# SINGLE VIEW HEAD POSE ESTIMATION

*Pedro Martins, Jorge Batista*

Institute for Systems and Robotics
Dep. of Electrical Engineering and Computers
University of Coimbra - Portugal

## ABSTRACT

This work addresses the problem of human head pose estimation from single view images. 3D rigid head pose is estimated combining Active Appearance Models(AAM) with Pose from Orthography and Scaling with ITerations(POSIT). AAM shape landmarks are tracked over time and used in POSIT for pose estimation. A statistical anthropometric 3D model is used as reference. Several experiences were performed comparing our approach with a planar ground truth. It was achieved average standard deviations about 2 degrees in orientation and 1 centimeter in distance.

***Index Terms***— Active Appearance Models, POSIT, Anthropometric Model

## 1. INTRODUCTION

In many Human Computer Interface(HCI) applications such as face recognition systems, teleconference, knowledge about gaze direction, video compression, etc, an accurate head pose (position and orientation) estimation is an important issue.

Traditionally there exists two classes of single view head pose estimation approaches: local methods that estimate the head pose [1] [2] from correspondences between image features and a model in order to extract the position and orientation of the subject, and global approaches that use the entire image to estimate head pose by template matching using several methods such as Gabor Wavelet [3] or Support Vector Machines [4]. The principal advantage of these methods is that they rely on just locating the face in the image, but have the disadvantage of relatively pour accuracy when compared to local approaches.

The work presented in this paper deals with the problem of estimate the tridimensional orientation and position of faces using a non-intrusive system. Our approach fits on local methods and is based on consider the human head as a rigid body. A statistical anthropometric 3D model is used combined with Pose from Orthography and Scaling with ITerations(POSIT) [5] algorithm for pose estimation. Since POSIT estimates pose by a set of 3D model points and 2D image projections correspondences, a way to extract facial characteristics is required. AdaBoost [6] is used primarily to locate the face in image and features like the position of eyes, eyebrows, mouth, nose, etc, are acquired using an Active Appearance Model(AAM) [7]. AAM is a statistical template matching method, can be used to track facial characteristics [8] and combined with POSIT solves the model/image registration problem.

This work was funded by FCT Project POSC/EEA-SRI/61150/2004

## 2. ACTIVE APPEARANCE MODELS

AAM is a statistical based segmentation method, where the variability of shape and texture is captured from a dataset. Building such a model allows the generation of new instances with photorealistic quality. In the search phase the model is adjusted to the target image by minimizing the texture residual. For futher details refer to [7].

### 2.1. Shape Model

The shape is defined as the quality of a configuration of points which is invariant under Euclidian Similarity transformations [9]. This landmark points are selected to match borders, vertexes, profile points, corners or other features that describe the shape. The representation used for a single $n$-point shape is a $2n$ vector given by $\mathbf{x} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)^T$. With $N$ shape annotations, follows a statistical analysis where the shapes are previously aligned to a common mean shape using a Generalised Procrustes Analysis (GPA) removing location, scale and rotation effects. Applying a Principal Components Analysis (PCA), we can model the statistical variation with $\mathbf{x} = \overline{\mathbf{x}} + \Phi_s \mathbf{b}_s$, where new shapes $\mathbf{x}$, are synthesised by deforming the mean shape, $\overline{\mathbf{x}}$, using a weighted linear combination of eigenvectors of the covariance matrix, $\Phi_s$. $\mathbf{b}_s$ is a vector of shape parameters which represents the weights. $\Phi_s$ holds the $t_s$ most important eigenvectors that explain a user defined variance.

### 2.2. Texture Model

For $m$ pixels sampled, the texture is represented by the vector $\mathbf{g} = [g_1, g_2, \ldots, g_{m-1}, g_m]^T$. Building a statistical texture model, requires warping each training image so that the control points match those of the mean shape. In order to prevent holes, the texture mapping is performed using the reverse map with bilinear interpolation correction. The texture mapping is performed, using a piece-wise affine warp, i.e. partitioning the convex hull of the mean shape by a set of triangles using the Delaunay triangulation. Each pixel inside a triangle is mapped into the correspondent triangle in the mean shape using barycentric coordinates, see figure 1. This procedure removes differences in texture due shape changes, establishing a common texture reference frame. To reduce the influence of global lighting variation a scaling, $\alpha$ and offset, $\beta$ is applied $\mathbf{g}_{norm} = (\mathbf{g}_i - \beta.\mathbf{1})/\alpha$. After the normalization we get

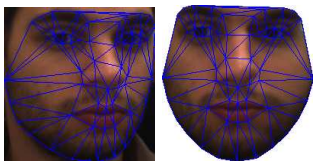


**Fig. 1**. Texture mapping example. Left) Original. Right) Warped texture

$\mathbf{g}_{norm}^T.1 = 0$ and $|\mathbf{g}_{norm}| = 1$. A texture model can be obtained by applying a low-memory PCA(since $m >> N$) on the normalized textures $\mathbf{g} = \bar{\mathbf{g}} + \Phi_g \mathbf{b}_g$, where $\mathbf{g}$ is the synthesized texture, $\bar{\mathbf{g}}$ is the mean texture, $\Phi_g$ contains the $t_g$ highest covariance texture eigenvectors and $\mathbf{b}_g$ is a vector of texture parameters.

### 2.3. Combined Model

The shape and texture from any training example is described by the parameters $\mathbf{b}_s$ and $\mathbf{b}_g$. To remove correlations between shape and texture model parameters a third PCA is performed to the following data:

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \Phi_s^T (\mathbf{x} - \bar{\mathbf{x}}) \\ \Phi_g^T (\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix} \quad (1)$$

Where $\mathbf{W}_s$ is a diagonal matrix of weights that measures the unit difference between shape and texture parameters. A simple estimate for $\mathbf{W}_s$ is to weight uniformly with ratio, $r$, of the total variance in texture and shape, i.e. $r = \sum_i \lambda_{gi} / \sum_i \lambda_{si}$. Then $\mathbf{W}_s = r\mathbf{I}$ [10]. As result, using again a PCA, $\Phi_c$ holds the $t_c$ highest eigenvectors, and we obtain the combined model $\mathbf{b} = \Phi_c \mathbf{c}$. Due the linear nature for the model, is possible to express shape, $\mathbf{x}$, and texture, $\mathbf{g}$, using the combined model by:

$$\mathbf{x} = \bar{\mathbf{x}} + \Phi_s \mathbf{W}_s^{-1} \Phi_{c,s} \mathbf{c} \quad and \quad \mathbf{g} = \bar{\mathbf{g}} + \Phi_g \Phi_{c,g} \mathbf{c} \quad (2)$$

where

$$\Phi_c = \begin{pmatrix} \Phi_{cs} \\ \Phi_{cg} \end{pmatrix} \quad (3)$$

$\mathbf{c}$ is a vector of appearance controlling both shape and texture. One AAM instance is built by generating the texture in the normalized frame and warping-it to the control points given by eq. 2. See figure 2.

**Fig. 2**. Building a AAM instance. Left) Shape control points. Center) Texture in normalized frame. Right) AAM instance.

### 2.4. Model Training

An AAM search can be treated as an optimization problem, where the texture difference between a model instace and a target image is minimized, $|\mathbf{I}_{image} - \mathbf{I}_{model}|^2$ updating the appearance parameters $\mathbf{c}$ and pose. This problem can be solved by learning offline the relation between the texture residual and the correspondent parameters change [7]. Additionally, are considered the similarity parameters for represent the 2D pose. To maintain linearity and keep the identity transformation at zero, these pose parameters are redefined to: $\mathbf{t} = (s_x, s_y, t_x, t_y)^t$ where $s_x = (s\cos(\theta) - 1)$, $s_y = s\sin(\theta)$ represents a combined scale, $s$, and rotation, $\theta$. The remaining parameters $t_x$ and $t_y$ are the usual translations. Now the complete model parameters, $\mathbf{p}$, (a $t_p = t_c + 4$ vector) are given by $\mathbf{p} = (\mathbf{c}^T | \mathbf{t}^T)^T$.

The initial AAM formulation uses the multivariate linear regression approach over the set of training texture residuals, $\delta\mathbf{g}$, and the correspondent model perturbations, $\delta\mathbf{p}$. The goal is to get the optimal prediction matrix, in the least square sense, satisfying the linear relation $\delta\mathbf{p} = \mathbf{R}\delta\mathbf{g}$. Later [7] it was suggested a better method, computing the gradient matrix $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$. The texture residual vector is defined as $\mathbf{r}(\mathbf{p}) = \mathbf{g}_{image}(\mathbf{p}) - \mathbf{g}_{model}(\mathbf{p})$ where the goal is to find the optimal update at model parameters to minimize $|\mathbf{r}(\mathbf{p})|$. A first order Taylor expansion leads to $\mathbf{r}(\mathbf{p} + \delta\mathbf{p}) \approx \mathbf{r}(\mathbf{p}) + \frac{\partial \mathbf{r}(\mathbf{p})}{\partial \mathbf{p}} \delta\mathbf{p}$, by minimizing in the least square sense, gives

$$\delta\mathbf{p} = -\left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T\right)^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \mathbf{r}(\mathbf{p}) \quad with \quad \mathbf{R} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}}\right)^{\dagger} \quad (4)$$

$\delta\mathbf{p}$ in eq. 4 gives the parameters probable update to fit the model. Regard that, since the sampling is always performed at the reference frame, the prediction matrix, $\mathbf{R}$, is considered fixed and it can be only estimated once.

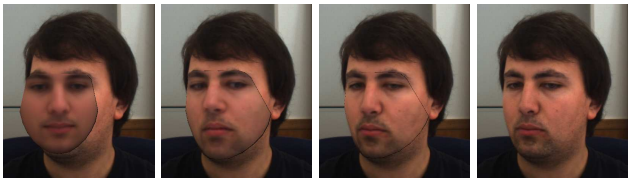
### 2.5. Iterative Model Refinement

For a given estimate $\mathbf{p}_0$, the model can be fitted by

**Algorithm 1** Iterative Model Refinement

Sample image at $\mathbf{x} \rightarrow \mathbf{g}_{image}$
Build an AAM instance AAM($\mathbf{p}$) $\rightarrow \mathbf{g}_{model}$
Compute residual $\delta\mathbf{g} = \mathbf{g}_{image} - \mathbf{g}_{model}$
Evaluate Error $E_0 = |\delta\mathbf{g}|^2$
Predict model displacements $\delta\mathbf{p} = \mathbf{R}\delta\mathbf{g}$
Set $k = 1$
Establish $\mathbf{p}_1 = \mathbf{p}_0 - k\delta\mathbf{p}$
If $|\delta\mathbf{g}_1|^2 < E_0$ accept $\mathbf{p}_1$ Else try $k = 1.5, k = 0.5, k = 0.25$

this procedure is repeated until no improvement is made to error $|\delta\mathbf{g}|$. Figure 3 shows a successful AAM search. Notice that, as better the initial estimate is, minor the risk of being trap in a local minimum. In this work AdaBoost [6] method is used to locate human faces.



(a) $1^{st}$ (b) $5^{th}$ (c) Final (d) Original

**Fig. 3**. Iterative model refinement.

## 3. POSE ESTIMATION METHOD

Pose from Orthography and Scaling with ITerations(POSIT) [5] is a fast and accurate, iterative algorithm for finding the pose (orientation and translation) of an 3D model or scene with respect to a camera given a set of 2D image and 3D object points correspondences.

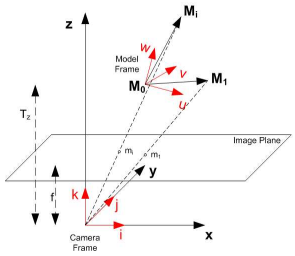



**Fig. 4**. Perspective projections $m_i$ for model points $M_i$

Figure 4 shows the pinhole camera model, with its center of projection $O$ and image plane at the focal length $f$ (focal Lent and image center are assumed to be known). In the camera referential the unit vectors are $i$, $j$ and $k$. A 3D model with feature points $M_0, M_1 \ldots, M_i, \ldots, M_n$ is positioned at camera *frustrum*. The model coordinate frame is centered at $M_0$ with unit vectors $u$, $v$ and $w$. A $M_i$ point has known coordinates $(U_i, V_i, W_i)$ in the model frame and unknown coordinates $(X_i, Y_i, Z_i)$ in the camera frame. The projections of $M_i$ are known and called $m_i$, having image coordinates $(x_i, y_i)$.

The pose matrix $\mathbf{P}$ gives the rigid transformation between the model and the camera frame:

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{T} \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{P}_3 & \mathbf{P}_4 \end{array}\right]^T \quad (5)$$

where $\mathbf{R}$ is the rotation matrix representing the orientation of the camera frame with respect to the model frame, $\mathbf{T} = (T_x, T_y, T_z)$ is the translation vector from the camera center to the model frame center. $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ and $\mathbf{P}_4$ are defined as the pose matrix rows.The rotation matrix $\mathbf{R}$ is the matrix whose rows are the coordinates of the unit vectors $(i, j, k)$ of camera frame expressed in the model coordinate frame $(M_0u, M_0v, M_0w)$. The rotation matrix transforms model coordinates of vectors $M_0M_i$ into coordinates defined in the camera system, for instance, the dot product $M_oM_i \cdot i$ between the vector $M_0M_i$ and the first row of the rotation matrix, provides the projection of this vector on the unit vector of the camera system, i.e. the coordinate $X_i$. To full compute $\mathbf{R}$ is only needed to compute $i$ and $j$ since $k = i \times j$. The translation vector $\mathbf{T}$ is the vector $OM_0$, has coordinates $(X_0, Y_0, Z_0)$ and is aligned with the vector $Om_o$, so, $\mathbf{T} = \frac{Z_0}{f} Om_0$. To compute the model translation form the camera center its just need $Z_0$ coordinate. Knowing $i$, $j$ and $Z_0$ the model pose becomes fully defined.

In a perspective projection model, a 3D point $(X_i, Y_i, Z_i)$ is projected in the image by $\left(x_i = f\frac{X_i}{Z_i}, y_i = f\frac{Y_i}{Z_i}\right)$. Under *weak perspective* (or also known *scaled orthographic*) projection model [11], a 3D image point projection can be written as $\left(x_i = \frac{f}{(1+\epsilon)}\frac{X_i}{T_z}, y_i = \frac{f}{(1+\epsilon)}\frac{Y_i}{T_z}\right)$. In scaled orthographic projection, a vector $M_0M_i$ in the model frame is projected by an orthographic projection over the plane $z = T_z$ followed by a perspective projection. The projected vector in the image plane has a scaling factor equals to $\frac{f}{Z_0}$.

### 3.1. Fundamental Equations

Defining the 4D vectors $\mathbf{I} = \frac{f}{T_z}\mathbf{P}_1$ and $\mathbf{J} = \frac{f}{T_z}\mathbf{P}_2$ and knowing that $(1 + \epsilon_i) = \frac{Z_i}{T_z}$, the fundamental equations that relate the row vectors $\mathbf{P}_1$, $\mathbf{P}_2$ of the pose matrix, the coordinates of the model features $M_0M_i$ and the coordinates $(x_i, y_i)$ from the correspondent images $m_i$ are:

$$M_0M_i \cdot \mathbf{I} = x'_i, \quad M_0M_i \cdot \mathbf{J} = y'_i \quad (6)$$

$$\mathbf{I} = \frac{f}{T_z}\mathbf{P}_1, \quad \mathbf{J} = \frac{f}{T_z}\mathbf{P}_2 \quad (7)$$

$$x'_i = x_i(1 + \epsilon_i), \quad y'_i = y_i(1 + \epsilon_i) \quad (8)$$

$$\epsilon_i = \mathbf{P}_3 \cdot M_0M_i/T_z - 1. \quad (9)$$

If values are given for $\epsilon_i$, eqs. 6 provide a linear system of equations with unknowns $\mathbf{I}$ and $\mathbf{J}$. Unit vectors $i$ and $j$ are found by normalizing $\mathbf{I}$ and $\mathbf{J}$. $T_z$ is obtained by the norms of either $\mathbf{I}$ and $\mathbf{J}$. This approach is called Pose from Orthography and Scaling(POS) [5], i.e. finding pose for fixed values of $\epsilon_i$. Once $i$ and $j$ have been computed, more refined values for $\epsilon_i$ can be found using again POS.

This method does not require an initial pose estimate, is very fast and robust with respect to image measurements and camera calibration errors, but in its original formulation it is required that the model origin image $m_0$ should be located. This means that we have restrictions building the 3D model. This situation can be solved by using POSIT in homogeneous form [12].

## 4. HEAD POSE ESTIMATION

Our framework is composed by the two parts previously described. The first part consists on AAM model fit for a given subject performing features tracking. The features used in this context are the AAM shape model landmarks location on the image over time. Notice that no temporal filter is used.

The second part is the head pose estimation using POSIT. By tracking features in each video frame combined with the landmark-based nature of AAMs we solve directly the image/3Dmodel registration problem.

As 3D model we use an anthropometric 3D rigid model of the human head. This is the best suitable rigid body model used to describe the face of several individuals and it was acquired by a frontal laser 3D scan of an physical model, selecting the equivalent 3D points of the AAM annotation procedure creating a sparse 3D model. Figure 5 illustrates this procedure.

## 5. EXPERIMENTAL RESULTS

The orientation of the estimated pose is represented by the Roll, Pitch and Yaw (RPY) angles. Figure 6 shows some samples of pose estimation. The pose estimated is represented by an animated 3DOF rotational OpenGL model showed at images top right. The evaluation of pose estimation accuracy is performed comparing the pose estimated by our framework with the estimated value obtained with the planar checkboard

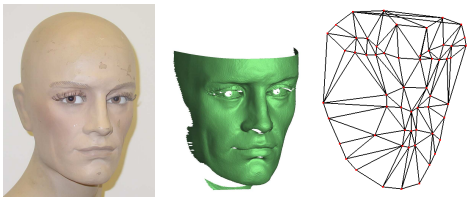
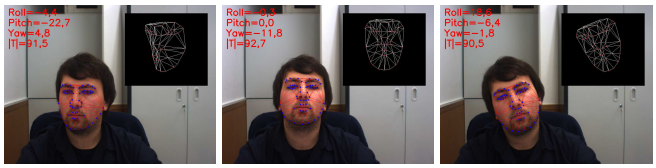


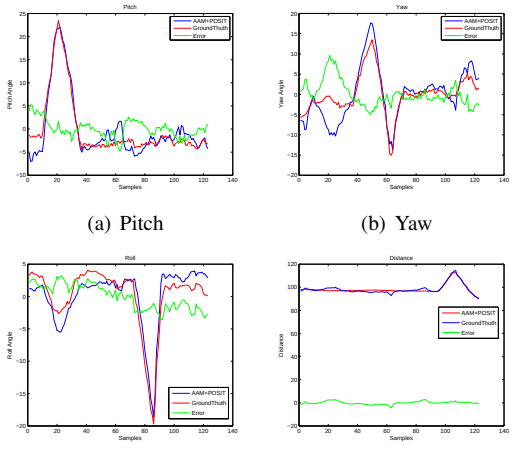
**Fig. 5**. Left) Physical anthropometric model. Center) 3D laser scan data acquired Right) Sparse OpenGL model built model using the AAM shape features.




(a) Pitch variation (b) Yaw variation (c) Roll variation

**Fig. 6**. Samples of pose estimation.

used as ground truth values. Figure 7 presents results from the pose estimated during a video sequence where the subject performs several human head movements, ranging from yaw, pitch and roll head rotations of several degrees. The experience begins by rotating head left, changing pitch angle, and recovering to frontal position, followed by a yaw angle, moving head up and down and again recovering to frontal position, and finally performing a head roll rotation. Near the end, after frame 95 the distance from camera is also changed. The individual parameters (Pitch, Yaw, Roll and distance) results are presented in figure 7-a, 7-b, 7-c and 7-d respectively. The graphical results show some correlations between Pitch and Yaw angles that result from the differences between the subject and the rigid 3D anthropometric model used.




(a) Pitch (b) Yaw

(c) Roll (d) Distance

**Fig. 7**. Angle Results.

Table 1 displays the error and average standard deviations of the pose parameters for several similar performed experiences with different individuals.

**Table 1**. Error standard deviation. The angle parametes are in degrees and the distance in centimeters.

| **Param.** | **Experiences error std** | | | | | | **Avg std** |
|---|---|---|---|---|---|---|---|
| Roll | 1.92 | 1.86 | 1.87 | 2.15 | 2.14 | 1.69 | $1.95^o$ |
| Pitch | 1.91 | 2.46 | 2.10 | 2.94 | 3.23 | 2.81 | $2.57^o$ |
| Yaw | 3.0 | 1.47 | 1.48 | 1.64 | 1.49 | 1.14 | $1.70^o$ |
| Distance | 1.29 | 1.72 | 1.37 | 1.50 | 1.30 | 0.85 | $1.33cm$ |

## 6. CONCLUSIONS

This work describes a single view solution to estimate the head pose of human subjects combining AAM and POSIT. AAM extract in each image frame the landmarks position. These selected features are tracked over time and used in conjunction with POSIT to estimate head pose. Required the use of a 3D rigid model, a statistical anthropometric model is selected since is the most suitable one. One of the major advantage of using combined AAM plus POSIT is that it solves directly the correspondences problem, avoiding the use of registration techniques. An accurate pose estimation is achieved with average standard deviations about 2 degrees in orientation and 1 centimeter in distance and subjects exhibiting a normal expression. The facial expression influence on pose estimation will be analyzed on future work.

## 7. REFERENCES


[1] Alex Waibel Rainer Stiefelhagen, Jie Yang, "A modelbased gaze trackeing system," *IEEE International Joint Symposia on Intelligence and Systems*, 1996.

[2] Shay Ohayon and Ehud Rivlin, "Robust 3d head tracking using camera pose estimation," *International Conference of Pattern Recognition*, 2006.

[3] V. uger, S. Bruns, and G. Sommer, "Efficient head pose estimation with gabor wavelet networks," 2000.

[4] J. Ng and S. Gong, "Multi-view face detection and pose estimation using a composite support vector machine across the view sphere," 1999.

[5] D. DeMenthon and L.S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, 1995.

[6] P.Viola and M. Jones, "Rapid object detection using a boosted cascate of simple features," *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[7] G.J. Edwards T.F.Cootes and C.J.Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[8] J. Ahlberg, "An active model for facial feature tracking," *EURASIP Journal on Applied Signal Processing*, 2002.

[9] T.F.Cootes and C.J.Taylor, "Statistical models of appearance for computer vision," Tech. Rep., Imaging Science and Biomedical Engineering - University of Manchester, 2004.

[10] Mikkel Bille Stegmann, "Active appearance models theory, extensions & cases," M.S. thesis, IMM Technical Univesity of Denmark, 2000.

[11] Ramani Duraiswami Philip David, Daniel DeMenthon and Hanan Samet, "Simultaneous pose and correspondence determination using line features," .

[12] D. DeMenthon, "Recognition and tracking of 3d objects by 1d search," .