Efficient Generic Face Model Fitting to Images and Videos

Luis Unzueta[a,*], Waldir Pimenta[b], Jon Goenetxea[a], Luís Paulo Santos[b], Fadi Dornaika[c,d]

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

Abstract

In this paper we present a robust and lightweight method for the automatic fitting of deformable 3D face models on facial images. Popular fitting techniques such as those based on statistical models of shape and appearance require a training stage based on a set of facial images and their corresponding facial landmarks, which have to be manually labeled. Therefore, new images in which to fit the model cannot differ too much in shape and appearance (including illumination variation, facial hair, wrinkles, etc) from those used for training. By contrast, our approach can fit a generic face model in two steps: (1) the detection of facial features based on local image gradient analysis and (2) the backprojection of a deformable 3D face model through the optimization of its deformation parameters. The proposed approach can retain the advantages of both learning-free and learning-based approaches. Thus, we can estimate the position, orientation, shape and actions of faces, and initialize user-specific face tracking approaches, such as Online Appearance Models (OAM), which have shown to be more robust

*Corresponding author.
Email addresses: lunzueta@vicomtech.org (Luis Unzueta), wpimenta@di.uminho.pt (Waldir Pimenta), jgoenetxea@vicomtech.org (Jon Goenetxea), psantos@di.uminho.pt (Luís Paulo Santos), fadi.dornaika@ehu.es (Fadi Dornaika)

than generic user tracking approaches. Experimental results show that our method outperforms other fitting alternatives under challenging illumination conditions and with a computational cost that allows its implementation in devices with low hardware specifications, such as smartphones and tablets. Our proposed approach lends itself nicely to many frameworks addressing semantic inference in face images and videos.

Keywords: Face model fitting, Head pose estimation, Facial feature detection, Face tracking

---

1. Introduction

Generic face model fitting has received much attention in the last decade. Face model fitting can be seen as a basic component in many Human-Computer Interaction applications since it enables facial feature detection, head pose estimation, face tracking, face recognition, and facial expression recognition. In general terms, two main kinds of approaches have been proposed: (i) learning-free and (ii) learning-based. The latter need a training stage with several images to build the model, and therefore depend on the selection of images for a good fitting in unseen images.

Learning-free approaches rely heavily on some radiometric and geometric properties associated with face images. These approaches exploit generic knowledge about faces, which often includes the position, symmetry, and edge shape of facial organs. They can locate facial features through low-level techniques (e.g. gradients, filtering), usually focusing on detecting individual face features (irises, nose, lips, ...) [1, 2, 3, 4]. Most of the learning-free approaches do not provide a full set of extracted face features, in contrast with learning-based techniques.

For instance, in [5], the authors exploit a range face image in order to detect the nose tip for frontal and non frontal faces. In [6], the authors attempt to detect eyes and mouth using the distance vector field that is formed by assigning to each pixel a vector pointing to the closest edge. Distance vector fields employ geometrical information and thus can help avoiding illumination problems in the critical step of eye and mouth region detection. In [7], a gradual confidence approach concerning facial feature extraction over real video frames is presented. The proposed methodology copes with large variations in the appearance of diverse subjects, as well as of the same subject in various frames within real video sequences. The system extracts the areas of

the face that statistically seem to be outstanding and forms an initial set of regions that are likely to include information about the features of interest. In this approach, primary facial features, such as the eyes and the mouth, are being consistently located. In [8], the authors divided the face feature extraction into three main steps. The first step is preprocessing. The goal of this step is to get rid of high intensity noise and to transform the input image into a binary one. The second step includes a labeling process and a grouping process. This step tries to generate facial feature candidates block by block. Finally, a geometrical face model is used to locate the actual position of a face. In [9], the authors showed that the eyes and mouth in facial images can be robustly detected. They used these points to normalize the images, assuming affine transformation, which can compensate for various viewing positions. In [10], a real-time face detection algorithm for locating faces, eyes and lips in images and videos is presented. The algorithm starts from the extraction of skin pixels based upon rules derived from a simple quadratic polynomial model in a normalized color space.

As can be seen, learning-free approaches can be very appealing. However, they suffer from some shortcomings. Firstly, most of them assume that some conditions are met (e.g., face images are taken in laboratory conditions, upright faces, etc). Secondly, most of these approaches focus on the detection of few facial features (mainly the eyes and the mouth). Very little attention was made to the detection of a rich set of facial features. Thirdly, accurate localization of the detected face features is still questionable.

On the other hand, learning-based approaches attempt to overcome the above mentioned shortcomings. Three subcategories are proposed: parameterized appearance models, discriminative approaches, and part-based deformable models.

Parameterized appearance models build a statistical model of shape and appearance on a set of manually labeled data [11, 12, 13, 14, 15]. In the 2D data domain, Active Shape Model (ASM) [16, 11], Active Appearance Model (AAM) [13, 14] and more recently, Active Orientation Model (AOM) [15] have been proposed. The ASM approach builds 2D shape models and uses their constraints along with some information on the image content near the 2D shape landmarks to locate points on new images. The AAM builds both the shape and the full texture variations [13, 14]. The AOM approach [15] follows the scope of AAM, but using gradient orientations instead of the texture and an improved cost function, which generalizes better to unseen identities. In the 3D data domain, 3D morphable models (3DMM) have been

proposed [17, 12], which include the 3D shape and texture models, built from 3D scans of data.

Discriminative approaches learn a correspondence between image features and landmark positions or motion parameters [18, 19, 20, 21]. Typical facial feature point detectors apply a sliding window-based search in a region of interest of the face [18]. However, this is a slow process, as the search time increases linearly with the search area. More recently, there have been a number of approaches aimed at eliminating the need for an exhaustive sliding window-based search, by using local image information and regression-based techniques built over the ASM framework [19, 20, 21], achieving state-of-the-art performance in the problem of 2D facial feature detection. In [22] discriminative methods and parameterized appearance models are unified through the proposed Supervised Descent Method (SDM) for solving Non-linear Least Squares problems, obtaining extremely fast and accurate fitting results.

Finally, part-based deformable models maximize the posterior likelihood of part locations given an image, in order to align the learned model [23, 24, 25, 26]. In recent years, the Constrained Local Model (CLM) approach has attracted interest since it circumvents many of the drawbacks of AAM, such as modeling complexity and sensitivity to lighting changes. CLM uses an ensemble of local detectors in combination with a statistical shape model, extending the basic idea of ASM. It obtains remarkable fitting results with previously unseen faces [24]. In [25] a component-based ASM and an interactive refinement algorithm are proposed, which provides more flexibility for handling images with large variation. In [26], a globally optimized tree shape model was presented, which not only finds face landmarks but also estimates the pose and the face image region, not as the previously mentioned methods, which all require a preliminary face detection stage [27] and do not estimate the head pose from 2D image data. In [28] a hybrid discriminative and part-based approach is proposed improving the results obtained by [24, 26] in the task of landmark localization.

In order to extend the face landmark estimation from 2D to 3D (2.5D), different alternatives have been proposed. In [29] a non-rigid structure-from-motion algorithm is proposed to construct the corresponding 3D shape modes of a 2D AAM during the training stage, in order to estimate the head pose angles from the 2D fitting procedure. However, the used 3D shape bases account simultaneously for shape variability (inter-person variability) and facial action (intra-person variability), with the two kinds of variability being

thus represented as interdependent, making the explicit recovery of only the facial actions impossible.

Recently, some works focused on combining AAM principles with some prior knowledge about 3D face shape [30, 31, 32, 33]. In [31], the authors proposed a 3D AAM which estimates the classic shape and appearance parameters together with the 3D pose parameters. The mean 3D face model is simply obtained from the mean 2D shape by exploiting an offline 3D face template. In [33], the authors proposed to minimize the sum of two criteria: the classic AAM criterion and the point-to-point discrepancy between the Candide-3 model [34] vertices and the 2D shape.

On the other hand, approaches based on Online Appearance Model (OAM) [35, 36] allow a more efficient person-specific face tracking without the need of a prior training stage. For instance, [35, 36] obtains a fast 3D head pose estimation and facial action extraction with sufficient accuracy for a wide range of applications, such as live facial puppetry, facial expression recognition, face recognition, etc. However, this approach requires an estimate for head pose estimation for the first frame in the video sequence so that the person-specific texture can be learned and then updated during the tracking (i.e., parameter fitting for the rest of the video sequence). In [32] a holistic method for the simultaneous estimation of two types of parameters (3D head pose and person specific shape parameters that are constant for a given subject) from a single image is proposed, using only a statistical facial texture model and a standard deformable 3D model. One advantage of the proposed fitting approach is that it does not require an accurate parameter initialization. However, this approach also requires a training stage, similar to the one of statistical shape and appearance models, with identical drawbacks.

In this paper, we propose a learning-free approach for detecting facial features that can overcome most of the shortcomings mentioned above. The proposed framework can retain the advantages of both learning-free and learning-based approaches. In particular, the advantages of learning-based approaches (i.e., rich set of facial features, real-time detection, accurate localization). In addition to these, the proposed approach will have the two advanges that are associated with learning free approaches[1]. First, there is no tedious learning phase. Second, unlike many learning approaches whose

[1]These are obvious advantages if the system should be used on mobile devices such as smart phones and tablets.

performance can downgrade if imaging conditions change, our proposed approach is training free and hence independent of training conditions. Our proposed approach has two main components. The first component is the detection of fiducial facial points using smoothed gradient maps and some prior knowledge about face parts in a given face region. The second component is the 3D fitting of a deformable 3D model to the detected points. In this component, a 3D fitting scheme is devised for estimating the 3D pose of the face as well as its deformable parameters (facial actions and shape variations) simultaneously. A byproduct of this fitting is that another subset of facial features can be obtained by simply projecting the 3D vertices of the adapted 3D model onto the image. We stress the fact that the deformable model used is a generic model having generic parameters allowing a 3D fitting to different persons and to different facial actions. Thus, we can estimate the position, orientation, shape and actions of faces, and initialize user-specific face tracking approaches, such as OAM, with better precision than the state-of-the-art approaches, under challenging illumination conditions, and with a computational cost that allows its implementation in devices with low hardware specifications, such as smartphones and tablets. The exploitation of a generic 3D deformable model is crucial for having an efficient and flexible fitting method.

Our proposed approach lends itself nicely to many frameworks addressing semantic inference in face images and videos, such as face and facial feature tracking, face recognition, face gesture analysis, and pose invariant dynamic facial expression recognition [37], to mention a few.

This paper is organized as follows. Section 2 gives insight on how to detect facial features from an image at a low computational cost. Section 3 explains the method we propose to locate and deform the 3D face object in order to fit the detected facial features. Section 4 shows the experimental results we obtain compared to state-of-the-art alternatives. Finally, in section 5 we discuss the obtained results and the future work. Additionally, Appendix A explains the 3D deformable face model we use in this work.

2. Lightweight facial features detection

Our approach for fitting 3D generic face models consists in two steps: (1) the detection of facial features on the image and (2) the adjustment of the deformable 3D face model such that the projection of its vertices onto the 2D plane of the image matches the locations of the detected facial fea-

tures. The latter is explained in section 3. For the first step, one can also apply learning-based approaches mentioned in section 1, which provide with a rich set of facial features, to a monocular image or even to a combination of monocular image and its corresponding depth map in order to measure 3D data. However, as we want to make our approach applicable to any person with any appearance in uncontrolled environments, we prefer to avoid using the learning-based techniques as they require a training stage with a set of facial images which constrain their applicability. In this work we do not consider profile views but those in which the perspective includes both eyes, even if they are occluded, for example, by eyeglasses. The proposed approach requires a initial stage of face detection, which, depending on the approach taken, might also require a facial training stage, such as [38, 39]. Nevertheless, these face detection techniques do not constrain the search as much as the facial features detection methods would do under different illumination conditions, facial expressions and appearances; therefore we consider them acceptable for our purpose. Furthermore, these approaches have been proved to be robust and do not need any retraining. We can also apply the same detection techniques (i.e., [38, 39]) for localizing facial parts such as eyes, nose and mouth, but we do not consider their detection as a *strict* requirement because we also consider low resolution facial images or partially occluded ones, which would prevent the detectors to find them properly. However, we include these as potential reinforcing checks, since they can locate the facial parts with higher precision in more favorable circumstances.

Figures 1 and 2 and algorithm 1 show the whole fitting process step by step, where the term *ROI* refers to a *region of interest* (the sought region) and *SROI* to a *search ROI*. The input data related to eyes, nose and mouth can be *ROI* or *SROI*, depending on whether they have already been detected by the corresponding object detector or not, as mentioned above. Algorithm 1 aims to detect 32 facial points in any input image (Fig. 3). These 32 points form a subset of Candide-3m vertices (Appendix A). Their 2D positions are fixed within their corresponding regions taking into account the scale of the regions and the in-plane rotation of the face (roll angle). Thus, by finding the *ROI* of a face part as well as the roll angle, the 2D points of that face part will be automatically and rapidly estimated. This process is good enough to allow an OAM tracker such as [36] to fit the 3D model on subsequent frames with a visually alike correlation between the model and the face images (Fig. 4). This is especially the case of contour points, which help in the initialization but do not match with real landmarks, which cannot be determined with

high certainty on a face image even by trained observers. Once a face region has been located on an image (e.g., using [38, 39]), all 32 point positions are always estimated, even if they are not really visible on the image, due to occlusions.

Algorithm 1 Lightweight facial features 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*) ▷ (ALGORITHM 2)
    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*) ▷ (ALGORITHM 2)
    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*) ▷ (ALGORITHM 2)
    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*) ▷ (ALGORITHM 5)
  return (*eyePoints* and *eyebrowPoints* and *mouthPoints* and *nosePoints* and *contourPoints*)
end procedure

First, the eye points are estimated, then the eyebrows, then the mouth, then the nose and finally the contour points. The search regions are derived from the detected face and eye regions (Fig. 1). In case *eyeROIs* have not been detected by an external detector (i.e., they have not been input to

algorithm 1), algorithms 2, 3 and 4 are applied to estimate their boundaries[2]. Then, we determine the eye point positions and the face projection *roll* angle θ, derived in a proportional and fixed way from the geometry of those *ROIs*. Fig. 5 shows that the eye center positions correspond to those of the *eyeROI* centers, that eye widths and heights are equal in both sides with a proportion obtained from the mean *ROI* sizes, where θ is measured, and how the rest of eye points are located. As we rely on face detectors, the *roll* angle variation has a limited range, and therefore the eyes have well-defined search regions. Thanks to the eyes displaying approximate radial symmetry we do not need the *roll* estimation for their localization.

For the estimation of the facial features in eyebrows, mouth and nose, their corresponding ROI boundaries are used as reference, also in a fixed way. These boundaries are also obtained through algorithms 2, 3 and 4, taking into account the influence of the *roll* angle θ. In the specific case of eyebrows, as some people have bangs occluding them, or even no eyebrows, we do not calculate the boundaries in X direction, but fix them according to the search region width and the expected eyebrow geometry in the 3D model. The parameters $peakValX$ and $peakValY$ are thresholds for the normalized gradient maps for detecting the horizontal and vertical boundaries. In our experiments we use $peakValX = 20$ and $peakValY = 50$ in all cases.

The double sigmoidal filtering applied to the search regions (algorithm 2) allows us to reduce the influence of directional illumination, while the squared sigmoidal gradient calculation accentuates the likely edges, and neglects the edge direction information, and considers only the edge strength [40]. The estimation of the contour point positions is done in a fixed way too, taking into account the eye and mouth positions. Algorithm 5 returns 8 contour points: the forehead center, the left and right cheek, 4 facial corners and the chin bottom point. Even though none of these points are fiducial points, they are useful for 3D model fitting and tracking. In the case of the facial side corners estimation, the image region that goes from the facial region boundary to its corresponding mouth corner is analyzed, assuming that a noticeable X gradient appears in that region in one of the sides but not in the other, when the subject exhibits a non-frontal pose, which corresponds to the face side boundary (e.g., see Fig. 2-(5)). For this we calculate the

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

squared sigmoidal gradient in X, and assume that those side points lie on it. These side points subsequently allow us to better estimate the *pitch* angle of the face. However, there might be cases in which both sides have a noticeable gradient in X, which may correspond not only to the face side boundary but to other features such as beard, or local shadows. In order to filter these cases we assume that the side that should take into account the gradient to estimate the X positions is that in which the mean positions are closer to the face region boundary, while for the other side the X positions are those of the boundary itself (see Fig. 2). The parameter *binThresh* is the binarization threshold for the normalized gradient map in X. In our experiments we use $binThresh = 150$.

Algorithm 2 ROI boundaries 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 boundaries 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 boundaries 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*) ← Locate the first value above *peakValX* starting from the left and right sides in *wHorProj*
  return (*leftX* and *rightX*)
end procedure

Algorithm 5 Contour features 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 ← Apply squared sigmoidal X gradient to *faceROI* and normalize between 0 and 255
  for each facial side **do**
    ssxFacialCornerROI ← Get region between *mouthCorner* and *faceROI* outer boundary
    binFacialCornerROI ← 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 ← Transform to global image coordinates
  end for
  facialCorners ← 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 ← 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

3. Deformable model backprojection

Once facial features have been located on the image, the next stage is to determine which position, orientation, shape units (SUs) and animation units (AUs) (Appendix A) fit them the best possible. The detected 32 facial features form a subset of the Candide-3m face model. We use the existing correspondence between the 3D model points and the 2D facial features to make the face model fitting more efficient. The 3D face model is given by the 3D coordinates of its vertices $\mathbf{P}_i$, $i = 1, ..., n$, where n is the number of vertices. Thus, 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 configuration of the 3D generic model 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).

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

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

A shape unit provides a way to deform the 3D model in order to adapt inter-person parameters such as the eye width, the eye separation distance, etc. (see Appendix A). Thus, the term $\mathbf{S}\tau_s$ accounts for shape or inter-person variability, while the term $\mathbf{A}\tau_a$ accounts for the facial or intra-person animation. Hence, in theory, for face tracking, the shape units would remain constant, while the animation units could vary. However, it is challenging to separate perfectly both kinds of variability defining the generic face model such as they would fit any kind of human face. This is due to the neutral facial expression which are significantly different from person to person. Therefore, in our initialization process we have to take into account both the shape and animation units, without an explicit distinction between them. Only, after the initialization we can assume that during the tracking stage the shape units remain constant. Furthermore, we consider a subset of the animation units in order to reduce the computational load [36].

In Eq. 1, the 3D shape is expressed in a local coordinate system. However, one should relate the 3D coordinates to the 2D image coordinate system. To

this end, we adopt the *weak perspective* projection model. We neglect the perspective effects since the depth variation of the face can be considered small, when compared to its absolute depth from the camera. Therefore, the mapping between the 3D face model and the image is given by a 2×4 matrix $\mathbf{M}$, encapsulating both the 3D face pose and the camera parameters. Thus, 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 $\mathbf{I}$ refers to the image), as defined in Eq. 3.

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

The projection matrix $\mathbf{M}$ 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) denote the 2D coordinates of the principle point, $\mathbf{r_1^T}$ and $\mathbf{r_2^T}$ are the first two rows of the 3D rotation matrix, and s is a global scale (the Candide model is given up to a scale).

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

The core idea behind our approach for deformable 3D model fitting is to estimate the 3D model configuration 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 counterparts in the projected model. Algorithm 6 shows the proposed method, which we call *deformable model backprojection*, as we are inferring the configuration of a 3D deformable model from sparse data corresponding to one of its 2D projections. The more data we detect on the image (32 points with the method proposed in section 2), the more shape and animation units we will be able to vary in the model. The minimal condition to be ensured is that the points to be matched should not lie on the same plane. Thus, our objective is to minimize Eq. 5, where $\mathbf{p}_j$ is the 2D projection of the 3D point $\mathbf{P}_j$. Its 2D coordinates depend on the model parameters (encapsulated in $\mathbf{b}$). 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 $\mathbf{d}_j$. This confidence depends on the method used for the detection of points. For our approach (section 2), we recommend to set the higher weights (e.g, 1) 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. We

apply the POS algorithm [3], in order to get an initial guess of the position and orientation of the face object, before the optimization procedure starts.

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

The degrees of freedom from the Candide model (to be optimized) are set in a normalized framework, so that their variations are not biased towards any in particular. Empirically, we found out that it was better to keep constant the translation estimated by POS algorithm since the sensitivity of LM is very high to these global parameters. For this reason we keep constant the position obtained through POS and optimize the rest of parameters, which can better accomplish this requirement.

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**

4. Experimental results and discussion

In order to evaluate the suitability of our approach for the initialization of an OAM-based 3D face tracking, we have used the CMU Pose, Illumination, and Expression (PIE) database [44]. In our experiments we have focused on the images in which the flash system was activated in order to get challenging illumination conditions while subjects maintained a neutral

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

facial expression. This database also contains other images in which subjects show different expressions under uniform and sufficient ambient light, i.e., without the flash system activated. We have ignored them because we are more interested in the challenging situation of the variate illumination conditions. 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. Nevertheless, we cannot ignore the high illumination variability, normally present in real world situations. For our study we have selected the cameras in which the subject has frontal-like views, considering the neighboring top, bottom and side views (each separated by about 22.5°) with respect to the frontal view (Fig. 6). Besides, we have also ignored the images with no illumination at all, which do not make sense in our test. Finally, we have used in total 7134 images for the test (68 subjects $\times$ 5 cameras $\times$ 21 flashlights $-$ 6 missing images in the database). We manually configured the Candide-3m model on each of the faces as ground truth data, 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. In the case that no face region is detected or one is incorrectly detected within an image we exclude that image from the evaluation. Note that the fitting error is computed for all vertices of Candide-3m.

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

We have compared six alternatives 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

[4]The implementations of CLM (`https://github.com/kylemcdonald/FaceTracker`) and SDM (`http://www.humansensing.cs.cmu.edu/intraface`) also provide with 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.

obtained by [42], (4) *FFBP* (Facial Feature Backprojection), our approach combining both the proposed facial features 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.

For all approaches we measured the fitting error obtained using all the Candide-3m points, not only those that are detected. The weights we used 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. Note that this challenging illumination test is unfavorable for the *HA* approach (fully appearance-based approach), which relies on a PCA model obtained from a training stage with a database of facial images. It is affected by illumination variation at the same level of pose variation. Therefore, in order to obtain the best possible results from this approach, we train user-specific PCA models with all the images corresponding to the same subject from the images in which we want to fit the face model. The initial guess is done in the same way as in [32], by setting the face model in a neutral configuration, with a position and scale directly related to the size of the detected face region. For the optimization we adopt a differential evolution strategy with 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, CR = 1. We also limit the random numbers to the range $[-0.5, 0.5]$. We assume that the position is obtained correctly in the initial guess and exclude it from the optimization.

In all the methods we solve the same number of shape and animation units (12 SUs and 3 AUs), maintaining the rest of Candide-3m parameters to a value of 0. 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*. Thus, the LM minimization in algorithm 6 attempts to simultaneously estimate 21 unknowns (3D pose and facial deformations).

Table 1 shows the results obtained in the test for the six considered alternatives. Thus, through this comparison we can evaluate the performance of our full approach (i.e., *FFBP*, which combines the feature detection and the deformable backprojection), but also the deformable backprojection itself

Table 1: Fitting errors 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

(i.e., the approaches that include the suffix *BP*), with respect to other alternatives. As can be seen, 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. On the other hand, this comparison also shows that our deformable backprojection approach improves the fitting quality (*CLMBP* vs *CLM* and *SDMBP* vs *SDM*). Below we will demonstrate that under a face tracking setting *FFBP* (with OAM) behaves better than *CLM* and is computationally lighter allowing its utilization on mobile devices.

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. Fig. 7 shows examples of 3D face model fitting obtained with *FFBP*, with its fitting stages shown in Fig. 8.

Additionally, we have done an evaluation of our approach combined with OAM (*FFBP-OAM*) in a tracking scenario using the camera of the iPad 2. In this test 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*

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

was too high compared to the others and the code of *SDM* was implemented exclusively for desktop computers, which prevented us to integrate them in the device. In this comparison, the users faces have severe occlusions at certain times, and the faces show different positions, orientations and expressions. Thus, we evaluate how the full system (initialization + tracking) behaves in these circumstances, (here the initialization refers to the proposed detection of features and the backprojection), where it has to (1) detect and fit the 3D model when a new face appears in front of the camera, (2) it has to track the face over time while it is visible and (3) detect the tracking lost (the face is occluded) and reinitialize the detection and tracking when the face becomes visible again. Fig. 9 shows how both approaches behave under a severe occlusion. In this case, *CLM* does not detect the occlusion properly, it does not restart the face detection process until the face is visible again, and keeps fitting the graphical model to neighboring regions not corresponding to the face. On the contrary, *FFBP-OAM* detects properly the occlusion time and stops tracking and then resets the tracking again once the face is visible again. The metrics inherently available in model-based tracking approaches, such as OAM, in order to better evaluate the differences between the reference model and the current observation is a clear advantage over other alternatives for this kind of situations. The full sequence, which includes more occlusion moments, is available as supplementary material.

The computation times obtained in this test are shown in table 3. For the initial fitting *CLM* needs an average time of 250 ms, whereas our approach needs an average time of about 60 ms, both with a detected face region of about 200×200. Moreover, during the tracking stage *CLM* needs an average of 88 ms to fit the model whereas the OAM tracking [36] requires only about 42 ms. Table 4 shows the computation times obtained on iPad 2 for the proposed facial feature detection and model backprojection separately. Fig. 10 shows images of our full system running on an iPhone 5. These results prove the better suitability of our approach for 3D deformable face model fitting when compared to other state-of-the-art alternatives.

Finally, we include another test in which we analyze the suitability of our

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

approach for the estimation of facial actions (intra-person variability) in a video sequence in which a face performs exaggerated facial expressions. In this test, the observed face starts with a neutral face and therefore our full approach combined with OAM (*FFBP-OAM*) can be used. We compare it to other two alternatives that involve the use of our backprojection, applied to every frame of the sequence, and that can estimate the positions of sufficient mouth contour points to infer facial actions in the lower face region, i.e., *CLMBP* and *SDMBP*. Thus, with these three approaches we estimate 26 unknowns (6 pose parameters, 12 SUs and 8 AUs) in the Candide-3m model. Considered SUs are the same as those in the test with the CMU database, while 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*.

Fig. 11 shows some samples of this comparison, while Fig. 12 shows the *Jaw Drop* AU and upper/lower lips distance variations. The full sequence, in which the frame-to-frame transition can be better observed, is available as supplementary material. In the three cases, the processing is done in images of resolution 320x240, and the results are visualized in images of 640x480. It can be observed how the three alternatives can estimate exaggerated AUs from the sequence. The trained CLM in *CLMBP* includes contour facial points, while the trained SDM from *SDMBP* does not, and therefore the Candide-3m model adjusts better to the real contour of the person in the former, when those contour points are well adjusted. However, the CLM was trained with not so big mouth variations and therefore the point adjustment is not accurate around the mouth, especially when the mouth is fully open. In any case, with the three alternatives the AU variation is distinguishable and therefore action activation moments can be detected with appropriate thresholds. The frame-to-frame transition in the case of *FFBP-OAM* is much smoother than in the other two cases, and is therefore better suited for video sequences.

5. Conclusions

In this work, we proposed a robust and lightweight method for the automatic fitting of 3D face models on facial images. Our approach consists of two steps: (1) the detection of facial features on the image and (2) the adjustment of the deformable 3D face model so that the projection of its vertices into the 2D plane of the image matches the locations of the detected facial features. For the first step, instead of using popular techniques such as those based on statistical models of shape and appearance, we propose using a filtered local image region gradient analysis, which has the following advantages: (1) it is computationally lightweight, (2) it does not require a previous training stage with a database of faces and therefore it is not biased by this circumstance, (3) it is efficient as the 32 estimated points correspond directly to a subset of the generic 3D face model to be fitted and (4) it can cope with challenging illumination conditions. For the second step, the core idea is to estimate the 3D model configuration by minimizing the distances between the detected facial points on the image and their counterparts in the projected model, through the assumption of weak perspective and a lightweight iterative approach that estimates all the considered face model variations.

We have proved the potential of our learning-free facial point detection and of our deformable backprojection approaches, by comparing their capabilities with respect to state-of-the-art alternatives with the challenging CMU PIE database illumination variation images and also in a tracking scenario using the camera of the iPad 2, in combination with OAM. Furthermore, we also have shown the possibility of integrating our approach in devices with low hardware specifications, such as smartphones and tablets, with state-of-the-art accuracy and improved performance when compared to other recent alternatives.

Our proposed approach needs as input one snapshot together with a detected face region. Therefore, it dispenses with tedious learning processes as well as the dependency on the associated learning conditions. The current limitations of the proposed method are purely related to the face pose. Indeed, relying on a face detector and the Candide-3m model, although the method does not require a frontal face, the 3D orientation of the face should not be arbitrary. We expect that the working ranges of the proposed method are around (-20°, +20°) for the roll angle and around (-30°, +30°) for the out-of-plane rotations.

Future work may focus on the extension of our system's scope to wider

head orientation angle ranges, lower image resolutions and also to other types of deformable objects apart from human faces, as well as handling partial occlusions.

6. Acknowledgements

We want acknowledge Fernando De la Torre and Jason Saragih for their respective clarifications about the implementations of their methods SDM and CLM for the experimental setup. We also thank Nerea Aranjuelo and Victor Goni from Vicomtech-IK4 for their aid in the experimental work. Luis Paulo Santos is partially funded by the FCT (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011. This work is partially supported by the Basque Government under the project S-PR13UN007.

Appendix A. Modifications to Candide-3

Our main interest in this work is to fit a 3D generic face model on a facial image under uncontrolled illumination using a learning-free computationally lightweight method. The low computation requirement comes from the fact that we expect to allow the final application to run in devices with low hardware specifications, such as smartphones and tablets. The 3D generic face model we adopt is a modified version of Candide-3 [34]. We will refer to this as Candide-3m. The modifications consist primarily in simplifying and streamlining the model in order to enhance the fitting and tracking capabilities of the original model. Fig. A.13 shows the Candide-3m geometry compared to the original model. Fig. A.14 shows the added and modified shape units (SUs) and animation units (AUs) with respect to Candide-3.

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

- The geometry around the eyes has been simplified by removing the vertices that form the eyelids.
- The triangulation around the eyes and mouth has been tweaked, to make the mesh density more uniform in those areas and to fit the new vertex list.

- The SUs have been changed in order to make them more appropriate for the initialization procedure proposed in this work: (1) *Cheeks Z*, *Chin Width* and *Eyes Vertical Difference* SUs have been removed, maintaining the rest, and (2) three more have been added, called *Eyebrow Width*, *Eyebrow Separation* and *Nose Width*.

- The AUs have also been changed in order to allow a more expressive tracking through an OAM approach such as [36]: (1) All MPEG-4 FAPs have been removed, (2) the *Upper Lip Raiser*, *Lid Tightener*, *Nose Wrinkler*, *Lip Presser* and *Upper Lid Raiser* animation unit vectors (AUVs) have been removed, and (3) the *Outer Brow Raiser* AUV has been splitted in left and right AUVs, and (4) the *Eyes Closed* AUV has been split into left and right AUVs and reorganized according to the new vertex list.

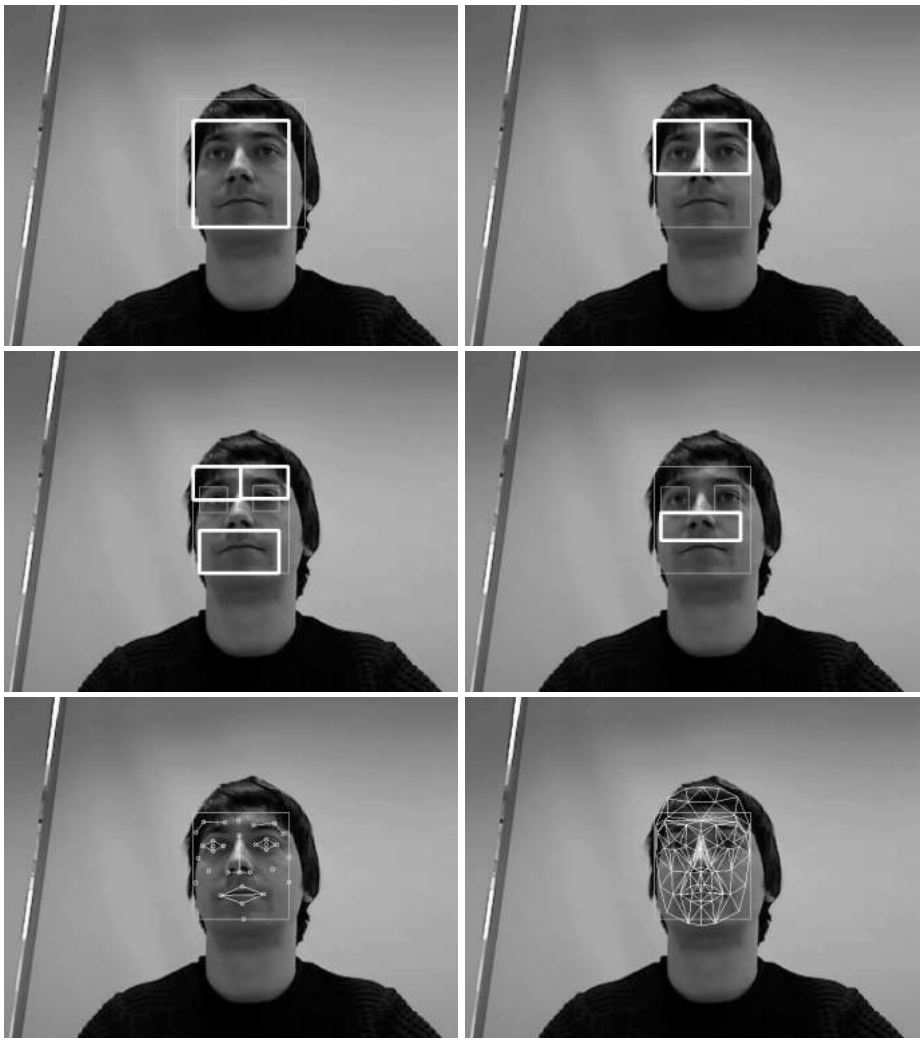
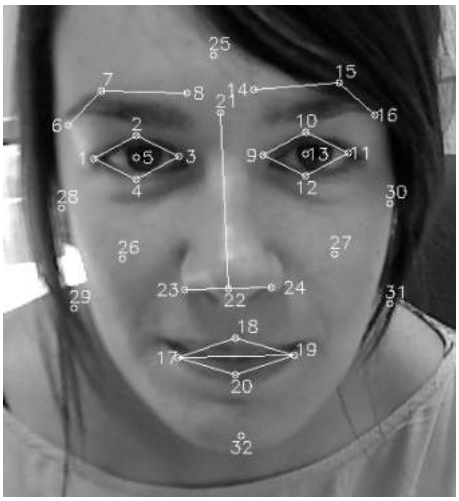


Figure 1: Proposed fitting approach. From left to right and top to bottom: (1) The detected face region and the *faceROI* derived from it (thicker line), (2) *faceROI* and the *eyeSROIs* derived from it (thicker line), (3) *faceROI*, the estimated *eyeROIs* and the *eyebrowSROIs* and *mouthSROI* derived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

Figure 2: Facial features detection procedure steps. From left to right and top down: (1) eye points detection, (2) eyebrow points detection, (3) mouth points detection, (4) nose points detection, (5) contour points detection and (6) face model fitting on the detected facial features.



DETECTED FACIAL POINTS

1. Left eye left corner
2. Left eye upper corner
3. Left eye right corner
4. Left eye lower corner
5. Left eye center
6. Left eyebrow left corner
7. Left eyebrow upper corner
8. Left eyebrow right corner
9. Right eye left corner
10. Right eye upper corner
11. Right eye right corner
12. Right eye lower corner
13. Right eye center
14. Right eyebrow left corner
15. Right eyebrow upper corner
16. Right eyebrow right corner
17. Mouth left corner
18. Mouth upper corner
19. Mouth right corner
20. Mouth lower corner
21. Nose top
22. Nose base
23. Left nostril
24. Right nostril
25. Forehead center
26. Left cheek
27. Right cheek
28. Left facial upper corner
29. Left facial lower corner
30. Right facial upper corner
31. Right facial lower corner
32. Chin bottom

Figure 3: The detected 32 facial points. Note that the words *left* and *right* are relative to the observer rather than the subject.

Figure 4: Images of the face model adjustment through OAM tracking [36] in a video, after applying the proposed initialization procedure in the first frame.

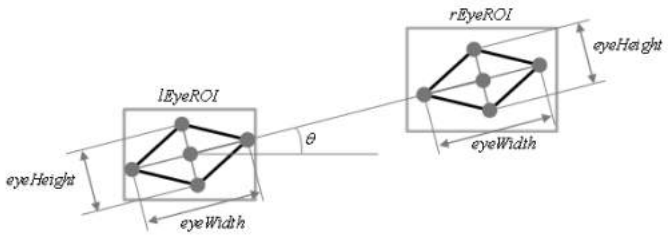


Figure 5: Eye points geometry derived in a fixed way from the estimated *eyeROIs*.

Figure 6: The considered viewpoints corresponding to subject 04000 of the CMU database with the flashlight 21 activated.

Figure 7: Examples of 3D face model fitting obtained with *FFBP* in the CMU PIE database illumination variation images.

Figure 8: *FFBP* fitting stages, from left to right: (1) Facial feature detection, (2) POS backprojection (only 3D pose) and (3) LM optimization (3D pose and 15 facial parameters.

Figure 9: Comparison between *CLM* and *FFBP-OAM* on an iPad 2 under a severe occlusion. The full sequence is available as supplementary material.

Figure 10: The full system running on an iPhone 5 at 24 FPS.

Figure 11: Comparison between *CLMBP*, *SDMBP* and *FFBP-OAM* in a video sequence with exaggerated facial expressions. The full sequence is available as supplementary material.

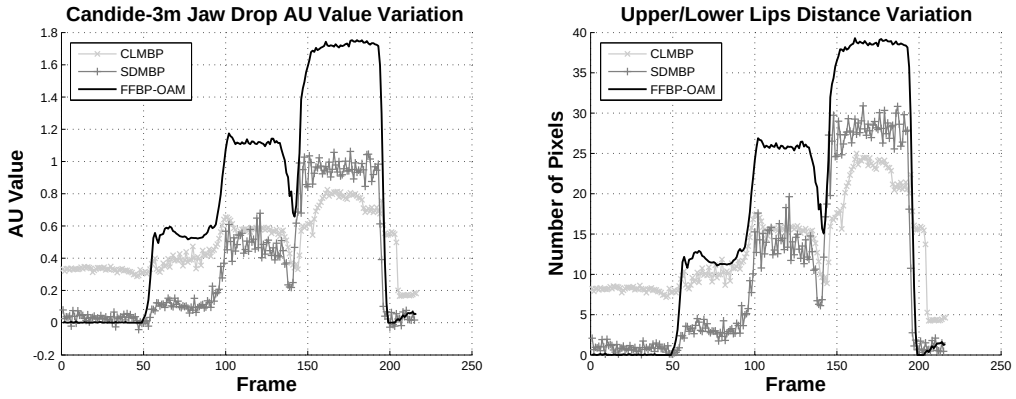


Figure 12: The *Jaw Drop* AU and upper/lower lips distance variations with *CLMBP*, *SDMBP* and *FFBP-OAM*.

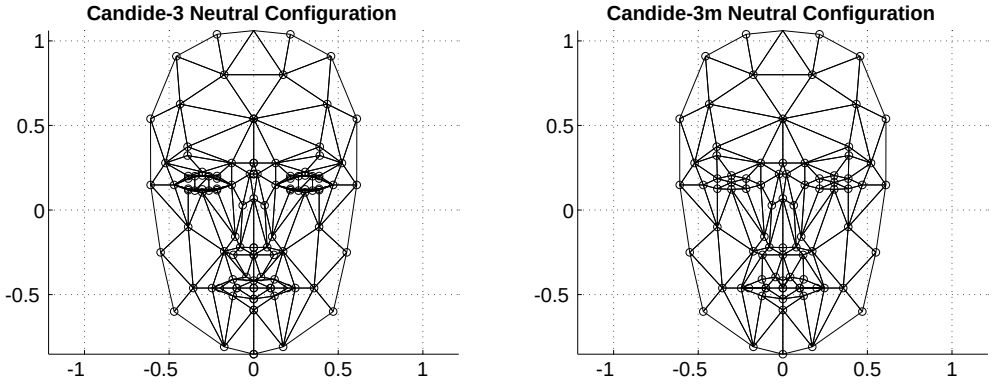


Figure A.13: The geometries of the Candide-3 and the Candide-3m face models.

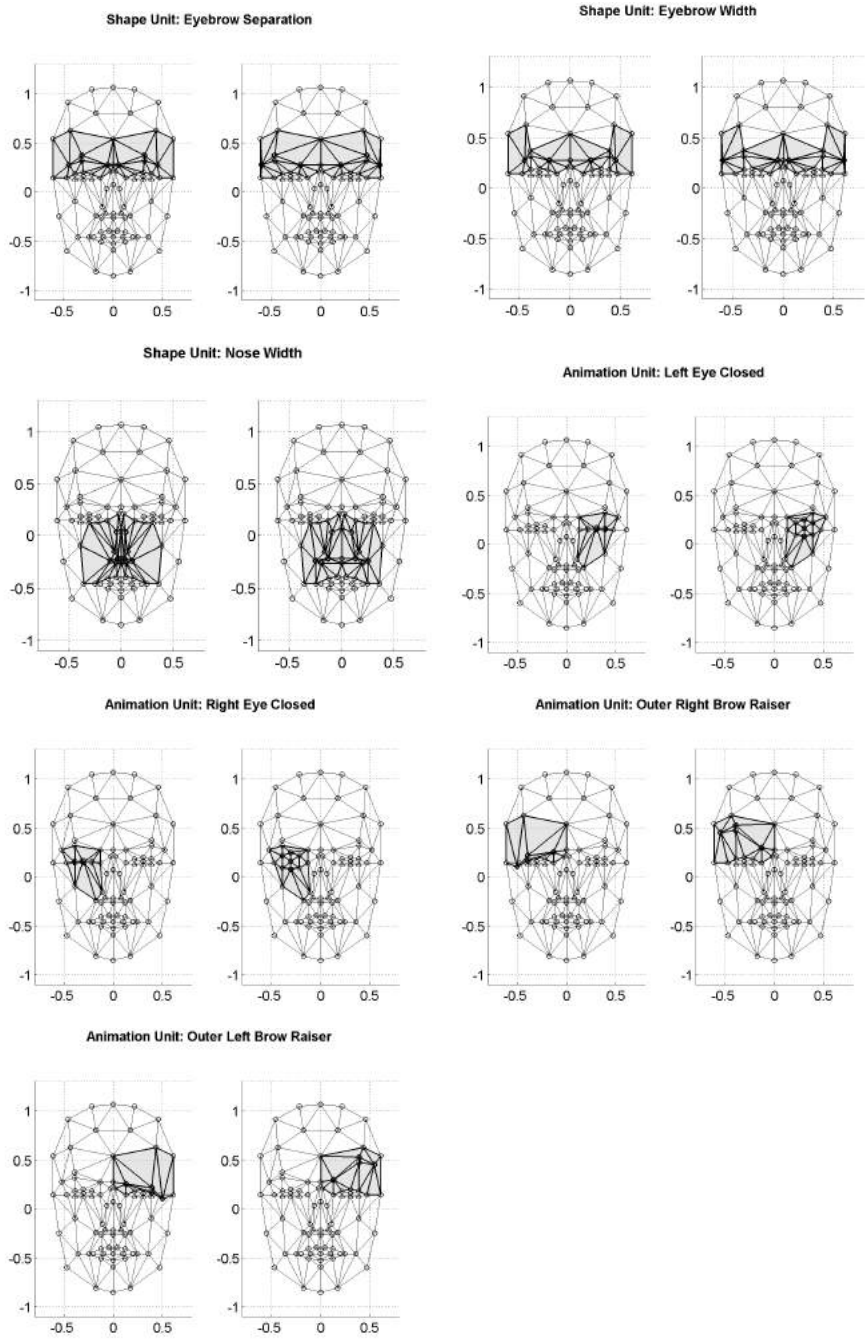


Figure A.14: The added and modified SUs and AUs in Candide-3m with respect to Candide-3, showing their variation from -1 to 1 values, where 0 corresponds to the neutral configuration.

References

[1] H. Kalbkhani, M. G. Shayesteh, S. M. Mousavi, Efficient algorithms for detection of face, eye and eye state, IET Computer Vision 7 (2013) 184–200.

[2] M. Perreira, V. Courboulay, A. Prigent, P. Estraillier, Fast, low resource, head detection and tracking for interactive applications, PsychNology Journal 7 (2009) 243–264.

[3] T. Hamada, K. Kato, K. Kawakami, Extracting facial features as in infants, Pattern Recognition Letters 21 (2000) 407–412.

[4] K.-W. Wong, K.-M. Lam, W.-C. Siu, An efficient algorithm for human face detection and facial feature extraction under different conditions, Pattern Recognition 34 (2001) 1993–2004.

[5] X. Peng, M. Bennamoun, A. S. Mian, A training-free nose tip detection method from face range images, Pattern Recognition 44 (2011) 544–558.

[6] S. Asteriadis, N. Nikolaidis, I. Pitas, Facial feature detection using distance vector fields, PR 42 (2009) 1388–1398.

[7] G. N. Votsis, A. I. Drosopoulos, S. D. Kollias, A modular approach to facial feature segmentation on real sequences, Signal Processing: Image Communication 18 (2003) 67–89.

[8] S. Jeng, H. Liao, C. Han, M. Chern, Y. Liu, Facial feature detection using geometrical face model: an efficient approach, Pattern Recognition 31 (1998) 273–282.

[9] D. Reisfeld, Y. Yeshurun, Preprocessing of face images: detection of features and pose normalization, Computer Vision and Image Understanding 71 (1998) 413–430.

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

[11] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, Active shape models - their training and application, Computer Vision and Image Understanding 61 (1995) 38–59.

[12] V. Blanz, P. Grother, P. J. Phillips, 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, volume 2, 2004, pp. 454–461.

[13] T. F. Cootes, G. V. Wheeler, K. N. Walker, C. J. Taylor, View-based active appearance models, Image and Vision Computing 20 (2002) 657–664.

[14] I. Matthews, S. Baker, Active appearance models revisited, International Journal of Computer Vision 60 (2004) 135–164.

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

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

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

[18] D. Vukadinovic, 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, volume 2, 2005.

[19] M. F. Valstar, B. Martinez, X. Binefa, 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, 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. F. Valstar, X. Binefa, M. Pantic, Local evidence aggregation for regression based facial point detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 1149–1163.

[22] X. Xiong, 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, T. F. Cootes, Feature detection and tracking with constrained local models, in: Proceedings of the British Machine Vision Conference, 2006, pp. 929–938.

[24] J. M. Saragih, S. Lucey, 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, T. S. Huang, Interactive facial feature localization, in: Proceedings of the IEEE European Conference on Computer Vision, 2012, pp. 679–692.

[26] X. Zhu, 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, Z. Zhang, A survey of recent advances in face detection, 2010.

[28] A. Asthana, S. Zafeiriou, S. Cheng, 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, 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, D. Kim, Pose robust face tracking by combining active appearance models and cylinder head models, International Journal of Computer Vision 80 (2008) 260–274.

[31] C. Chen, 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, 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, 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. D. Jepson, D. J. Fleet, T. F. El-Maraghi, Robust online appearance models for visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 1296–1311.

[36] F. Dornaika, F. Davoine, On appearance based face and facial action tracking, IEEE Transactions on Circuits and Systems for Video Technology 16 (2006) 1107–1124.

[37] F. Dornaika, F. Davoine, Simultaneous facial action tracking and expression recognition in the presence of head motion, International Journal of Computer Vision 76 (2008) 257–281.

[38] P. Viola, 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, volume 1, 2001, pp. 511–518.

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

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

[41] S. Suzuki, K. Be, Topological structural analysis of digitized binary images by border following, Computer Vision, Graphics, and Image Processing 30 (1985) 32–46.

[42] D. F. DeMenthon, L. S. Davis, Model-based object pose in 25 lines of code, International Journal of Computer Vision 15 (1995) 123–141.

[43] J. J. Moré, The Levenberg-Marquardt algorithm: implementation and theory, in: G. A. Watson (Ed.), Numerical Analysis, Lecture Notes in Mathematics 630, 18, Springer-Verlag, 1977, pp. 105–116.

[44] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 1615–1618.