	–	–	–
Te.	Set4	–	–	–	962	–	–
	Set5	–	–	–	–	570	–
	Set6	–	–	–	–	–	1482

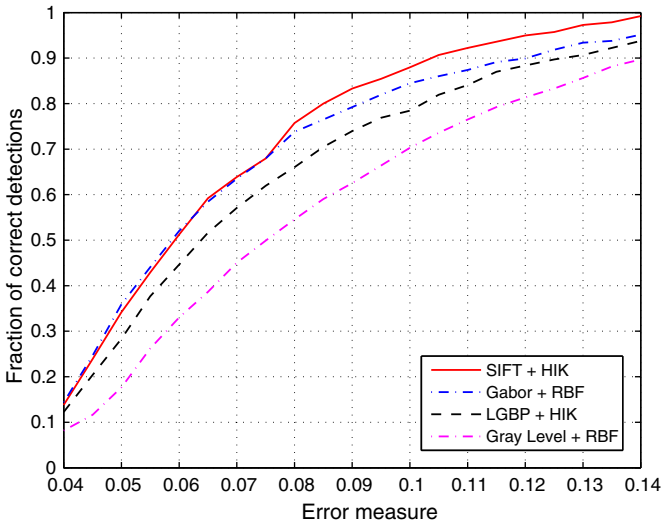

Fig. 4. Comparison on Set 3 of the cumulative error performance of point-to-point errors for feature evaluation.

- BioID [21]: This contains 1521 frontal face images that vary in illumination, background, face size and slight head pose variation. Based on results from the literature [30,11,40,46], this database is considered as challenging. A test on BioID thus constitutes a benchmark comparison with the existing state-of-the-art methods.

Table 1 depicts the different sets used for training and evaluation, and it shows the characteristics of each database. Set 1 is used to train the first stage of our framework. As this classifier is trained over a high number of locations, we could not use many images. In contrast, the second stage is only trained on a selected subset of locations. We used more images from different databases, adding variability to our training samples (Set 2). Set 3 is our cross-validation set used to select the best combination of features and kernels (Section 6.4). Set 4 is used to evaluate the robustness of our methods w.r.t. facial expressions (Section 6.6.1). Set 5 is used to analyze the performance of our method on non-frontal faces (Section 6.6.2). Finally, Set 6 is used to compare the proposed system with state-of-the-art methods (Section 6.7).

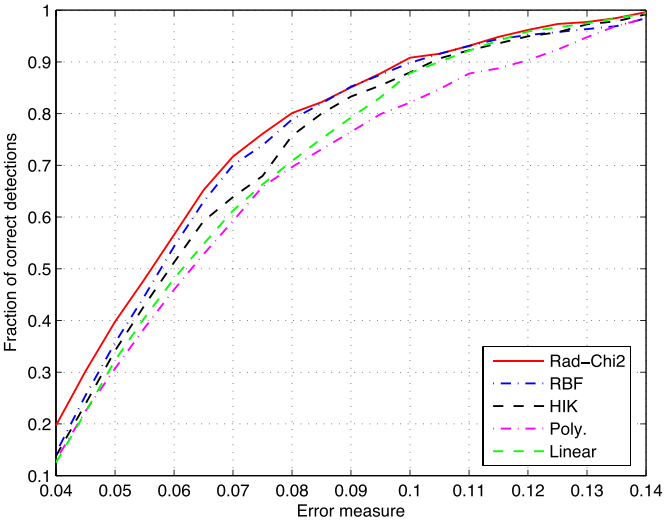

Fig. 5. Comparison on Set 3 of the cumulative error performance of point-to-point errors for kernel evaluation.

6.3. Performance measures

As a performance measure, we used the distance from the detected point to the manually labeled ground truth. This distance is then normalized by the interocular distance, regularly used in state-of-the-art studies. The detection error of point i is defined as the Euclidian distance d_i between the point and its manually labeled ground truth. The average is given thus:

$$m_e = \frac{1}{nd_{IOD}} \sum_{i=1}^{n} d_i \tag{24}$$

where d_{IOD} is the Interocular Distance, dened as the distance between the eye centers (computed using eye corners), and n the total number of images. Typically, a location is accepted if the distance to the ground truth is lower than 10% of the interocular distance ($m_e < 0.10$).

6.4. Two-stage classifier evaluation

The following sections report validation results on the LFPW database, obtained by associating different features/kernels couple. We first test several descriptors to find the optimal one and evaluate the selected feature using several kernels. Each following curve depicts the cumulative distribution error obtained by only taking the global maximum of each response map as final detections. Any constraint between points is thus applied.

6.4.1. Features

For this evaluation, we have used the features presented in Section 4.1. Spatial features (gray level and Gabor magnitude images) are combined with the classical RBF kernel. For the histogram-based features (SIFT and LGBP), we have used the Histogram intersection kernel (HIK). Fig. 4 presents the results of this evaluation performed on Set3.

This study shows that the worst results are obtained using gray levels as features. SIFT and Gabor features present roughly identical error measures with small error, and 65% of the images have an error of less than 7.5%. Beyond this error, the SIFT descriptor gives better results and shows higher robustness.

6.4.2. Kernels

The second evaluation round now concerns selecting the best kernel to use with the SIFT descriptor. We decided to compare the results obtained using the RBF, histogram intersection, polynomial and exponential-χ^2 kernels, introduced in Section 4.2. This evaluation is again performed on the cross-validation dataset (Set 3). Fig. 5 presents the results of this study.

This figure shows that the polynomial kernel gives the worst results among all evaluated kernels, especially for an error of 10%. The others perform identically up to an error of 10%. Classical radial and radial-χ^2 kernel results are quite similar, with a small advantage for the radial-χ^2 kernel, especially for the precision around 4%. It is also important to note that, even if the linear kernel is not as precise as a radial-χ^2 kernel, it would be much more suitable for real-time tracking since the computational cost is clearly reduced. In this case, the precision issue could be avoided by reducing the size of the ROI.

6.4.3. Multiple-Kernel learning

The following experiment considers the multiple-kernel aspect and multi-resolution features used. Fig. 6 shows the impact that multi-resolution patches have on facial point detection. Decision maps obtained with kernels $k_1,\ldots,k_4$ show that each kernel k_i emphasizes different information level. Kernel k_1 handles this problem using a local perspective, whereas kernel k_4 uses global information. The final decision map, corresponding to the SVM output, uses these different information levels to give a confidence index to each candidate pixel over the search region.

As we have one SVM per facial landmark, we find one set of weights $\beta_1,\beta_2,\beta_3,\beta_4$ for each facial landmark. Table 2 reports the mean weights learned for the points belonging to the same group of facial landmarks.

For the eyes, brows and mouth the biggest β corresponds to the kernel using the most local information (first scale), followed by the kernel using the most global information (last scale). These points are defined by a discriminative texture with high contrast, explaining the high weights on the local kernel. In contrast, the global kernel helps attenuate the possible ambiguities induced by the local support of the first scale, which may confuse the brow and eye, as clearly visible in Fig. 6. For the chin, a higher weight is applied on the global

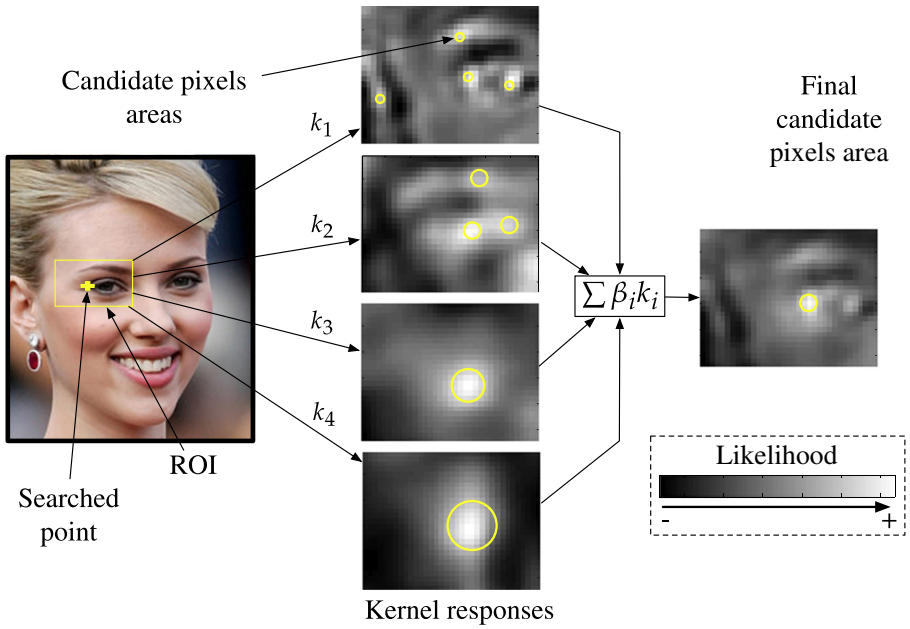

Fig. 6. Decision map of each kernel for detecting the outer corner of the right eye.

Table 2
Mean weights associated with each kernel for different facial areas.

Facial landmarks	β_1	β_2	β_3	β_4
Brows (6 points)	0.35	0.20	0.17	0.28
Eyes (10 points)	0.40	0.20	0.20	0.20
Nose (3 points)	0.29	0.23	0.13	0.35
Mouth (4 points)	0.31	0.23	0.18	0.28
Chin (1 point)	0.14	0.26	0.25	0.35

information. This stems from the aperture effect: for this type of point, a texture extracted at a low scale is not relevant (applying a horizontal translation on the patch does not change the local texture). In contrast, a higher scale captures some clues in the face image i.e., the location of the mouth or the nose, thus helping the system locating this landmark.

6.5. Fitting evaluation

In this section, we propose studying the interest of using a deformable model to constrain points. For this evaluation, we first compare the point detection accuracy:

- at the output of classifier $\mathcal{S}_2$ by taking the global maximum of each response map (Fig. 5).
- at the model initialization step, corresponding to the mean model obtained using previous maxima.
- at the end of the alignment process.

For a deeper study, we also approximate the human error using annotations on LFPW from three different workers. We use the mean of these annotation sets as ground truth, compute the error for each worker and take the mean as human error. Fig. 7 presents the evaluation results.

Several conclusions can be drawn from Fig. 7. First, initializing a model from the response of $\mathcal{S}_2$ introduces a bias, resulting in much less accurate detections between 2% and 10%. However, the robustness of the results obtained at 10% from the initialization model and $\mathcal{S}_2$ is quite similar because the model fixes all the non-plausible detections. To evaluate the alignment and contribution of our method, we compare the curve obtained from the $\mathcal{S}_2$ maxima and that obtained after the fitting process. After $\mathcal{S}_2$, 89% of the images have an average

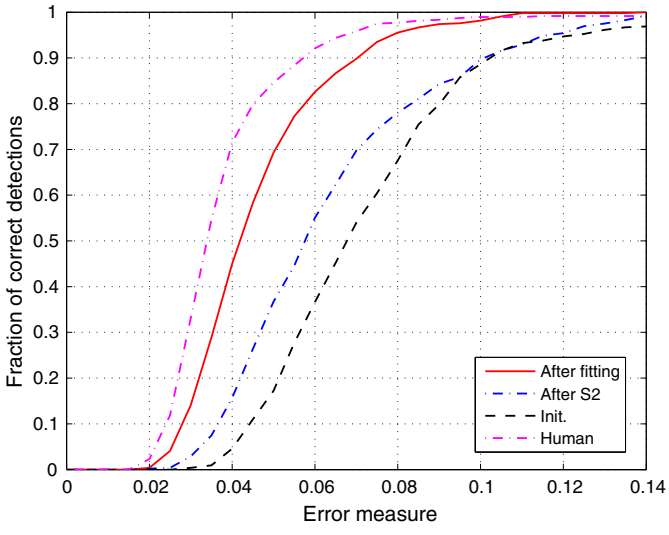

Fig. 7. Comparison on Set 3 of the cumulative error performance of point-to-point error for fitting evaluation.

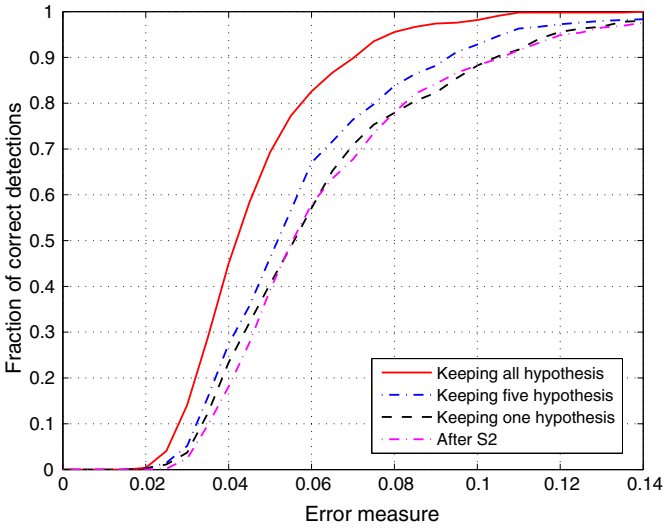

Fig. 8. Comparison on Set 3 of the cumulative error performance of point-to-point error depending on the number of candidates kept.

point error under 10%. The figure shows that the proposed fitting process clearly improves the results with 97% of the images having an error of less than 10%. We can also see that the human error is very close to that obtained with our system. Moreover, this human error curve emphasizes the difficulty that humans face when annotating difficult images. We can indeed see that almost no images are annotated with an error of less than 2%, which corresponds to an error of roughly 1 pixel for a mean interocular distance of 50 pixels.

During a second evaluation round, we compute and compare the cumulative error obtained by reducing the number of kept hypotheses during optimization process (Fig. 8).

The results of this study show that we reduce model flexibility if we only keep the global maximum of each map during the optimization process. This lack directly impacts the cumulative error, which tends toward the one obtained from the output of the second stage. Moreover, this figure also shows that adding more hypothetical locations gives a better cumulative error distribution. Indeed, these adequately weighted locations give the system more options and avoid outliers. The system flexibility is thus clearly improved without any loss in computational time.

6.6. Robustness evaluation

In the following experiments, we propose evaluating the system robustness on two current facial deformations: expressions and poses.

6.6.1. Effect of expressions

This study aims to evaluate our system on expressive images. We apply our detector on Set4, which contains neutral and expressive images. Considering shape, the effects of expressions on facial images are

Table 3
Results for expression robustness on Set 4 for expressive and neutral images. Each point is grouped by sub-part: LE = Left Eye, LEb = Left Eyebrow, RE = Right Eye, REb = Right Eyebrow and M = Mouth.

Method	Data	Mean error and standard deviation (%) for each model				
		LE	LEb	RE	REb	M
MuKam	Neu.	3.5 (1.4)	5.2 (2.4)	3.2 (1.5)	5.3 (2.8)	4.9 (2.7)
	Exp.	3.5 (1.7)	5.4 (3.0)	3.3 (1.6)	5.6 (3.1)	7.4 (3.5)
Saragih et al.	Neu.	6.0 (1.5)	7.2 (3.0)	5.4 (1.6)	6.6 (3.3)	5.2 (2.4)
	Exp.	6.4(1.8)	7.8 (4.3)	6.8 (3.6)	7.9 (3.8)	8.8 (4.5)

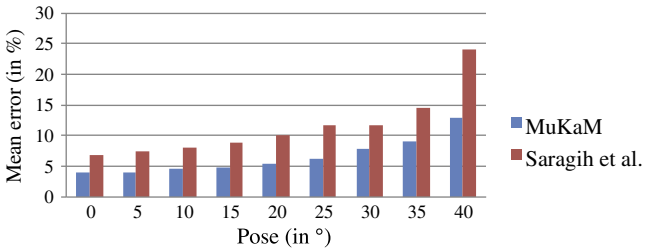

Fig. 9. For face images presenting different poses (Set 5).

mostly observable on the eyebrows and mouth, resulting in larger model deformations. To correctly evaluate our system on expressive faces, we regroup our measures by face-parts, giving errors relevant to expressive deformations. We calculate the mean error on 5 face-parts: right eye, left eye, right eyebrow, left eyebrow and mouth. For comparison purpose, we also evaluate the work of [38] considered to be the state-of-the-art method for CLM. However, it is important to note that the provided code is primarily designed for tracking and not really well suited for image alignment task. Nevertheless, we slightly modify it to get a frame by frame detector (region of interest have been increased, and the parameters are initialized for each image) and to proceed to the fairest comparison. Table 3 depicts the evaluation results.

Our system presents accurate results and a good ability to generalize on expressive images. For each sub-model, the results obtained on expressive images are close to those obtained on neutral faces. Our system locates points with an error of less than roughly 2 pixels for the eyes and 2.5 pixels for the eyebrows. The mouth is more difficult to detect because of the possible deformations. As expressive faces, we choose frames that correspond to the apex of the expression where deformations are often exaggerated (especially in Cohn-Kanade database), which are thus not necessarily admissible for the shape model.

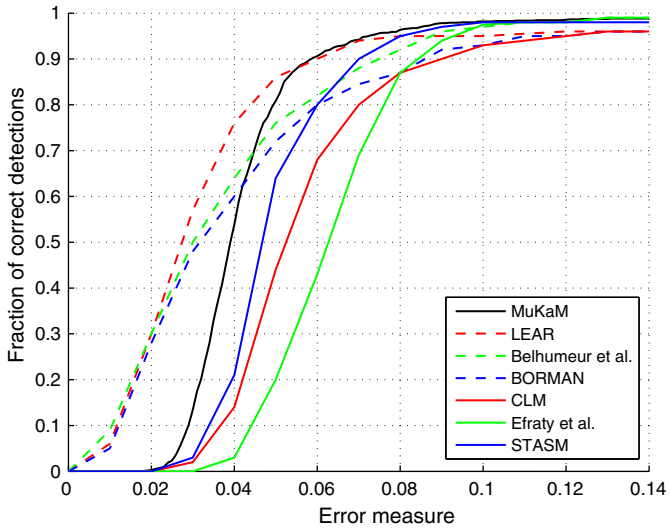

Fig. 11. Comparison on Set 3 of the cumulative error performance of point-to-point error for fitting evaluation.

6.6.2. Effect of pose

For this evaluation, we apply our detector on Set 5 containing faces presenting pose varying along the yaw angle from −40° to +40°. As for the expression robustness evaluation, we also show the results obtained by the work of [38] in a frame by frame detector mode. Figs. 9 and 10 show some quantitative and qualitative results of this study.

The Histogram represented in Fig. 9 demonstrates the efficiency of our method when handling non-frontal faces up to 25°, where the mean errors stay below 6.1%. The mean error is lower than 5% (resp. 10%) for a pose up to 15° (resp. 35°). The system shows some difficulties over this range because we only estimate an in-plane rotation during the

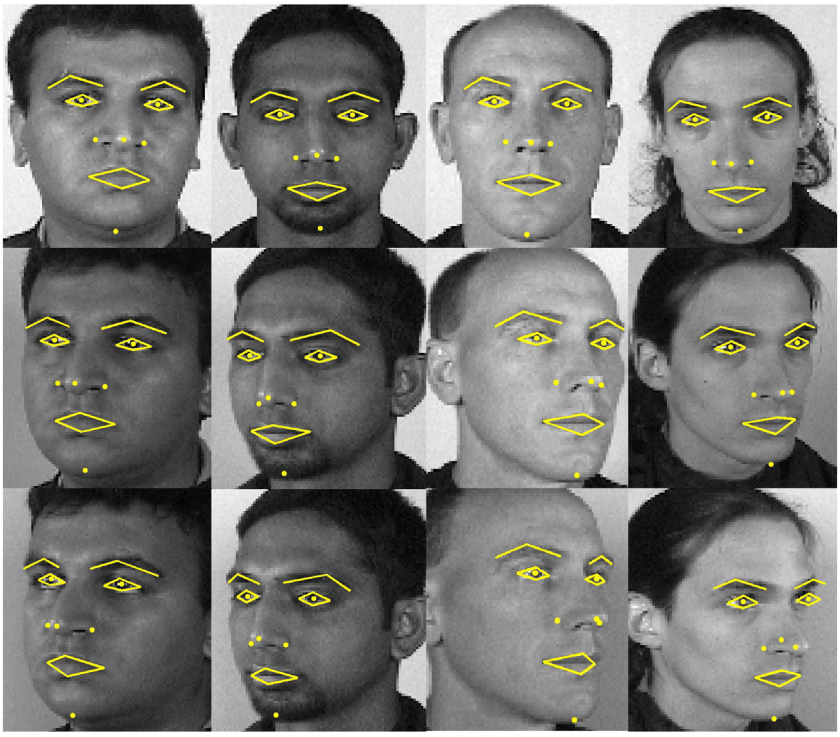

Fig. 10. Results obtained for face images presenting different poses: 0, 25 and 40°.

model initialization. When the face presents out-of-plane rotations, the model initialization may thus be far from the solution. Moreover, because we train our deformable model on near-frontal faces, it is not especially able to handle this type of large variation. One solution to overcome this weakness would be to task an upstream system to give a prior information on facial deformation induced by poses as proposed by [18]. But this kind of system should necessarily be very robust to not steer the alignment process in a wrong direction. We could also think about an iterative process, in which the pose parameters are repeatedly estimated, and the corresponding transformation is then applied to the patch expert. This way, the pose information is directly introduced in the score computation. But this process may increase processing time because of the necessity to compute the score maps at each new pose parameters. Such an approach would be much more appropriate in a tracking framework [38], where the pose parameters of the previous frame could be used.

However, the performances of our method up to 35° show that this system is well suited for applications designed for human computer interaction, in which the user generally faces the system.

6.7. Results & comparison

The following experiment compares our method with the current state-of-the-art methods. This evaluation is performed on Set 6, containing BioID faces on which numerous systems have been tested. This is a particularly challenging dataset as it is captured under cluttered backgrounds and various real-world illumination environments. We thus compare our system with several recent methods: the original CLM from [11], LEAR from [27], BoRMaN from [46], STASM from [30] and the system of [2] and [14]. For a fair comparison, we use the error measure for only 17 points, as [11] explains. Fig. 11 reports the cumulative error distribution of the *me*17 error measures on BioID database.

The reported results show that our method can be compared favorably with state-of-the-art experimental results. As pointed by [5], it is important to note that the curves of the works of [27] and [2] are hardly comparable as they re-annotated some of the data.

Fig. 12 shows some qualitative results on representative images from two different databases: the first row for LFPW (Set3) and the second row for BioID (Set 6). These images depict the good ability the system has to handle different deformations (induced by pose variations or expressions) and partial occlusions.

6.8. Processing times

To reduce complexity and offer the system the ability to quickly annotate an image, the proposed detector uses a two-stage classifier as a detection step. The first stage $\mathcal{S}_1$ uses a linear classifier; the maxima selection is thus performed with convolution operations. Using a MATLAB code on Intel i7 2.8 GHz, this step requires less than 0.10 s per image. The second stage estimate the candidate likelihoods in under 1.2 s per image. For comparison, running the last stage directly on the whole region (i.e., extracting the descriptor from each pixel in the region and computing the kernel function with all examples) would require more than 5 min per image. Finally, our system annotates the 24 points in approximately 1.5 s. We are currently implementing the algorithm in C++ to achieve real-time localizations.

7. Conclusion

Accurate and fully automatic facial landmark detection would have many applications and remains an essential step in many face analysis systems. In this work, we have focused on two main aspects of facial landmarking: the detection step, which locates each point independently, and the optimization step, which brings constraints to these locations.

The Multi-Kernel Appearance Model has been designed to overcome the inherent limitation of PAMs. Indeed, MuKAM is based on a detection step, during which several hypotheses are generated, and used to estimate the parameters of a deformable model. Several key attributes contribute to system robustness and accuracy. The approach is based on multi-resolution features to encode more comprehensive information. These descriptors are then combined with multiple-kernel SVM using a well suited and highly discriminative kernel function. A bootstrap strategy has also been implemented, providing relevant examples during the training process. Finally, to reduce computational time, we propose a two-stage classifier especially designed to reduce the amount of data computed by a time-consuming kernel machine: the first stage based on gray-level features and linear kernels quickly select a set of hypothetical locations, while the second stage proposes a more complex focus on these hypotheses and estimate their likelihood.

Moreover, a final step introduces constraints between detections. It is based on an alignment process that finds the optimal parameters of a statistical shape model: a Gauss–Newton algorithm optimizes a cost function especially to handle the sparse observation given by the second stage of our classifier.

In this paper, we investigate key aspects of our method by performing complete experiments at different levels. We also analyze the robustness of our system against usual face variation aspects: pose and expression. The results of these extensive cross-database evaluations show the good ability of our system to cope with these perturbations. Finally, comparisons with other methods performed on the standard BioID database show that our detector can be compared favorably with state-of-the-art methods.

As our current works concern emotion recognition and human behavior analysis, the accuracy and robustness of this facial landmark detector will allow us to consider interactions in real-world conditions.

Fig. 12. Qualitative results on Set 3 and Set 4 (LFPW and BioID databases respectively).

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) and the Cap Digital Business cluster for digital content.

References

[1] Kevin Bailly, Maurice Milgram, Philippe Phothisane, Erwan Bigorgne, Learning global cost function for face alignment, Proc. IEEE Int'l Conf on Pattern Recognition (ICPR'12), 2012, pp. 1112–1115.
[2] P. Belhumeur, D. Jacobs, D. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR'11), 2011, pp. 545–552.
[3] Xudong Cao, Yichen Wei, Fang Wen, Jian Sun, Face alignment by explicit shape regression, Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'12), 2012, p. 2887.
[4] T. Cootes, G. Edwards, C. Taylor, Active appearance models, Proc. IEEE European Conference on Computer Vision (ECCV'98), 1998, p. 484.
[5] T. Cootes, M. Ionita, C. Lindner, P. Sauer, Robust and accurate shape model fitting using random forest regression voting, Proc. IEEE European Conference on Computer Vision (ECCV'12), 2012, pp. 278–291.
[6] T. Cootes, C. Taylor, Active shape models-smart snakes, Proc. British Machine Vision Conf. (BMVC'92), 1992, pp. 9–18.
[7] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape models-their training and application, Proc. IEEE Conf. Comp. Vision and, Pattern Recognition (CVPR'95), 1995, p. 38.
[8] D. Cristinacce, T. Cootes, Facial feature detection using adaboost with shape constraints, Proc. British Machine Vision Conference (BMVC'03), 2003, pp. 231–240.
[9] D. Cristinacce, T. Cootes, Feature detection and tracking with constrained local models, Proc. British Machine Vision Conf. (BMVC'06), 2006, pp. 929–938.
[10] D. Cristinacce, T. Cootes, Boosted regression active shape models, Proc. British Machine Vision Conf. (BMVC'07), 2007, pp. 880–889.
[11] D. Cristinacce, T. Cootes, Automatic feature localisation with constrained local models, Pattern Recogn. 41 (10) (2008) 3054–3067.
[12] S. Duffner, C. Garcia, A connexionist approach for robust and precise facial feature detection in complex scenes, Int. Symposium on Image and Signal Processing and Analysis (ISPA'05), 2005, p. 316–312.
[13] M. Eckhardt, I. Fasel, J. Movellan, Towards practical facial feature detection, Int. J. Comput. Vis. Recognit. Artif. Intell. (IJCV'09) 23 (3) (2009) 379.
[14] B. Efraty, C. Huang, S. Shah, I. Kakadiaris, Facial landmark detection in uncontrolled conditions, Proc. Int'l Joint Conf. Biometrics (IJCB'11), 2011, pp. 1–8.
[15] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, Int. J. Comput. Vis. 61 (1) (2005) 55–79.
[16] C. Goodall, Procrustes methods in the statistical analysis of shape, J. R. Stat. Soc. Ser. B Methodol. (1991) 285–339.
[17] N. Gourier, D. Hall, J. Crowley, Facial features detection robust to pose, illumination and identity, IEEE Int'l Conf. on Systems, Man and Cybernetics (SMC'04), vol. 1, IEEE, 2004, pp. 617–622.
[18] L. Gu, T. Kanade, 3d alignment of face in a single image, IEEE Int'l Conf. Computer Vision and, Pattern Recognition (CVPR'06), 2006, pp. 1305–1312.
[19] L. Gu, T. Kanade, A generative shape regularization model for robust face alignment, Proc. of the European Conference on Computer Vision (ECCV'08), 2008, pp. 413–426.
[20] S. Hanif, L. Prevost, R. Belaroussi, M. Milgram, Real-time facial feature localization by combining space displacement neural networks, Pattern Recogn. Lett. 29 (8) (2008) 1094–1104.
[21] O. Jesorsky, K. Kirchberg, R. Frischholz, Robust face detection using the hausdorff distance, Audio-and Video-Based Biometric Person Authentication, 2001, p. 90.
[22] T. Kanade, Y. Tian, J. Cohn, Comprehensive database for facial expression analysis, Proc. IEEE Conf. Face and Gesture Recognition (FG'00), 2000, p. 46.
[23] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, M. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 27–72.
[24] G. Little, S. Krishna, J. Black, S. Panchanathan, A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose angle and illumination angle, Proc. IEEE Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP'05), vol. 2, 2005, pp. 89–92.
[25] X. Liu, Discriminative face alignment, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 1941–1954.
[26] D. Lowe, Object recognition from local scale-invariant features, Proc. IEEE Int'l Conf. on Computer Vision (ICCV'99), vol. 2, IEEE, 1999, pp. 1150–1157.
[27] B. Martinez, M. Valstar, X. Binefa, M. Pantic, Local evidence aggregation for regression based facial point detection, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI'12) 35 (5) (2012) 1149–1163.
[28] I. Matthews, S. Baker, Active appearance models revisited, Int. J. Comput. Vis. (IJCV'04) 60 (2) (2004) 135–164.
[29] S. Milborrow, J. Morkel, F. Nicolls, The MUCT landmarked face database, Pattern Recognition Association of South Africa, 2010.
[30] S. Milborrow, F. Nicolls, Locating facial features with an extended active shape model, Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR'08), 2008, p. 504.
[31] J. Movellan, Tutorial on gabor filters, Open Source Document, 2002.
[32] M. Nguyen, J. Perez, F. De la Torre, Facial feature detection with optimal pixel reduction SVMs, Proc. IEEE Conf. Face and Gesture Recognition (FG'08), 2008.
[33] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognit. 29 (1) (1996) 51–59.
[34] S. Pizer, E. Amburn, J. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. Zimmerman, K. Zuiderveld, Adaptive histogram equalization and its variations, Comput. Vis. Graph. Image Process. (CVGIP'87) 39 (3) (1987) 355–368.
[35] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Adv. Large Margin Classifiers 10 (3) (1999) 61–74.
[36] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, Simplemkl, J. Mach. Learn. Res. 9 (2008) 2491–2521.
[37] V. Rapp, T. Senechal, K. Bailly, L. Prevost, Multiple kernel learning SVM and statistical validation for facial landmark detection, Proc. IEEE Int'l Conf. Face and Gesture Recognition (FG'11), 2011, pp. 265–271.
[38] J. Saragih, S. Lucey, J. Cohn, Deformable model fitting by regularized landmark mean-shift, Int. J. Comput. Vis. (2011) 200–215.
[39] B. Scholkopf, A. Smola, Learning with kernels, MIT Press, Cambridge, 2002.
[40] T. Senechal, L. Prevost, S. Hanif, Neural network cascade for facial feature localization, Fourth Int'l Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR'10), 2010, p. 141.
[41] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (PIE) database, Proc. IEEE Int'l on Face and Gesture Recognition (FG'02), IEEE, 2002, pp. 46–51.
[42] V. Sreekanth, A. Vedaldi, A. Zisserman, C. Jawahar, Generalized rbf feature maps for efficient detection, Proc. British Machine Vision Conf. (BMVC'10), 2012.
[43] E. Sudderth, A. Ihler, W. Freeman, A. Willsky, Nonparametric belief propagation, IEEE Int'l Conf. Computer Vision and, Pattern Recognition (CVPR'03), vol. 1, June 2003, pp. I-605–I-612, (vol. 1).
[44] P. Tresadern, M. Ionita, T. Cootes, Real-time facial feature tracking on a mobile device, Int. J. Comput. Vis. (2011) 1–10.
[45] Georgios Tzimiropoulos, Joan Alabort-i-Medina, Stefanos Zafeiriou, Maja Pantic, Generic active appearance models revisited, Proc. Asian Conf. Computer Vision (ACCV'12), 2012.
[46] M. Valstar, B. Martinez, X. Binefa, M. Pantic, Facial point detection using boosted regression and graph models, Proc. IEEE Conf. Comp. Vision and Pattern Recognition (CVPR'10), 2010.
[47] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, Proc. IEEE Int'l Conf on Computer Vision (ICCV'09), IEEE, 2009, pp. 606–613.
[48] A. Vedaldi, A. Zisserman, Structured output regression for detection with partial truncation, Proc. Advances in Neural Information Processing Systems (NIPS'09), 2009.
[49] P. Viola, M. Jones, Robust real-time object detection, Int. J. Comput. Vis. 57 (2) (2002) 137.
[50] D. Vukadinovic, 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, 2005, p. 1692.
[51] Y. Wang, S. Lucey, J. Cohn, Enforcing convexity for improved alignment with constrained local models, Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR'08), 2008, pp. 1–8.
[52] M. Wimmer, F. Stulp, S. Pietzsch, B. Radig, Learning local objective functions for robust face model fitting, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI'08) 30 (8) (2008) 1357–1370.
[53] W. Zhang, S. Shan, W. Gao, X. Chen, H. Zhang, Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition, Proc. Int'l Conf. on Computer Vision (ICCV'05), vol. 1, IEEE, 2005, pp. 786–791.
[54] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'12), 2012, pp. 2879–2886.