

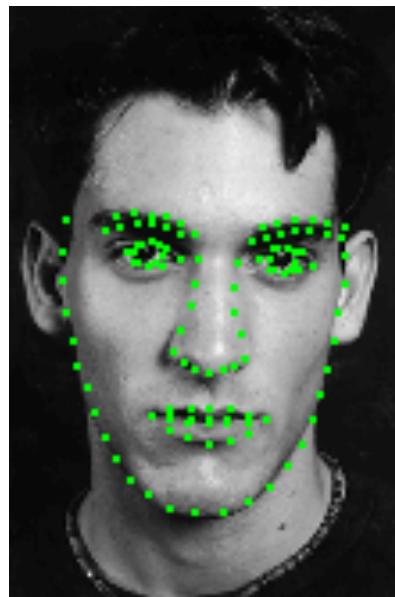
Robert Collins
CSE586

CSE 586, Spring 2015

Shape Analysis
Statistical Shape Models

Recall : Building Shape Models

- Require labeled training images
 - landmarks represent correspondences



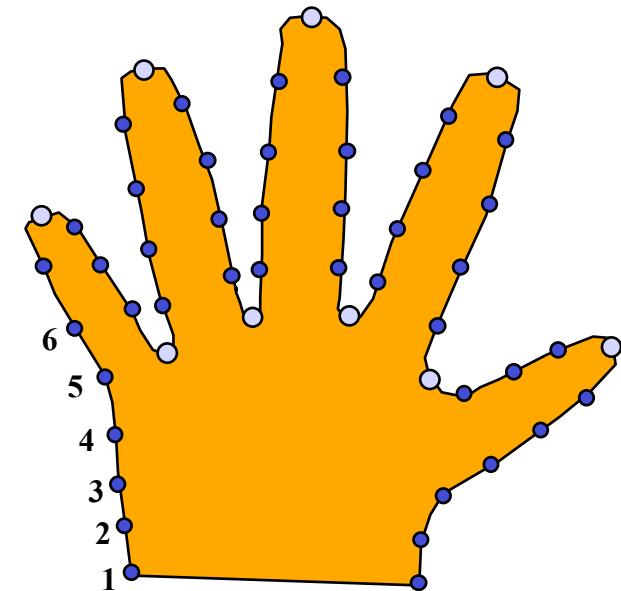
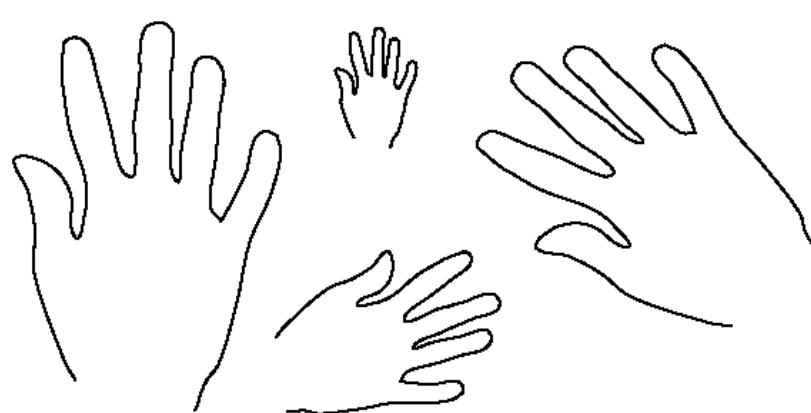
Recall: Generalized Procrustes Analysis

Algorithm for generalized orthogonal Procrustes analysis:

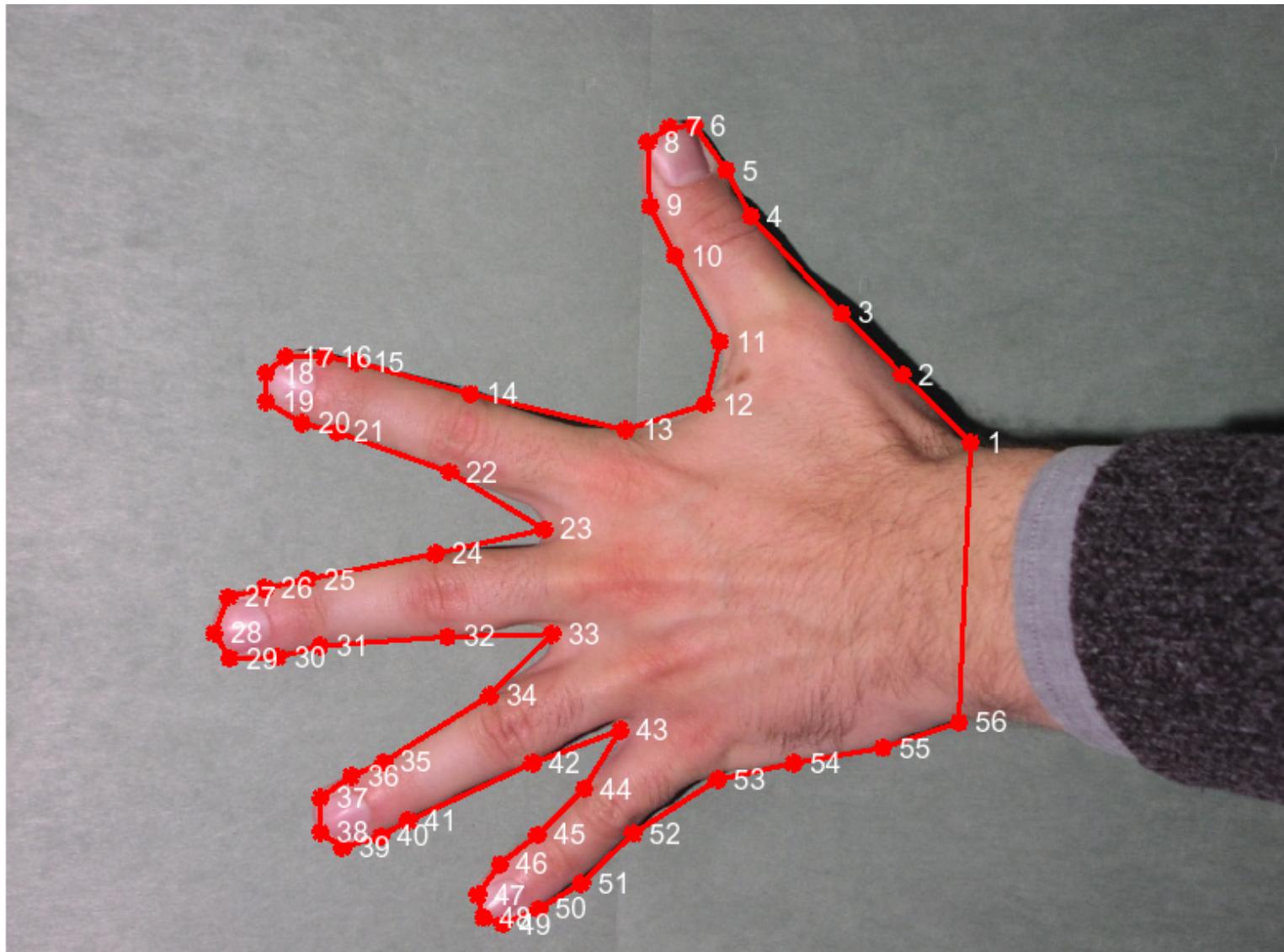
1. *Select one shape to be the approximate mean shape (i.e. the first shape in the set).*
2. *Align the shapes to the approximate mean shape.*
 - a. *Calculate the centroid of each shape (or set of landmarks).*
 - b. *Align all shapes centroid to the origin.*
 - c. *Normalize each shapes **centroid size**.*
 - d. *Rotate each shape to align with the newest approximate mean.*
3. *Calculate the new approximate mean from the aligned shapes.*
4. *If the approximate mean from steps 2 and 3 are different the return to step 2, otherwise you have found the true mean shape of the set.*

Examlpe: Hand shape model

- 72 points placed around boundary of hand
 - 18 hand outlines obtained by thresholding images of hand on a white background
 - Primary landmarks chosen at tips of fingers and joint between fingers
 - Other points placed equally between

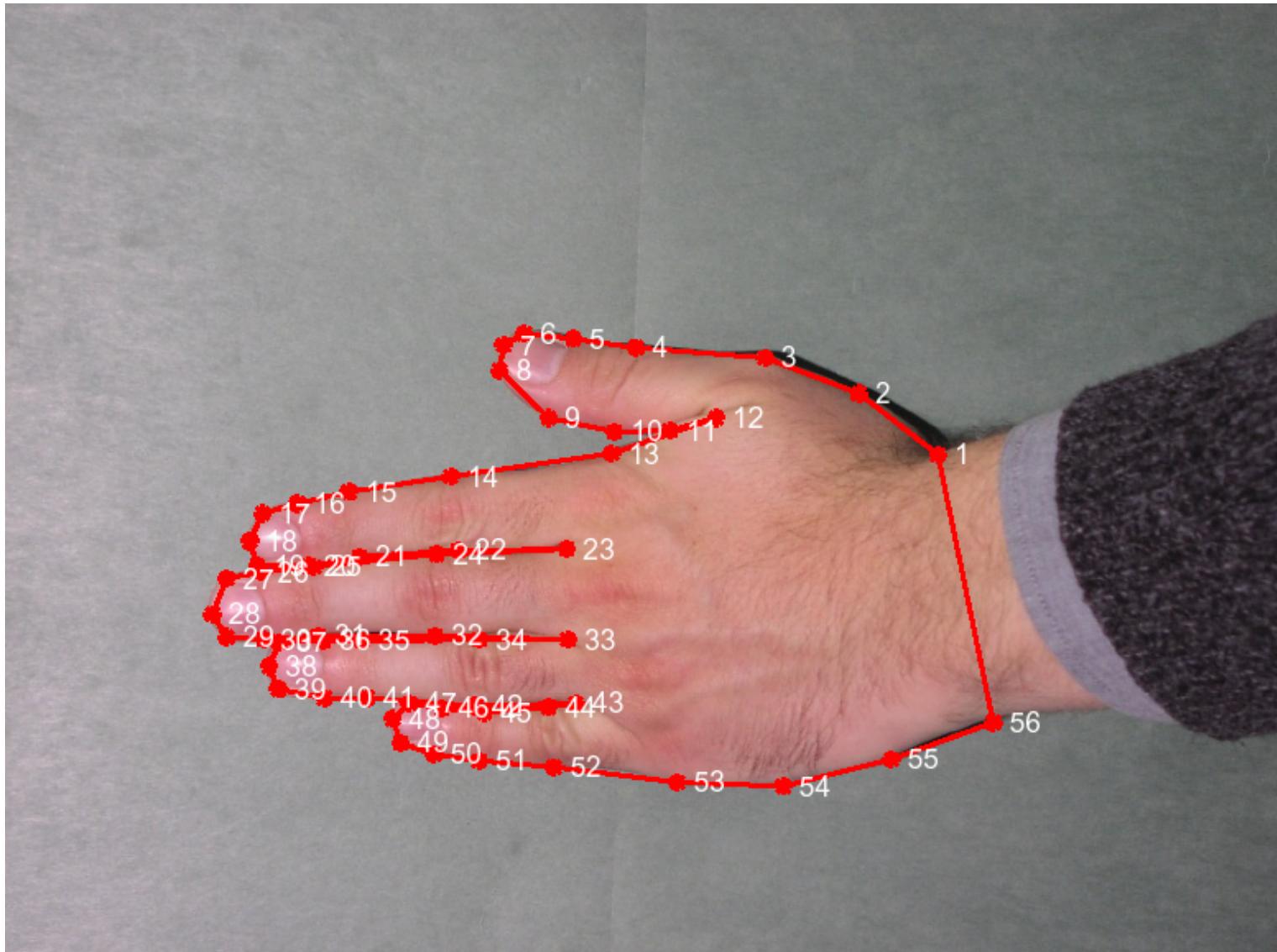


Labeling Examples



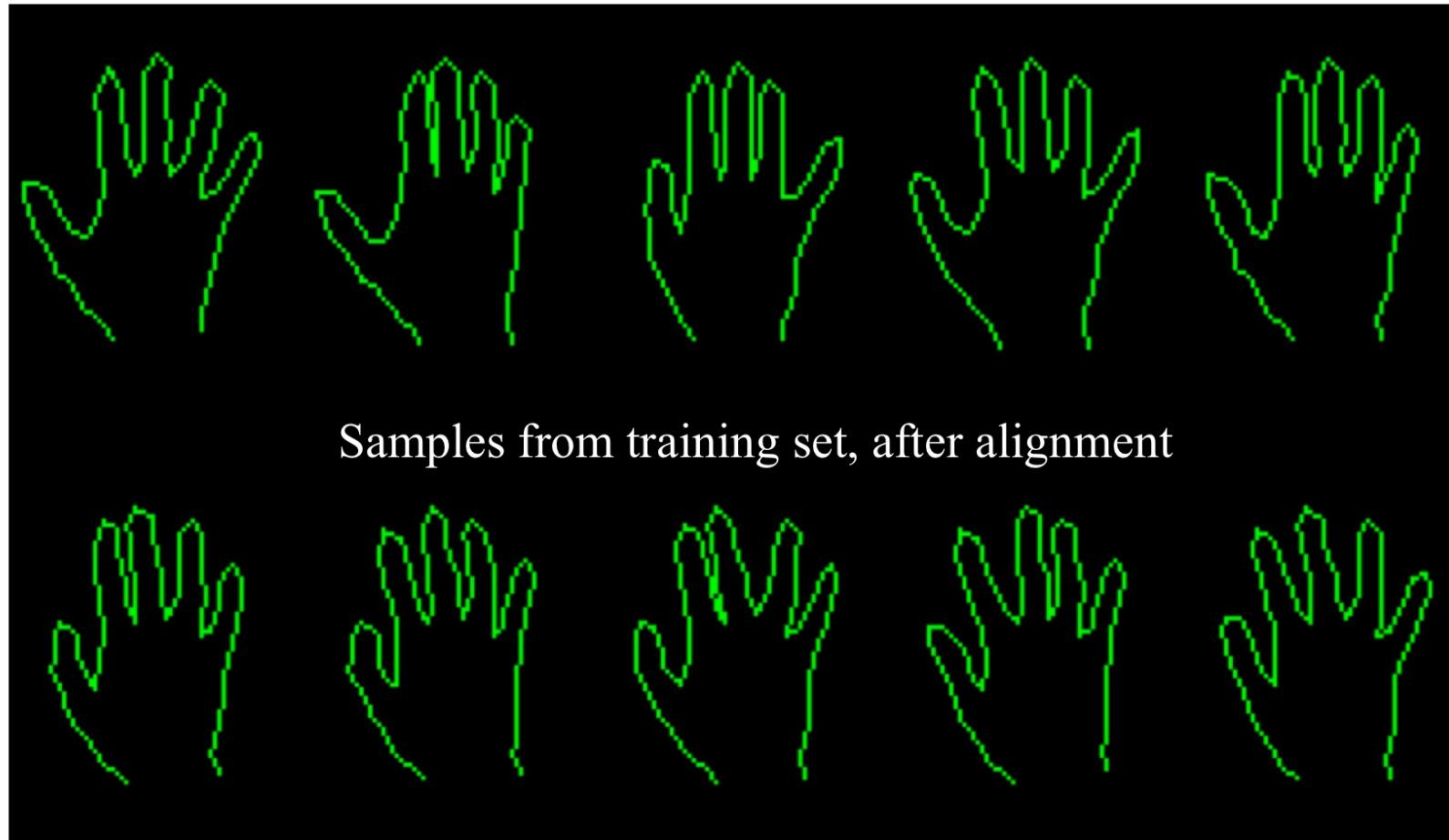
From Stegmann and Gomez

Labeling Examples



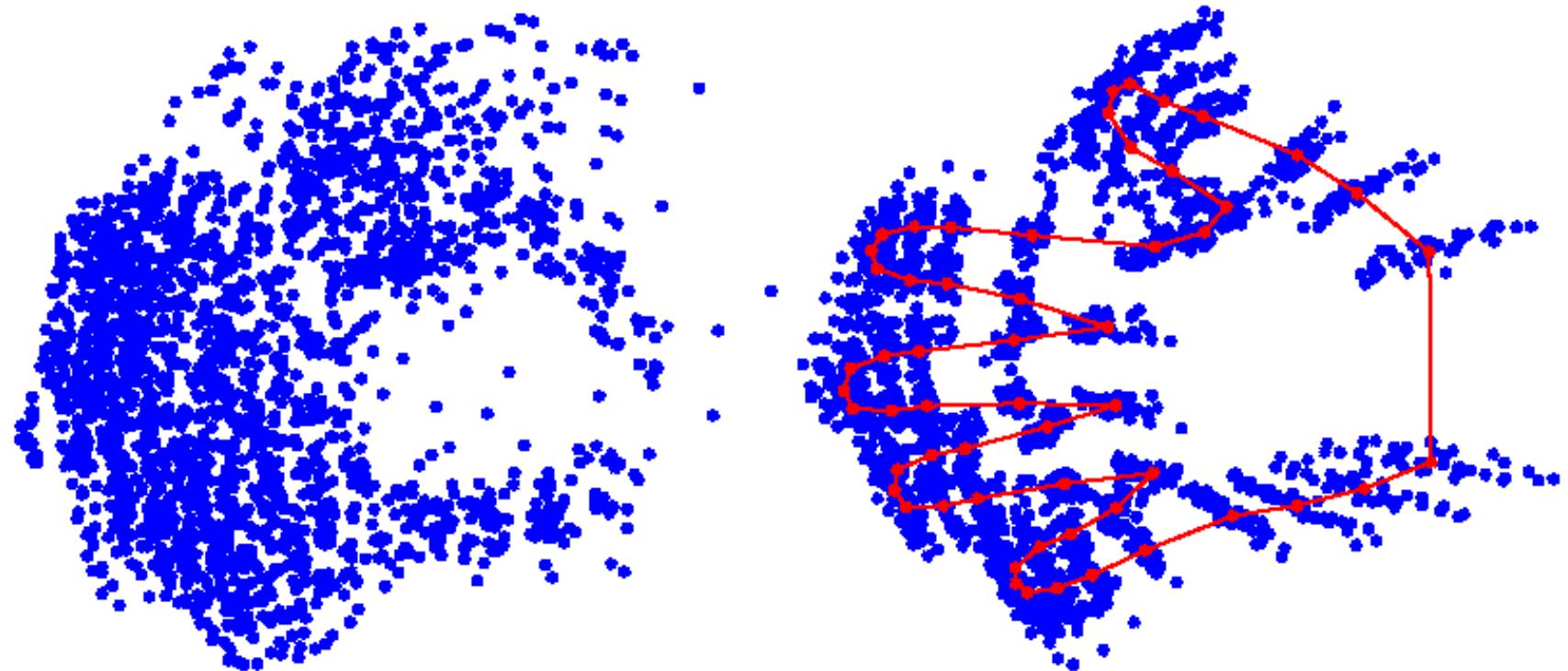
From Stegmann and Gomez

Aligned Training Examples



Point Alignment

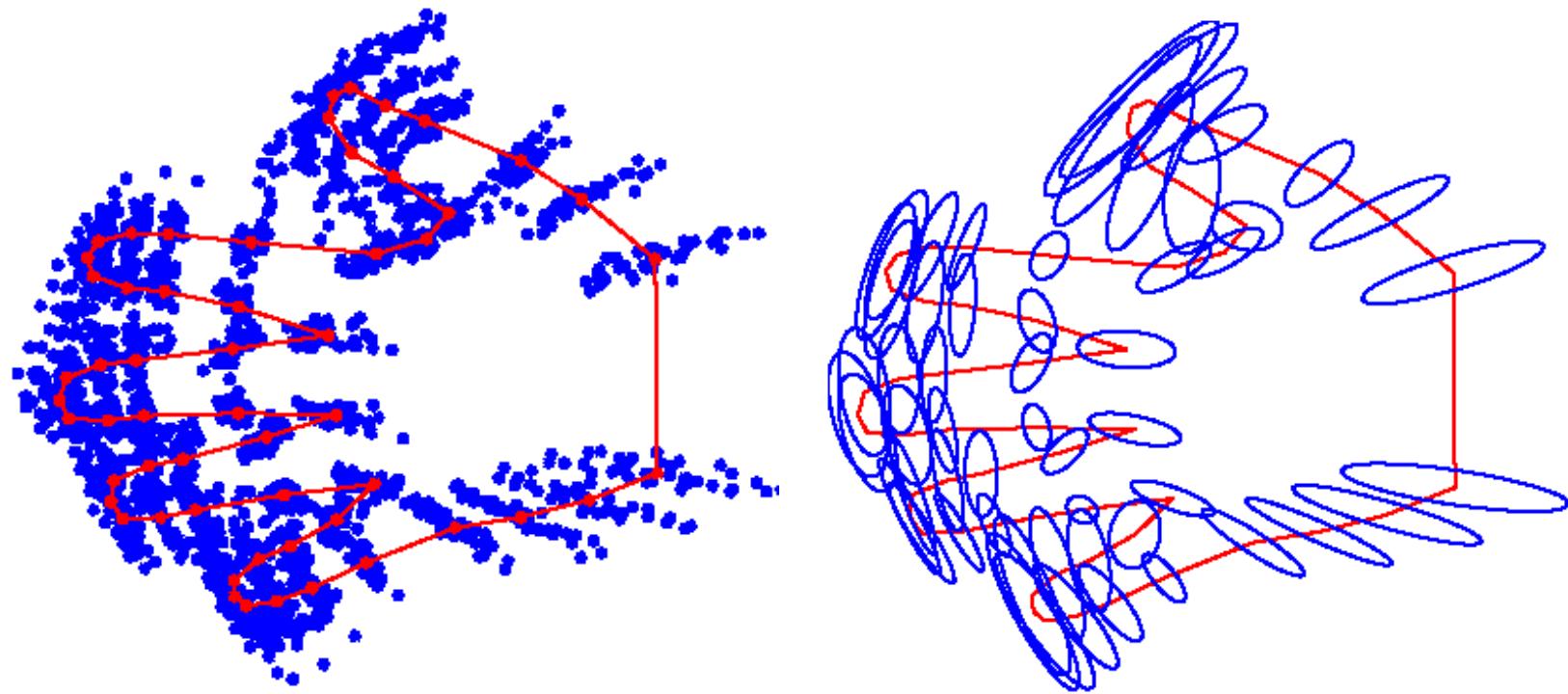
Training points before and after alignment



Note that each point looks like it varies in position by a Gaussian distribution.

Simple Shape Analysis

Fit a Gaussian (mean and covariance) to the cluster of aligned points for each landmark separately.



Problem: Each point DOES NOT move independently from all the other points. In fact, the movement of points is often highly correlated when a shape undergoes deformation.

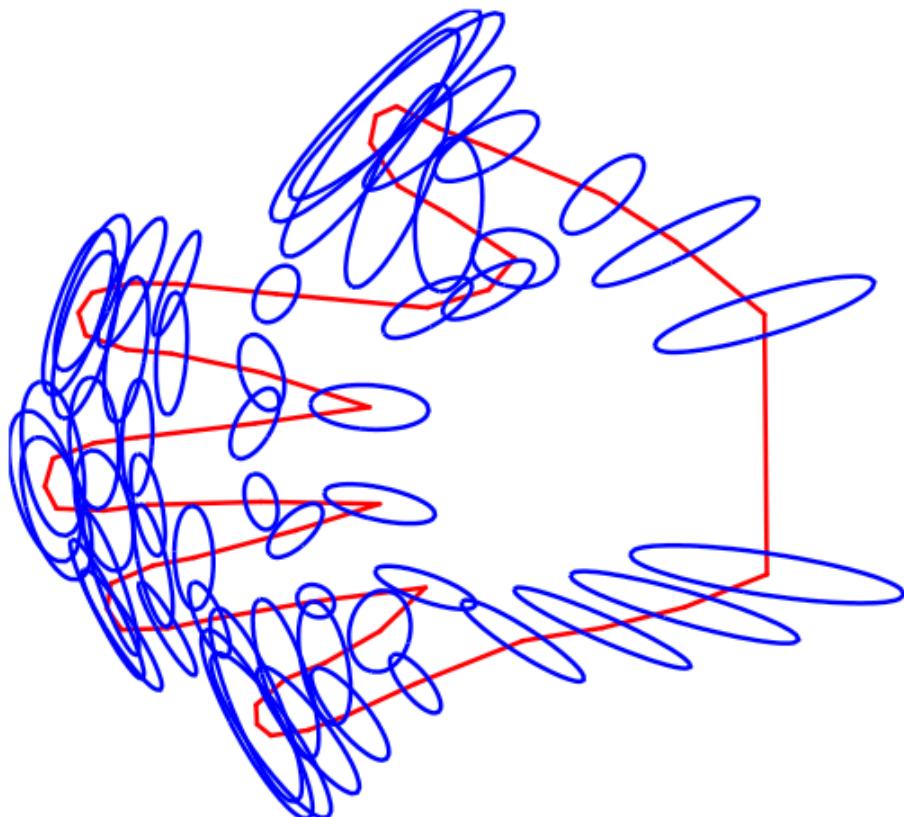
Statistical Shape Analysis

Goal: Model how the entire collection of points varies.

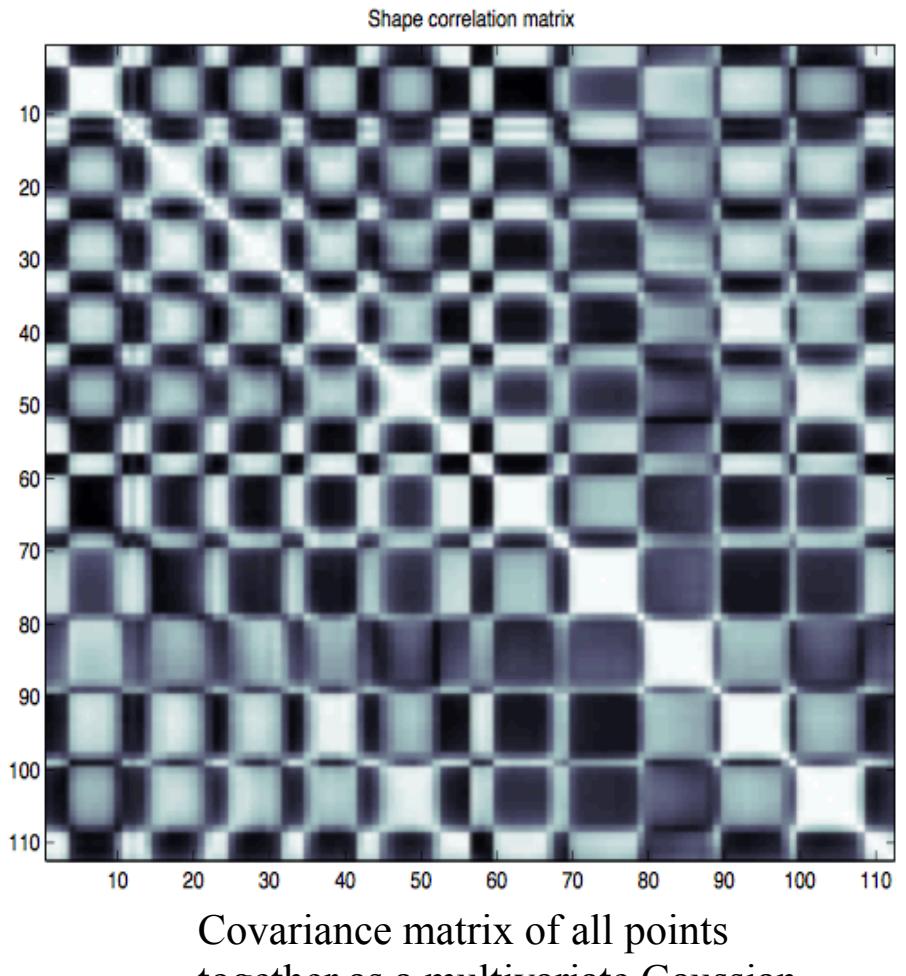
We will concatenate all points from each training example into a single vector in a high dimensional space. The full hand shape model is then built by computing PCA on all model points collectively, which captures correlations of motion between points.

Motivation

There is information stored in the correlation matrix about the joint correlation of all point movements.



Covariance ellipses of each point independently



Covariance matrix of all points together as a multivariate Gaussian

Shape Space

Consider K landmark points (x_i, y_i) in 2D

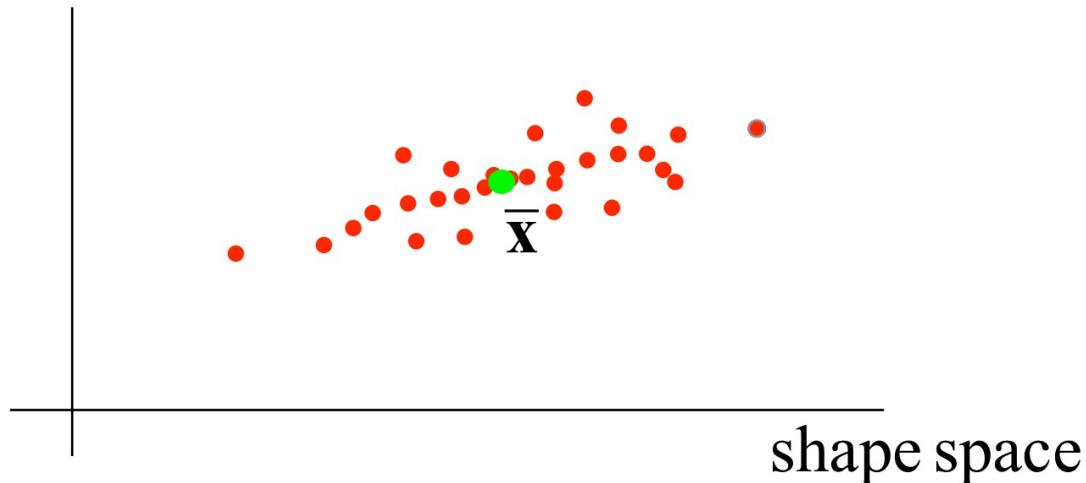
Consider each shape as a concatenated vector \mathbf{x} ,
viewed as a point in $2*K$ dimensional space.

(note: the “shape” really lives in $2K-4$ dimensional space, but that is outside the scope of this lecture)

We would like a parameterized model that describes the class of shapes we have observed.

Shape Space

- Each observed shape is now a point (vector) x in $2*K$ dimensional space.
- The “mean shape” is the center of mass of all these points



Shape Models

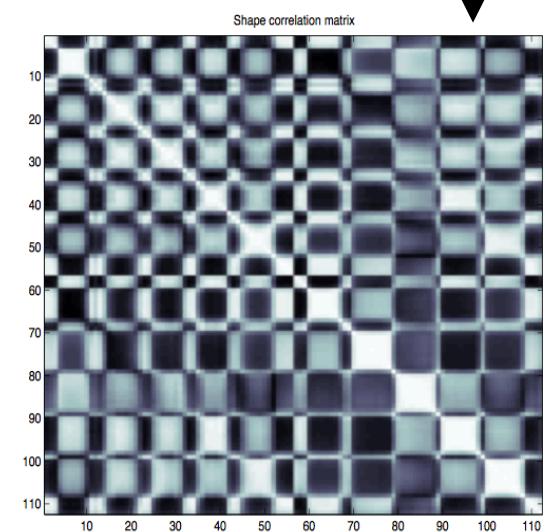
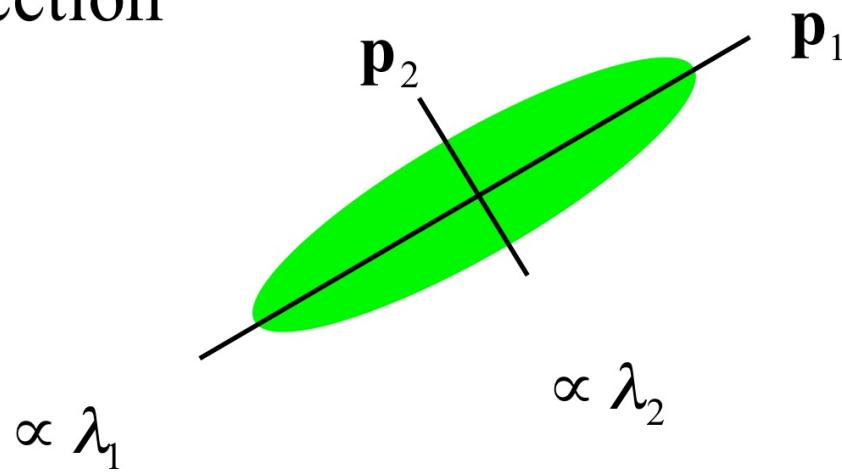
- For shape synthesis
 - Parameterised model preferable
 - We will use a linear model

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$$

- That is,
$$\mathbf{x} = \bar{\mathbf{x}} + b_1 \mathbf{p}_1 + b_2 \mathbf{p}_2 + \dots + b_m \mathbf{p}_m$$

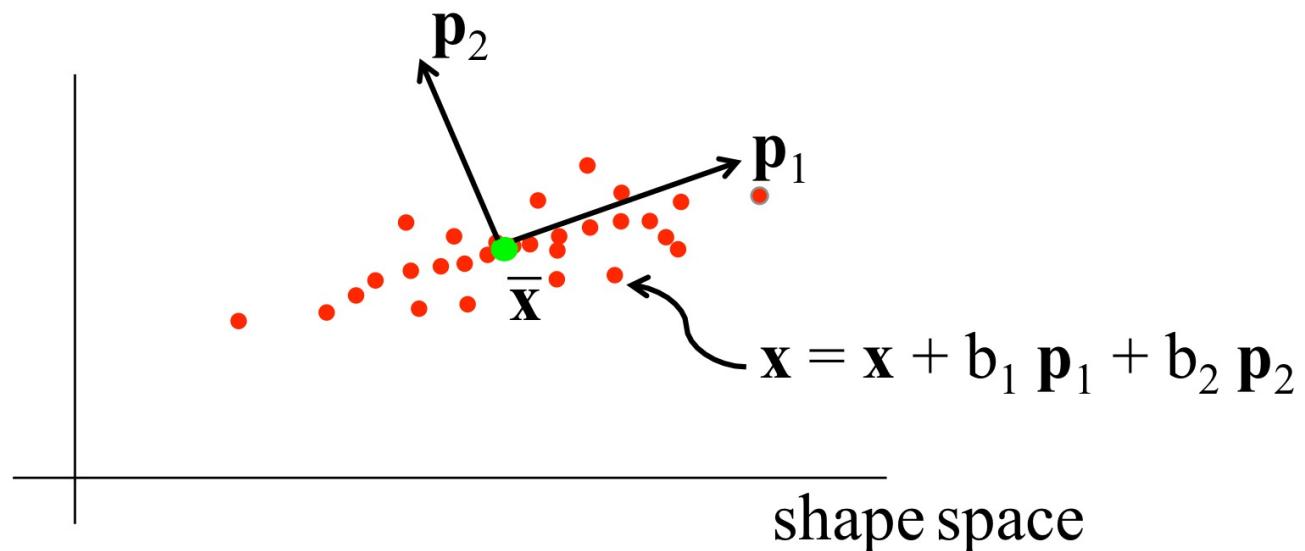
Principal Component Analysis

- Matrix X contains each shape as one column
- Compute eigenvectors of scatter matrix $X^T X$
- Eigenvectors : main directions
- Eigenvalue : spread or variation along that direction



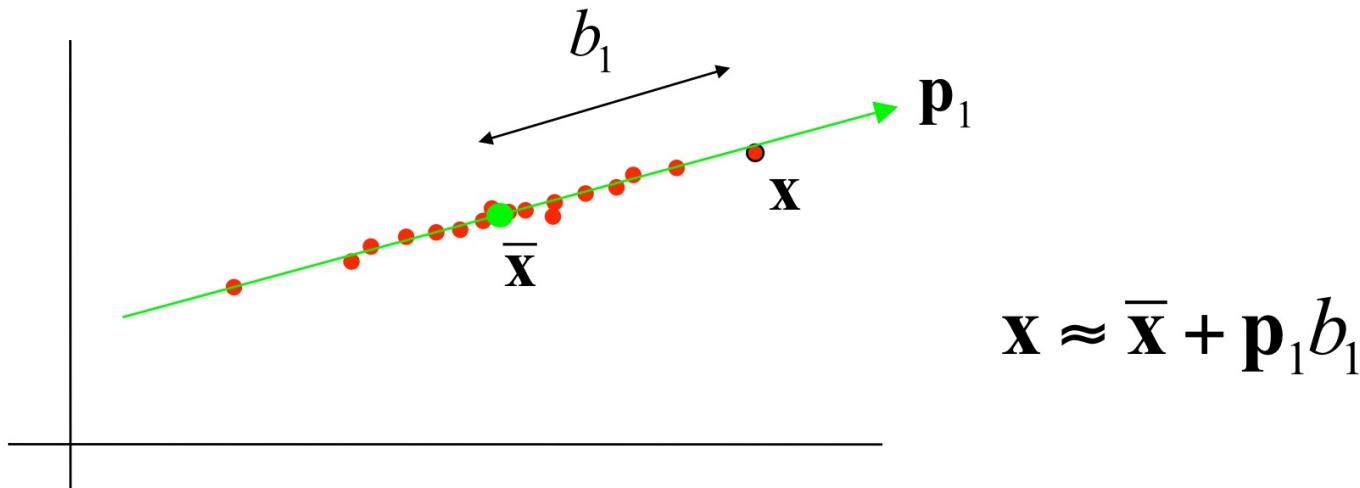
Shape Space

- The eigenvectors p_1, p_2 form a new (rotated) basis set of coordinate axes
- Each shape x can now be assigned coordinates (b_1, b_2) according to the eigenvector axes



Dimensionality Reduction

- Coordinates often highly correlated
- Some dimensions account for most of the observed spread (variation) of the data
- We don't lose much by approximating the shape using only those dimensions

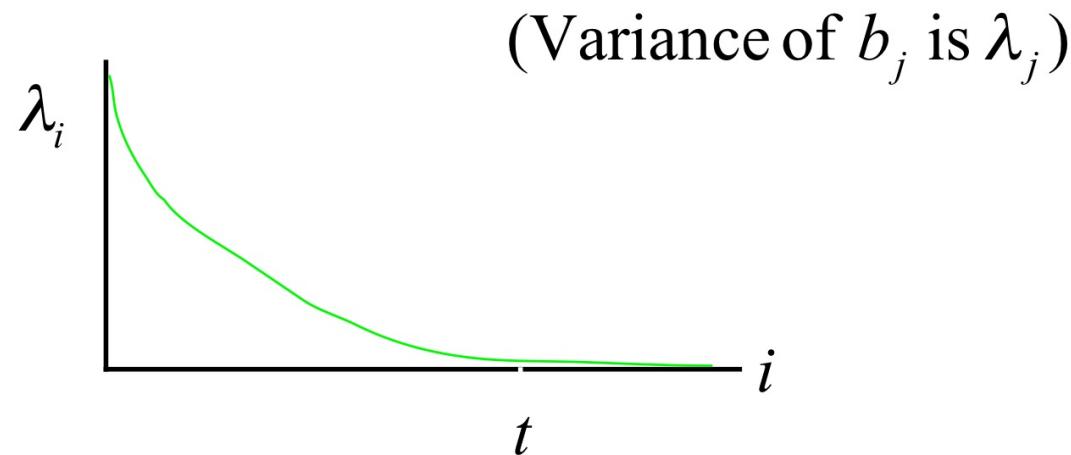


Dimensionality Reduction

- Data lies in subspace of reduced dim.

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{p}_1 b_1 + \dots + \mathbf{p}_n b_n$$

- However, for some t , $b_j \approx 0$ if $j > t$



Building Shape Models

- Given aligned shapes, $\{\mathbf{x}_i\}$
- Apply PCA

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$$

- \mathbf{P} – First t eigenvectors of covar. matrix
- \mathbf{b} – Shape model parameters = coordinates of shape along eigenvector axes

Statistical Shape Models

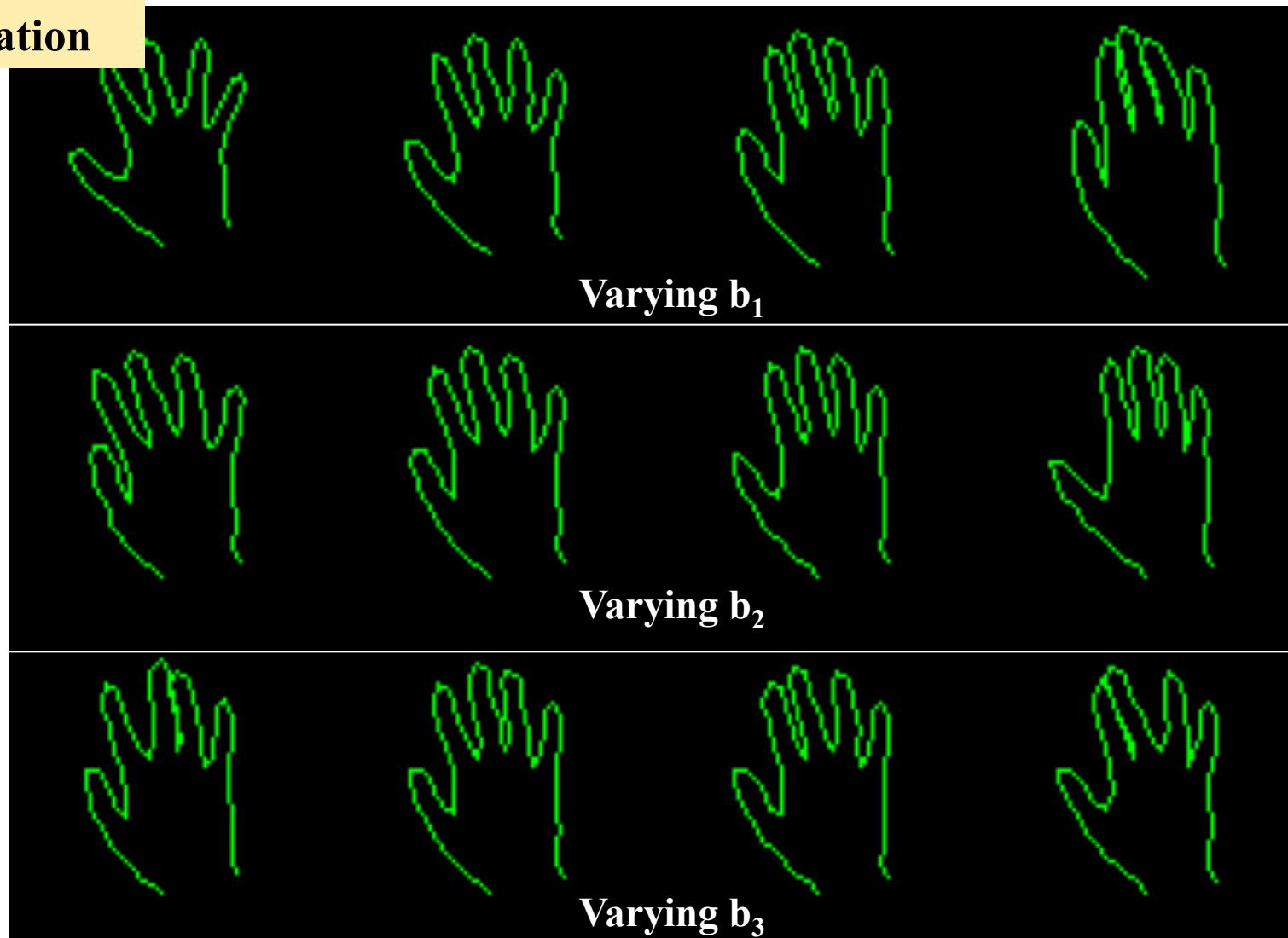
- Represent likelihood of observing a given shape with a probability distribution
- Learn $p(\mathbf{b})$ from training set
- If \mathbf{x} multivariate gaussian, then
 - \mathbf{b} gaussian with diagonal covariance

$$\mathbf{S}_b = \text{diag}(\lambda_1 \cdots \lambda_t)$$

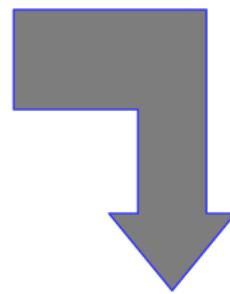
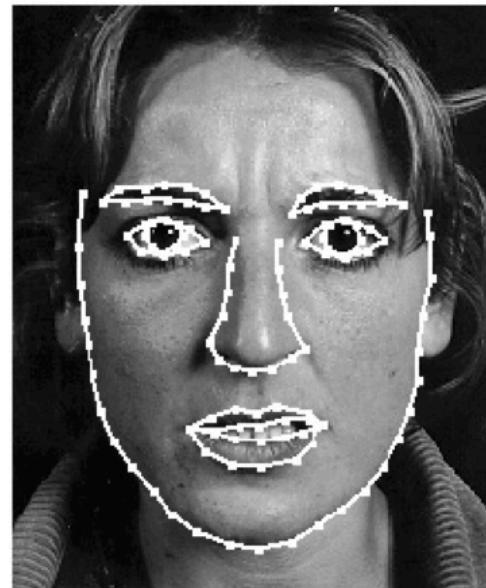
- Or, can use mixture model for $p(\mathbf{b})$

Hand Shape Model

First 3 modes
of variation



Face Shape Example



Training instances



Modes 1-3



Distribution of Parameters

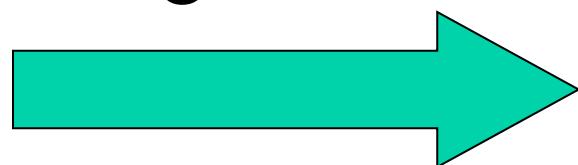
- Learn $p(\mathbf{b})$ from training set
- If \mathbf{x} multivariate gaussian, then
 - \mathbf{b} gaussian with diagonal covariance

$$\mathbf{S}_b = \text{diag}(\lambda_1 \dots \lambda_t)$$

- Can use mixture model for $p(\mathbf{b})$

Summary so far...

- We can build statistical models of shape change
- Require point correspondences across training set
- Get compact model (few parameters) by doing Principle Components Analysis (PCA)
- Next: Matching models to images

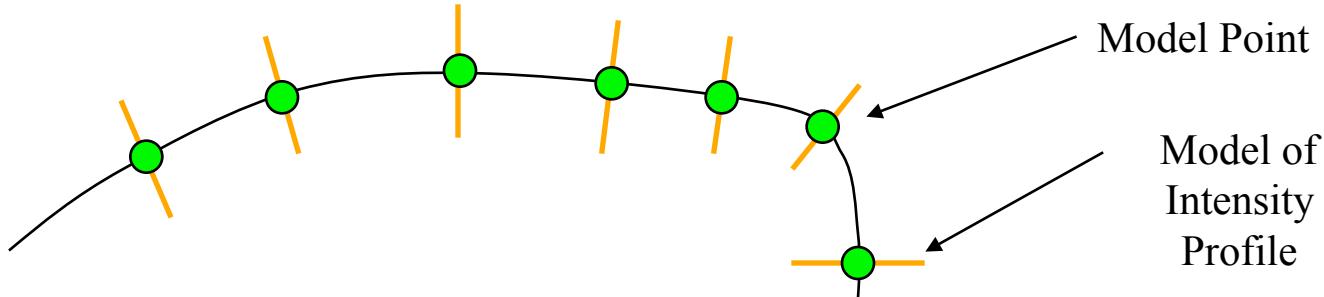


Active Shape Models

- 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 (similar to active contours)

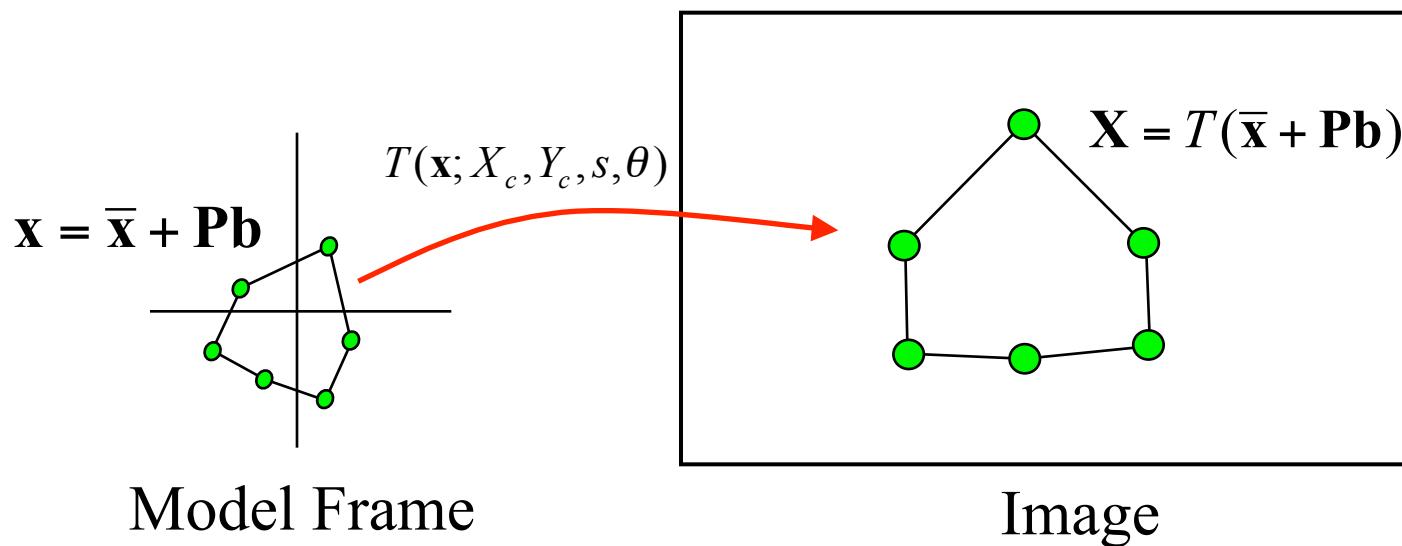
Active Shape Models

- Match shape model to new image
- Require:
 - Statistical shape model
 - Model of image structure at each point



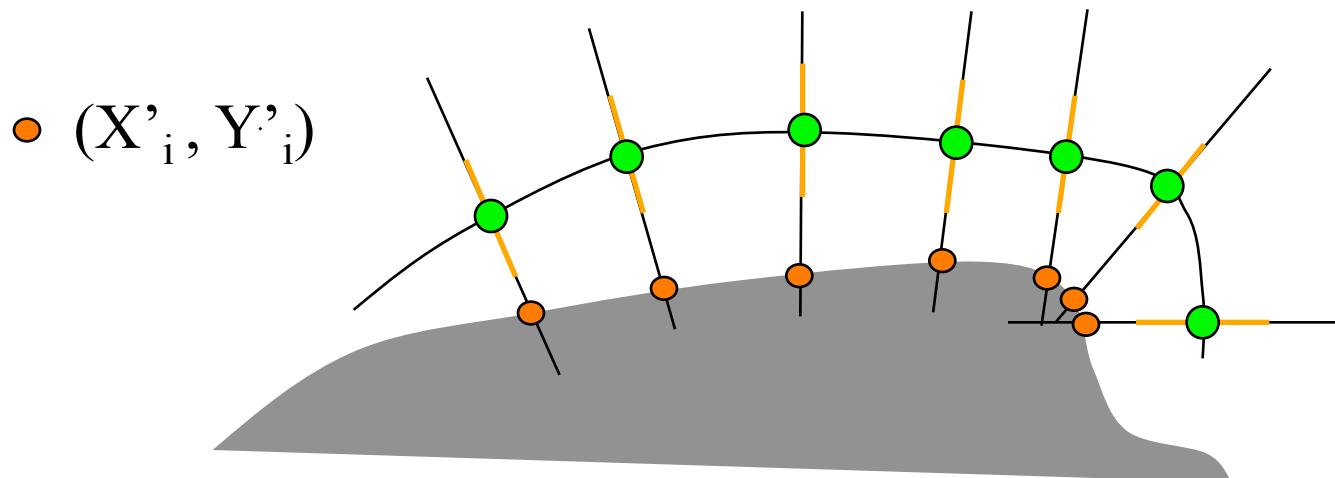
Placing model in image

- The model points are defined in a model co-ordinate frame
- Must apply global transformation, T , to place it in the image



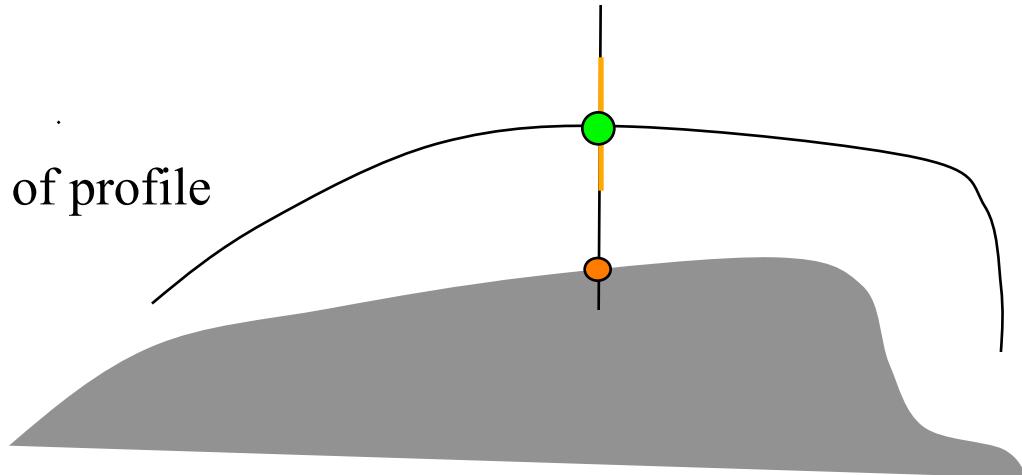
ASM Search Overview

- Local optimisation
- Initialise near target
 - Search along profiles for best match X'
 - Update parameters to move towards X' .

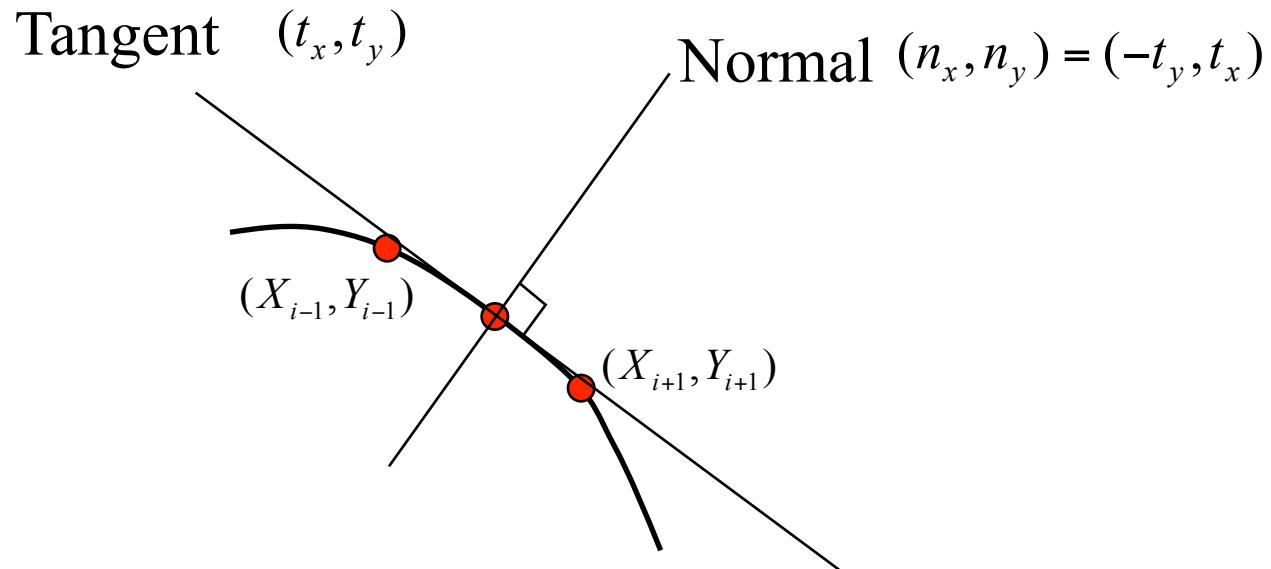


Local Structure Models

- Need to search for local match for each point
- Options
 - Strongest edge
 - Correlation
 - Statistical model of profile



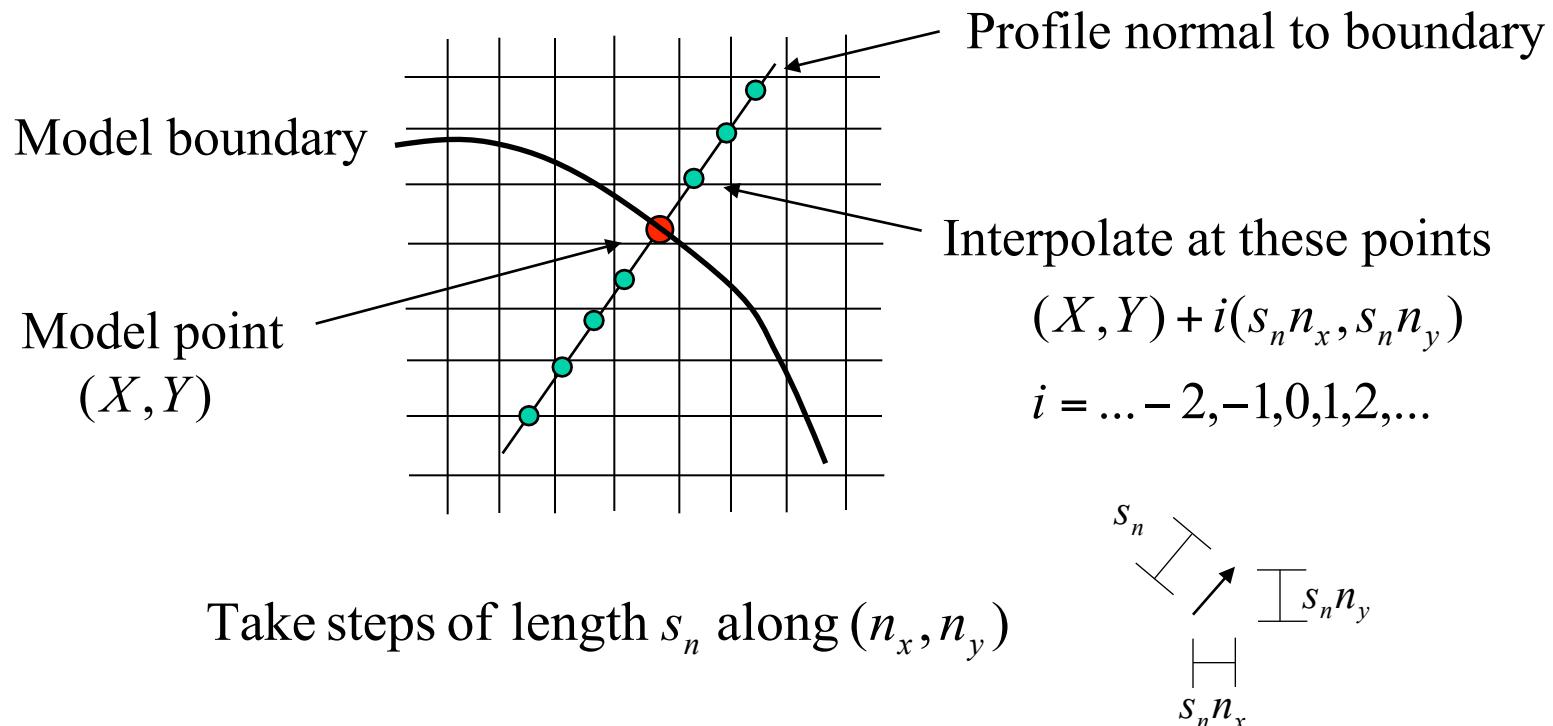
Computing Normal to Boundary



$$(t_x, t_y) \approx \frac{(d_x, d_y)}{\sqrt{d_x^2 + d_y^2}} \quad \begin{aligned} d_x &= X_{i+1} - X_{i-1} \\ d_y &= Y_{i+1} - Y_{i-1} \end{aligned}$$

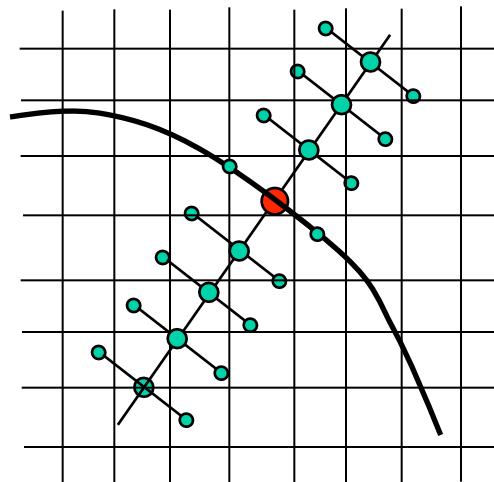
(Unit vector)

Sampling along profiles



Noise reduction

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



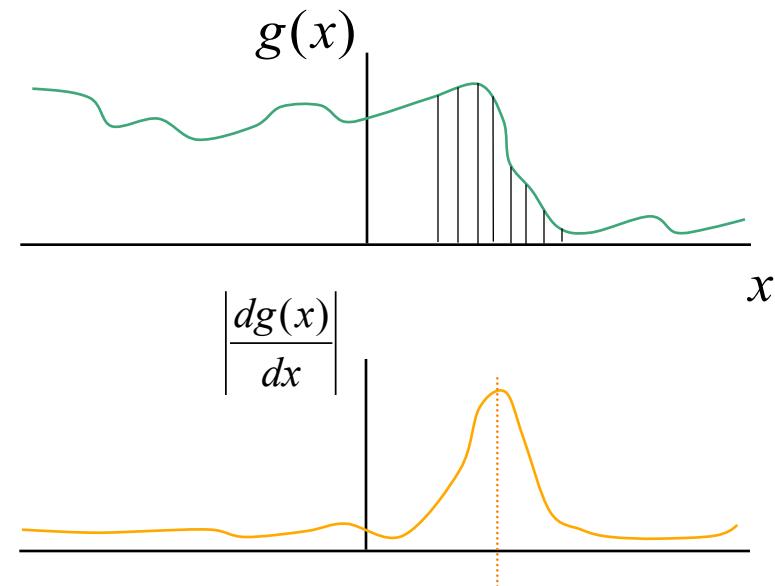
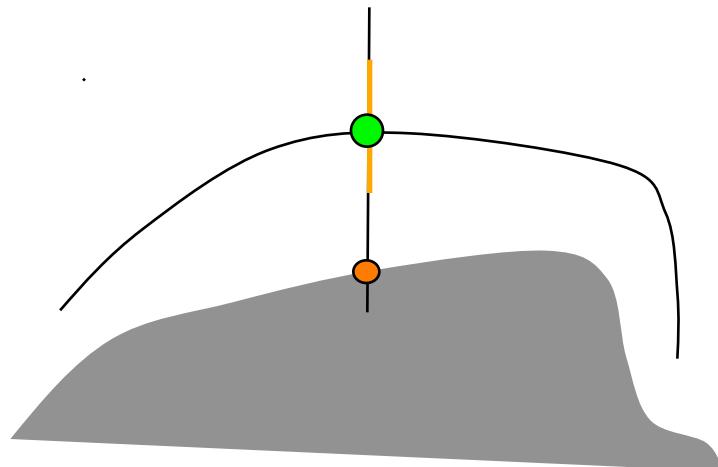
$$g_{i1} \quad g_{i2} \\ g_{i3}$$

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

Sampled profile is

$$\mathbf{g} = (\dots, g_{-2}, g_{-1}, g_0, g_1, g_2, \dots)$$

Searching for strong edges

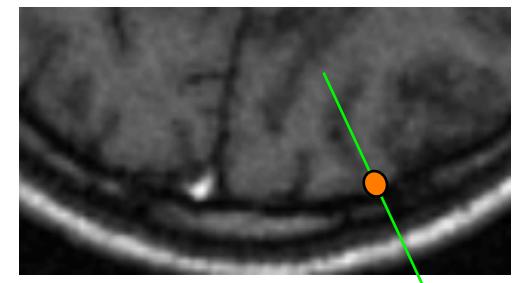
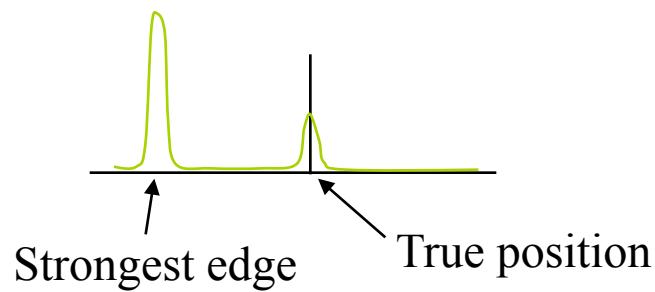
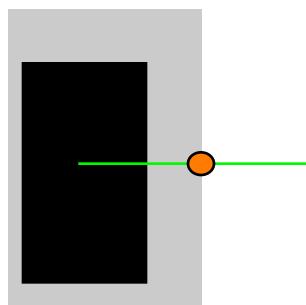


$$\frac{dg(x)}{dx} = 0.5(g(x+1) - g(x-1))$$

Select point along profile at strongest edge

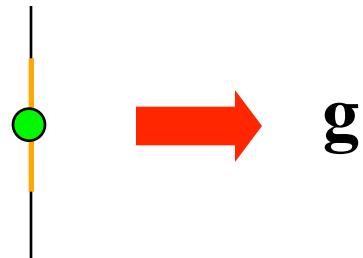
Profile Models

- Sometimes true point not on strongest edge
- Model local intensity profile structure to help locate the point



Statistical Profile Models

- Estimate p.d.f. for sample on profile



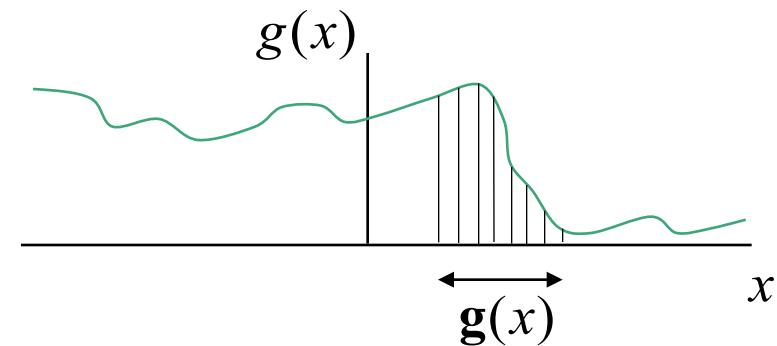
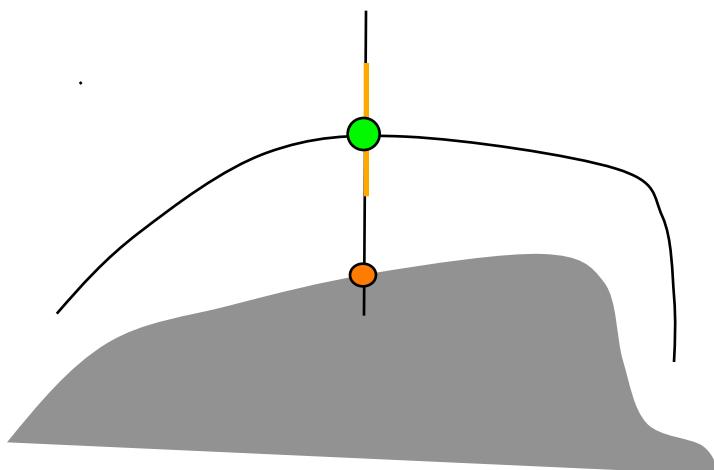
- Normalise to allow for global lighting variations
- From training set learn $p(\mathbf{g})$

Profile Models

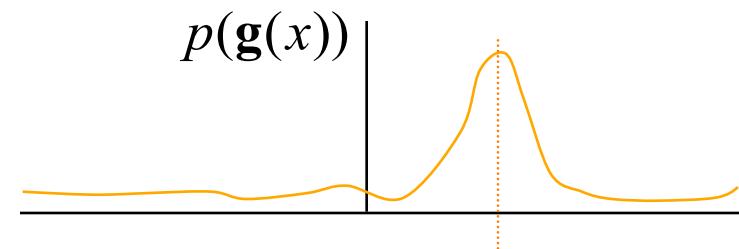
- For each point in model
 - For each training image
 - Sample values along profile
 - Normalise
 - Build statistical model
 - eg Gaussian PDF using eigen-model approach

Searching Along Profiles

- During search we look along a normal for the best match for each profile



Form vector from samples about x

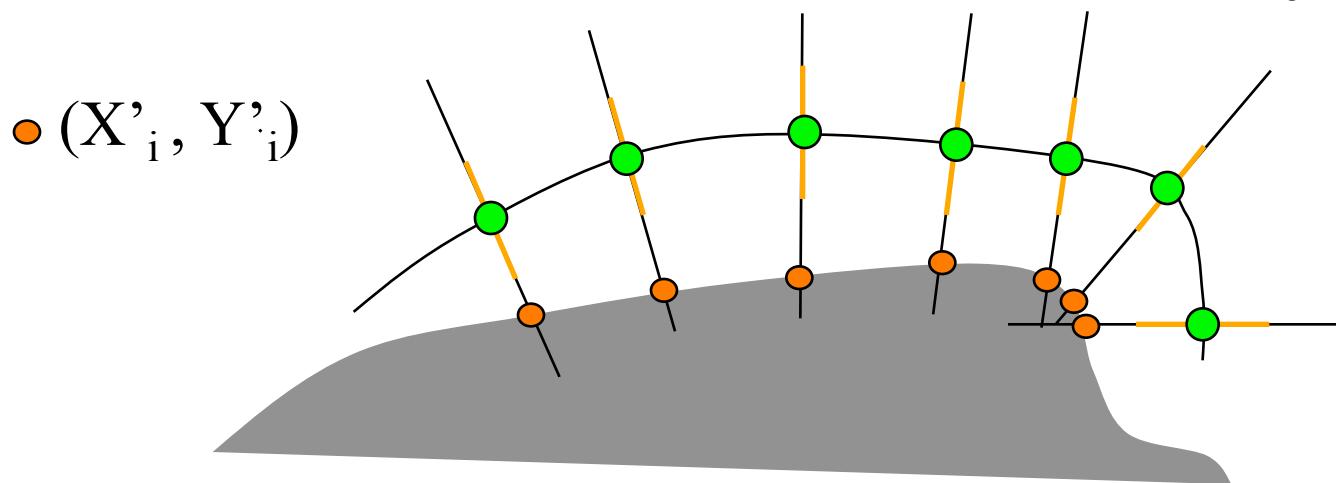


Search algorithm

- Search along profile
- Update global transformation, T , and parameters, \mathbf{b} , to minimise

$$| \mathbf{X} - T(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b}) |^2$$

Recall from last lecture that we can solve this in closed form!



Updating parameters

- Find pose and model parameters to minimise

$$f(\mathbf{b}, X_c, Y_c, s, \theta) = \| \mathbf{X} - T(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b}; X_c, Y_c, s, \theta) \|^2$$

shape scale

translation rotation

Updating Parameters

Iterative Algorithm

Repeat until convergence:

Fix \mathbf{b} and find (X_c, Y_c, s, θ) which minimise $| \mathbf{X} - T(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b}) |^2$

Analytic solution exists

Fix (X_c, Y_c, s, θ) and find \mathbf{b} which minimises $| \mathbf{X} - T(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b}) |^2$

Problem: solving for \mathbf{b}
directly by least squares
does not necessarily lead
to an “acceptable” shape.

Update step for \mathbf{b}

- Hard constraints

Minimise $| \mathbf{X} - T(\mathbf{x} + \mathbf{P}\mathbf{b}) |^2$ subject to $p(\mathbf{b}) < p_t$

$$\text{e.g. } |b_i| \leq 3\sqrt{\ddot{e}_i}$$

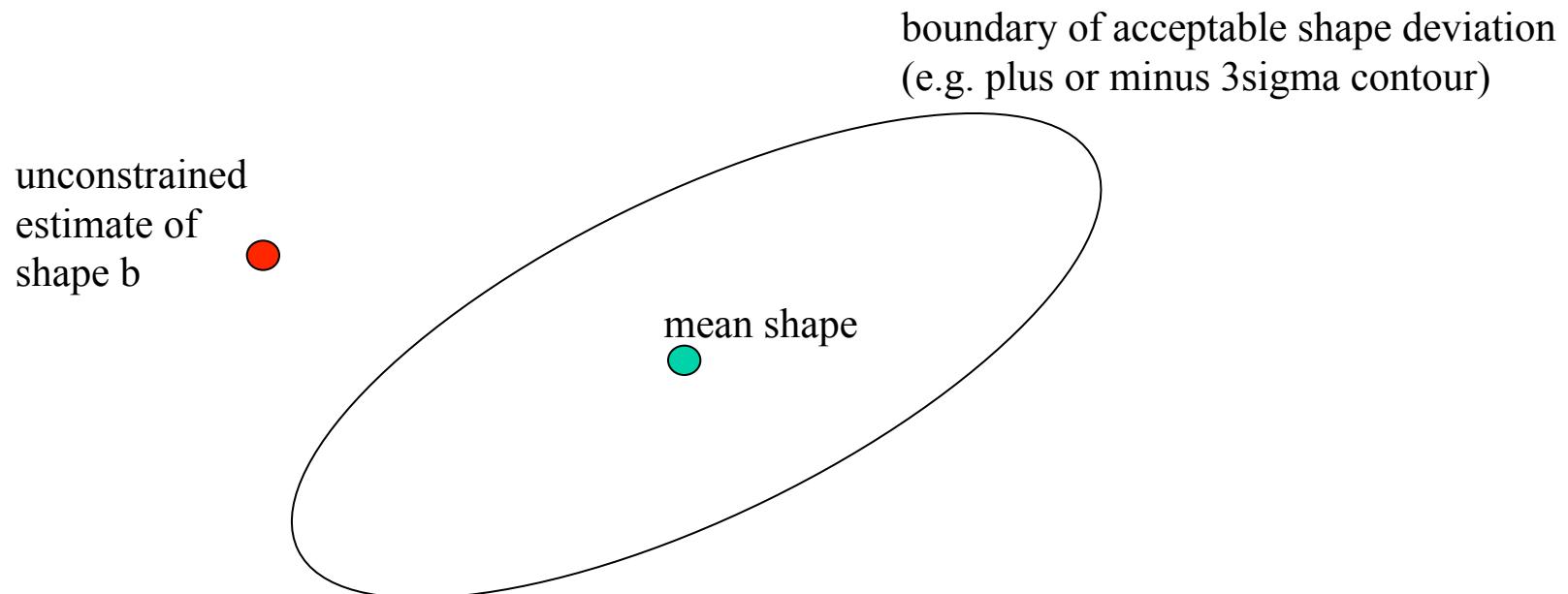
- Soft constraints

Minimise $| T^{-1}(\mathbf{X}) - (\mathbf{x} + \mathbf{P}\mathbf{b}) |^2 / \sigma_r^2 + \log(p(\mathbf{b}))$

- Can also weight by quality of local match

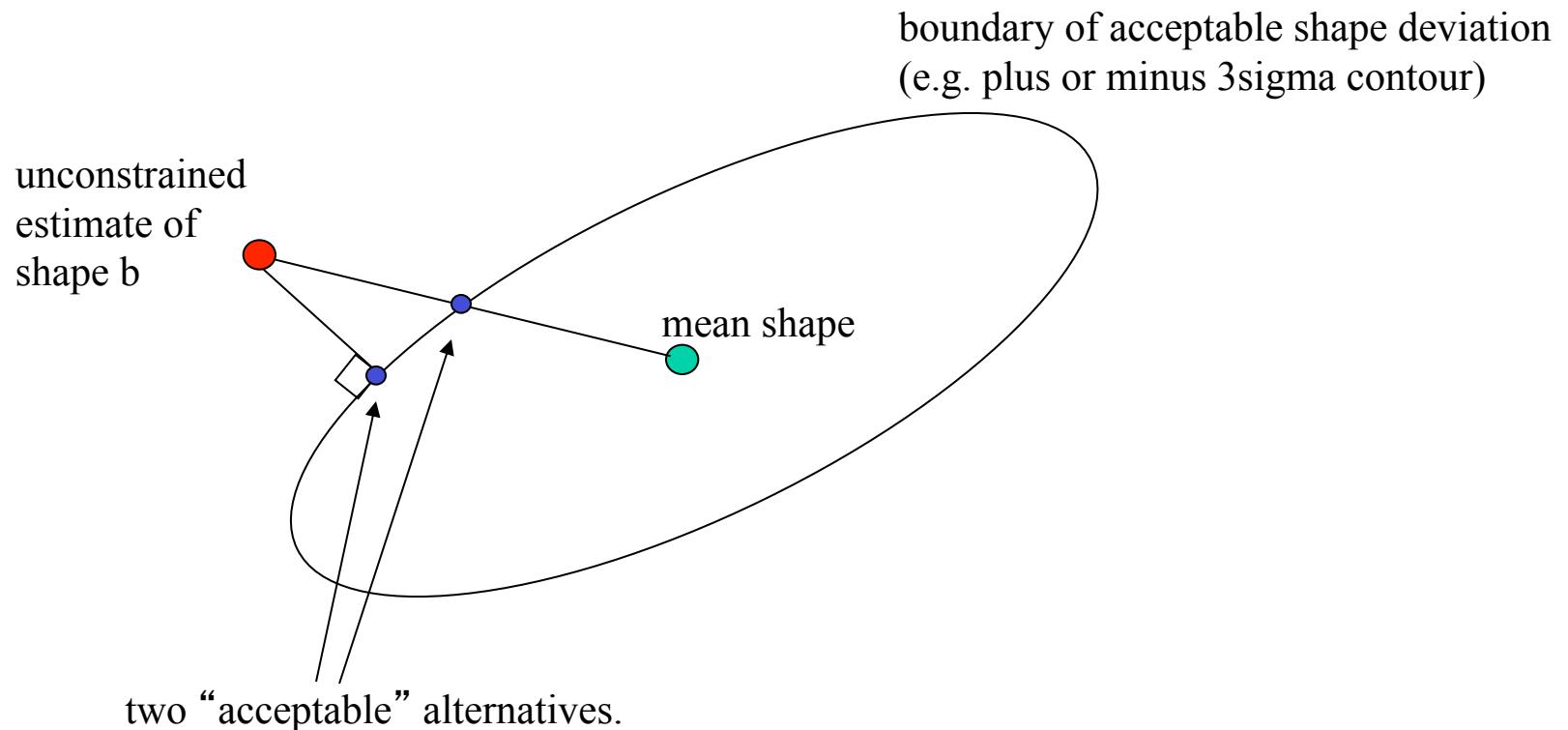
Update Step for b

- Intuition: project b onto closest “acceptable” shape.



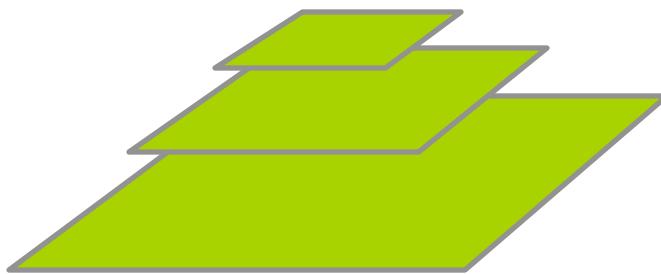
Update Step for b

- Intuition: project b onto closest “acceptable” shape.



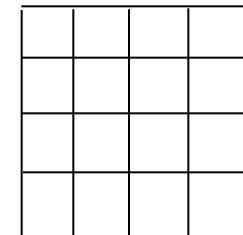
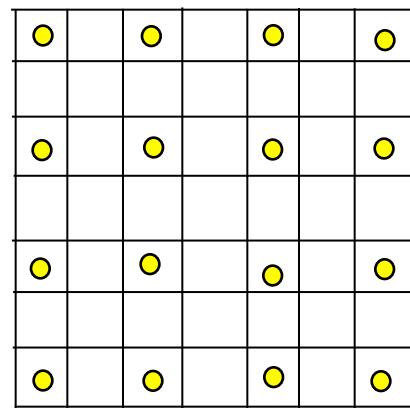
Multi-Resolution Search

- 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



Gaussian Pyramids

- To generate image at level L
 - Smooth image at level L-1 with gaussian filter (eg $(1 \ 5 \ 8 \ 5 \ 1)/20$)
 - Sub-sample every other pixel

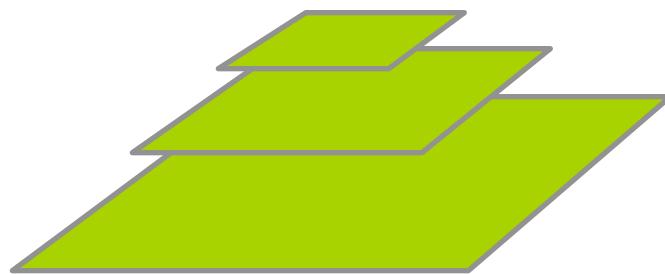


Each level half
the size of the
one below

Implementation Detail

Multi-Resolution Search

- Start at coarse resolution
- For each resolution
 - Search along profiles for best matches
 - Update parameters to fit matches
 - (Apply constraints to parameters)
 - Until converge at this resolution



Example



Initial pose



After 5 iterations



Convergence

Active Shape Models

- Advantages
 - Fast, simple, accurate
 - Efficient to extend to 3D
- Disadvantages
 - Only sparse use of image information
 - Treat local models as independent

Active Appearance Models

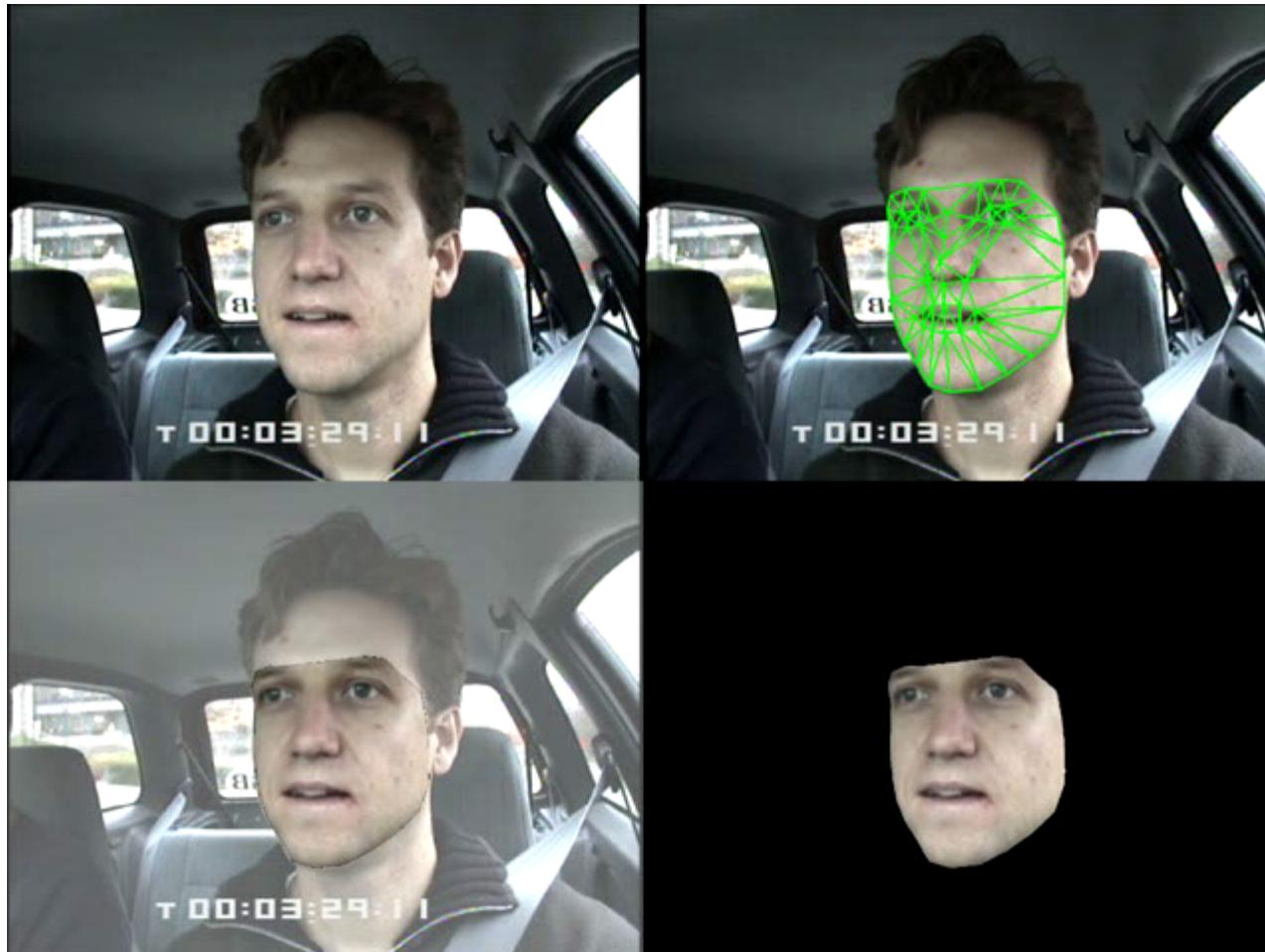
Combined shape and appearance modeling:

- Triangulated mesh model
- Model shape variation using Active Shape Model
- also model intensity variation within each patch



Baker and Matthews, http://www.ri.cmu.edu/projects/project_448.html

Active Appearance Models



Baker and Matthews, http://www.ri.cmu.edu/projects/project_448.html

Active Appearance Models

