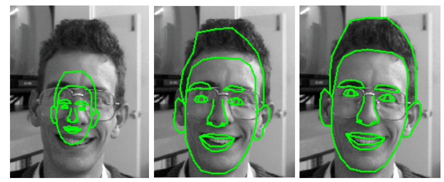
CSE 586, Spring 2010
Advanced Computer Vision

Procrustes Shape Analysis

Credits

lots of slides are due to



Lecture material from Tim Cootes University of Manchester.
For more info, see http://www.isbe.man.ac.uk/~bim/
(includes code for exploring active shape/appearance models).

Overview

Statistical Shape Models: a method of modelling shape and shape variation

Active Shape Models: an active contour method with constraints on shape

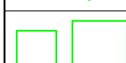
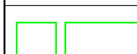
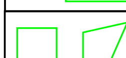
Shape

- Need to model the variability in shape
- What is shape?
  - *Geometric information that remains when location, scale and rotational effects removed* (Kendall)

Shape

- More generally
  - *Shape is the geometric information invariant to a particular class of transformations*
- Transformations:
  - Euclidean (translation + rotation)
  - Similarity (translation+rotation+scaling)
  - Affine
  - Projective

Shape

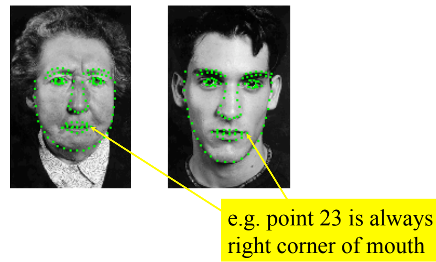
Shapes	Euclidean	Similarity	Affine	Projective
	✓			
		✓		
			✓	
				✓

Shape Models

- We will represent the shape using a set of points
- We will model the variation by computing the PDF of the distribution of shapes in a training set (Gaussian)
- This allows us to generate new shapes similar to the training set

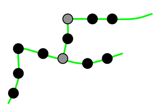
Point-based Shape Models

- Require labeled training images
  - point "landmarks" represent correspondences



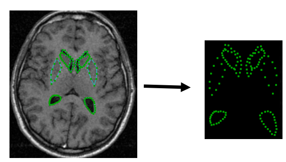
Suitable Landmarks

- Define correspondences
  - Well defined corners
  - `T' junctions
  - Easily located biological landmarks
  - Use additional points along boundaries to define shape more accurately



Building Shape Models

- For each example



$\mathbf{x} = (x_1, y_1, \ldots, x_n, y_n)^T$

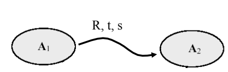
note, we are considering only 2D points here, but the approach generalizes to nD.

Statistical Shape Models

- Given a set of shapes:
- Align shapes into common frame
  - Procrustes analysis
- Estimate shape distribution p(x)
  - Single gaussian often sufficient
  - Mixture models sometimes necessary

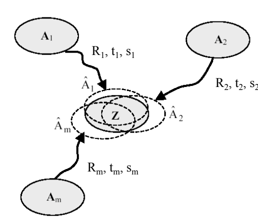
Procrustes Analysis

Procrustes Analysis



Align one shape with another
(not symmetric)

General Procrustes Analysis



Align a set of shapes with respect to some unknown "mean" shape
(independent of ordering of shapes)

Why "Procrustes"?

http://www.procrustes.nl/gif/illustr.gif

Aligning Two Shapes

- Procrustes analysis:
  - Find transformation which minimises

$$|\mathbf{x}_1 - T(\mathbf{x}_2)|^2$$

  - T is a particular type of transformation that you want "shape" to be invariant to

Go to board. Insert my handwritten notes here.

Aligning Two Shapes

- Procrustes analysis:
  - Find transformation which minimises

$$|\mathbf{x}_1 - T(\mathbf{x}_2)|^2$$

  - If T is a similarity transformation, the resulting shapes have
    - Identical center of mass
    - approximately the same scale and orientation

Steps in Similarity Alignment

Given a set of K points: Configuration

Translation normalization: Centered Configuration
(center of mass at origin)

Scale normalization: Pre-shape
(divide by Sqrt of SSQ centered coordinates)

Rotation normalization: Shape
(rotate to alignment with ref shape)

Aligning Pre-Shapes by Rotation

Sketch:

A, B are two Kx2 preshapes, R is unknown rotation

Want to minimize ||(AR - B)^2|| subject to $R^TR=I$

After some manipulation, we get
$(A^TB)(A^TB)^T = R\,(A^TB)^T(A^TB)R^T$

Note, both sides are symmetric and have same eigenvalues. This is a job for SVD!

Aligning Pre-Shapes by Rotation

Sketch continued:

$(A^TB)(A^TB)^T = R\,(A^TB)^T(A^TB)R^T$

SVD ↓ SVD ↓

$V\,D\,V^T = R\,W\,D\,W^T\,R^T$

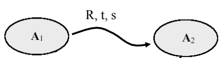
So... $V = R\,W$

and therefore $R = V\,W^T$

Robert Collins
CSE586

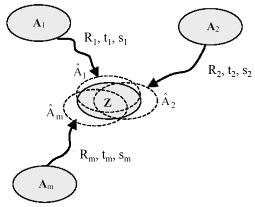
Recall:

Procrustes Analysis



Align one shape
with another
(not symmetric)

General Procrustes Analysis



Align a set of shapes with respect
to some unknown "mean" shape
(independent of ordering of shapes)

Robert Collins
CSE586

Aligning a Set of Shapes

- Generalised Procrustes Analysis
  - Find the transformations T_i which minimise

$$\sum |\mathbf{m} - T_i(\mathbf{x}_i)|^2$$

  - Where $\mathbf{m} = \frac{1}{n}\sum T_i(\mathbf{x}_i)$
  - Under the constraint that $|\mathbf{m}| = 1$

Robert Collins
CSE586

Aligning Shapes : Algorithm

- Normalise all so center of mass is at origin, and size=1
- Let $\mathbf{m} = \mathbf{x}_1$
- Align each shape with **m** (via a rotation)
- Re-calculate $\mathbf{m} = \frac{1}{n}\sum T_i(\mathbf{x}_i)$
- Normalise **m** to default size, orientation
- Repeat until convergence