3D Alignment of Face in a Single Image

Leon Gu
Carnegie Mellon University

Presented at CVPR 2006, NYC
3D Face Alignment

Locate detailed facial feature points: 3D Geometry + 3D Pose
Face Representation

3D Geometry

View-based Patches
Feature Detector

- **Descriptor**
  - Compute image gradient
  - Normalize the $L_1$ norm of the gradient vector to one.
  - Fit a Gaussian

- **Detector**
  - Initialized by face detector
  - Search in a small local region
Individual Feature Detection

Initialization

Detection Result
Individual Feature Detection

Initialization

Detection Result
3D Alignment - Formulation

- Deform and transform a 3D parametric shape model to fit a set of noisy 2D feature points.
3D Alignment - Problems

- “Fill-in” the missing information
  - Self-occluded parts
  - Depth

- De-noising

- Handle outliers
Training Data?

- 3D Faces (geometry, texture)
- Correspondence
**Generate Faces for Training**

**Database (A)**
- 720 images (83 Points)

**Database (B)**
- 100 laser-scans; Semi-automatic registration. (8895 vertices)
One Example

Image-plane Coordinates;
Texture

Depth
Multiple Virtual Faces

one 2D image + different laser-scans.
Database of Virtual Faces
3D Shape Prior
View-based Feature Detectors

(a) left eye corner  
(b) left eyebrow  
(c) right eyeball center  
(d) the lowest point on mouth contour  
(e) the center of left lower cheek  
(f) the point on the contour of right cheek
Model Matching

2D Observation

3D Shape Matching
Shape Augmentation

2D Observation

3D "Observation"

Current 3D Shp Estimate

Another View of 3D Observation
De-Noising

- Projection and Shrinkage

- Decompose 3D observation into PCA space
- Shrink deformation coefficients towards mean shape.
- Reconstruct the 3D shape
Projection-Shrinkage

Observation
\[ X \]

Projection
\[ b = \Phi_r^T X \]

Shrinkage
\[ \hat{b}_i = \alpha_i b_i \]
Shrinking Factor

- Depends on model variance and noise variance.
- Preserve major deformations
- Penalize minor deformations

\[ \alpha_i = \frac{\text{eigval}}{\text{eigval} + \text{noise_var}} = \frac{\lambda_i}{\lambda_i + \sigma^2}, \]

\[ \sigma^2 = \frac{1}{n} \sum_{j=r+1}^{n} \lambda_j \]
Shrinkage

3D Observation

Shrunk Shape
Hand

Observations

Shrunken Shapes
Human Body

Observations

Shrunken Shapes
Outliers?

Observations

Shrunken Shapes
Iterative Re-weighting

Identify outliers

Penalize outliers

\[ X = (1 - p) \times \begin{pmatrix} \text{3D Obs} \end{pmatrix} + p \times \begin{pmatrix} \text{3D Shp Estimate} \end{pmatrix} \]
Comparison

Shrinkage

Re-weighting + Shrinkage
Pose Estimation

\[ \theta = \arg \min \sum_i w_i \left( T_{\theta}\left( \right) - \right)^2 \]

- Weak perspective projection \( \theta = \{s, R, t\} \).
- Only visible points are involved.
The Algorithm

1. Face detection; view based feature point detection
2. Fitting 3D model to 2D points (EM iteration)
   1. Shape augmentation: fill in missing information; penalize outliers by weighted average.

\[ X = (1-p) \times \left( \begin{array}{c}
\end{array} \right) + p \times \left( \begin{array}{c}
\end{array} \right) \]

2. Shape estimation: projection and shrinkage.


\[ \theta = \arg \min \sum_i w_i (T_\theta(\ldots) - \ldots)^2 \]

3. Re-weighting; update visibility mask.
4. Repeat until convergence
Summary

• Trained from synthetic 3D faces
  • Correspondence established automatically

• 3D face alignment
  • Fully automatic (initialized by multi-view face detector)
  • Handle large range of view changes
  • Well generalized to new, unseen faces

• 3D shape/pose estimation from a single 2D observation
  • Self-occlusion, depth (shape augmentation)
  • Observation noise (shrinkage)
  • Outliers (iterative re-weighting)