Supplementary Material
Generative Face Alignment Through 2.5D Active Appearance Models

Pedro Martins, Rui Caseiro, Jorge Batista

Institute of Systems and Robotics, University of Coimbra, DEEC - Polo II, 3030-290 Coimbra, Portugal

1. Introduction

This document provides three extra sections with additional details that were excluded from the main document to meet length requirements. Section Appendix A describes the details envolving the efficient warping procedure. Section Appendix B provides the derivation of eq.B.7 that appears as part of the solution of the Simultaneous Forwards Additive (SFA) algorithm in section 3.1 of the main document. Finally, Appendix C describes the used approach to build the 3D PDM using a stereo pair of cameras.

For the sake of notation, we rewrite the equations that are required for this supplementary material section:

- The 3D Point Distribution Model (PDM), including the full pose variation, is defined by

$$s = s_0 + \sum_{i=1}^{n} p_i \phi_i + \sum_{j=1}^{6} q_j \psi_j + s_\psi \tag{1}$$

- The 3D shape s is projected into the image space using a full perspective camera, as

$$\begin{bmatrix} w(x_1 \cdots x_v) \\ w(y_1 \cdots y_v) \\ w \cdots w \end{bmatrix} = \begin{bmatrix} f_x & \alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{c|c} \mathbf{R}_0 & \mathbf{t}_0 \end{array} \right] \underbrace{\begin{bmatrix} s^{x_1} \ldots s^{x_v} \\ s^{y_1} \ldots s^{y_v} \\ s^{z_1} \ldots s^{z_v} \\ 1 \cdots 1 \end{bmatrix}}_{\text{PDM shape (eq.1)}} \tag{2}$$

Email address: `pedromartins@isr.uc.pt Telephone (+351) 239 796383 Fax (+351) 239 406672` (Pedro Martins)
URL: `http://www.isr.uc.pt/~pedromartins` (Pedro Martins)

- The piecewise affine warp function is given by

$$\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}) = \mathbf{x}_{\mathbf{p}_i} + \alpha \left(\mathbf{x}_{\mathbf{p}_j} - \mathbf{x}_{\mathbf{p}_i} \right) + \beta \left(\mathbf{x}_{\mathbf{p}_k} - \mathbf{x}_{\mathbf{p}_i} \right), \forall \text{ triangles} \in s_{0\mathbf{p}} \tag{3}$$

  where $\mathbf{x}_{\mathbf{p}_i}$, $\mathbf{x}_{\mathbf{p}_j}$, $\mathbf{x}_{\mathbf{p}_k}$ are triangle vertex's coordinates and α, β are the barycentric coordinates for the projected pixel $\mathbf{x_p}$.

Appendix A. Piecewise Affine Warp

The piecewise affine warp is composed by sets of affine warps between corresponding triangles of the mesh. The base triangles are found by partitioning the convex hull of the projected mean shape, $s_{0\mathbf{p}}$, using the Delaunay triangulation, and each pixel belonging to a given triangle is mapped to its correspondent triangle using barycentric coordinates.

As mentioned, two meshes are involved in the warping procedure: the projected base mesh $s_{0\mathbf{p}}$ (that is fixed) with the triangle vertexes $< (x^0_{p_i}, y^0_{p_i})^T$, $(x^0_{p_j}, y^0_{p_j})^T$, $(x^0_{p_k}, y^0_{p_k})^T >$ and the current projected mesh $s_{\mathbf{p}}$ with the triangles vertexes coordinates $< (x_{p_i}, y_{p_i})^T$, $(x_{p_j}, y_{p_j})^T$, $(x_{p_k}, y_{p_k})^T >$, being $(i, j, k = \#\text{ triangles})$.

The barycentric coordinates α, β, used in eq.3, are given by

$$\alpha = \frac{(x_p - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y_p - y^0_{p_i})(x^0_{p_k} - x^0_{p_i})}{(x^0_{p_j} - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y^0_{p_j} - y^0_{p_i})(x^0_{p_k} - x^0_{p_i})} \tag{A.1}$$

$$\beta = \frac{(y_p - y^0_{p_i})(x^0_{p_j} - x^0_{p_i}) - (x_p - x^0_{p_i})(y^0_{p_j} - y^0_{p_i})}{(x^0_{p_j} - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y^0_{p_j} - y^0_{p_i})(x^0_{p_k} - x^0_{p_i})}, \tag{A.2}$$

and the eqs. 3, A.1, A.2, 1 and 2 can be combined into a single per-triangle affine warp, as

$$\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}) = (a_1 + a_2 x_p + a_3 y_p, a_4 + a_5 x_p + a_6 y_p)^T \tag{A.3}$$

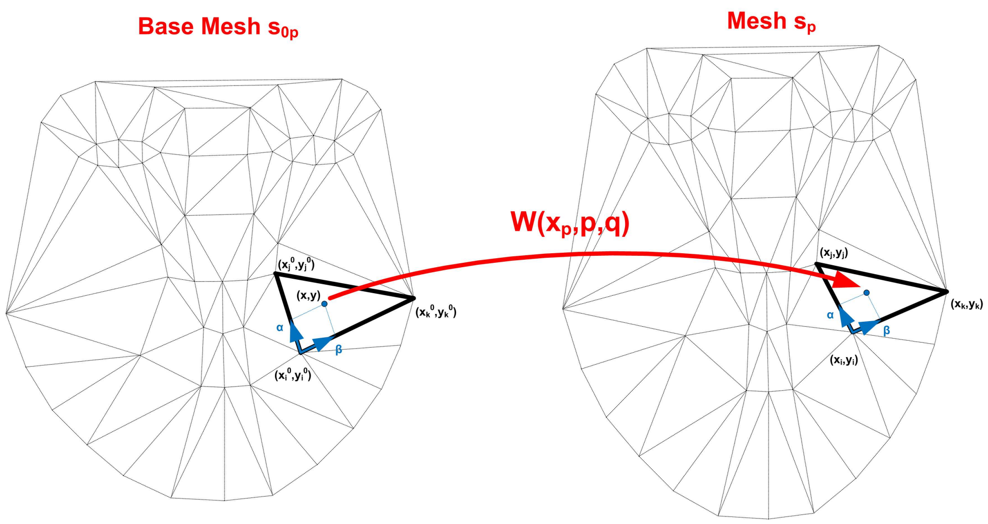

Figure A.1: Computing the piecewise affine warp $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$. Each pixel $\mathbf{x_p}$ belonging to a given triangle $< (x^0_{p_i}, y^0_{p_i})^T$, $(x^0_{p_j}, y^0_{p_j})^T$, $(x^0_{p_k}, y^0_{p_k})^T >$ in the projected base mesh $s_{0\mathbf{p}}$ is mapped to the correspondent triangle $< (x_{p_i}, y_{p_i})^T$, $(x_{p_j}, y_{p_j})^T$, $(x_{p_k}, y_{p_k})^T >$ of the current projected mesh $s_\mathbf{p}$ using barycentric coordinates (α, β).

where a_1, a_2, a_3, a_4, a_5 and a_6 are the affine parameters that are given by

$$\begin{aligned}
a_1 &= (x_{p_i}(x^0_{p_j} y^0_{p_k} - y^0_{p_j} x^0_{p_k}) + x^0_{p_i}(x_{p_k} y^0_{p_j} - y^0_{p_k} x_{p_j}) + y^0_{p_i}(x^0_{p_k} x_{p_j} - x^0_{p_j} x_{p_k}))/\Delta \\
a_2 &= (y^0_{p_k}(x_{p_j} - x_{p_i}) + y^0_{p_i}(x_{p_k} - x_{p_j}) + y^0_{p_j}(x_{p_i} - x_{p_k}))/\Delta \\
a_3 &= (x^0_{p_k}(x_{p_i} - x_{p_j}) + x^0_{p_j}(x_{p_k} - x_{p_i}) + x^0_{p_i}(x_{p_j} - x_{p_k}))/\Delta \\
a_4 &= (y_{p_i}(x^0_{p_j} y^0_{p_k} - y^0_{p_j} x^0_{p_k}) + x^0_{p_i}(y_{p_k} y^0_{p_j} - y^0_{p_k} y_{p_j}) + y^0_{p_i}(x^0_{p_k} y_{p_j} - x^0_{p_j} y_{p_k}))/\Delta \\
a_5 &= (y^0_{p_k}(y_{p_j} - y_{p_i}) + y^0_{p_i}(y_{p_k} - y_{p_j}) + y^0_{p_j}(y_{p_i} - y_{p_k}))/\Delta \\
a_6 &= (x^0_{p_k}(y_{p_i} - y_{p_j}) + x^0_{p_j}(y_{p_k} - y_{p_i}) + x^0_{p_i}(y_{p_j} - y_{p_k}))/\Delta \\
&\text{with} \\
\Delta &= (x^0_{p_j} - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y^0_{p_j} - y^0_{p_i})(x^0_{p_k} - x^0_{p_i}).
\end{aligned} \tag{A.4}$$

The affine parameters $a_1, \ldots, a_6$ need only to be computed once per triangle, not once per pixel. Also, and since the projected base mesh is fixed (i.e. there is always a constant warping frame), a lookup table that encodes the triangle identity speeds up the entire warping procedure.

Figure A.2 shows a warping example from an input image $\mathbf{I}(\mathbf{x_p})$ to $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$ using the warp $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$ and the described triangle lookup table.

(a) Input image $\mathbf{I}(\mathbf{x_p})$ (b) $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$ (c) Triangle lookup table

Figure A.2: Piecewise affine warping example. a) Shows the input image $\mathbf{I}(\mathbf{x_p})$. b) The warped image $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$ using the warp $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$. c) The triangle lookup table that encodes the triangle identity. Each pixel position holds the number of the triangle it belongs to.

The algorithm 1 summarizes this section by showing the list of steps required to perform the piecewise affine warp.

1 **Precompute:** The triangle lookuptable (see figure A.2-c)

2 Evaluate the current mesh s from **p** and **q** using eq.1

3 Find the full perspective mesh projection $s_{\mathbf{p}}$ with eq.2

4 Compute the affine parameters $(a_1, a_2, a_3, a_4, a_5, a_6)$ for each triangle using eqs.A.4

5 For each pixel $\mathbf{x_p}$ in the projected base mesh $s_{0_\mathbf{p}}$, lookup the triangle where $\mathbf{x_p}$ lies in and then lookup the corresponding values of $(a_1, \ldots, a_6)$

6 Evaluate $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$ from eq.A.3 and bilinear interpolate to find $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$

Algorithm 1: Piecewise affine warp.

Appendix B. SFA Derivation

The nonlinear optimization

$$\arg\min_{\mathbf{p},\mathbf{q},\boldsymbol{\lambda}} \sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \left[\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) \right]^2 \tag{B.1}$$

can be solved by gradient descent using additive updates to the parameters as

$$\sum_{\mathbf{x}\in s_{0\mathbf{p}}} [\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} (\lambda_i + \Delta\lambda_i)\mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p} + \Delta\mathbf{p}, \mathbf{q} + \Delta\mathbf{q}))]^2. \tag{B.2}$$

Using a first order Taylor expansion, the last term can be expressed as

$$\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p} + \Delta\mathbf{p}, \mathbf{q} + \Delta\mathbf{q})) \approx \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) + \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{p}}\Delta\mathbf{p} + \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{q}}\Delta\mathbf{q}, \tag{B.3}$$

and, the chain rule can be used on part of the second term of eq.B.3, giving

$$\frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{p}} = \left[\frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial x}\frac{\partial \mathbf{W}_x(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}} + \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial y}\frac{\partial \mathbf{W}_y(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}}\right]. \tag{B.4}$$

Rearranging the terms, results

$$\frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{p}} = \underbrace{\left[\begin{array}{cc} \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial x} & \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial y} \end{array}\right]}_{\nabla \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))} \underbrace{\left[\begin{array}{ccc} \frac{\partial \mathbf{W}x(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_1} & \dots & \frac{\partial \mathbf{W}x(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_n} \\ \frac{\partial \mathbf{W}y(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_1} & \dots & \frac{\partial \mathbf{W}y(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_n} \end{array}\right]}_{\text{Jacobian of the Warp } \frac{\partial \mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}}}, \tag{B.5}$$

being $\nabla \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$ the gradients of the image $\mathbf{I}(\mathbf{x_p})$ evaluated at $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$ and the term $\frac{\partial \mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}}$ the Jacobian of the warp w.r.t. the shape parameters, $\mathbf{p}$.

Similarly for the pose parameters, $\mathbf{q}$, part of the last term of eq.B.3 can be written as

$$\frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{q}} = \nabla \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))\frac{\partial \mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{q}}. \tag{B.6}$$

Finally the eq.B.2, can be written as

$$\sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \left[\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i\mathbf{A}_i(\mathbf{x_p}) + \sum_{i=1}^{m+2} \Delta\lambda_i\mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) - \nabla \mathbf{I}\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\Delta\mathbf{p} - \nabla \mathbf{I}\frac{\partial \mathbf{W}}{\partial \mathbf{q}}\Delta\mathbf{q}\right]^2. \tag{B.7}$$

Appendix C. Building The 3D PDM From Stereo Data

The 3D shape model (PDM) can be acquired from several ways such as using laser range scans[1], time-of-flight (ToF) cameras[2], structure from motion (SfM) techniques[3][4] and of course multi-camera networks. The 3D PDM used in this work, was built using a fully calibrated stereo system where the 2D shape on each view was extracted by fitting a 2D AAM[5] using $v = 58$ landmarks. See figure C.3.

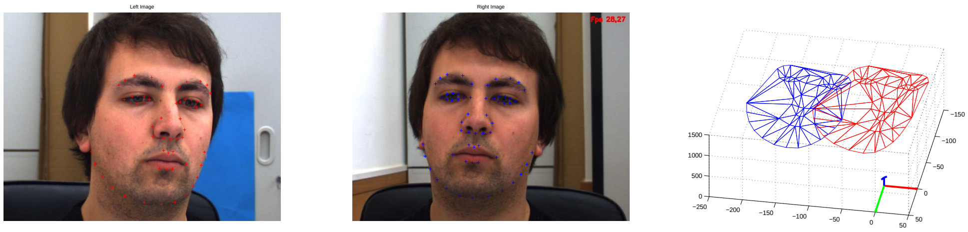

Figure C.3: Left and right images captured by a calibrated stereo system. Each shape annotation results from applying a 2D AAM. The 3D recovered structures (for each camera) are shown on the right picture. Red and blue colors respectively.

The classical triangulation algorithm was used to recover the 3D structure for each view. In short, the triangulation algorithm consists in finding the depths Z_l and Z_r from the normalized perspective projections $(x_l, y_l) = (\frac{X_l}{Z_l}, \frac{Y_l}{Z_l})$ and $(x_r, y_r) = (\frac{X_r}{Z_r}, \frac{Y_r}{Z_r})$ with (X_l, Y_l, Z_l) and (X_r, Y_r, Z_r) being the coordinates of the same 3D point in the left and right camera frame, all this, knowing the rotation $\mathbf{R}$ and translation $\mathbf{t}$ between cameras. The least-squares solution, using all the v points in each shape annotation, is given by

$$\begin{bmatrix} Z_{l_1} & \cdots & Z_{l_v} \\ Z_{r_1} & \cdots & Z_{r_v} \end{bmatrix} = \left[-\mathbf{R} \begin{pmatrix} x_{r_1} & \cdots & x_{r_v} \\ y_{r_1} & \cdots & y_{r_v} \\ 1 & \cdots & 1 \end{pmatrix} \quad \begin{pmatrix} x_{l_1} & \cdots & x_{l_v} \\ y_{l_1} & \cdots & y_{l_v} \\ 1 & \cdots & 1 \end{pmatrix} \right]^{\dagger} \begin{bmatrix} \mathbf{t} & \cdots & \mathbf{t} \end{bmatrix}. \tag{C.1}$$

Using eqs.C.1, the 3D shape mesh samples from pairs of 2D image annotations can be retrieved, as illustrated in figure C.3. However, these mesh coordinates are expressed w.r.t. the camera coordinate frame and therefore the user head rotations are not correctly modeled. To overcome this problem, the PDM was converted into the base pose $(\mathbf{R}_0, \mathbf{t}_0)$ coordinate frame (as included in eq.2)[1], by firstly removing the mean from s_0, centering the mean shape around de origin[2] and then $\mathbf{R}_0$ and $\mathbf{t}_0$ were found by solving the following

[1]Expressing the PDM w.r.t. another coordinate frame requires only changes on the rigid motion (s_0).

[2]It would be convenient to center s_0 around the neck axis, where the true head rotations are made. However, estimating the true neck coordinate frame is not in the scope of this work. We simply move the center of gravity of s_0 back and down 50mm as $s_0 \leftarrow (s_0^{x_i}, s_0^{y_i} - 50, s_0^{z_i} - 50), i = 1, \ldots, v$.

optimization problem:

$$\arg\min_{\theta,\gamma,t_z} \mathbf{K} \left[\mathbf{R}_{pan}(\theta)\mathbf{R}_{roll}(\gamma) \quad \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix} \right] \begin{bmatrix} s_0^{x_1} \cdots s_0^{x_v} \\ s_0^{y_1} \cdots s_0^{y_v} \\ s_0^{z_1} \cdots s_0^{z_v} \\ 1 \cdots 1 \end{bmatrix} \tag{C.2}$$

where $\mathbf{R}_{pan}(\theta)$ and $\mathbf{R}_{roll}(\gamma)$ represent the pan and roll rotations matrices by θ and γ amount, respectively, that changes the 3D orientation of s_0. The t_z parameter is the translation along the camera optical axis from the centroid of the mean shape s_0.

The optimization in eq.C.2 is performed in four steps. First t_z is found by setting a desirable 2D mesh projection width over the image plane (p.e. 200 pixels) holding θ and γ equal to zero. This width value defines the base mesh projection size that is related to all the fitting algorithms computational complexity. The base mesh projection size define the constant warping frame described in the texture model section and consequently the size of all the Steepest Descent images. Then θ and γ are optimized independently in order to hold a symmetric mesh projection. A symmetric shape is desirable to balance the model fitting, otherwise the AAM will perform better for user head rotations where the texture model holds more pixels.

Finally, the last step consist in optimize again for t_z using the previously found values of θ and γ, just to hold the desirable 2D mesh projection width. The base pose is then given by

$$\mathbf{R}_0 = \mathbf{R}_{pan}(\theta)\mathbf{R}_{roll}(\gamma) \text{ and } \mathbf{t}_0 = \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix}. \tag{C.3}$$

References

[1] V.Blanz, T.Vetter, A morphable model for the synthesis of 3d faces, in: SIGGRAPH 99, 1999, pp. 187–194.

[2] Y. Cuil, S. Schuon, D. Chan, S. Thrun, C.Theobalt, 3d shape scanning with a time-of-flight camera, in: IEEE Computer Vision and Pattern Recognition, 2010.

[3] C.Tomasi, T.Kanade, Shape and motion from image streams under orthography -a factorization method, International Journal on Computer Vision 9 (1) (1992) 137–154.

[4] J.Xiao, J.Chai, T.Kanade, A closed-form solution to non-rigid shape and motion recovery, in: European Conference on Computer Vision, 2004.

[5] I.Matthews, S.Baker, Active appearance models revisited, International Journal of Computer Vision 60 (1) (2004) 135–164.