CHAPTER 5

Efficient Deformable 3D Face Model Fitting to Monocular Images

Luis Unzueta*[1], Waldir Pimenta[2], Jon Goenetxea[1], Luís Paulo Santos[2] and Fadi Dornaika[3,4]

[1]Vicomtech-IK4, Paseo Mikeletegi, 57, Parque Tecnológico, 20009, Donostia, Spain
[2]Departamento de Informática, University of Minho. Campus de Gualtar, 4710-057, Braga, Portugal
[3]Computer Engineering Faculty, University of the Basque Country EHU/UPV, Manuel de Lardizabal, 1, 20018, Donostia, Spain
[4]Ikerbasque, Basque Foundation for Science, Alameda Urquijo, 36-5, Plaza Bizkaia, 48011, Bilbao, Spain

Abstract: In this work we present a robust and lightweight approach for the automatic fitting of deformable 3D face models to facial pictures. Well known fitting methods, for example those taking into account statistical models of shape and appearance, need a training stage based on a set of facial landmarks, manually tagged on facial pictures. In this manner, new pictures in which to fit the model cannot differ excessively in shape and appearance (including illumination changes, facial hair, wrinkles, and so on) from those utilized for training. By contrast, our methodology can fit a generic face model in two stages: (1) the localization of facial features based on local image gradient analysis and (2) the backprojection of

*Corresponding author Luis Unzueta: Vicomtech-IK4, Paseo Mikeletegi, 57, Parque Tecnológico, 20009, Donostia, Spain; Tel: +[34] 943 30 92 30; Fax: +[34] 943 30 93 93; E-mail: lunzueta@vicomtech.org

Fadi Dornaika (Ed)

a deformable 3D face model through the optimization of its deformation parameters. The proposed methodology preserves the advantages of both learning-free and learning-based methodologies. Subsequently, we can estimate the position, orientation, shape and actions of faces, and initialize user-specific face tracking approaches, such as Online Appearance Models (OAMs), which have demonstrated to be more robust than generic user tracking methodologies. Experimental results demonstrate that our strategy outperforms other fitting methods under challenging illumination conditions and with a computational footprint that permits its execution in gadgets with reduced computational power, such as cell phones and tablets. Our proposed methodology fits well with numerous systems addressing semantic inference in face images and videos.

Keywords: 2D shape landmarks, 3D face model, deformable model backprojection, facial actions, facial expression recognition, facial feature extraction, facial parts, face gesture analysis, face model fitting, face recognition, face tracking, gradient maps, head pose estimation, learning-free, Levenberg-Marquardt, monocular image, online appearance model, shape variations, sigmoidal filter, weak perspective.

INTRODUCTION

Generic face model fitting has been a hot research topic during the last decade. It can be seen as an essential part in numerous Human-Computer Interaction applications since it allows face tracking, head pose estimation, identification, and face gesture analysis. In general terms, two types of methods have been proposed: (i) learning-free and (ii) learning-based. The latter require a training stage with many pictures to construct the model, and therefore rely on the choice of pictures for a good fitting in unseen pictures.

Learning-free methodologies depend intensely on some radiometric and geometric properties present in face pictures. These methodologies rely on generic knowledge about faces, which usually incorporates the position, symmetry, and edge profile of facial organs. They can place facial features using low-level methods (e.g. filtering, gradients), typically relying on recognizing individual face features (lips, nose, irises, ...) [1–4]. A large portion of the learning-free methodologies do not produce a full collection of extracted face features, contrary to learning-based strategies.

For example, in [5], the authors exploit a range facial scan in order to automatically distinguish the nose tip for both frontal and non frontal poses. In [7], an incremental certainty methodology regarding the extraction of facial features over real video frames is explained. The proposed procedure adapts to large varieties of subject appearances, including frame-to-frame changes within video sequences. The framework identifies the zones of the

face that are measurably exceptional and assembles an initial set of regions that are expected to incorporate data about the features of interest. In this methodology, core facial features, for example the eyes and the mouth, are in effect reliably identified. In [6], the authors try to recognize the eyes and mouth utilizing the separation vector field that is structured by atributing a vector to every pixel indicating its nearest edge. Separation vector fields are based on geometrical structure, and consequently can help in evading illumination issues in the location of the eyes and mouth areas. In [9], the authors demonstrated that the eyes and mouth in facial pictures can be robustly identified. They used their locations to normalize the pictures, assuming affine transformation, which can make up for different viewpoints. In [10], real-time face detection algorithm for searching faces, eyes and lips in pictures and videos is explained. The calculation builds upon the extraction of skin pixels based on rules derived from a straightforward quadratic polynomial model in a normalized color space. In [8], the authors separated the facial feature extraction into three core steps. The initial step is preprocessing. The objective of this step is to get rid of high intensity noise and to binarize the input picture. The second step incorporates a labeling procedure and an aggregation procedure. This step tries to create facial feature candidates block by block. Finally, a geometrical face model is utilized to detect the face position.

As can be seen, learning-free methodologies have apealing characteristics. Nonetheless, they present a few deficiencies. Firstly, the majority of them makes the assumption that a few conditions are met (for instance, that face pictures are taken in controled conditions and in an upright orientation). Furthermore, they usually depend on the discovery of few facial features (primarily the eyes and the mouth). Almost no consideration is given to the assembly of an extensive collection of facial features. Thirdly, accurate localization of the detected face features is still faulty.

Learning-based methodologies, on the other hand, aim to overcome these deficiencies. Three subcategories can be identified: parameterized appearance models, part-based deformable models and discriminative methodologies.

Parameterized appearance models generate a statistical model of shape and appearance from a collection of manually marked data [11–15]. In the 2D data domain, Active Shape Models (ASM) [11, 16], Active Appearance Models (AAM) [13, 14] and more recently, Active Orientation Models (AOM) [15] have been proposed. The ASM methodology generates 2D shape models and relies on motion constraints in conjunction with some image data from the

regions near the 2D shape landmarks to find features on new pictures. The AAM uses both the shape and the texture [13,14]. The AOM approach [15] takes is similar to AAM, differing in the utilization of gradient orientations rather than the texture and an enhanced cost function, which generalizes better to unknown faces. In the 3D data domain, 3D morphable models (3DMMs) have been proposed [12,17], which incorporate the 3D shape and texture models, assembled from 3D scans.

Part-based deformable models maximize the posterior likelihood of part areas given a picture, so as to adjust the learned model [23–26]. Recently, the Constrained Local Model (CLM) methodology has attracted interest since it bypasses a large number of the disadvantages of AAM, for example, demonstrating robustness to lighting changes. CLM utilizes a set of several local detectors combined with a statistical shape model, amplifying the ASM approach. It achieves remarkable fitting results with unseen images [24]. In [25] a part based ASM and a semi-automatic refinement calculation are proposed, which results in more adaptability for facial pictures with large variation. In [26], a globally optimized tree shape model was introduced, which discovers facial points of interest as well as estimates the pose and the face image region, unlike the mentioned approaches, which all depend on a preparatory face localization stage [27] and do not assess the head posture from 2D picture information. In [28] a hybrid discriminative and part-based methodology is proposed enhancing the outcomes achieved by [24,26] in the location of feature points.

Finally, discriminative methodologies build a correspondence between image features and motion parameters or feature point positions [18–21]. Facial landmark detectors usually apply a sliding window-based scanning throughout various regions in facial images [18]. Nonetheless, this is a time-consuming procedure, as the scanning time increases proportionally with the size of the search zone. In recent times, various methodologies have been proposed that aim at alleviating this, by utilizing local image information and regression-based techniques applied to the ASM approach [19–21], obtaining state-of-the-art performance in the field of 2D facial feature detection. In [22] discriminative techniques and parameterized appearance models are bound together through the proposed Supervised Descent Method (SDM) for solving Non-direct Least Squares problems, obtaining significantly quick and precise fitting results.

On the other hand, Online Appearance Models (OAM) [35,36] permit a more effective person-specific face tracking without the need for an earlier training stage. They compute a quick 3D head posture estimation and facial

action extraction with sufficient precision for an extensive variety of uses – for example, live facial puppetry, facial expression recognition, and face recognition. Nonetheless, this methodology demands an initial head posture estimation in the first frame so that the person-specific texture can be learned and subsequently updated. In [32] a holistic technique for the simultaneous estimation of two sorts of parameters (3D head pose and person-specific shape parameters that are consistent for a given subject) from a single picture is proposed, utilizing just a statistical facial texture model and a generic deformable 3D model. One advantage of the proposed fitting methodology is that it does not require a precise parameter initialization. Nevertheless, this methodology needs a training stage, with the same disadvantages as in the case of statistical shape and appearance models.

In this work, we propose a learning-free approach for identifying facial features, which can overcome the majority of the inadequacies specified previously. The proposed system can preserve the positive aspects of both learning-free and learning-based methodologies. Specifically, the advantages of learning-based methodologies (i.e., rich sets of facial features, accurate and real-time estimation) are preserved in our proposed methodology. Additionally, the proposed methodology will have the two advantages that are connected with learning-free approaches[1]. To start with, there is no learning stage. Second, unlike numerous learning methodologies whose execution can degrade if imaging conditions change, our proposed methodology is training free and subsequently free from the influence of training conditions. Our proposed methodology has two primary parts. The initial step is the recognition of fiducial facial features using smoothed gradient maps and some prior knowledge about face geometry. The second part is the 3D fitting of a deformable 3D model to the detected feature points. In this step, a 3D fitting method is designed for extracting the 3D pose and its deformable parameters (facial actions and shape variations) at the same time. A result of this fitting is that additional facial features can be acquired by basically projecting the 3D vertices of the adjusted 3D model onto the picture. The deformable model used is a generic model with a set of parameters permitting a 3D fitting to different people and to diverse facial actions. In this way, we can estimate the position, orientation, shape and facial actions, and initialize person-specific face tracking procedures, such as OAM, with higher accuracy than state-of-the-art approaches, under difficult illumination conditions, and sufficiently low processing power requirements as to permit its execution in

[1]These are clearly favorable features if the framework is to be utilized on portable equipment, for example PDAs and tablets.

gadgets with lesser capabilities, such as cell phones and tablets. The use of a generic 3D deformable model is vital for having a efficient and adaptable fitting system.

This chapter is organized as follows. Section 2 explains the proposed learning-free approach for detecting facial features from an image. Section 3 describes the proposed approach to fit the deformable 3D facial shape to the detected 2D features. Section 4 presents the obtained experimental results compared to state-of-the-art techniques. Finally, in section 5, these results and future work are discussed. In addition, appendix A explains the 3D deformable face model used in this work.

LIGHTWEIGHT FACIAL FEATURE DETECTION

Our methodology for fitting 3D generic face models comprises two stages: (1) detect facial features on the picture and (2) adjust the deformable 3D face model such that the projection of a set of key vertices onto the 2D plane of the picture matches the positions of the corresponding facial features. In this work we consider perspectives in which both eyes can be seen, regardless of the possibility that they are occluded, for instance, by eyeglasses. The proposed methodology requires an initial step of face detection, which, depending on the methodology taken, may require a facial training stage, for example, [38,39]. We can likewise apply the same detection methods (i.e., [38, 39]) for locating facial parts, for example, the eyes, nose and mouth, although we do not consider their identification as a *strict* prerequisite because we also include low resolution facial pictures or partially occluded ones, which would prevent the detectors to discover the features appropriately.

The entire fitting methodology, step by step, is shown in Fig. 1 and algorithm 1, where the term *ROI* alludes to a *region of interest* (the sought region) and *SROI* to a *search ROI*. Depending on whether they have already been detected by the corresponding object detector or not, the input data related to the eyes, nose and mouth can be either *ROI* or *SROI*, as specified previously. Algorithm 1 attempts to identify 32 facial features in the input monocular image (Fig. 2). These 32 features correspond to a subset of vertices in the Candide-3m model (index A). Their 2D positions are settled inside their corresponding regions considering the size of the areas and the in-plane face rotation (sideways head tilt, or roll angle). This way, by finding the *ROI* of a face part and the roll angle, the 2D points of that face part will be quickly and automatically located. This methodology is adequate to initialize an OAM tracker, for example [36], to fit the 3D model frame-by-frame with the

correlation between the model and the face pictures. This is particularly the case of contour points, which help in the initialization despite not matching with real landmarks, and thus cannot be located with high confidence on a face picture even by expert observers. When a face region has been found on a picture (e.g., utilizing [38,39]), each of the 32 point positions are calculated, even if some are occluded.

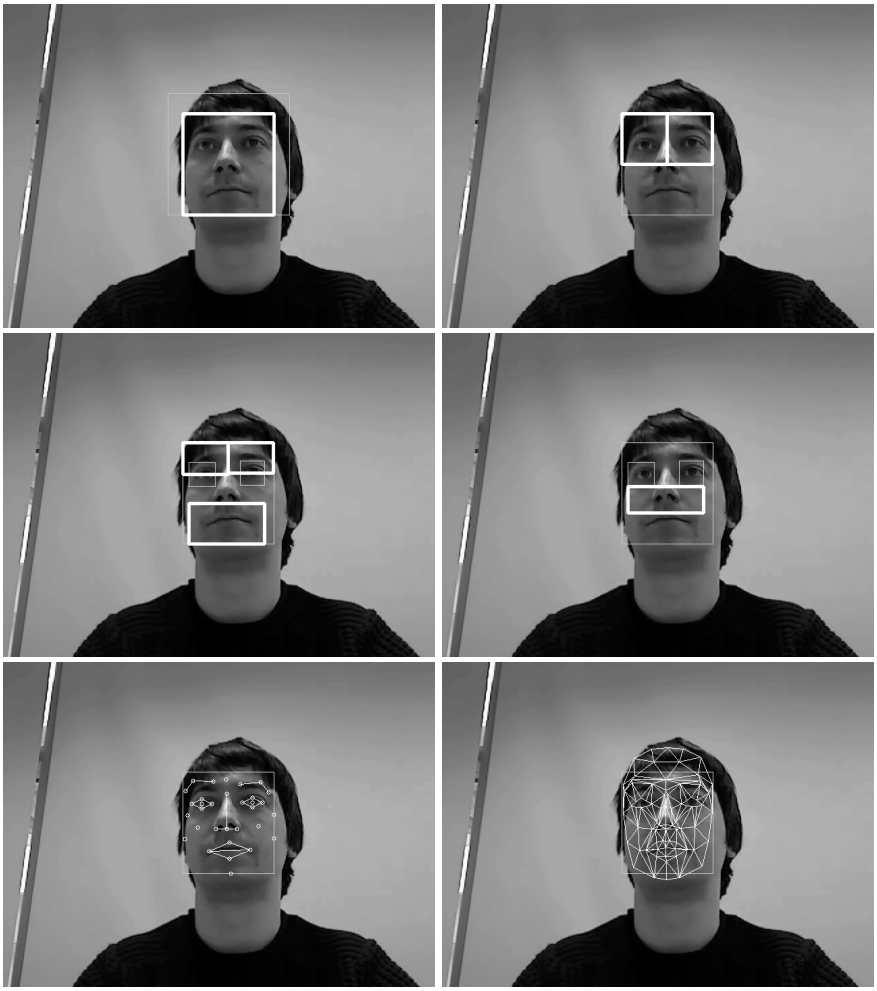

Figure 1: Proposed fitting approach. From left to right and top to bottom: (1) The detected face region and the *faceROI* derived from it (thicker line), (2) *faceROI* and the *eyeSROIs* derived from it (thicker line), (3) *faceROI*, the estimated *eyeROIs* and the *eyebrowSROIs* and *mouthSROI* derived from them (thicker lines), (4) *faceROI*, the estimated *eyeROIs* and the *noseSROI* derived from them (thicker line), (5) the detected facial features and (6) the fitted 3D face model projection.

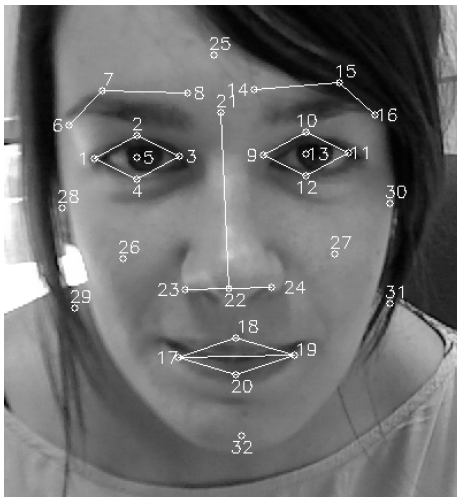

Figure 2: The 32 detected facial points. Note that the words *left* and *right* are relative to the observer rather than the subject.

Algorithm 1 Lightweight facial feature detection algorithm

procedure FACIALPOINTDETECTION(*faceROI*, *lEye(S)ROI*, *rEye(S)ROI*, *nose(S)ROI*, *mouth(S)ROI*, *peakValX*, *peakValY*, *binThresh*)
  for each eye **do**
    if ¬ *eyeROI* **then**
      eyeROI ← ROIBOUNDDETECTION(*eyeSROI*, *peakValX*, *peakValY*)
    end if
  end for
  θ ← Estimate *roll* rotation angle derived from *eyeROIs*
  eyePoints ← Estimate eye point positions in a fixed way derived from (*eyeROIs* and θ)
  for each eyebrow **do**
    rotEyebrowSROI ← Get the eyebrow search region derived from (*faceROI* and *eyeROI*) and rotate it $(-\theta)$
    rotEyebrowROI ← ROIBOUNDDETECTION(*rotEyebrowSROI*, NOT_USED, *peakValY*)
    eyebrowPoints ← Estimate eyebrow point positions in a fixed way derived from *rotEyebrowROI* and apply θ rotation and transform to global image coordinates
  end for
  for mouth and nose **do**
    if ¬ *partROI* **then**
      rotPartSROI ← Rotate *partSROI* $(-\theta)$
      rotPartROI ← ROIBOUNDDETECTION(*rotPartSROI*, *peakValX*, *peakValY*)
    else
      rotPartROI ← Rotate *partROI* $(-\theta)$
    end if
    partPoints ← Estimate part point positions in a fixed way derived from *rotPartROI* and apply θ rotation and transform to global image coordinates
  end for
  contourPoints ← CONTOURPOINTDETECTION(*faceROI*, *eyeCenters*, *lEyeLCorner*, *rEyeRCorner*, *mouthCorners*, *binThresh*)
  return (*eyePoints* and *eyebrowPoints* and *mouthPoints* and *nosePoints* and *contourPoints*)
end procedure

The search regions are estimated from the detected face and eye regions (Fig. 1). In case external detectors were not used to find the *eyeROIs* (i.e., they have not been input to algorithm 1), we apply algorithms 2, 3 and 4 to estimate their boundaries[2]. Next, the eye point positions and the face projection *roll* angle θ are set, derived in a proportional and fixed way from the geometry of those *ROIs*. Particularly, the *eyeROI* centers correspond to eye center positions, the eye widths and heights are the same in both sides relying on the mean *ROI* sizes, where θ is measured, and the remaining eye points are located around the centers. Using face detectors, there is a limited *roll* angle range, and subsequently the eyes have well-defined search regions.

The corresponding ROI boundaries of eyebrows, mouth and nose are used as a reference, also in a fixed way, for the estimation of their corresponding facial features. Algorithms 2, 3 and 4 are also used to obtain these boundaries, taking into account the influence of the *roll* angle θ. In the specific case of eyebrows, we do not calculate the boundaries in the X direction, but fix them according to the search region width and the expected eyebrow geometry in the 3D model, as some people have bangs occluding them, or even lack eyebrows altogether. The parameters $peakValX$ and $peakValY$ are thresholds for detecting the horizontal and vertical boundaries in the normalized gradient maps. In our experiments we use $peakValX = 20$ and $peakValY = 50$ in all cases.

We can reduce the influence of directional illumination by applying the double sigmoidal filtering applied to the search regions (algorithm 2), while the candidate edges are accentuated through the squared sigmoidal gradient calculation, which considers only the edge strength and neglects the edge direction information [40]. The contour point positions are estimated in a fixed way too, relying on the eye and mouth positions. Algorithm 5 returns 8 contour points: the forehead center, the left and right cheeks, the 4 facial corners and the chin bottom point. Although none of them are fiducial points, they are useful for 3D model fitting and tracking. In the case of the facial side corner estimation, we analyze the image region that goes from the facial region boundary to its corresponding mouth corner, assuming that in that region a noticeable X gradient appears only in one of the sides, when the subject exhibits a non-frontal pose, corresponding to the face side boundary. The squared sigmoidal gradient in X is calculated, assuming that those side points lie on it. Then, these side points allow us to better estimate the *pitch* angle of the face. Nonetheless, it can occur that both sides have a noticeable

[2]Note that the ROI boundaries of the eyebrows, nose and mouth are also estimated through algorithms 2, 3 and 4. Algorithm 2 invokes both algorithms 3 and 4.

gradient in X, such as in the case of the existence of other features such as local shadows or a beard. In order to circumvent these conditions, we assume that the side that should have the gradient applied to estimate the X positions is the one in which the mean positions are closer to the face region boundary, while for the other side the X positions correspond to those of the boundary itself. The parameter *binThresh* is the binarization threshold for the normalized gradient map in X. In our experiments we use $binThresh = 150$.

Algorithm 2 ROI boundary detection algorithm

procedure ROIBOUNDDETECTION(*SROI*, *peakValX*, *peakValY*)
  dsSROI ← Apply double sigmoidal filter to *SROI*
  ssySROI ← Apply squared sigmoidal Y gradient to *dsSROI*
  (*bottomY* and *topY*) ← YBOUNDDETECTION(*ssySROI*, *peakValY*) ▷ (ALGORITHM 3)
  (*leftX* and *rightX*) ← XBOUNDDETECTION(*ssySROI*, *peakValX*, *bottomY*, *topY*) ▷ (ALGORITHM 4)
  return (*leftX* and *rightX* and *bottomY* and *topY*)
end procedure

Algorithm 3 ROI Y boundary detection algorithm

procedure YBOUNDDETECTION(*ssySROI*, *peakValY*)
  for each *row* in *ssySROI* **do**
    $w \leftarrow (ssySROI_{height}/2 - |ssySROI_{height}/2 - y|) \cdot peakValY$
    $wVertProj_{row} \leftarrow (w \cdot \sum_{x=1}^{width} ssySROI_x)$
  end for
  Normalize *wVertProj* values from 0 to 100
  maxLowY ← Locate the local maximum above *peakValY* with the lowest position in *wVertProj*
  $topY \leftarrow (maxLowY + ssySROI_{height}/4)$
  $bottomY \leftarrow (maxLowY - ssySROI_{height}/4)$
  return (*bottomY* and *topY*)
end procedure

Algorithm 4 ROI X boundary detection algorithm

procedure XBOUNDDETECTION($ssySROI$, $bottomY$, $topY$, $peakValX$)
  for each col in $ssySROI$ **do**
    $w \leftarrow (ssySROI_{width}/2 - |ssySROI_{width}/2 - x|) \cdot peakValX$
    $wHorProj_{col} \leftarrow (w \cdot \sum_{y=bottomY}^{topY} ssySROI_y)$
  end for
  Normalize $wHorProj$ values from 0 to 100
  ($leftX$ and $rightX$) $\leftarrow$ Locate the first value above $peakValX$ starting from the left and right sides in $wHorProj$
  return ($leftX$ and $rightX$)
end procedure

Algorithm 5 Contour feature detection algorithm

procedure CONTOURPOINTDETECTION($faceROI$, $eyeCenters$, $lEyeLCorner$, $rEyeRCorner$, $mouthCorners$, $binThresh$)
  $faceVector \leftarrow (lEyeCenter + rEyeCenter - mouthLCorner - mouthRCorner)/2$
  $foreheadCenter \leftarrow (lEyeCenter + rEyeCenter + faceVector)/2$
  $lCheek \leftarrow (lEyeLCorner + lEyeCenter - faceVector)/2$
  $rCheek \leftarrow (rEyeRCorner + rEyeCenter - faceVector)/2$
  $ssxFaceROI \leftarrow$ Apply squared sigmoidal X gradient to $faceROI$ and normalize between 0 and 255
  for each facial side **do**
    $ssxFacialCornerROI \leftarrow$ Get region between $mouthCorner$ and $faceROI$ outer boundary
    $binFacialCornerROI \leftarrow$ Binarize $ssxFacialCornerROI$ with $binThresh$ and remove clusters (obtained through [41]) with $area < 0.8 \cdot ssxFacialCornerROI_{height}$
    $facialUCorner_y \leftarrow 0.75 \cdot ssxFacialCornerROI_{height}$
    $facialUCorner_x \leftarrow$ Get X centroid of white pixels at $facialUCorner_y$ in $binFacialCornerROI$
    $facialLCorner_y \leftarrow 0.25 \cdot ssxFacialCornerROI_{height}$
    $facialUCorner_x \leftarrow$ Get X centroid of white pixels at $facialUCorner_y$ in $binFacialCornerROI$
    $facialCorners \leftarrow$ Transform to global image coordinates
  end for
  $facialCorners \leftarrow$ Check which side from $facialCorners$ mean X position is further from its corresponding face region boundary, and then set their X positions in the boundary
  $chinBottom \leftarrow$ Calculate the intersection between the bottom of $faceROI$ and the line traced by $faceVector$
  return ($foreheadCenter$ and $lCheek$ and $rCheek$ and $facialCorners$ and $chinBottom$)
end procedure

DEFORMABLE MODEL BACKPROJECTION

The next stage is to determine the position, orientation, shape units (SUs) and animation units (AUs) (appendix A) which best fit the 32 detected facial features. In order to make the face model fitting more efficient, we use the existing correspondence between the 2D facial features and the 3D model points. The 3D generic model is given by the 3D coordinates of its vertices

$\mathbf{P}_i$, $i = 1, ..., n$, where n is the number of vertices. This way, the shape, up to a global scale, can be fully described by a $3n$-vector $\mathbf{g}$, the concatenation of the 3D coordinates of all vertices (Eq. 1), where $\overline{\mathbf{g}}$ is the standard shape of the model, the columns of $\mathbf{S}$ and $\mathbf{A}$ are the shape and animation units, and $\tau_s \in \mathbb{R}^m$ and $\tau_a \in \mathbb{R}^k$, are the shape and animation control vectors, respectively.

The 3D generic model configuration is given by the 3D face pose parameters (rotations and translations in the three axes) and the shape and animation control vectors, τ_s and τ_a. These define the parameter vector $\mathbf{b}$ (Eq. 2).

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{S}\tau_s + \mathbf{A}\tau_a \tag{1}$$

$$\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \tau_s, \tau_a]^T \tag{2}$$

Inter-person parameters, such as the eye width and the eye separation distance, can be controled through shape units (see appendix A). The term $\mathbf{S}\tau_s$ accounts for the shape or inter-person variability, while the term $\mathbf{A}\tau_a$ accounts for the facial or intra-person animation. Thus, in theory, the shape units would remain constant for face tracking, while the animation units could vary. Nevertheless, as they are meant to fit any kind of human face, it is challenging to perfectly separate both kinds of parameters, because the neutral facial expression can be significantly different from person to person. Hence, we have to take into account both the shape and animation units in our initialization process, without an explicit distinction between them. After the initialization we can assume that the shape units remain constant. Moreover, in order to reduce the computational burden we consider a subset of the animation units [36].

The 3D shape in Eq. 1 is expressed in a local coordinate system, but this should be related to the 2D image coordinate system. Thus, we adopt the *weak perspective* projection model. The perspective effects can be neglected since the depth variation of the face is small, when compared to its absolute depth from the camera viewpoint. The mapping between the image and the 3D face model is given by a 2×4 matrix $\mathbf{M}$, encapsulating both the camera parameters and the 3D face pose. Therefore, as defined in Eq. 3, a 3D vertex $\mathbf{P}_i = [X_i, Y_i, Z_i]^T \subset \mathbf{g}$ will be projected onto the image point $\mathbf{p}_i = [u_i, v_i]^T \subset \mathbf{I}$

(where **I** refers to the image).

$$\mathbf{p}_i = [u_i, v_i]^T = \mathbf{M}[X_i, Y_i, Z_i, 1]^T \tag{3}$$

The projection matrix **M** is given by Eq. 4, where α_u and α_v are the camera focal length expressed in vertical and horizontal pixels, respectively. (u_c, v_c) represent the principal point 2D coordinates, s is a global scale and $\mathbf{r_1^T}$ and $\mathbf{r_2^T}$ are the first two rows of the 3D rotation matrix.

$$\mathbf{M} = \begin{bmatrix} \frac{\alpha_u}{t_z}\, s\, \mathbf{r_1^T} & \alpha_u \frac{t_x}{t_z} + u_c \\ \frac{\alpha_v}{t_z}\, s\, \mathbf{r_2^T} & \alpha_v \frac{t_y}{t_z} + v_c \end{bmatrix} \tag{4}$$

The core idea of our approach is to estimate the 3D model parameters by minimizing the distances between the detected facial points ($\mathbf{d}_j = [x_j, y_j]^T \subset \mathbf{I}$, where $j = 1, ..., q$ and $q \leq n$) and their corresponding projected vertices from the 3D model. Algorithm 6 shows the procedure, called *deformable model backprojection*. The more points are detected on the image (32 with the proposed learning-free method), the more shape and animation units can vary in the model. The minimal requirement is that the points to be matched must not be coplanar. This way, the objective is to minimize Eq. 5, where $\mathbf{p}_j$ is the 2D projection of the 3D point $\mathbf{P}_j$. Its 2D coordinates rely on the model parameters (encapsulated in **b**). These coordinates are obtained via equations 1 and 3. The weight elements w_j refer to confidence values ($0 \leq w_j \leq 1$) for their corresponding $\mathbf{d}_j$, and depend on the approach used for facial point detection. For our method (section 2), higher weights (e.g, 1) should correspond to eye points, mouth points, nose top and base points, and the forehead center point; in a second level (e.g., 0.8) the eyebrow points and the rest of contour points; and finally in a third level (e.g., 0.2) the left and right nostrils. In order to get an initial guess of the position and orientation of the face object, before the optimization procedure starts, the POS algorithm [3] is applied.

$$\mathbf{b}^* = arg \min_{\mathbf{b}} \sum_{j=1}^{q} w_j \cdot [(\{\mathbf{d}_j\}_x - \{\mathbf{p}_j(\mathbf{b})\}_x)^2 + (\{\mathbf{d}_j\}_y - \{\mathbf{p}_j(\mathbf{b})\}_y)^2] \tag{5}$$

The degrees of freedom of the Candide model (to be optimized) are initially normalized, so that their values are not biased towards any of them

[3] POS is a pose solver based on a linearization of the perspective projection equations, which corresponds to a single iteration of POSIT [42].

in particular. Empirically, we observed that it was recommendable to keep the translation estimated by POS constant because of the high sensitivity of the Levenberg-Marquardt (LM) algorithm to these global parameters. Therefore, we keep the position from POS constant, and optimize the rest of parameters.

Algorithm 6 Deformable model backprojection algorithm

1: **procedure** MODELBACKPROJECTION($\overline{\mathbf{g}}$, $\mathbf{w}$, $\mathbf{S}$, $\mathbf{A}$, $\mathbf{d}$)
2: (θ_x^0 and θ_y^0 and θ_z^0 and t_x^0 and t_y^0 and t_z^0) $\leftarrow$ Apply POS algorithm [42] to $\overline{\mathbf{g}}$ with $\mathbf{d}$
3: $\mathbf{b} \leftarrow$ Starting from (θ_x^0 and θ_y^0 and θ_z^0 and t_x^0 and t_y^0 and t_z^0 and $\tau_s = 0$ and $\tau_a = 0$) minimize Eq. 5 through the Levenberg-Marquardt algorithm [43], taking into account equations 1 and 3 for the update in the iterative optimization process. The position is kept constant ($t_x = t_x^0, t_y = t_y^0, t_z = t_z^0$).
4: **return b**
5: **end procedure**

EXPERIMENTAL RESULTS AND DISCUSSION

We have used the CMU Pose, Illumination, and Expression (PIE) database [44], in order to evaluate the suitability of our approach for the initialization of an OAM-based 3D face tracking. We have used the images where the flash system was activated, in order to get challenging illumination conditions while subjects maintained a neutral facial expression. In our context, in which we expect to fit the face model for a posterior OAM-based tracking, we can assume that in the first frame the person will have the mouth closed, which is valid for many applications. For this experiment we have selected the images where the subject has frontal or near-frontal views. In total, we have used 7134 images for the test (68 subjects $\times$ 5 cameras $\times$ 21 flashlights $-$ 6 missing images in the database). We created the ground truth by manually configuring the Candide-3m model on each of the faces, then applied the automatic fitting approach (described in sections 2 and 3) and measured the fitting error with respect to the ground truth as a percentage, in the same way as [18,19]. This is described by Eq. 6, where $\mathbf{p}_i^{\text{fit}}$ and $\mathbf{p}_i^{\text{gt}}$ correspond to the fitted and ground truth projections of point i respectively, and $\mathbf{l}^{\text{gt}}$ and $\mathbf{r}^{\text{gt}}$ to the ground truth left and right eye center projections. If no face region was detected, or one was incorrectly detected, we excluded that image from the evaluation. All vertices of Candide-3m are used for computing the fitting

error.

$$e = \frac{\sum_{i=1}^{n} \|\mathbf{p}_i^{\text{fit}} - \mathbf{p}_i^{\text{gt}}\| / n}{\|\mathbf{l}^{\text{gt}} - \mathbf{r}^{\text{gt}}\|} \cdot 100 \tag{6}$$

Six alternatives are compared in the test: (1) *HA* (Holistic Approach) [32], (2) *CLM* (Constrained Local Model) [24] with head orientation obtained by [29] [4], (3) *SDM* (Supervised Descent Method) [22] with head orientation obtained by [42], (4) *FFBP* (Facial Feature Backprojection), our approach combining both the proposed facial feature detector and the backprojection, (5) *CLMBP*, the CLM approach but replacing its estimated orientation by our full backprojection approach and (6) *SDMBP* the SDM approach but with our full backprojection approach.

We used all the Candide-3m points in order to measure the fitting error, for all approaches. The used weights for the partial backprojection in *CLM* and *SDM* and the full backprojection in *CLMBP* and *SDMBP* are all equal to 1, except for the eyebrows and contours, which have 0.8. This challenging illumination test is unfavorable for the *HA* approach (fully appearance-based approach), as it relies on a PCA model obtained from a training stage. Hence, we train user-specific PCA models from the images in which we want to fit the face model, in order to obtain the best possible results from this approach. For the optimization a differential evolution strategy is adopted with an exponential crossover, a random-to-best vector to be perturbed, one difference vector for perturbation and the following parameter values: maximum number of iterations = 10, population size = 300, F = 0.85, and CR = 1. The random numbers are set to the range $[-0.5, 0.5]$.

We solve the same number of shape and animation units (12 SUs and 3 AUs) in all the methods, maintaining the rest of Candide-3m parameters to a value of 0. The considered SUs correspond to *Eyebrows Vertical Position*, *Eyes Vertical Position*, *Eyes Width*, *Eyes Height*, *Eyes Separation Distance*, *Nose Vertical Position*, *Mouth Vertical Position*, *Mouth Width*, *Eyebrow Width*, *Eyebrow Separation*, *Nose Width* and *Lip Thickness*, while the selected AUs correspond to *Brow Lowerer*, *Outer Left Brow Raiser* and *Outer Right Brow Raiser*. This way,

[4] The implementations of CLM (`https://github.com/kylemcdonald/FaceTracker`) and SDM (`http://www.humansensing.cs.cmu.edu/intraface`) also provide the head orientation, obtained through [29] for CLM and [42] for SDM. In these two methods, given the 2D points and the head orientation, we apply the rest of our backprojection approach to place the 3D object, i.e. we only adjust the head position and the facial deformations to the 2D detections, not the orientation. The orientation would be that of [29] and [42], respectively.

Table 1: Fitting error comparison obtained in the CMU PIE database illumination variation images.

	C05		C07		C09		C27		C29		GLOBAL	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
FFBP	16.02	7.28	12.48	5.84	16.83	7.52	13.57	6.34	15.93	8.58	14.93	7.35
CLMBP	11.55	9.74	8.52	5.12	10.96	7.18	8.73	6.07	11.49	9.72	10.23	7.87
SDMBP	9.13	3.76	8.24	3.05	9.06	4.87	8.23	2.83	9.24	3.63	8.78	3.72
CLM	18.29	8.97	13.44	5.11	12.27	6.82	11.32	5.80	12.11	9.57	13.44	7.82
SDM	9.79	4.10	10.18	3.44	8.03	4.99	7.25	2.67	10.05	4.24	9.05	4.14
HA	37.60	20.20	31.06	16.40	30.26	15.80	32.06	16.54	31.39	15.67	32.42	17.16

Table 2: Fitting errors of facial parts obtained with *FFBP* in the CMU PIE database illumination variation images.

	C05		C07		C09		C27		C29		GLOBAL	
	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
Eyes	8.62	6.65	7.56	5.46	8.85	6.91	7.14	5.52	8.06	8.67	8.03	6.75
Eyebrows	12.54	6.65	11.53	6.02	13.69	6.41	10.98	5.39	12.40	9.22	12.21	6.91
Nose	12.42	7.42	9.28	6.29	11.00	8.14	8.84	6.21	10.88	8.58	10.46	7.48
Mouth	12.75	10.19	9.97	8.02	11.93	9.40	10.10	9.11	11.02	10.58	11.13	9.54

the LM minimization in algorithm 6 attempts to simultaneously approximate 21 unknowns (3D pose and facial deformations).

The obtained results for the six considered alternatives are shown in Table 1. This comparison allows us to evaluate the not only the relative performance of our full approach (i.e., *FFBP*, which combines the feature detection and the deformable backprojection), but also the deformable backprojection itself (i.e., the approaches that include the suffix *BP*), with respect to other alternatives. The results we obtain with the full approach (*FFBP*) have less error than *HA* and have similar values to those of *CLM*, with the advantage of not being dependent on the quality of a trained model for the fitting. Moreover, this comparison also shows that our deformable backprojection approach improves the fitting quality (*CLMBP* vs *CLM* and *SDMBP* vs *SDM*). Next we will show that under a face tracking setting *FFBP* (with OAM) behaves better than *CLM* and is computationally less intensive, allowing its utilization in gadgets with lower computational capabilities.

Table 2 shows the fitting errors obtained with *FFBP* for the points corresponding to each facial part separately. It can be observed that the lowest errors correspond to the eyes. This was expected, since eye regions can be found in a specific area which usually presents significant gradient levels with similar patterns from face to face. This is in contrast to other facial regions such as the mouth.

We have also evaluated our approach in combination with OAM (*FFBP*-

Table 3: Average computation times (in ms) obtained with *FFBP-OAM* and *CLM* [24] on iPad 2.

	Initialization	Frame-to-Frame Tracking
FFBP-OAM	60	42
CLM [24]	250	88

OAM) in a tracking scenario using the camera of the iPad 2. We have only integrated in the device *FFBP-OAM* and *CLM* in its original form (i.e., with its own model, without transferring its tracked points and orientations to Candide-3m). The computation power required by *HA* was too high compared to the others and the code of *SDM* was implemented exclusively for desktop computers, which prevented us to integrate it in the device. In this test, the faces have severe occlusions at certain times, and they adopt different positions, orientations and expressions. We evaluate how the full system (initialization + tracking) behaves in these circumstances, where it has to (1) detect and fit the 3D model when a new face appears, (2) track the face while it is visible and (3) detect when the tracking is lost and reinitialize the tracking when a face becomes visible again. Fig. 3 shows how both approaches behave under severe occlusion. In this case, *CLM* does not detect the occlusion correctly, does not restart the face detection process until the face is visible again, and repeatedly fits the graphical model to neighboring regions not corresponding to the face. On the contrary, *FFBP-OAM* properly detects the occlusion time and stops tracking, and then restarts the tracking once the face is visible again. The metrics inherently available in model-based tracking approaches, such as OAM, to better evaluate the current observation's divergence from the reference model, present a clear advantage over other alternatives for this kind of situations.

Table 3 shows the computation times obtained in this test. *CLM* needs an average time of 250 ms for the initial fitting with a detected face region of about 200×200, whereas our approach needs an average time of about 60 ms. During the tracking stage, *CLM* needs an average of 88 ms whereas the OAM tracking [36] requires only about 42 ms to fit the model. Table 4 shows the computation times obtained for the proposed facial feature detection and model backprojection separately, on the iPad 2. Fig. 4 shows images of our full system running on an iPhone 5. These results demonstrate the better suitability of our approach when compared to other state-of-the-art alternatives for 3D deformable face model fitting.

Finally, we analyze the suitability of our approach for the estimation of facial actions (intra-person variability) in a video sequence in which a face

Table 4: Average computation times (in ms) obtained with *FFBP* in the facial feature detection and model backprojection stages on iPad 2.

	Facial Feature Detection	Model Backprojection
FFBP	22	38

performs exaggerated facial expressions. In this experiment, the observed face starts with a neutral face, which allows our full approach combined with OAM (*FFBP-OAM*) to be used. It is compared to other two alternatives that involve the use of our backprojection, applied to every frame of the sequence, and that can infer facial actions in the lower face region by estimating the positions of sufficient mouth contour points, i.e., *CLMBP* and *SDMBP*. We estimate 26 variables (6 pose parameters, 12 SUs and 8 AUs) in the Candide-3m mode with these three approaches. The considered SUs are those used in the test with the CMU database, while the AUs correspond to *Jaw Drop*, *Lip Stretcher*, *Brow Lowerer*, *Lip Corner Depressor*, *Outer Left Brow Raiser*, *Outer Right Brow Raiser*, *Left Eye Closed* and *Right Eye Closed*.

Figure 3: Comparison between *CLM* and *FFBP-OAM* on an iPad 2 under a severe occlusion.

Figure 4: The full system running on an iPhone 5 at 24 FPS.

Some samples of this comparison are shown in Fig. 5, while Fig. 6 shows the *Jaw Drop* AU and upper/lower lip distance variations. In the three cases, images with a resolution of 320×240 are used for processing, and the results are visualized in images of size 640×480. It can be seen how exaggerated AUs can be estimated from the sequence by the three alternatives. The trained CLM in *CLMBP* includes contour facial points, while the trained SDM from *SDMBP* does not, and therefore, when those contour points are well adjusted, the Candide-3m model adjusts better to the real contour of the person in the former. Nevertheless, the CLM was trained with limited mouth variations, and therefore, especially when the mouth is fully open, the point adjustment is not accurate around the mouth. Nevertheless, the AU variation is distinguishable with the three alternatives and therefore action activation moments can be detected with appropriate thresholds. The frame-to-frame transition in the case of *FFBP-OAM* is better suited for video sequences as it is much smoother than in the other two cases.

Figure 5: Comparison between *CLMBP*, *SDMBP* and *FFBP-OAM* in a video sequence with exaggerated facial expressions.

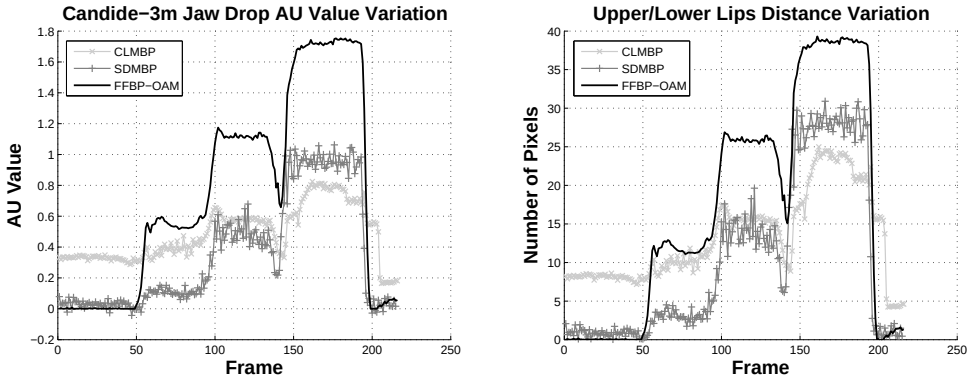

Figure 6: The *Jaw Drop* AU and upper/lower lip distance variations with *CLMBP*, *SDMBP* and *FFBP-OAM*.

CONCLUSIONS

In this work, we proposed a robust and lightweight procedure for the automatic fitting of 3D face models on facial pictures. Our approach is divided in two stages: (1) the detection of facial features on the picture and (2) the adjustment of the deformable 3D face model. The adjustment is performed by projecting its vertices into the 2D plane of the picture, and assessing the matching accuracy between the projected locations and the points of the detected facial features. For the first step, we propose a gradient analysis using filtered local image regions instead of popular techniques such as those based on statistical models of shape and appearance. This approach has the following benefits: (1) lower computational cost (2) non-reliance on a preliminary training stage, which avoidis the biased result provided by pre-trained statistical databases, (3) efficient matching as the 32 detection points are directly related to a subset of the generic 3D face model and (4) robust handling of challenging illumination. For the second step, we propose to use the detected facial points to estimate the 3D model configuration by minimizing the distances between the paired points across both point sets: those in the detected facial point set and their counterparts in the projected model. This approach assumes a camera with weak perspective, and uses a lightweight iterative approach to estimate the considered face model variations.

We have demonstrated the capability of our learning-free facial point detection and of our deformable backprojection approaches, by contrasting their performance with respect to state-of-the-art approaches. The challenging CMU PIE database was used for testing due to its illumination variation images. Similarly, videos captured with the camera of an iPad2 were used

to test tracking scenarios. Furthermore, we also have tested the integration of our method in low hardware capacity devices such as smartphones and tablets, with similar accuracy than in state-of-the-art methods but with an improved performance when compared to other recent alternatives.

Our proposed approach only needs an input snapshot image combined with the detected face features. Accordingly, it gets rid of tedious learning processes and also of the dependency on the related learning conditions. The current limitations of the proposed strategy are only related to the face pose. Although the method does not need a frontal face, the 3D orientation of the face should not be arbitrary. We estimate that the working ranges of the proposed method are around $(-20^\circ, +20^\circ)$ for the roll angle and around $(-30^\circ, +30^\circ)$ for the out-of-plane rotations.

Future work may concentrate on increasing the head orientation angle ranges, using lower image resolutions, and also the inclusion of other types of deformable objects apart from human faces. The management of the partial occlusions is another possible improovement for this method.

APPENDIX A: MODIFICATIONS TO CANDIDE-3

Our fundamental interest for this work is to fit a 3D generic face model on a facial picture under uncontrolled light conditions utilizing a computationally lightweight system, and avoiding previous learning phases. The objective of a computationally light processing is to permit the last application to run in gadgets with low computational capacities, for example, smart-phones and tablets. We are using an adaptation of Candide-3 [34] as a 3D generic face model. We call to this modification Candide-3m. This new Candide version is more simple is streamlined model so as to improve the fitting and tracking abilities of the original one.

The Candide-3m model has the following modifications with respect to Candide-3:

- Some vertices were removed from the geometry around the eyes to simplify the shape of the eyelids.
- The mesh around the eyes and mouth is more uniform. This is made tweaking the triangulation in those areas.
- The SUs have been changed to adapt them for the proposed initialization procedure:(1) *Cheeks Z*, *Chin Width* and *Eyes Vertical Difference* SUs have been removed, and (2) three more have been added, called *Eyebrow Width*, *Eyebrow Separation* and *Nose Width*.

- The AUs have been changed to increase the expressiveness of the tracking through an OAM approach such as [36]: (1) All MPEG-4 FAPs have been deleted, (2) the *Upper Lip Raiser*, *Lid Tightener*, *Nose Wrinkler*, *Lip Presser* and *Upper Lid Raiser* animation unit vectors (AUVs) have been deleted, and (3) the *Outer Brow Raiser* AUV has been split two different AUs (one for each eyebrow), and (4) the *Eyes Closed* AUV has been split in two different AUVs (one for each eye).

ACKNOWLEDGMENTS

We want to thank Nerea Aranjuelo and Victor Goni from Vicomtech-IK4 for their aid in the experimental work. We want to thank also Jason Saragih and Fernando De la Torre for their explanations about the implementations of their methods CLM and SDM for the experimental setup. Luís Paulo Santos is partially funded by the FCT (Portuguese Foundation for Science and Technology) within Project Scope UID/CEC/00319/2013. This work is partially supported by the Basque Government under the project S-PR13UN007. Finally, we want to thank Elsevier for allowing us to reuse material for this chapter from our following paper: Unzueta, L., Pimenta, W., Goenetxea, J., Santos, L.P. and Dornaika, F., "Efficient Generic Face Model Fitting to Images and Videos." Image and Vision Computing, 32(5), 321-334, 2014, doi:10.1016/j.imavis.2014.02.006.

CONFLICT OF INTEREST

We declare no conflict of interest regarding this publication.

References

[1] H. Kalbkhani, M. Shayesteh, and S. Mousavi, “Efficient algorithms for detection of face, eye and eye state,” *IET Computer Vision*, vol. 7, no. 3, pp. 184–200, 2013.

[2] M. Perreira, V. Courboulay, A. Prigent, and P. Estraillier, “Fast, low resource, head detection and tracking for interactive applications,” *PsychNology Journal*, vol. 7, no. 3, pp. 243–264, 2009.

[3] T. Hamada, K. Kato, and K. Kawakami, “Extracting facial features as in infants,” *Pattern Recognition Letters*, vol. 21, no. 5, pp. 407–412, 2000.

[4] K.-W. Wong, K.-M. Lam, and W.-C. Siu, "An efficient algorithm for human face detection and facial feature extraction under different conditions," *Pattern Recognition*, vol. 34, no. 10, pp. 1993–2004, 2001.

[5] X. Peng, M. Bennamoun, and A. Mian, "A training-free nose tip detection method from face range images," *Pattern Recognition*, vol. 44, no. 3, pp. 544–558, 2011.

[6] S. Asteriadis, N. Nikolaidis, and I. Pitas, "Facial feature detection using distance vector fields," *PR*, vol. 42, no. 7, pp. 1388–1398, 2009.

[7] G. Votsis, A. Drosopoulos, and S. Kollias, "A modular approach to facial feature segmentation on real sequences," *Signal Processing: Image Communication*, vol. 18, pp. 67–89, 2003.

[8] S. Jeng, H. Liao, C. Han, M. Chern, and Y. Liu, "Facial feature detection using geometrical face model: an efficient approach," *Pattern Recognition*, vol. 31, no. 3, pp. 273–282, 1998.

[9] D. Reisfeld and Y. Yeshurun, "Preprocessing of face images: detection of features and pose normalization," *Computer Vision and Image Understanding*, vol. 71, no. 3, pp. 413–430, 1998.

[10] C. Chiang, W. Tai, M. Yang, Y. Huang, and C. Huang, "A novel method for detecting lips, eyes and faces in real time," *Real-Time Imaging*, vol. 9, pp. 277–287, 2003.

[11] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, 1995.

[12] V. Blanz, P. Grother, P. Phillips, and T. Vetter, "Face recognition based on frontal views generated from non-frontal images," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 454–461.

[13] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "View-based active appearance models," *Image and Vision Computing*, vol. 20, no. 9–10, pp. 657–664, 2002.

[14] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, pp. 135–164, 2004.

[15] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic, "Generic active appearance models revisited," in *Proceedings of the Asian Conference on Computer Vision*, vol. LNCS 7726, 2013, pp. 650–663.

[16] T. Cootes and C. Taylor, "Active shape models - smart snakes," in *Proceedings of the British Machine Vision Conference*, 1992, pp. 266–275.

[17] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1–12, 2003.

[18] D. Vukadinovic and M. Pantic, "Fully automatic facial feature point detection using Gabor feature based boosted classifiers," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, 2005.

[19] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial point detection using boosted regression and graph models," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2729–2736.

[20] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2887–2894.

[21] B. Martinez, M. Valstar, X. Binefa, and M. Pantic, "Local evidence aggregation for regression based facial point detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1149–1163, 2013.

[22] X. Xiong and F. De la Torre, "Supervised descent method and its application to face alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[23] D. Cristinacce and T. Cootes, "Feature detection and tracking with constrained local models," in *Proceedings of the British Machine Vision Conference*, 2006, pp. 929–938.

[24] J. Saragih, S. Lucey, and J. Cohn, "Face alignment through subspace constrained mean-shifts," in *Proceedings of the International Conference of Computer Vision*, 2009.

[25] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. Huang, "Interactive facial feature localization," in *Proceedings of the IEEE European Conference on Computer Vision*, 2012, pp. 679–692.

[26] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.

[27] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," 2010.

[28] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[29] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2D+3D active appearance models," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2004, pp. 535–542.

[30] J. Sung, T. Kanade, and D. Kim, "Pose robust face tracking by combining active appearance models and cylinder head models," *International Journal of Computer Vision*, vol. 80, pp. 260–274, 2008.

[31] C. Chen and C. Wang, "3D active appearance model for aligning faces in 2D images," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3133–3139.

[32] F. Dornaika and B. Raducanu, "Simultaneous 3D face pose and person-specific shape estimation from a single image using a holistic approach," in *Proceedings of the Workshop on Applications of Computer Vision*, 2009, pp. 1–6.

[33] M. Zhou, Y. Wang, and X. Huang, "Real-time 3D face and facial action tracking using extended 2D+3D AAMs," in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2010, pp. 3963–3966.

[34] J. Ahlberg, "Candide-3 - an updated parameterized face," 2001.

[35] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[36] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1107–1124, 2006.

[37] ——, "Simultaneous facial action tracking and expression recognition in the presence of head motion," *International Journal of Computer Vision*, vol. 76, no. 3, pp. 257–281, 2008.

[38] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 511–518.

[39] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, 2002, pp. 900–903.

[40] M. Zhou, Y. Wang, X. Feng, and X. Wang, "A robust texture preprocessing for AAM," in *Proceedings of the International Conference on Computer Science and Software Engineering*, vol. 2, 2008, pp. 919–922.

[41] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[42] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, pp. 123–141, 1995.

[43] J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical Analysis, Lecture Notes in Mathematics 630*, G. A. Watson, Ed. Springer-Verlag, 1977, no. 18, pp. 105–116.

[44] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, 2003.