

# Face Alignment Through 2.5D Active Appearance Models

Pedro Martins  
pedromartins@isr.uc.pt

Rui Caseiro  
ruicaseiro@isr.uc.pt

Jorge Batista  
batista@isr.uc.pt

Institute of Systems and Robotics  
University of Coimbra  
Coimbra, Portugal

---

## Abstract

This work addresses the fitting of 3D deformable face models from a single view through 2.5D Active Appearance Models (AAM). The main contribution of this paper is the use of 2.5D AAM that combines a 3D *metric* Point Distribution Model (PDM) and a 2D appearance model whose control points are defined by *full perspective* projections of the PDM. The advantage is that, assuming a calibrated camera, 3D metric shapes can be retrieved from single view images. Two algorithms and computationally efficient approximations are proposed, the Simultaneous Forwards Additive (SFA) and the Normalization Forwards Additive (NFA), both based on the Lucas Kanade framework. The SFA algorithm is computationally expensive but more accurate. It searches for shape and appearance parameters simultaneously whereas the NFA projects out the appearance from the error image and searches only for the shape parameters. Expanded solutions for the SFA and NFA are also proposed in order to take into account head self occlusions. An extensive performance evaluation is presented. The frequency of convergence for the SFA, NFA and their efficient approximation is evaluated, showing that the 2.5D model can outperform 2D based methods. The Robust extensions to occlusion were tested on a synthetic sequence showing that the model can deal robustly with large head rotation.

## 1 Introduction

Facial image alignment is the key aspect in many computer vision applications, such as advanced human computer interaction, face recognition, head pose estimation or realistic graphical animation. Detecting and tracking faces in video is a challenging task because faces are non rigid and their images have a high degree of variability in shape, texture, pose and imaging conditions.

The Active Appearance Model (AAM), introduced by [24], is one of the most effective face alignment techniques with respect to fitting accuracy and efficiency. AAMs are intrinsically 2D models, combining a 2D Point Distribution Model (PDM) and a 2D appearance model in a single formulation using a fitting process that rely on a precomputed regression matrix. Later, Matthews *et al.* [6] reformulate the AAM with true analytical derived

gradients achieving a better fitting accuracy and real-time performances using the Inverse Compositional (IC)[14] approach.

Natural extensions to 3D had been proposed [9][10][18], being the 3D Morphable Model (3DMM) [2] one of the most popular. The main differences between AAMs and 3DMMs are that 3DMMs are usually constructed to be denser and incorporates a reflectance model (Phong model). The large amount of data due to the density of the 3DMMs makes the algorithm quite slow, requiring several seconds to fit to a frame. Efficient 3DMMs [19], based on the IC algorithm, has also been proposed. Still, its Jacobian and Hessian are only locally valid and take an average of 30s per frame making it impracticable for real-time applications.

This paper addresses the fitting of 3D deformable face models from a single view through 2.5D AAM. The 2.5D model can be viewed as a 3D sparse PDM whose projections define 2D control points for the 2D appearance. This means that 2.5D data has components of both 2D image data and 3D volumetric shape data. Face alignment in this 2.5D dimensional space will carry an extra level of complexity since the IC approach is invalid in this case [17]. Matthews *et al.* [7] proposed a 2D+3D AAM work around by exploiting the 2D and 3D shape models simultaneously. The shape instance generated by 2D AAM is constrained to be consistent with a projection of 3D *affine* shape (they use a 3D PDM built from non rigid structure from motion [8]). This constraint is formulated as a part of the cost function, where a balancing weight is added. The value of this weighting constant is determined manually.

Our approach uses a single 3D metric PDM combined with a *full perspective model*, using an analytically derived gradient. The use of a full perspective model carry an important advantage over the state of the art solutions since it can track a face by a camera with a short focal length and strong radial distortion (p.e. a low cost webcam). Assuming a calibrated camera this solution allows the estimation of 3D euclidean shapes from a single image. Compared to [7], no balancing weight is required since the approach is based on a single 3D PDM. Two fitting algorithms are proposed: the Simultaneous Forward Additive (SFA) and the Normalization Forward Additive (NFA), both based on the Lucas-Kanade forward additive [16] update step. The xFA requires evaluating several components per iteration, however efficient approximations are proposed namely for the Jacobian of the warp computation. Self and partial occlusion are handled using robust fitting techniques. Expanded solutions for the SFA and NFA are proposed, not accounting for invisible pixels by combining outlier estimation with clues extracted from 3D pose.

## 2 2.5D Parametric Models

Our aim is to build a 2.5D AAM by combining a 3D metric Point Distribution Model (PDM) with a 2D appearance model whose control points are defined by full perspective projections of the PDM. The 3D PDM is modeled by the shape and pose parameters,  $\mathbf{p}$  and  $\mathbf{q}$  respectively, that uniquely defines a shape  $s$  in the 3D space whose projection into the image space sets 2D control points where the generated texture ( $\lambda$ ) is held.

### 2.1 Shape Model - The PDM

The shape of a non-rigid object can be expressed as a linear combination of a set of  $n$  basis shapes stored in a matrix  $\Phi$  plus a mean vector. This representation is also known as a Point Distribution Model (PDM)[20]. A 3D  $v$ -point shape is defined by the vertex locations of

a mesh as  $s = (X_1, \dots, X_v, Y_1, \dots, Y_v, Z_1, \dots, Z_v)^T$  and the PDM is represented by the linear model  $s = s_0 + \Phi \mathbf{p}$ , where  $\mathbf{p}$  is a vector of shape configuration weights,  $s_0$  is the mean shape (also referred as the *base mesh*) and the basis  $\Phi = [\phi_1 \dots \phi_n]$  represents the allowed modes of deformation.

In this work, the 3D PDM, including the full pose variation, is defined by

$$s = s_0 + \sum_{i=1}^n p_i \phi_i + \sum_{j=1}^6 q_j \psi_j + \underbrace{\int_0^{t-1} \sum_{j=1}^6 q_j \psi_j \partial t}_{s_\psi} \quad (6)$$

where  $\mathbf{p}$  are the previous shape parameters,  $\mathbf{q}$  are the pose parameters and  $s_\psi$  is the contribution of pose increments over time  $t$ . The first two terms represent PDM modes of deformation, the third term is the current estimated pose, and the last term ( $s_\psi$ ) acts as an offset that accumulates pose increments from previous time frames. Note that  $\psi_1, \dots, \psi_6$  are a special set of eigenvectors that are expressed w.r.t. the base mesh,  $s_0$ , and are only valid for small changes in pose.

This constraint on the pose changes results from the linear parametrization of the 3D pose parameters. Expressing a rotation of  $\theta$  radians around an arbitrary axis  $\mathbf{w} = (w_x, w_y, w_z)$  by the Rodrigues formula  $\mathbf{R} = \mathbf{I}_3 + \mathbf{A} \sin(\theta) + \mathbf{A}^2(1 - \cos(\theta))$ , with  $\mathbf{A}$  being a skew symmetric matrix, under the assumption of small rotations  $\mathbf{R}$  can be approximated by  $\mathbf{R} \approx \mathbf{I}_3 + \mathbf{A} \theta$ <sup>1</sup>. In a similar way, the small changes in pose that drive the 3D mesh points  $\mathbf{P} = (X, Y, Z)$  into  $\mathbf{P}'$  can be linearly parametrized by the vector  $\mathbf{q} = [w_x, w_y, w_z, t_x, t_y, t_z]^T$  as

$$\mathbf{P}' = \mathbf{P} + \underbrace{\begin{bmatrix} 0 & Z & -Y & 1 & 0 & 0 \\ -Z & 0 & X & 0 & 1 & 0 \\ Y & -X & 0 & 0 & 0 & 1 \end{bmatrix}}_{\psi_1, \dots, \psi_6} \mathbf{q} \quad (2)$$

where  $(t_x, t_y, t_z)$  are the translation components.

Using a *full perspective* camera, the 3D shape  $s$  generated by the PDM (eq. 1) is projected into the image space as

$$\begin{bmatrix} w(x_1 \dots x_v) \\ w(y_1 \dots y_v) \\ w \dots w \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} \mathbf{R}_0 & | & \mathbf{t}_0 \end{bmatrix}}_{\text{Base Pose}} \begin{bmatrix} s^{x_1} \dots s^{x_v} \\ s^{y_1} \dots s^{y_v} \\ s^{z_1} \dots s^{z_v} \\ 1 \dots 1 \end{bmatrix} \quad (3)$$

where  $\mathbf{K}$  is the camera matrix, assumed to be known (radial distortion can also be included).  $\mathbf{R}_0$  and  $\mathbf{t}_0$  define an extra referential, the *base pose*, with the purpose of setting a head reference (since rotation around the camera is not a head rotation).  $\mathbf{R}_0$  and  $\mathbf{t}_0$  are estimated in the PDM building process.

## 2.2 Texture Model

The texture model is almost identical to the traditional 2D formulation [21]. In our approach each training image is texture-warped into a common reference using a warping function  $\mathbf{W}$ . This function  $\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})$  is a piecewise affine warp and is a function of the shape and pose parameters that defines the 2D texture control points by means of the perspective projection

<sup>1</sup>The  $\theta$  coefficient can be dropped by relaxing the constraint that  $\mathbf{w}$  is unit length.

of the mesh  $s$  (eq.3). The warping is defined for all the projected pixels  $\mathbf{x}_p$ <sup>2</sup> contained in  $s_{0p}$  and is given by

$$\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}) = \mathbf{x}_{p_i} + \alpha (\mathbf{x}_{p_j} - \mathbf{x}_{p_i}) + \beta (\mathbf{x}_{p_k} - \mathbf{x}_{p_i}), \forall \text{ triangles} \in s_{0p} \quad (4)$$

where  $\mathbf{x}_{p_i}$ ,  $\mathbf{x}_{p_j}$ ,  $\mathbf{x}_{p_k}$  are triangle vertex's coordinates and  $\alpha$ ,  $\beta$  are the barycentric coordinates for the pixel  $\mathbf{x}_p$ . The appearance model is obtained by applying a low memory PCA on all the warped training images and it is represented by a base appearance,  $\mathbf{A}_0(\mathbf{x}_p)$ , plus a linear combination of  $m$  eigen images  $\mathbf{A}_i(\mathbf{x}_p)$ , as  $\mathbf{A}(\mathbf{x}_p) = \mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^m \lambda_i \mathbf{A}_i(\mathbf{x}_p)$ ,  $\mathbf{x}_p \in s_{0p}$ .  $\lambda_i$  are the appearance parameters. To model the gain and illumination offset effects, two extra appearance images are added ( $\mathbf{A}_{m+1}(\mathbf{x}_p) = \mathbf{A}_0(\mathbf{x}_p)$ ,  $\mathbf{A}_{m+2}(\mathbf{x}_p) = \mathbf{1}$ ) which imposes the need for orthonormalization.

### 3 Model Fitting

Fitting the AAM consists in finding the best set of parameters,  $\mathbf{p}$ ,  $\mathbf{q}$  and  $\lambda$  that describe the face in the target image. Since the IC approach [14] was proved in [15] to be invalid for the 2.5D AAM, the additive formulation proposed by Lucas-Kanade[16][17][18] was adopted. Two algorithms are proposed and described on the paper: the Simultaneous Forwards Additive (SFA) and the Normalization Forwards Additive (NFA). Both formulations include the 6DOF embedded in the PDM and just like the solutions initially proposed in [15][16] the SFA searches for all the parameters simultaneously whereas the NFA projects out the appearance from the error image. In section 3.3 it is shown how to maintain the fitting efficiency by making a simple approximation and precomputing a couple of terms. The experimental evaluation proved that the proposed solution substantially improves the fitting performance.

#### 3.1 Simultaneous Forwards Additive (SFA)

The SFA goal is to minimize the squared difference between the current instance of the appearance and the target warped image. The optimization consists in solving

$$\sum_{\mathbf{x}_p \in s_{0p}} \left[ \mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x}_p) - \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})) \right]^2 \quad (5)$$

simultaneously for the shape, pose and appearance parameters,  $\mathbf{p}$ ,  $\mathbf{q}$  and  $\lambda$  respectively.  $\mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))$  represents the input image  $\mathbf{I}(\mathbf{x}_p)$  warped by  $\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})$ . The nonlinear optimization in eq.5 can be solved by gradient descent using additive updates to the parameters as  $\sum_{\mathbf{x}_p \in s_{0p}} [\mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^{m+2} (\lambda_i + \Delta\lambda_i) \mathbf{A}_i(\mathbf{x}_p) - \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p} + \Delta\mathbf{p}, \mathbf{q} + \Delta\mathbf{q}))]^2$ . Expanding and holding the first order Taylor terms gives

$$\sum_{\mathbf{x}_p \in s_{0p}} \left[ \mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x}_p) + \sum_{i=1}^m \Delta\lambda_i \mathbf{A}_i(\mathbf{x}_p) - \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})) - \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{q}} \Delta\mathbf{q} \right]^2 \quad (6)$$

where  $\nabla \mathbf{I} = (\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y})$  is the gradient of image  $\mathbf{I}(\mathbf{x}_p)$  evaluated at  $\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})$  ( $\nabla \mathbf{I}$  is computed in the coordinate frame of  $\mathbf{I}(\mathbf{x}_p)$  and then warped back using  $\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})$ ),  $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$  and  $\frac{\partial \mathbf{W}}{\partial \mathbf{q}}$  are the Jacobians of the warp (see section 4).

<sup>2</sup>During the remaining of the paper,  $\mathbf{x}_p = [x, y]^T$  defines a projected 3D point into the 2D image space, by eq.3.

Defining, in eq.6, all the parameters as  $\mathbf{r} = [\mathbf{p}^T \mathbf{q}^T \lambda^T]^T$  and denoting the  $(n + 6 + m + 2)$  Steepest Descent images  $\mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}$  as

$$\mathbf{SD}(\mathbf{x}_p)_{\text{sfa}} = \left[ \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}_1} \dots \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}_n} \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{q}_1} \dots \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{q}_6} \mathbf{A}_1(\mathbf{x}_p) \dots \mathbf{A}_{m+2}(\mathbf{x}_p) \right], \quad (7)$$

then, by taking the partial derivative w.r.t.  $\Delta \mathbf{r}$  and make-it equal to zero comes the closed from solution

$$\Delta \mathbf{r} = \mathbf{H}_{\text{sfa}}^{-1} \sum_{\mathbf{x}_p \in s_{0p}} \mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}^T \mathbf{E}(\mathbf{x}_p)_{\text{sfa}} \quad (8)$$

where  $\mathbf{H}_{\text{sfa}} = \sum_{\mathbf{x}_p \in s_{0p}} \mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}^T \mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}$  is the Gauss-Newton approximation to the Hessian matrix and  $\mathbf{E}(\mathbf{x}_p)_{\text{sfa}} = \mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x}_p) - \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))$  is the error image. Finally the parameters are additively updated as  $\mathbf{r} \leftarrow \mathbf{r} + \Delta \mathbf{r}$  until  $\Delta \mathbf{r} \leq \varepsilon$  or a maximum number of iterations has reached.

The SFA algorithm is computationally expensive since for each iteration it is required the reevaluation of the error image, the gradients after the warp,  $\nabla \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))$ , the Jacobians,  $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ ,  $\frac{\partial \mathbf{W}}{\partial \mathbf{q}}$ , that depend on  $\mathbf{p}$  and  $\mathbf{q}$  respectively, the  $\mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}$  images, the Hessian matrix and its inverse. This makes SFA algorithm rather slow but very accurate since it searches for shape, pose and appearance parameters simultaneously. Nevertheless, some components of the Jacobians  $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ ,  $\frac{\partial \mathbf{W}}{\partial \mathbf{q}}$  are constant and can be precomputed (see section 4).

### 3.2 Normalization Forwards Additive (NFA)

A slightly different algorithm to minimize the expression in eq.5 is the NFA algorithm. An alternative way of dealing with the linear appearance variation is to project out the appearance images  $\mathbf{A}_i(\mathbf{x}_p)$  from the error image [13]. Denoting the appearance into a single image by  $\mathbf{A}(\mathbf{x}_p, \lambda) = \mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x}_p)$ , eq.5 can be written as  $\sum_{\mathbf{x}_p \in s_{0p}} [\mathbf{A}(\mathbf{x}_p, \lambda) - \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))]^2$ . Supposing now that there is no appearance variation, and  $\mathbf{A}(\mathbf{x}_p, \lambda) = \mathbf{A}_0(\mathbf{x}_p)$ , the modified  $\mathbf{SD}_{\text{nfa}}(\mathbf{x}_p)$  are represented as

$$\mathbf{SD}(\mathbf{x}_p)_{\text{nfa}} = \left[ \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}_1} \dots \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}_n} \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{q}_1} \dots \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{q}_6} \right] \quad (9)$$

and the Hessian is  $\mathbf{H}_{\text{nfa}} = \sum_{\mathbf{x}_p \in s_{0p}} \mathbf{SD}(\mathbf{x}_p)_{\text{nfa}}^T \mathbf{SD}(\mathbf{x}_p)_{\text{nfa}}$ . In this framework the error image is given by  $\mathbf{E}(\mathbf{x}_p)_{\text{lk}} = \mathbf{A}_0(\mathbf{x}_p) - \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))$  and in order to include the full appearance variation a *normalization* step is required. It is accomplished in the following two steps:

(1) Project the error image,  $\mathbf{E}(\mathbf{x}_p)_{\text{lk}}$ , into the appearance basis by estimating the appearance parameters using  $\lambda = \sum_{i=1}^{m+2} \mathbf{A}_i(\mathbf{x}_p) \mathbf{E}(\mathbf{x}_p)_{\text{lk}}$ .

(2) Remove the component of the error image in the direction of  $\mathbf{A}_i(\mathbf{x}_p)$  finding the normalized error image  $\mathbf{E}_{\text{nfa}}(\mathbf{x}_p) = \mathbf{E}(\mathbf{x}_p)_{\text{lk}} - \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x}_p)$ .

The NFA method consist in normalizing the error image (that has appearance  $\mathbf{A}(\mathbf{x}_p, \lambda)$ ) so that the component of the error image in the direction  $\mathbf{A}_i(\mathbf{x}_p)$  is zero, this step has the advantage of estimate the appearance parameters  $\lambda$ . Finally the parameters updates are given

$$\text{by } \begin{bmatrix} \Delta \mathbf{p} \\ \Delta \mathbf{q} \end{bmatrix} = \mathbf{H}_{\text{nfa}}^{-1} \sum_{\mathbf{x}_p \in s_{0p}} \mathbf{SD}(\mathbf{x}_p)_{\text{nfa}}^T \mathbf{E}(\mathbf{x}_p)_{\text{nfa}}.$$

The NFA algorithm is less computationally expensive than the SFA, since it projects out the appearance from the error image and searches only for the shape and pose parameters. Each iteration requires reevaluating the error image,  $\mathbf{E}(\mathbf{x}_p)_{\text{lk}}$ , the normalized error image

$\mathbf{E}(\mathbf{x}_p)_{\text{nfa}}$ ,  $\nabla \mathbf{I}$ ,  $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ ,  $\frac{\partial \mathbf{W}}{\partial \mathbf{q}}$ ,  $\mathbf{SD}(\mathbf{x}_p)_{\text{nfa}}$  and  $\mathbf{H}_{\text{nfa}}^{-1}$ . However, note that the  $\mathbf{SD}(\mathbf{x}_p)_{\text{nfa}}$  images are much smaller in number than the  $\mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}$ , i.e. ( $n \ll m$ ). The typical values of  $n$  are about 4 – 5 and  $m$  about 40 – 60.

### 3.3 Efficient Approximations to SFA and NFA

Some computational load can be reduced by eliminating the need to recompute image gradients at each iteration. Following the idea proposed by Hager *et al.* [10], and assuming that we have good estimates for all the parameters  $\mathbf{p}$ ,  $\mathbf{q}$  and  $\lambda$  (in eq.5), the error image  $\mathbf{E}(\mathbf{x}_p)_{\text{sfa}}$  will be  $\approx \mathbf{0}$  and we can say that:

$$\left( \mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x}_p) \right) \approx \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})) \Rightarrow \underbrace{\left( \nabla \mathbf{A}_0(\mathbf{x}_p) + \sum_{i=1}^{m+2} \lambda_i \nabla \mathbf{A}_i(\mathbf{x}_p) \right)}_{\nabla \mathbf{A}_i(\mathbf{x}_p, \lambda)} \approx \nabla \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})). \quad (10)$$

Under this approximation, the Efficient SFA/NFA Steepest Descent images from eq.7 and eq.9 can be written as

$$\mathbf{SD}(\mathbf{x}_p)_{\text{esfa}} = \left[ \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{p}_1} \dots \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{p}_n} \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{q}_1} \dots \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{q}_6} \mathbf{A}_1(\mathbf{x}_p) \dots \mathbf{A}_{m+2}(\mathbf{x}_p) \right], \quad (11)$$

$$\mathbf{SD}(\mathbf{x}_p)_{\text{enfna}} = \left[ \nabla \mathbf{A}_0(\mathbf{x}_p) \frac{\partial \mathbf{W}}{\partial \mathbf{p}_1} \dots \nabla \mathbf{A}_0(\mathbf{x}_p) \frac{\partial \mathbf{W}}{\partial \mathbf{p}_n} \nabla \mathbf{A}_0(\mathbf{x}_p) \frac{\partial \mathbf{W}}{\partial \mathbf{q}_1} \dots \nabla \mathbf{A}_0(\mathbf{x}_p) \frac{\partial \mathbf{W}}{\partial \mathbf{q}_6} \right]. \quad (12)$$

This approximation, besides providing extra computation efficiency (the gradients of the template can be precomputed), it has the great advantage of providing better stability to noise sensitivity since it avoids the reevaluation of the gradients in the input image  $\nabla \mathbf{I}(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))$  (and both warps  $\mathbf{I}_x(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))$ ,  $\mathbf{I}_y(\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q}))$ ) at each iteration.

### 3.4 Robust Fitting

Both SFA and NFA are data driven algorithms and the error image continuously drives the models in further updates. In the case of occlusion, the error image accounts for all the pixels equally (L2 norm) leading the model to diverge. To overcome this problem, occlusion can be modeled as outlier pixels in the appearance model and handled by robust fitting methods [13] [14], namely by Iteratively Reweighted Least Squares (IRLS) where outliers are not accounted for the parameters updates.

Robust fitting seek to minimize  $\sum_{\mathbf{x}_p} \rho(\mathbf{E}(\mathbf{x}_p)_{\text{sfa}}^2, \sigma)$  where  $\rho(\cdot)$  is a robust function and  $\sigma$  is the scale parameter that can be estimated from the error image. The derivation of the Robust version of SFA algorithm (RSFA) is similar to section 3.1, where the parameters update is given by  $\Delta \mathbf{r} = \mathbf{H}_{\text{rsfa}}^{-1} \sum_{\mathbf{x} \in s_{0p}} \rho(\mathbf{E}(\mathbf{x}_p)_{\text{sfa}}^2) \mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}^T \mathbf{E}(\mathbf{x}_p)_{\text{sfa}}$  with  $\rho(\mathbf{E}(\mathbf{x}_p)_{\text{sfa}}^2)$  being the weight mask that measures the confidence of each pixel over the base mesh and the Hessian is  $\mathbf{H}_{\text{rsfa}} = \sum_{\mathbf{x} \in s_{0p}} \rho(\mathbf{E}(\mathbf{x}_p)_{\text{sfa}}^2) \mathbf{SD}(\mathbf{x}_p)_{\text{sfa}}^T \mathbf{SD}_{\text{sfa}}(\mathbf{x}_p)$ . In the same way, the Robust version of NFA (RNFA) includes a weight mask in the Steepest Descent images,  $\mathbf{SD}(\mathbf{x}_p)_{\text{nfa}}$  and in the Hessian,  $\mathbf{H}_{\text{rnfa}}$ . Just like in the NFA algorithm, the RNFA requires a robust appearance normalization. A slightly modified solution of the normalization step proposed in [16] was used. The efficient approximations presented in section 3.3, are also valid for the robust fitting versions.

#### 3.4.1 Modified Robust Error Function

The 2.5D model has the advantage that is able to estimate visible areas (say mesh triangles) in the image projection model. Invisible triangles by the camera can be dropped, setting

them as outliers and not taking them into consideration in the fitting process.

The *modified* robust error function is based on the Talwar function and the scale parameter  $\sigma$  is estimated by using information about the triangles visibility over the base mesh through Back-face Culling (figures 1-d-e)).

## 4 The Jacobian of the Warp

The Jacobians of the warp measure the rate of change of the destination in the warp  $\mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})$  w.r.t. the parameters  $\mathbf{p}$  and  $\mathbf{q}$ . Two Jacobians must be derived  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}}$  and  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{q}}$ , w.r.t. shape and pose parameters respectively.

The Jacobian for the shape parameters can be decomposed by the chain rule as

$$\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}} = \sum_{k=1}^v \left[ \frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{x}_k} \frac{\partial \mathbf{x}_k}{\partial \mathbf{p}} + \frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{y}_k} \frac{\partial \mathbf{y}_k}{\partial \mathbf{p}} \right]. \quad (13)$$

Taking eq.4, comes that  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{x}_k} = (1 - \alpha - \beta, 0)$  and  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{y}_k} = (0, 1 - \alpha - \beta)$ . These Jacobians are images w.r.t. a particular vertex and have the same size of the projected base mesh  $s_{0p}$ . Figure 1-a)-b)-c) shows examples of these images for some landmarks (note the  $x$  and  $y$  components). The Jacobians are only non zero around the neighbors triangles of vertex  $k^{th}$ , taking the maximum value of 1 at the vertex location and decaying linearly with a rate of  $1 - \alpha - \beta$  to the other surrounding vertex's.

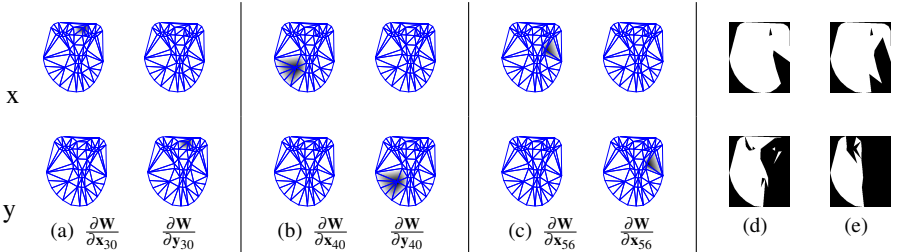


Figure 1: a)-b)-c) Shows  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{x}_k}$  and  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{y}_k}$  for the landmarks 30, 40 and 56, respectively. Top and bottom rows represent  $\mathbf{W}_x(\mathbf{x}_p, \mathbf{p}, \mathbf{q})$  and  $\mathbf{W}_y(\mathbf{x}_p, \mathbf{p}, \mathbf{q})$  components. d)-e) Triangle visibility by Back-face Culling for a head pitch variation of  $40^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $120^\circ$  (top-left to bottom-right) w.r.t the base pose. Nonvisible triangles (in black) are not used to update the parameters.

The remaining terms  $\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}}$  and  $\frac{\partial \mathbf{y}_k}{\partial \mathbf{p}}$  are both scalars, found by combining eq.3 and eq.1, as

$$\begin{bmatrix} w\mathbf{x}_k \\ w\mathbf{y}_k \\ w \end{bmatrix} = \mathbf{K} \left[ \mathbf{R}_0 \mid \mathbf{t}_0 \right] \begin{bmatrix} s_0^{xk} + p_i \phi_i^{xk} \\ s_0^{yk} + p_i \phi_i^{yk} \\ s_0^z + p_i \phi_i^z \\ 1 \end{bmatrix} \quad (14)$$

with  $i = 1, \dots, n$  parameters and  $k = 1, \dots, v$  landmarks. To compute  $\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}}$  we take the differential  $\frac{\partial}{\partial \mathbf{p}} \left( \frac{w\mathbf{x}_k}{w} \right)$  from eq.14 and the same for  $\frac{\partial \mathbf{y}_k}{\partial \mathbf{p}} = \frac{\partial}{\partial \mathbf{p}} \left( \frac{w\mathbf{y}_k}{w} \right)$ . Results are omitted due the lack of space.

The same approach is taken to evaluate the Jacobian of the warp for the pose parameters,  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{q}}$ . A chain rule decomposition is made and the new terms  $\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}}$  and  $\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}}$ , again both

scalars, are found by combining eq.3 with eq.1, leading to

$$\begin{bmatrix} w\mathbf{x}_k \\ w\mathbf{y}_k \\ w \end{bmatrix} = \mathbf{K} \left[ \mathbf{R}_0 \mid \mathbf{t}_0 \right] \begin{bmatrix} s_0^{x_k} + q_i \psi_i^{x_k} + s_\psi^{x_k} \\ s_0^{y_k} + q_i \psi_i^{y_k} + s_\psi^{y_k} \\ s_0^z + q_i \psi_i^z + s_\psi^z \\ 1 \end{bmatrix} \quad (15)$$

with  $i = 1, \dots, 6$  and  $k = 1, \dots, v$ . In the same way,  $\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \left( \frac{w\mathbf{x}_k}{w} \right)$  and  $\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \left( \frac{w\mathbf{y}_k}{w} \right)$ .

The Jacobians of the warp depend on  $\mathbf{p}$  and  $\mathbf{q}$ , so they are required to be recomputed at every iteration. However, both components  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{x}_k}$  and  $\frac{\partial \mathbf{W}(\mathbf{x}_p, \mathbf{p}, \mathbf{q})}{\partial \mathbf{y}_k}$  depend only on the configuration of the base mesh and thus can be precomputed and efficiently stored as sparse matrices, reducing the overall computation. At the fitting stage only the computation of  $\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}}$ ,  $\frac{\partial \mathbf{y}_k}{\partial \mathbf{p}}$ ,  $\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}}$  and  $\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}}$  is required, being all scalar values.

## 5 Experimental Results

The 2.5D AAM was constructed from a set of 7 human faces. The 3D PDM was built using a fully calibrated stereo system where the 2D shape on each view was extracted by fitting a 2D AAM[9] using  $v = 58$  landmarks. The 3D shape was recovered using the classical triangulation algorithm on a total of 20 images for each individual (10 left + 10 right) exhibiting several expressions and head poses. The AAM holds  $n = 5$  shape parameters,  $m = 47$  eigen faces and uses about 74600 grey level pixels. The initial 6DOF estimate is provided by combining a face detection (Adaboost [10]) with the Pose from Orthography and Scaling with Iterations (POSIT)[11], using the base mesh  $s_0$  as the required 3D rigid model.

### 5.1 Fitting Robustness and Rate of Convergence

To evaluate the fitting robustness and the rate of convergence of the proposed solutions, we follow the performance evaluation scheme, presented in [9][12]. Figure 2 shows the results obtained by comparing the fitting robustness and rate of convergence of all the non robust 2.5D algorithms discussed and the 2D state of the art leading approach (Simultaneous Inverse Compositional - SIC [13]). These experiments measure the performance of the algorithms in

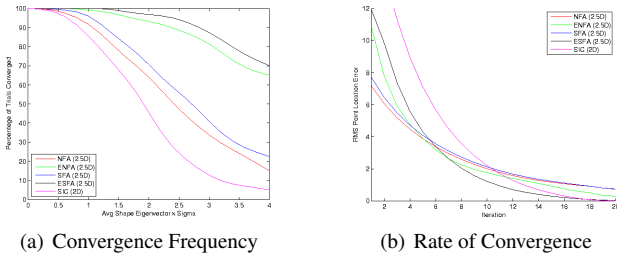


Figure 2: Robustness fitting and convergence comparison between 2.5D and 2D algorithms.

two ways: (1) the average frequency of convergence i.e. the number of times each algorithm has converged vs. initial perturbation; (2) the average rate of convergence i.e. the 2D RMS error in the mesh point location vs. iteration number (if convergence was accomplished).



For these experiments, the AAM was perturbed from a set of ground truth parameters using independent Gaussian distributions with variance equal to a multiple of a given eigenvalue mode, and tested for convergence. The ground truth data was generated using the same AAM by a combination of tracking (say fitting in every frame) / hand initialization / visual confirmation on several small sequences of each individual. A subset of 10 random selected frames, from each sequence, were used for further testing, accounting a total of 70 frames. For each testing frame we generate a number of 20 trials by perturbing the shape and appearance parameters simultaneously from the ground-truth. All the algorithms are executed and its convergence ability was evaluated by comparing the final 2D RMS error shape with the ground-truth. A threshold of 1.0 RMS pixels was used to define convergence.

Analyzing figure 2, we can conclude that 2.5D fitting algorithms are more robust than 2D algorithms and they converge faster, taking less iterations to converge. The 3D PDM is inherently higher dimensional than the 2D PDM, however, it uses less 3D shape parameters than the 2D PDM to represent the same visual phenomenon. The 3D PDM is also less prone to local minima because an 2D model can easily generate a physically unfeasible shape.

The results also show that the efficient versions proposed perform better than the standard formulations. The main reason for this performance increase is the reduced noise influence that comes out from avoiding the reevaluation of the gradients of the input image in each iteration. The Efficient-SFA, that searches simultaneously for all the parameters, has proved to be the best algorithm w.r.t. convergence speed with high fitting success rates even from far initial estimates.

## 5.2 Robust Methods Evaluation

The robust fitting methods proposed in this work intend to improve the performance w.r.t. self occlusion due to 3D head motion. To evaluate these algorithms, namely the RNFA, the RSFA and the Efficient versions ERNFA and ERNFA, three synthetic sequences were created. A set of images with an individual standing in near frontal position was used. The current 3D mesh location  $s$  was found by fitting the 2.5D AAM (using ESFA). Then, ranging the 3D mesh from  $-90^\circ$  to  $90^\circ$  degrees in both roll, pitch and yaw angles, using one degree of resolution, the fixed appearance image is projected into the camera and stored (figure 3-left). Finally, all the fitting algorithms were evaluated using these sequences, starting from the frontal position. In all the algorithms the scale parameter,  $\sigma$ , is found by assuming

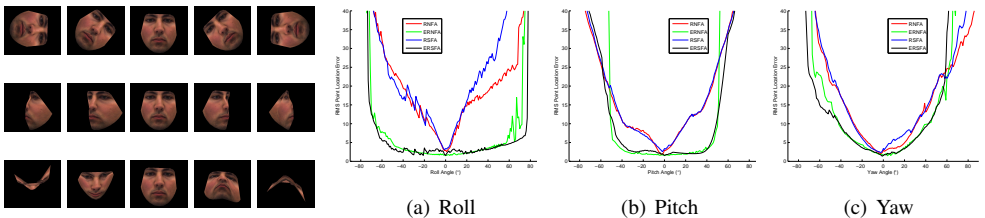


Figure 3: Robust algorithms evaluation on the synthetic sequences at left figure. The graphics show the RMS error due to roll, pitch and yaw angles ranging from  $-90^\circ$  to  $90^\circ$ , respectively.

that exists 20% of outliers. Figure 3-bottom shows the RMS error in point location for all the algorithms. Once again the Efficient versions of the algorithms (ERNFA and ERSFA) outperform their standard versions (RNFA and RSFA). Also the ERSFA performs slightly

better than the ERNFA, as expected, due to the parameters search strategy. These experiments show that, using the efficient algorithms, the model can deal with rotations in about  $\pm 70^\circ$  roll,  $\pm 50^\circ$  pitch and  $\pm 45^\circ$  in yaw angles.

### 5.3 Head Pose Estimation

Although the proposed methods are not explicitly oriented for pose estimation, the updates on the pose parameters,  $\Delta \mathbf{q}$ , can be analyzed and used for this purpose. In this section, only the ESFA algorithm has been used, since it was proved to be the most accurate.

Figure 4 shows the estimated rigid head pose during a video sequence where the subject performs several human head movements. It consists in making (independent) head rotations first in pitch, then in yaw and finally in roll orientations, returning always to frontal position. Finally, the distance to the camera was also evaluated. The pose estimation accuracy is performed comparing the pose estimated with the one estimated from a planar checkerboard, used as ground truth reference values. The error standard deviation found was about  $1.6^\circ$ ,  $1.7^\circ$ ,  $3.5^\circ$ ,  $28.8\text{mm}$  in roll, pitch and yaw angles and distance, respectively.

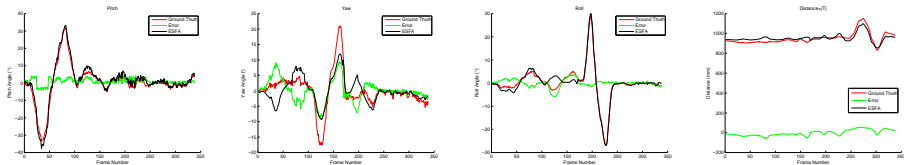


Figure 4: Head pitch, yaw, roll and distance evaluation. The horizontal axis is the time line.

## 6 Conclusions

In this paper we perform 3D facial image alignment from single view 2D images through a 2.5D AAM. The 2.5D AAM combines a 3D metric PDM with a full perspective projection that define the 2D appearance. The model is able to recover 3D Euclidian shapes by assuming a calibrated camera. Two algorithms and computational efficient approximations are proposed, both are based on the Lucas and Kanade framework. The SFA, compared with NFA, is the most accurate algorithm although is also the most computationally expensive. Their efficient versions have shown a substantial improvement in the fitting performance, being more robust to noise and able to converge from far initial estimates, requiring less computational effort. To make the model able to deal with self or partial occlusion, robust extensions to SFA and NFA are also proposed. Again, their efficient approximations perform much better than the basic versions. Several performance evaluation carried on real and synthetic data demonstrated that the 2.5D AAM outperform the traditional 2D AAM and accurately handle face pose variations.

## Acknowledgements

This work was supported by the Portuguese Science Foundation (FCT) by the project “Dinâmica Facial 4D para Reconhecimento de Identidade” with grant PTDC/EIA-CCO/108791/2008. Pedro Martins also acknowledges the FCT for support through the grant SFRH/BD/45178/2008.

## References

- [1] B.Lucas and T.Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.
- [2] D. DeMenthon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, June 1995.
- [3] F.Dornaika and J.Ahlberg. Fast and reliable active appearance model search for 3d face tracking. In *International Conference on Model-based Imaging, Rendering, Image Analysis and Graphical Special Effects (MIRAGE)*, pages 113–122, March 2003.
- [4] F.Dornaika and J.Ahlberg. Fitting 3d face models for tracking and active appearance model training. *Image and Vision Computing*, 24:1010–1024, September 2006.
- [5] G.Hager and P.Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–39, October 1998.
- [6] I.Matthews and S.Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(1):135–164, November 2004.
- [7] I.Matthews, J.Xiao, and S.Baker. 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *International Journal of Computer Vision*, 75(1):93–113, October 2007.
- [8] J.Xiao, J.Chai, and T.Kanade. A closed-form solution to non-rigid shape and motion recovery. In *European Conference on Computer Vision*, May 2004.
- [9] M.G.Roberts, T.F.Cootes, and J.E.Adams. Robust active appearance models with iteratively rescaled kernels. In *British Machine Vision Conference*, volume 1, pages 302–311, 2007.
- [10] P.Lucey, S.Lucey, M.Cox, S.Sridharan, and J.F.Cohn. Comparing object alignment algorithms with appearance variation: Forward-additive vs inverse-composition. In *IEEE International Workshop on Multimedia Signal Processing - MMSP*, pages 337–342, 2008.
- [11] P.Viola and M.Jones. Robust real-time object detection. *International Journal of Computer Vision*, July 2001.
- [12] R.Gross, I.Matthews, and S.Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(1):1080–1093, November 2005.
- [13] R.Gross, I.Matthews, and S.Baker. Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604, 2006.
- [14] S.Baker and I.Matthews. Equivalence and efficiency of image alignment algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, December 2001.

- [15] S.Baker and I.Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(1):221–255, March 2004.
- [16] S.Baker, R.Gross, and I.Matthews. Lucas kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, CMU Robotics Institute, November 2003.
- [17] S.Baker, R.Patil, K.M.Cheung, and I.Matthews. Lucas kanade 20 years on: A unifying framework: Part 5. Technical Report CMU-RI-TR-04-64, CMU Robotics Institute, November 2004.
- [18] S.E.Ayala-Raggi and J.C.Enriquez L.A.Robles. Recovering 3d shape and albedo from a face image under arbitrary lighting and pose by using a 3d illumination-based aam model. In *International Conference on Image Analysis and Recognition*, pages 584–593, 2009.
- [19] S.Romdhani and T.Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *IEEE International Conference on Computer Vision*, pages 59–66, 2003.
- [20] T.F.Cootes and C.J.Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engineering, University of Manchester, 2004.
- [21] T.F.Cootes, G.J.Edwards, and C.J.Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.
- [22] V.Blanz and T.Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH 99*, pages 187–194, 1999.

## A The Jacobian of The Warp Partial Differentials

Defining the elements  $m_{0,j}$  of the base projection matrix  $\mathbf{M}_0$  as

$$\underbrace{\begin{bmatrix} m_{011} & m_{012} & m_{013} & m_{014} \\ m_{021} & m_{022} & m_{023} & m_{024} \\ m_{031} & m_{032} & m_{033} & m_{034} \end{bmatrix}}_{\mathbf{M}_0} = \mathbf{K} \begin{bmatrix} \mathbf{R}_0 & \mathbf{t}_0 \end{bmatrix} \quad (16)$$

the quantities  $\xi_1, \dots, \xi_6$  and  $\Xi_1, \dots, \Xi_6$  are scalar values given by

$$\begin{aligned} \xi_1 &= m_{011} \phi_i^{xk} + m_{012} \phi_i^{yk} + m_{013} \phi_i^{zk} & \xi_4 &= m_{011} \psi_j^{xk} + m_{012} \psi_j^{yk} + m_{013} \psi_j^{zk} \\ \xi_2 &= m_{021} \phi_i^{xk} + m_{022} \phi_i^{yk} + m_{023} \phi_i^{zk} & \xi_5 &= m_{021} \psi_j^{xk} + m_{022} \psi_j^{yk} + m_{023} \psi_j^{zk} \\ \xi_3 &= m_{031} \phi_i^{xk} + m_{032} \phi_i^{yk} + m_{033} \phi_i^{zk} & \xi_6 &= m_{031} \psi_j^{xk} + m_{032} \psi_j^{yk} + m_{033} \psi_j^{zk} \end{aligned} \quad (17)$$

$$\begin{bmatrix} \Xi_1 \\ \Xi_2 \\ \Xi_3 \end{bmatrix} = \mathbf{M}_0 \begin{bmatrix} s_0^{xk} + p_i \phi_i^{xk} + \sum_{j \neq i}^n p_j \phi_j^{xk} + \sum_{j=1}^6 q_j \psi_j^{xk} + s_\psi^{xk} \\ s_0^{yk} + p_i \phi_i^{yk} + \sum_{j \neq i}^n p_j \phi_j^{yk} + \sum_{j=1}^6 q_j \psi_j^{yk} + s_\psi^{yk} \\ s_0^{zk} + p_i \phi_i^{zk} + \sum_{j \neq i}^n p_j \phi_j^{zk} + \sum_{j=1}^6 q_j \psi_j^{zk} + s_\psi^{zk} \\ 1 \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} \bar{\mathfrak{E}}_4 \\ \bar{\mathfrak{E}}_5 \\ \bar{\mathfrak{E}}_6 \end{bmatrix} = \mathbf{M}_0 \begin{bmatrix} s_0^{x_k} + \sum_{i=1}^n p_i \phi_i^{x_k} + q_j \psi_j^{x_k} + \sum_{i \neq j}^6 q_i \Psi_i^{x_k} + s_{\Psi}^{x_k} \\ s_0^{y_k} + \sum_{i=1}^n p_i \phi_i^{y_k} + q_j \psi_j^{y_k} + \sum_{i \neq j}^6 q_i \Psi_i^{y_k} + s_{\Psi}^{y_k} \\ s_0^{z_k} + \sum_{i=1}^n p_i \phi_i^{z_k} + q_j \psi_j^{z_k} + \sum_{i \neq j}^6 q_i \Psi_i^{z_k} + s_{\Psi}^{z_k} \\ 1 \end{bmatrix}. \quad (19)$$

## ERRATA

Dear reader, unfortunately, we found some mistakes after the publication:

- Minus operator missing in eqs.7 and 11 in the appearance images

$$\mathbf{SD}(\mathbf{x}_p)_{\text{sfa}} = \left[ \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}_1} \dots \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}_n} \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{q}_1} \dots \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{q}_6} - \mathbf{A}_1(\mathbf{x}_p) \dots - \mathbf{A}_{m+2}(\mathbf{x}_p) \right]$$

$$\mathbf{SD}(\mathbf{x}_p)_{\text{esfa}} = \left[ \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{p}_1} \dots \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{p}_n} \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{q}_1} \dots \nabla \mathbf{A}_i(\mathbf{x}_p, \lambda) \frac{\partial \mathbf{W}}{\partial \mathbf{q}_6} - \mathbf{A}_1(\mathbf{x}_p) \dots - \mathbf{A}_{m+2}(\mathbf{x}_p) \right]$$

- Missing some shape components in eqs.14

$$\begin{bmatrix} w\mathbf{x}_k \\ w\mathbf{y}_k \\ w \end{bmatrix} = \mathbf{K} \underbrace{[\mathbf{R}_0 \mid \mathbf{t}_0]}_{\mathbf{M}_0} \begin{bmatrix} s_0^{x_k} + p_i \phi_i^{x_k} + \sum_{j \neq i}^n p_j \phi_j^{x_k} + \sum_{j=1}^6 q_j \Psi_j^{x_k} + s_{\Psi}^{x_k} \\ s_0^{y_k} + p_i \phi_i^{y_k} + \sum_{j \neq i}^n p_j \phi_j^{y_k} + \sum_{j=1}^6 q_j \Psi_j^{y_k} + s_{\Psi}^{y_k} \\ s_0^{z_k} + p_i \phi_i^{z_k} + \sum_{j \neq i}^n p_j \phi_j^{z_k} + \sum_{j=1}^6 q_j \Psi_j^{z_k} + s_{\Psi}^{z_k} \\ 1 \end{bmatrix}$$

$$\frac{\partial \mathbf{x}_k}{\partial p_i} = \frac{\xi_1 \bar{\mathfrak{E}}_3 - \bar{\mathfrak{E}}_1 \xi_3}{(\bar{\mathfrak{E}}_3)^2} \quad \text{and} \quad \frac{\partial \mathbf{y}_k}{\partial p_i} = \frac{\xi_2 \bar{\mathfrak{E}}_3 - \bar{\mathfrak{E}}_2 \xi_3}{(\bar{\mathfrak{E}}_3)^2}$$

with  $i = 1, \dots, n$  (shape parameters) and  $k = 1, \dots, v$  (landmarks).

- Also in eq.15

$$\begin{bmatrix} w\mathbf{x}_k \\ w\mathbf{y}_k \\ w \end{bmatrix} = \mathbf{M}_0 \begin{bmatrix} s_0^{x_k} + \sum_{i=1}^n p_i \phi_i^{x_k} + q_j \psi_j^{x_k} + \sum_{i \neq j} q_i \Psi_i^{x_k} + s_{\Psi}^{x_k} \\ s_0^{y_k} + \sum_{i=1}^n p_i \phi_i^{y_k} + q_j \psi_j^{y_k} + \sum_{i \neq j} q_i \Psi_i^{y_k} + s_{\Psi}^{y_k} \\ s_0^{z_k} + \sum_{i=1}^n p_i \phi_i^{z_k} + q_j \psi_j^{z_k} + \sum_{i \neq j} q_i \Psi_i^{z_k} + s_{\Psi}^{z_k} \\ 1 \end{bmatrix}$$

$$\frac{\partial \mathbf{x}_k}{\partial q_j} = \frac{\xi_4 \bar{\mathfrak{E}}_6 - \bar{\mathfrak{E}}_4 \xi_6}{(\bar{\mathfrak{E}}_6)^2} \quad \text{and} \quad \frac{\partial \mathbf{y}_k}{\partial q_j} = \frac{\xi_5 \bar{\mathfrak{E}}_6 - \bar{\mathfrak{E}}_5 \xi_6}{(\bar{\mathfrak{E}}_6)^2}$$

with  $j = 1, \dots, 6$  (pose parameters) and  $k = 1, \dots, v$  (landmarks).

We apologize for the inconvenience.