Monocular Head Pose Estimation

Pedro Martins, Jorge Batista

Institute for Systems and Robotics
http://www.isr.uc.pt

Department of Electrical Engineering and Computers
University of Coimbra
Portugal

AI_FI – Advance Interaction using Facial Information
FCT Project POSC/EEA-SRI/61150/2004
http://aifi.isr.uc.pt
Introduction

- Single View 6DOF Pose Estimation
  - Human Computer Interface (HCI)
  - Face Recognition Systems
  - Knowledge about gaze direction
  - Video Compression

Agenda

- Active Appearance Models (AAM)
  - Shape, Texture and Combined Models
  - Model Training
  - Model Fitting

- Monocular Pose Estimation
  - Pose from Orthography and Scaling with Iterations (POSIT)
  - Anthropometric 3D Model
  - Pose Evaluation
  - Augmented Reality
Face Model

A set of input parameters generate a face image output

Shape Model  Texture Model  Combined Model
Active Appearance Models

- Active Appearance Models (AAM) is a statistical based template matching method, where the variability of shape and texture is captured from a representative training set.
- Able to extract relevant face information without background interference
- Describes facial characteristics in a reduced model
Shape Model

- Shape is defined as a Set of Landmarks Points
  - Invariant over Euclidian Similarity transformations
  - No landmark connectivity information is given

\[ x = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)^T \]
Shape Model - Generalized Procrustes Analysis

- Remove location, scale and rotation effects

Raw Data

Aligned Data
Shape Model

- Applying a PCA

\[ x = \bar{x} + \Phi_s b_s \]

- \( x \) is the synthesized shape
- \( \bar{x} \) is the mean shape
- \( \Phi_s \) contains the highest covariance shape eigenvectors
- \( b_s \) is a vector of shape parameters representing the weights
Texture Model

- For $m$ pixels sampled, the texture is represented by:
  \[ g = (g_1, g_2, ..., g_{m-1}, g_m)^T \]

- Required warping each image to a common reference frame
  - Delaunay Triangulation
  - Each pixel is mapped by barycentric coordinates
Hardware Assisted Texture

- Modern graphics cards provide hardware based solutions
- Texture mapping using OpenGL API
- Delaunay Triangles
- Orthographic Projection Model
- Load warped image from the FrameBuffer

<table>
<thead>
<tr>
<th></th>
<th>MatLab</th>
<th>C/C++</th>
<th>OpenGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>2.7 s</td>
<td>200 ms</td>
<td>5 ms</td>
</tr>
</tbody>
</table>
Texture Mapping Video
Texture Mapping Examples
Photometric Normalization

- Histogram Equalization in each of the 3 Color Channels
Texture Model

- Applying a LowMemory PCA

\[ g = \bar{g} + \Phi_g b_g \]

- \( g \) is the synthesized texture
- \( \bar{g} \) is the mean texture
- \( \Phi_g \) contains the highest covariance texture eigenvectors
- \( b_g \) is a vector of texture parameters representing the weights
Combined Shape + Texture Model

- To remove correlations between bs and bg a third PCA is performed

\[
 b = \left( \frac{W_s b_s}{b_g} \right) = \left( \frac{W_s \Phi_s^T(x - \bar{x})}{\Phi_g^T(g - \bar{g})} \right)
\]

- Uniformly weight with ratio \( r \)

\[
 W_s = rI \quad r = \frac{\sum \lambda_{si}}{\sum \lambda_{sj}}
\]

- Combined model

\[
 b = \Phi_c c
\]
AAM Instance Examples
AAM Model Training

- **Optimization Problem**
  - Minimize texture difference between mode and the beneath part of the target image that it covers

- **Include pose parameters**
  \[ t = \begin{bmatrix} S_x & S_y & T_x & T_y \end{bmatrix}, S_x = s \cos(\theta) - 1, S_y = s \sin(\theta) \]

- **Full parameters**
  \[ p = \begin{bmatrix} c^T & t^T \end{bmatrix} \]

- **Learning the correlations between AAM model instances and texture residuals**

Find the optimal prediction matrix
\[ \delta p = R\delta g \]
AAM Model Training(2)

- Residual \( r(p) = g_{image} - g_{model} \)
- Minimize \( |r(p)|^2 \)
- Expanding in Taylor Series \( r(p + \delta p) \approx r(p) + J\delta p \)
- So \( |r(p + \delta p)|^2 \) leads to

\[
\delta p = -(J^T J)^{-1} J^T r
\]

\[
J = \begin{bmatrix}
\frac{\delta r_1}{\delta p_1} & \ldots & \frac{\delta r_1}{\delta p_t} \\
\vdots & \ddots & \vdots \\
\frac{\delta r_m}{\delta p_1} & \ldots & \frac{\delta r_m}{\delta p_t}
\end{bmatrix}_{m \times t_p}
\]

\[
J = \frac{\delta r}{\delta p} = \Delta g \cdot \Delta p^{-1}
\]
AAM Model Fitting

- AdaBoost Initial Location Estimate
- Damped Gauss-Newton Steepest Descent method

Sample Image
\((x, y) \in g_{image}\)

Build AAM Instance
\(AAM(p) = (x_{model}, y_{model}, g_{model})\)

Compute Texture Residual
\(\delta g = g_{image} - g_{model}\)

Update Model Displacements
\(p_{k+1} = p_k + \alpha (J^T J)^{-1} J^T \delta g\)

Until No Improvement is made to the error
IMM Database AAM Fitting
AAM Model Fitting
AAM Model Fitting Failure
Monocular Head Pose Estimation

- Single View Head Pose Estimation
- POSIT - Pose from Orthography and Scaling with Iterations
- Rigid 3D Face Surface Model
**POSIT** - Pose from Orthography and Scaling with Iterations

- Perspective Projection Model

\[
\begin{bmatrix}
   u \\
   v \\
   w \\
\end{bmatrix} =
K
\begin{bmatrix}
   1 & 0 & 0 & 0 \\
   0 & 1 & 0 & 0 \\
   0 & 0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
   r_{11} & r_{12} & r_{13} & T_x \\
   r_{21} & r_{22} & r_{23} & T_y \\
   r_{31} & r_{32} & r_{33} & T_z \\
\end{bmatrix}
\begin{bmatrix}
   X \\
   Y \\
   Z \\
   1 \\
\end{bmatrix}
\]

- Using normalized image coordinates

\[
\begin{bmatrix}
   u \\
   v \\
   w \\
\end{bmatrix} = K^{-1}
\begin{bmatrix}
   u' \\
   v' \\
   w' \\
\end{bmatrix} =
\begin{bmatrix}
   u \\
   v \\
   w \\
\end{bmatrix} =
\begin{bmatrix}
   X \\
   Y \\
   Z \\
   1 \\
\end{bmatrix}
\]

Defining \( r_1, r_2 \) and \( r_3 \) as

\[
\begin{align*}
   r_1 &= \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \end{bmatrix}, \\
   r_2 &= \begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \end{bmatrix}, \\
   r_3 &= \begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \end{bmatrix}
\end{align*}
\]
POSIT - Pose from Orthography and Scaling with Iterations (2)

Dividing all elements by $T_z$

$$\begin{bmatrix}
  u \\
v \\
w
\end{bmatrix} = \begin{bmatrix}
  r_1^T/T_z & T_x/T_z \\
r_2^T/T_z & T_y/T_z \\
r_3^T/T_z & 1
\end{bmatrix}\begin{bmatrix}
  X \\
Y \\
Z
\end{bmatrix}$$

Applying the transpose on the remaining eqs

$$\begin{bmatrix}
u \\
v
\end{bmatrix} = \begin{bmatrix}
  X & Y & Z & 1
\end{bmatrix}\begin{bmatrix}
r_1/T_z & r_2/T_z \\
T_x/T_z & T_y/T_z
\end{bmatrix}$$

Extending for $n$ points

$$\begin{bmatrix}
u_1 & v_1
\end{bmatrix} = \begin{bmatrix}
  X_1 & Y_1 & Z_1 & 1
\end{bmatrix}\begin{bmatrix}
r_1/T_z & r_2/T_z \\
T_x/T_z & T_y/T_z
\end{bmatrix}$$

$$\begin{bmatrix}
u_2 & v_2
\end{bmatrix} = \begin{bmatrix}
  X_2 & Y_2 & Z_2 & 1
\end{bmatrix}\begin{bmatrix}
r_1/T_z & r_2/T_z \\
T_x/T_z & T_y/T_z
\end{bmatrix}$$

$$\vdots$$

$$\begin{bmatrix}
u_{n-1} & v_{n-1}
\end{bmatrix} = \begin{bmatrix}
  X_{n-1} & Y_{n-1} & Z_{n-1} & 1
\end{bmatrix}\begin{bmatrix}
r_1/T_z & r_2/T_z \\
T_x/T_z & T_y/T_z
\end{bmatrix}$$

$$\begin{bmatrix}
u_n & v_n
\end{bmatrix} = \begin{bmatrix}
  X_n & Y_n & Z_n & 1
\end{bmatrix}\begin{bmatrix}
r_1/T_z & r_2/T_z \\
T_x/T_z & T_y/T_z
\end{bmatrix}$$

**POSIT Algorithm**

Normalize Image Coordinates

$$u_i = u_i - \frac{c_x}{f}, v_i = v_i - \frac{c_y}{f}$$

Assume $w_i = 1$

Get Scaled Orthographic coordinates

$$(u_i, v_i) = w_i(u_i, v_i)$$

Compute

$$\begin{bmatrix}
r_1/T_z & r_2/T_z \\
T_x/T_z & T_y/T_z
\end{bmatrix} = M^{-1}\begin{bmatrix}
u_1 & v_1 \\
\vdots & \vdots \\
u_n & v_n
\end{bmatrix}$$

Find $T_z, T_x, T_y, r1$ and $r2$

Compute $r3$ by the cross product

$$r_3 = r_1 \times r_2$$

Update

$$w_i = 1 + \frac{r_3}{T_z}(X_i, Y_i, Z_i)$$

Until Pose Converge
3D Anthropometric Model

Physical Anthropometric Model

3D laser scan data

Sparse 3D model (OpenGL)

One-to-One 2D/3D Correspondences
Head Pose Estimation - Demo
Pose Evaluation – Pose From a Plane

- Knowing the camera matrix, $K$, the Homography holds,

$$H = K[R_1 | R_2 | T]$$

- $R_1$, $R_2$ – first 2 columns of rotation matrix $R$
- $T$ – translation vector

- The full pose can be retrieved using the following normalization

Compute $W = K^{-1}H$

$$R_1 = \frac{W_1}{l} \quad R_2 = \frac{W_2}{l} \quad T = \frac{W_3}{l}$$

$$l = \sqrt{|W_1| \cdot |W_2|}$$

The vectors $c$, $p$ and $d$ are defined as

$$c = R_1 + R_2 \quad p = R_1 \times R_2 \quad d = c \times p$$

$$R_1' = \frac{1}{\sqrt{2}} \left( \frac{c}{|c|} + \frac{d}{|d|} \right) \quad R_2' = \frac{1}{\sqrt{2}} \left( \frac{c}{|c|} - \frac{d}{|d|} \right) \quad R_3' = R_1' \times R_2'$$

$$R = [R_1' \mid R_2' \mid R_3']$$
Pose Estimation Evaluation - Demo
Pose Estimation Evaluation

- AAM+POSIT Head Pose Compared with a planar checkboard pose

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Avg std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll</td>
<td>1.94 deg</td>
</tr>
<tr>
<td>Pitch</td>
<td>2.57 deg</td>
</tr>
<tr>
<td>Yaw</td>
<td>1.7 deg</td>
</tr>
<tr>
<td>Distance</td>
<td>1.33cm</td>
</tr>
</tbody>
</table>

Correlations between Pitch and Yaw angles
3D Glasses Augmentation

- Augmented Reality (AR) is the overlay of artificial computer graphics images on the physical world.
3D Glasses Augmentation - Demo
Final Notes

- Single View Solution to estimate the 6DOF Head Pose
- Combines AAM Features Extracting + POSIT Pose Estimation
- Easy 2D/3D image registration
- Average std errors in about 2 degree in orientation and 1 cm in position

Advantages

- Rigid 3D Head Model
- Identity Differences
- Facial Expression Influence

Weeknesses