Active Shape Model

Xiaoming Liu
Michigan State University

* slides credit to Dr. Cootes
Problem Definition

Moving and deforming a *template* to minimize the *distance* between the template and an image.

Note the difference between image alignment and image registration.
Applications

- Face fitting [Baker & Matthews’ 04IJCV]
- Expression analysis/recognition [Zhu & Ji’ 06CVPR]
- Image coding [Baker et al.’ 04PAMI]
- Tracking [Hager & Belhumeur’ 98PAMI]
- Image mosaicing [Shum & Szeliski’ 00IJCV]
- Medical image interpretation [Mitchell et al.’ 02TMI]
- Dynamic texture [Doretto & Soatto’ 06PAMI]
- Industrial inspection [Rolfe et al.’ 01IME]
Active Shape Models

Suppose we have a statistical shape model
- Trained from sets of examples

How do we use it to interpret new images?

Use an “Active Shape Model”

Iterative method of matching model to image
Building Models

Require labelled training images

- landmarks represent correspondences
Building Shape Models

Given aligned shapes, \( \{ x_i \} \)

Apply PCA

\[
x = \bar{x} + P b
\]

\( P \) – First \( t \) eigenvectors of covar. matrix

\( b \) – Shape model parameters
Hand Shape Model

Varying $b_1$  Varying $b_2$  Varying $b_3$
Active Shape Models

Match shape model to new image

Require:

- Statistical shape model
- Model of image structure at each point
Placing Model in Image

The model points are defined in a model coordinate frame.

Must apply global transformation, $T$, to place in image.

$x = \bar{x} + P_b$

$X = T(\bar{x} + P_b)$
ASM Search Overview

Local optimisation
Initialise near target
• Search along profiles for best match,$X'$
• Update parameters to match to $X'$. 

$(X_i, Y_i)$
Local Structure Models

Need to search for local match for each point

Model

- Strongest edge
- Correlation
- Statistical model of profile
Computing Normal to Boundary

Tangent \((t_x, t_y)\)

Normal \((n_x, n_y) = (-t_y, t_x)\)

\[
(t_x, t_y) \approx \left( \frac{d_x, d_y}{\sqrt{d_x^2 + d_y^2}} \right)
\]

\[
d_x = X_{i+1} - X_{i-1}
\]

\[
d_y = Y_{i+1} - Y_{i-1}
\]

[Unit vector]
Sampling Along Profiles

Model boundary

Model point $(X, Y)$

Profile normal to boundary

Interpolate at these points

$$(X, Y) + i(s_n n_x, s_n n_y)$$

$i = \ldots -2, -1, 0, 1, 2, \ldots$

Take steps of length $s_n$ along $(n_x, n_y)$
Noise Reduction

In noisy images, average orthogonal to profile
• Improves signal-to-noise along profile

Use $g_i = 0.25g_{i1} + 0.5g_{i2} + 0.25g_{i3}$

Sampled profile is
$g = (..., g_{-2}, g_{-1}, g_0, g_1, g_2, ...)$. 
Searching for Strong Edges

\[
\frac{dg(x)}{dx} = 0.5(g(x+1) - g(x-1))
\]

Select point along profile at strongest edge
Profile Models

Sometimes true point not on strongest edge

Model local structure to help locate the point
Statistical Profile Models

Estimate p.d.f. for sample on profile

Normalise to allow for global lighting variations

From training set learn

\[ p(g) \]
Profile Models

For each point in model

• For each training image
  – Sample values along profile
  – Normalise

• Build statistical model
  – eg Gaussian PDF using eigen-model approach
Searching Along Profiles

During search we look along a normal for the best match for each profile.
Search Algorithm

Search along profile

Update global transformation, $T$, and parameters, $b$, to minimise

$$| X - T(\bar{x} + Pb) |^2$$

$(X_i, Y_i)$
Updating Parameters

Find pose and model parameters to minimise

Either

\[ f(b, X_c, Y_c, s, \theta) = |X - T(\bar{x} + Pb; X_c, Y_c, s, \theta)|^2 \]

• Put into general optimiser
• Use two stage iterative approach
Updating Parameters

\[ f(b, X, Y, s, \theta) = \| X - T(\bar{x} + Pb; X, Y, s, \theta) \|^2 \]

Repeat until convergence:

Fix \( b \) and find \((X, Y, s, \theta)\) which minimise \( \| X - T(\bar{x} + Pb) \|^2 \)

Analytic solution exists (see notes)

Fix \((X, Y, s, \theta)\) and find \( b \) which minimises \( \| X - T(\bar{x} + Pb) \|^2 \)

\[ b = P^T (T^{-1}(X) - \bar{x}) \]
Multi-Resolution Search

Train models at each level of pyramid
- Gaussian pyramid with step size 2
- Use same points but different local models

Start search at coarse resolution
- Refine at finer resolution
Lessons learned

ASM is relatively fast

ASM too simplistic; not robust when new images are introduced

May not converge to good solution

Key insight: ASM does not incorporate all gray-level information in parameters
Extensions

Active Appearance Models

Applications
Active Appearance Models (AAM)
Shape Models

\[ s = (x_1, y_1, x_2, y_2, \ldots, x_v, y_v)^T. \]

\[ s = s_0 + \sum_{i=1}^{n} p_i s_i. \]
Appearance Models

\[ A(x) = A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) \quad \forall x \in s_0 \]
Model Instantiation

Appearance, $A$

$A_0 + 3559A_1 + 351A_2 - 256A_3 \ldots$

$W(x; p)$

AAM Model Instance

$M(W(x; p))$

Shape, $s$

$s_0 - 54s_1 + 10s_2 - 9.1s_3 \ldots$
Fitting AAM

\[
\sum_{x \in S_0} \left[ A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(W(x; p)) \right]^2
\]

Initial: 21.8  
3 iterations: 18.0  
6 iterations: 11.9  
10 iterations: 0.69  
15 iterations: 0.09  
20 iterations: 0.09
3DMM

M: Projection matrix

Shape

= + 4x  + 7x

Texture

= - 5x  + 2x
3DMM

$M$: Projection matrix

$$\begin{bmatrix} 1 \\ 4 \\ 7 \\ \vdots \end{bmatrix} \begin{bmatrix} \text{Shape} \\ \text{Texture} \end{bmatrix} = \begin{bmatrix} \text{Shape} \\ \text{Texture} \end{bmatrix}$$

Matrix multiplication = Single fully connected layer
Nonlinear 3DMM

M: Projection matrix

Embedding basis in CNN

MLP

CNN

Texture

Shape
Texture REPRESENTATION

- **Point cloud:**
  - ✓ Completed
  - ✗ High dimension
  - ✗ No/Little spatial relation

- **2D Frontal Face**
  - ✗ Limited information of 2 sides
  - ✓ Convolvable

- **Unwarped 2D texture**
  - ✓ Completed
  - ✓ Convolvable
Nonlinear 3DMM FRAMEWORK

Model fitting

Model learning

Rendering Layer

$I$

$E$

$f_S$

$m$

$L_m$

$D_S$

$D_T$

$S$

$T$

$L_S$

$L_T$

$L_{adv}$

$\hat{I}$

$L_{rec}$
RENDERING LAYER
Framework for Generic Objects

Model fitting

Model learning

Input image

\( I \)

\( E \)

\( f_S \)

\( D_S \)

Occupancy field

Normal

Color Field

\( f_A \)

\( D_A \)

Normal

Reconstruction

Albedo

Shading

Self-supervised Loss
Shape Representation

\[ D_S : \mathbb{R}^3 \times \mathbb{R}^{d_S} \rightarrow V \]

Max pooling

Probability of occupancy
Albedo Representation

$D_S \rightarrow \text{Max pooling} \rightarrow o$

$D_A \rightarrow \text{Max pooling} \rightarrow o$

$k = \text{argmax}_i(o_i)$

Albedo Decoder

<table>
<thead>
<tr>
<th>Branch #1</th>
<th>Branch #2</th>
<th>Branch #3</th>
<th>Branch #4</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Car Image]</td>
<td>![Leaves Image]</td>
<td>![Leaf Image]</td>
<td>![Car Image]</td>
</tr>
</tbody>
</table>

Final shape

Final albedo
Rendering

Camera → Image

View Ray

Surface point $x_j$

$$I_j = A_j \cdot \sum_{b=1}^{B^2} \gamma_b H_b(n_j)$$

$$n_j = \sigma \left( \frac{\delta D_S(x_j)}{\delta x_j} \right)$$

Albedo Decoder

$$A_j = D_A(f_A, f_S, x_j)$$
Model Training

- Supervised prior learning with synthetic images
  - Learning shape decoder
  - Learning albedo decoder and encoder
- Unsupervised joint modeling

\[ \mathcal{L} = \mathcal{L}_{\text{img}} + \lambda_{\text{sil}} \mathcal{L}_{\text{sil}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} \]

\[ D_S(\mathcal{E}_S, \mathcal{E}_P^{-1} \begin{bmatrix} u \\ v \\ d \end{bmatrix}) - \]

\[ 0.5 \]

\[ d \]

\[ 0.5 \]
Experiments - Expressiveness

Latent Space

3D Shape
Experiments --- Single-view 3D Reconstruction

(Pix2Mesh) Wang et al. Pixel2mesh: Generating 3d mesh models from single RGB images. In ECCV 2018
# Experiments --- Reconstruction on Real Images

<table>
<thead>
<tr>
<th>Input image</th>
<th>ShapeHD</th>
<th>Proposed</th>
<th>Input image</th>
<th>ShapeHD</th>
<th>Proposed</th>
<th>Ground-truth</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Pascal 3D+ dataset" /></td>
<td><img src="image2" alt="ShapeHD" /></td>
<td><img src="image3" alt="Proposed" /></td>
<td><img src="image4" alt="Input image" /></td>
<td><img src="image5" alt="ShapeHD" /></td>
<td><img src="image6" alt="Proposed" /></td>
<td><img src="image7" alt="Ground-truth" /></td>
</tr>
</tbody>
</table>

(PshapeHD) Wu et al. Learning shape priors for single-view 3D completion and reconstruction. In ECCV 2018
Experiments --- Quantitative Evaluation

Chamfer Distance (CD) on PASCAL 3D+ database

<table>
<thead>
<tr>
<th>Category</th>
<th>3D-R2N2</th>
<th>DRC</th>
<th>ShapeHD</th>
<th>DAREC</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>plane</td>
<td>0.305</td>
<td>0.112</td>
<td>0.094</td>
<td>0.108</td>
<td>0.102</td>
</tr>
<tr>
<td>car</td>
<td>0.305</td>
<td>0.099</td>
<td>0.129</td>
<td>0.101</td>
<td>0.113</td>
</tr>
<tr>
<td>chair</td>
<td>0.238</td>
<td>0.158</td>
<td>0.137</td>
<td>0.135</td>
<td>0.119</td>
</tr>
<tr>
<td>couch</td>
<td>0.347</td>
<td>0.169</td>
<td>0.176</td>
<td>-</td>
<td>0.148</td>
</tr>
<tr>
<td>table</td>
<td>0.321</td>
<td>0.162</td>
<td>0.153</td>
<td>-</td>
<td>0.127</td>
</tr>
</tbody>
</table>

Mean     | 0.303   | 0.140 | 0.138   | -     | 0.122    |

CD on Pix3D database

<table>
<thead>
<tr>
<th>Category</th>
<th>3D-R2N2</th>
<th>DRC</th>
<th>ShapeHD</th>
<th>DAREC</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>chair</td>
<td>0.239</td>
<td>0.160</td>
<td>0.123</td>
<td>0.112</td>
<td>0.091</td>
</tr>
<tr>
<td>couch</td>
<td>0.307</td>
<td>0.178</td>
<td>0.137</td>
<td>-</td>
<td>0.114</td>
</tr>
<tr>
<td>table</td>
<td>0.289</td>
<td>0.163</td>
<td>0.133</td>
<td>-</td>
<td>0.127</td>
</tr>
</tbody>
</table>

Mean     | 0.278   | 0.167 | 0.131   | -     | 0.110    |

(3D-R2N2) Choy et al. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In ECCV 2016
(DRC) Tulsiani et al. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In CVPR 2017
(ShapeHD) Wu et al. Learning shape priors for single-view 3D completion and reconstruction. In ECCV 2018
3DMM for ImageNet?
Applications

https://www.youtube.com/watch?v=M1iu viJN8

https://www.youtube.com/watch?v=rOAcXbLEypU

Hand tracking (MATLAB example)