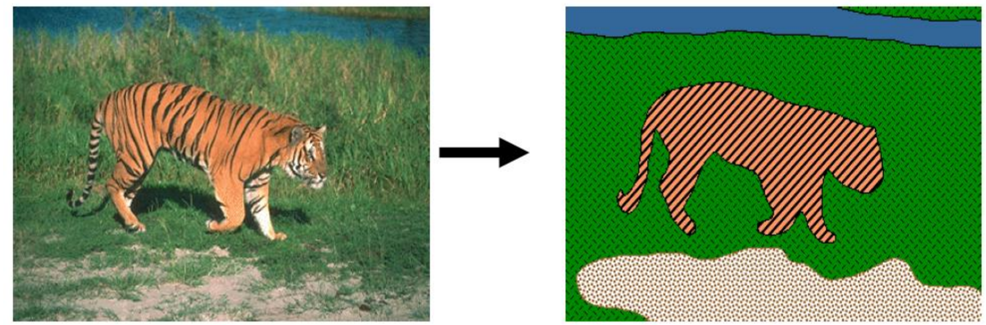
Medical Image Analysis
Lecture 09

Statistical Prior Based Segmentation

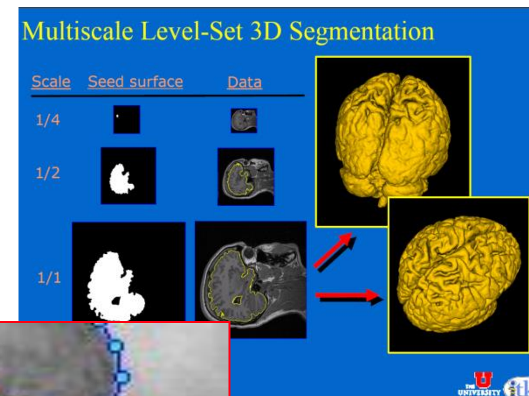
Martin Urschler

Image Segmentation Overview

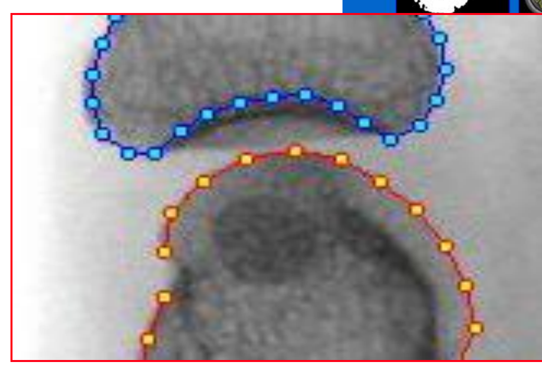
- Low-Level Methods



- High-Level
  Deformable Models

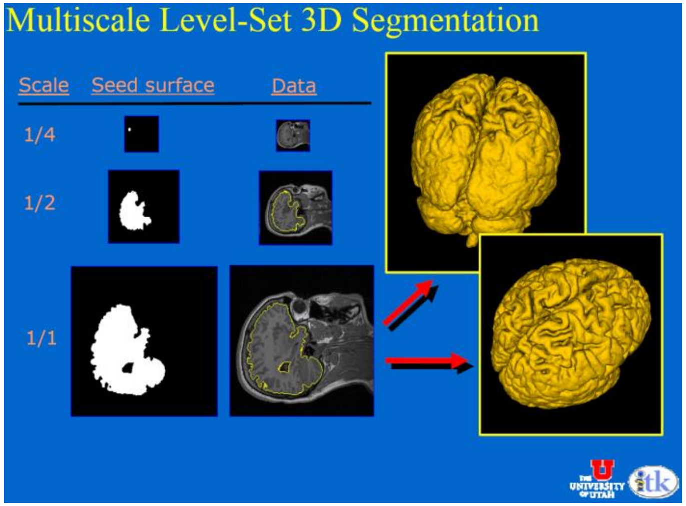


- Shape & Appearance
  Prior based
  Deformable Models



Model Based Segmentation

- High-Level Deformable Models (AC, LS)
  - „Weak“ Model Knowledge:
  - Segmentation is connected
  - Contour/Surface is „smooth“



- Strong Model Knowledge:
  - Explicitly train on prior instances!
  - Subspace Models of Appearance & Shape

Model Based Segmentation - Idea

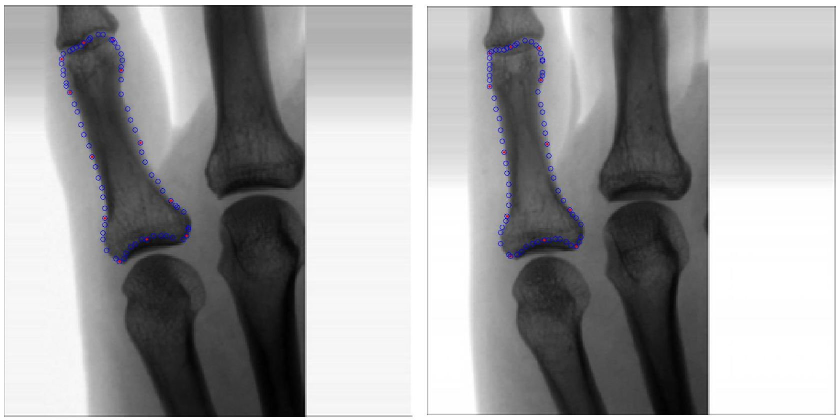
Sample Objects (Training Instances)

Generate Parameterized Model

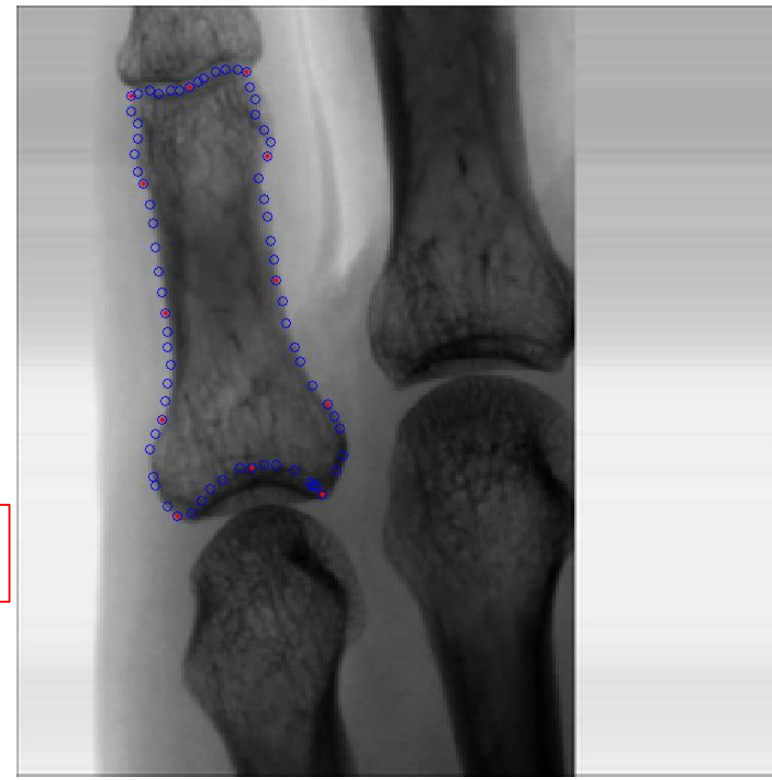
Training Instance $\mathbf{x} = \bar{\mathbf{g}} + \mathbf{P}\mathbf{b}$ Parameters

Model Based Segmentation - Idea

Training Instances



New Unseen Image



Parameterized Model

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

Modify Model Parameters

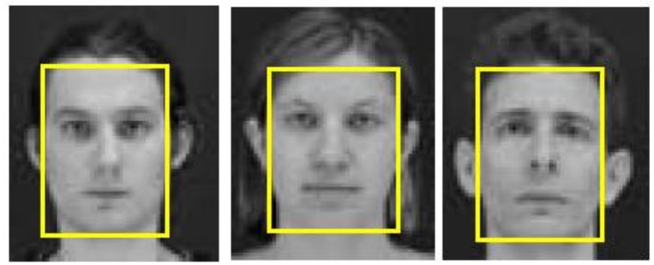
Synthetic Object

Model Based Segmentation

- Subspace Models
  - Patch Based Approaches („EigenFace“)
    - Basic Approach for ASM, AAM
  - Statistical Shape Models & Active Shape Models (ASM)
  - Statistical Appearance Models & Active Appearance Models (AAM)

EigenFace (EigenPatch) Models

Mark face region on training set



Initial Registration!

Sample region

$\mathbf{g}'$

Normalize (global lighting)

$\mathbf{g}$

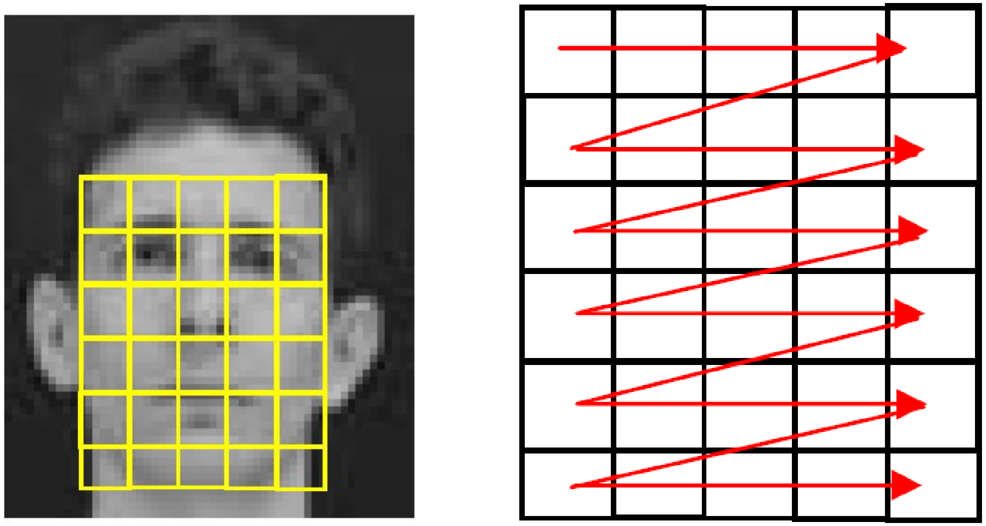
$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{P}\mathbf{b}$$

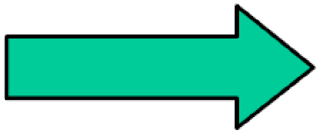
Statistical Analysis

Compact model of face patch variations

Representing Regions

- Represent each region as a (feature) vector
  - Raster scan values k x m region:
    n=k*m vector -> n-dimensional feature vector g'

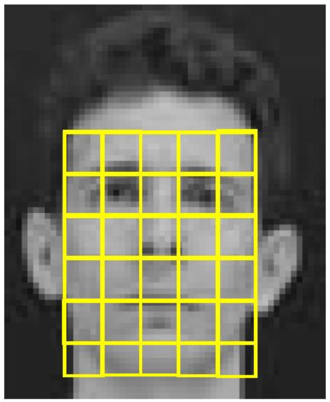




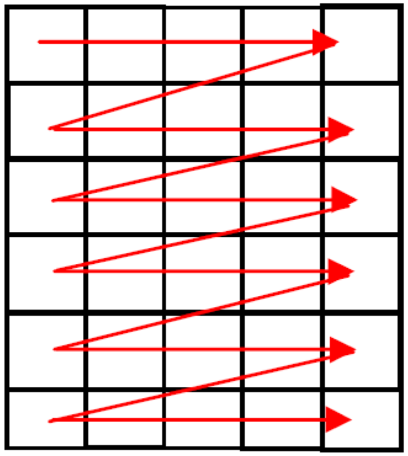
g'

Lighting Normalization

Investigate n-dim. Feature Space



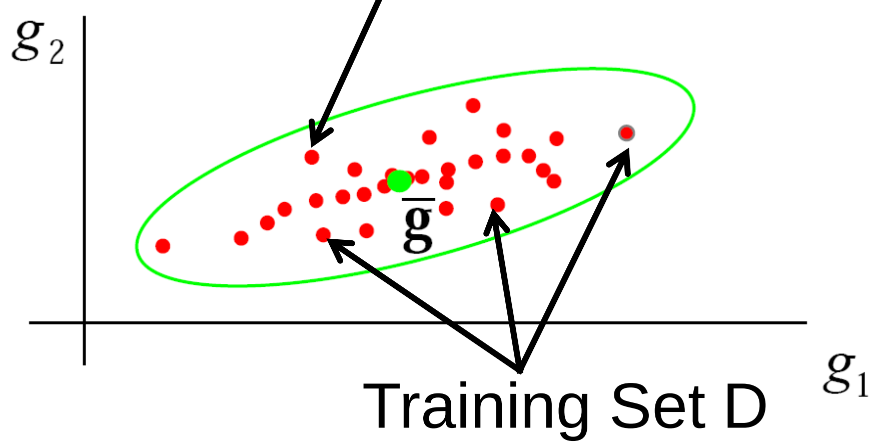
$\mathbf{g}^1$



Training Set $\mathbf{D} = \left(\mathbf{g}^1 \mathbf{g}^2 \ldots \mathbf{g}^s\right)$

$$\boldsymbol{g} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

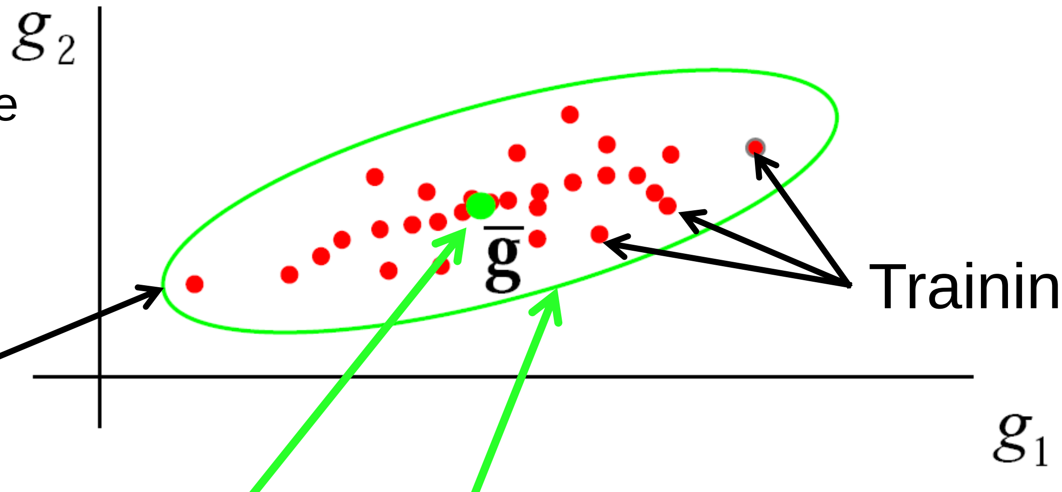
Here we visualize only 2 of the n dimensions!



Gauss Model Assumption (Generative)

Q: How do we compute the parameters of the Gauss Model?

A: Mean is simple. Hyperellipsoid from Principal Component Analysis (PCA)



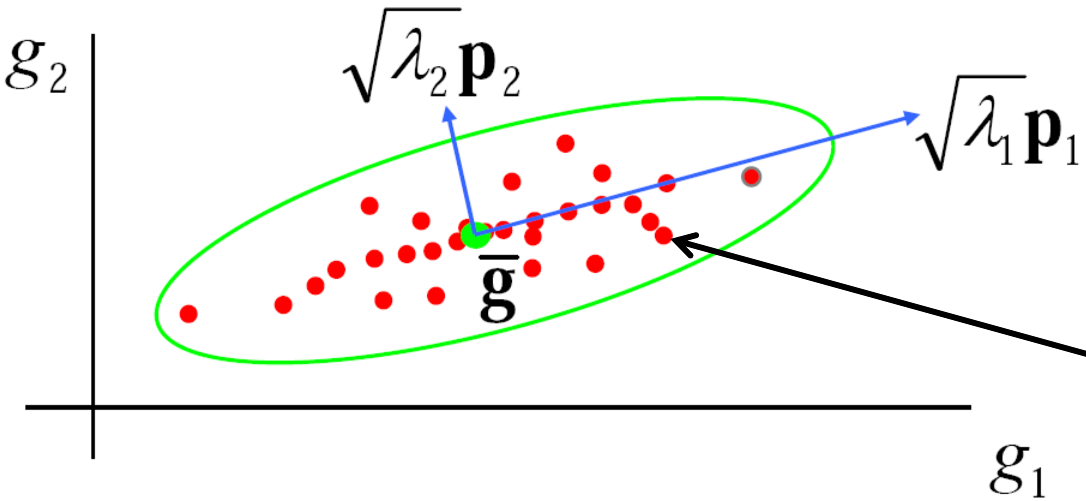
Mean and covariance matrix C of s data vectors define a Gaussian model in n-dim. space! -> Parametric Density Estimation

$$\bar{\mathbf{g}} = \frac{1}{s}\mathbf{D}\mathbf{1} \quad \mathbf{C} = \frac{1}{s-1}\hat{\mathbf{D}}\hat{\mathbf{D}}^T$$

$$\hat{\mathbf{D}} = \left\{ \mathbf{g}^1 - \bar{\mathbf{g}} \quad \mathbf{g}^2 - \bar{\mathbf{g}} \quad \cdots \quad \mathbf{g}^s - \bar{\mathbf{g}} \right\}$$

Principal Component Analysis

- We need projection lines that maximize the variance of the projected data -> Least Squares Problem
- To solve, we compute eigenvectors **P** of covariance **C**
- Eigenvalue: variance of projected data along eigenvector
- Eigenvectors **P**: main directions sorted by variance

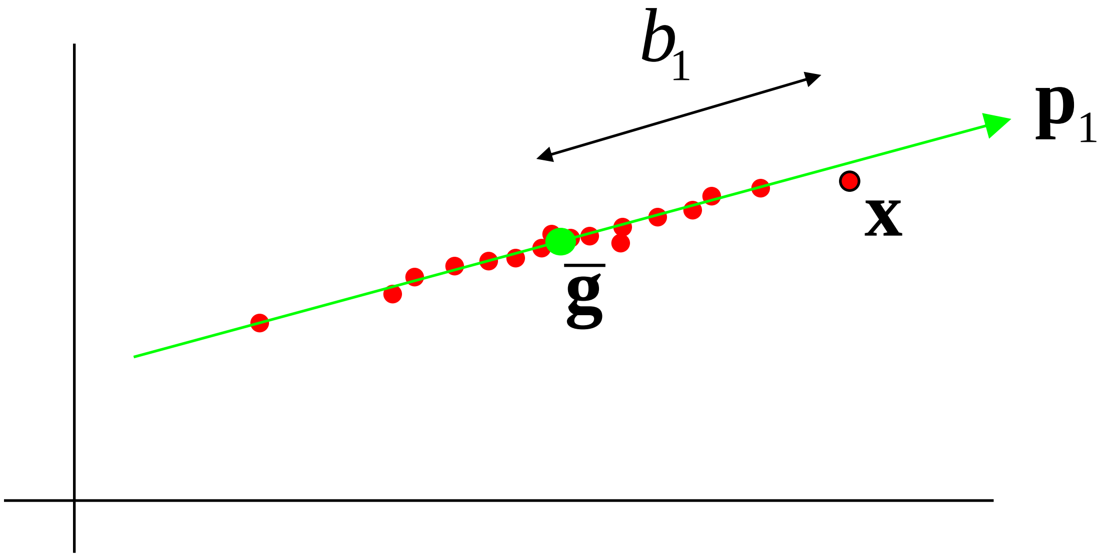


Projections of g

$$\mathbf{b} = \mathbf{P}^T(\mathbf{g} - \overline{\mathbf{g}})$$

Dimensionality Reduction

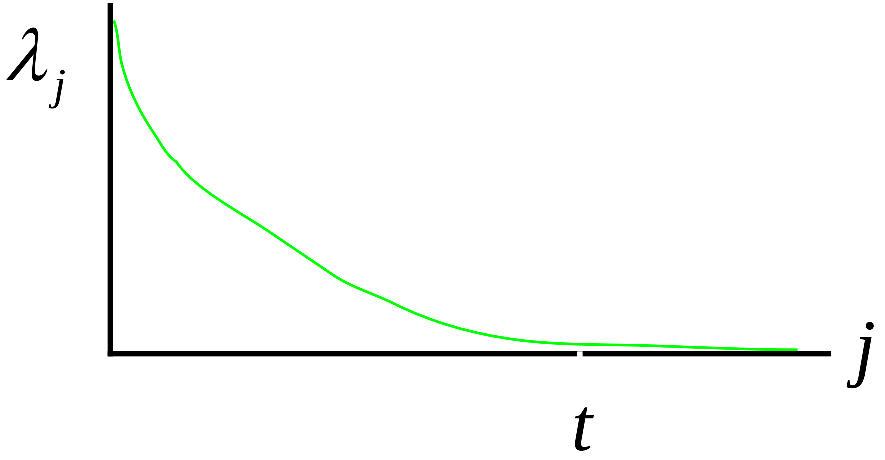
- Subspace Representation
- Project to Eigenvector with large variance!



$$\mathbf{x} \approx \overline{\mathbf{g}} + \mathbf{p}_1 b_1$$

Dimensionality Reduction

- General $$\mathbf{x} = \overline{\mathbf{g}} + \mathbf{Pb} = \overline{\mathbf{g}} + \mathbf{p}_1 b_1 + \cdots + \mathbf{p}_n b_n$$
- However, for some *t*, $b_j \approx 0 \text{ if } j > t$
  - Variance corresponding to b_j is λ_j



Building Eigen-Models

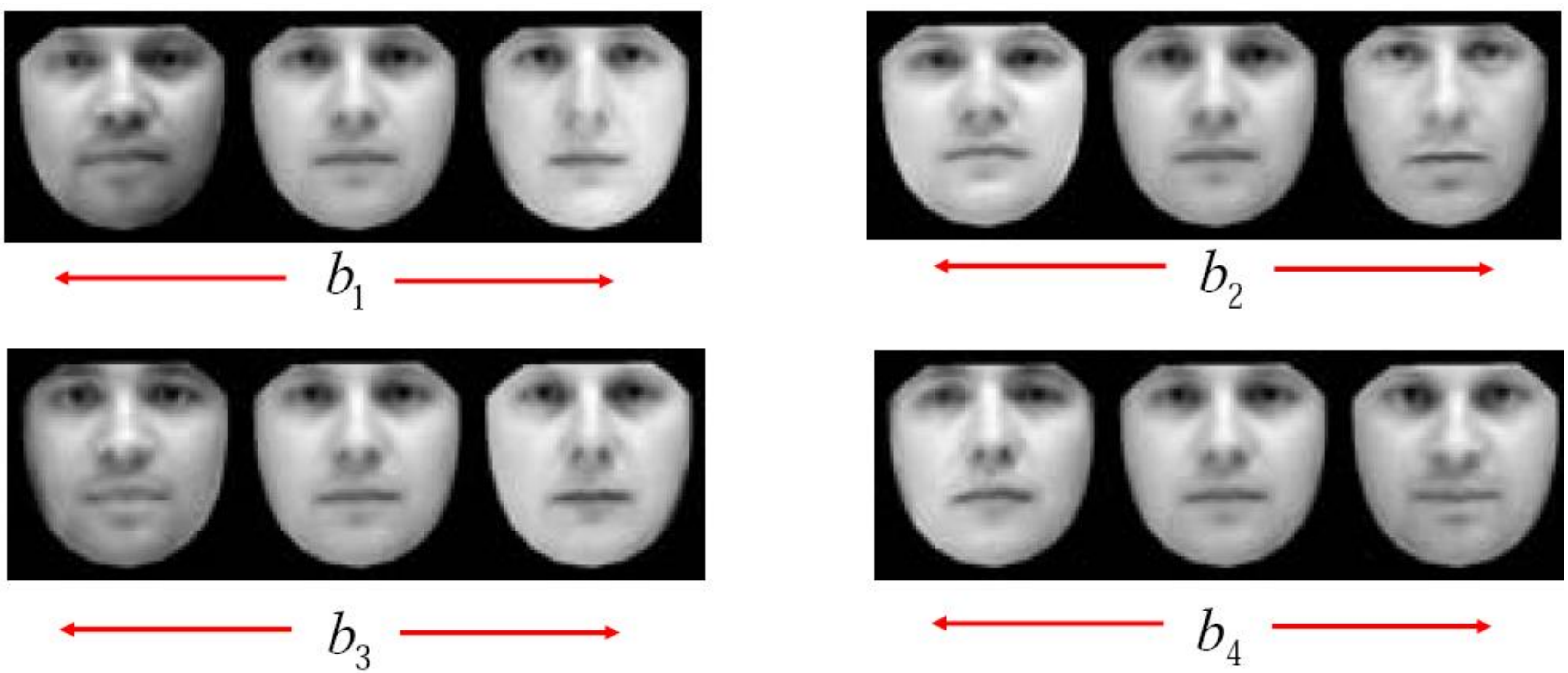
- Given set of training examples $\{\mathbf{g}_i\}$
- Compute mean and eigenvectors P of covariance matrix C
- Model is then

$$\mathbf{g} \approx \overline{\mathbf{g}} + \mathbf{P}_t \mathbf{b}_t$$

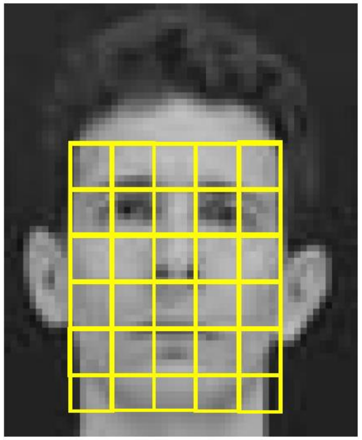
- $\mathbf{P_t}$ – First t eigenvectors of covariance matrix
- $\mathbf{b_t}$ – Model parameters (projection of each training example into subspace)

Generative Eigen-Face models

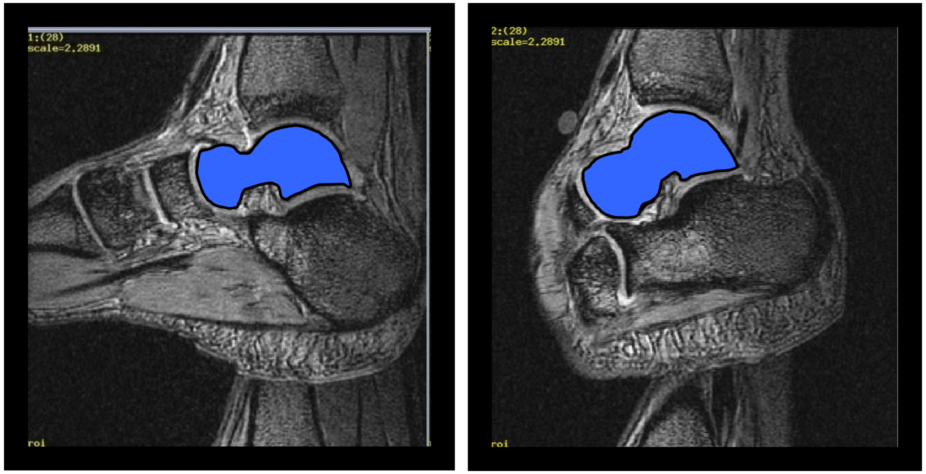
- Model of variation in region $\mathbf{g} \approx \bar{\mathbf{g}} + \mathbf{P}\mathbf{b}$



Appearance Models



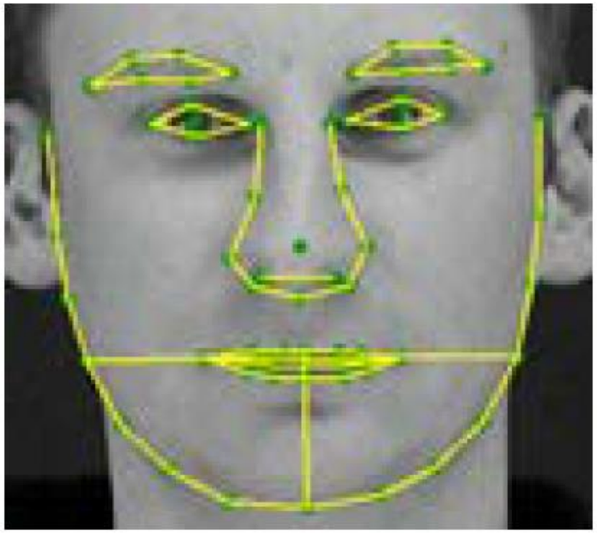
- Now we have a model for appearances
- However: registration and face deformation (expression) are problems
- Shape variance and the registration problem leads to statistical shape models!
- Combination leads to statistical models of shape and appearance!

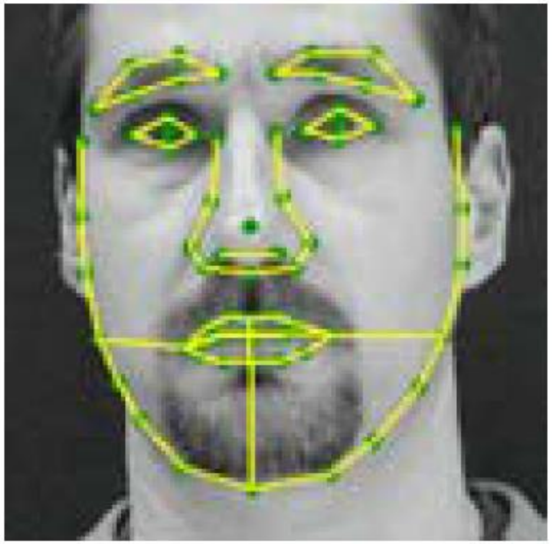


Statistical Shape Models

Building Models - Landmarks

- Different Idea: Model based on landmarks instead of face patches
- Represent shape using a set of points
- Requires labeled training images
  - Landmarks represent correspondences



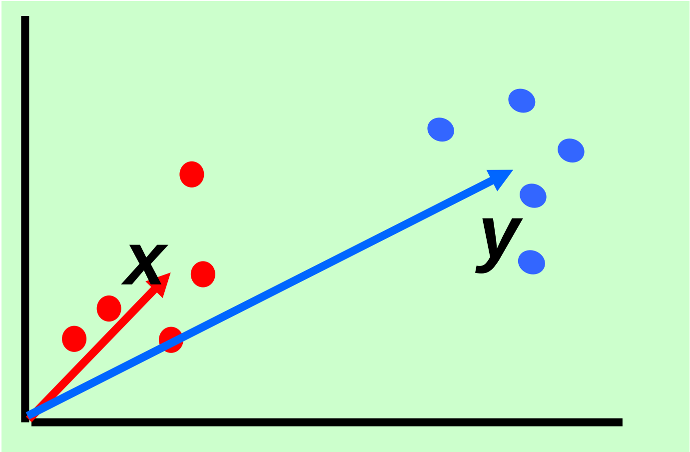


Statistical Shape Models

- Given set of training shapes (s point sets with N corresponding points)
- Align shapes into common coordinate frame to remove similarity transform (Registration)
  - Generalized Procrustes Analysis
  - Only differences due to shape deformations remain
- Estimate shape distribution
  - Single Gaussian often sufficient (again PCA model)

Aligning Two Shapes

- Procrustes Analysis:
  - Find transformation which minimizes L2 norm of the differences between first and transformed second point set
  - Resulting shapes have
    - Identical center of gravity (CoG)
    - Approximately the same scale and orientation
  - Align Shapes minimizing $D = \frac{1}{N} \sum_{i=1}^{N} | R\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i |^2$



Point-Based Registration

- The Procrustes Problem:
  - Given two sets of *N* corresponding points **P** = {$\mathbf{p}_i$} and **Q** = {$\mathbf{q}_i$}
  - Find the rigid-body transformation (rotation matrix ***R*** and translation vector **t**) that minimizes the mean squared distance between the points:

$$\mathrm{E}_{\mathrm{Procr.}} = \frac{1}{N}\sum_{i=1}^{N} |\,\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\,|^2$$

  - E is the *Procrustes registration error*
  - **P** and **Q** represented as matrices

$$\mathbf{P} = \left[\mathbf{p}_1^T, \ldots, \mathbf{p}_i^T, \ldots, \mathbf{p}_N^T\right] = \begin{bmatrix} p_{1,x} & p_{1,y} \\ p_{2,x} & p_{2,y} \\ \ldots & \ldots \\ p_{N,x} & p_{N,y} \end{bmatrix}$$

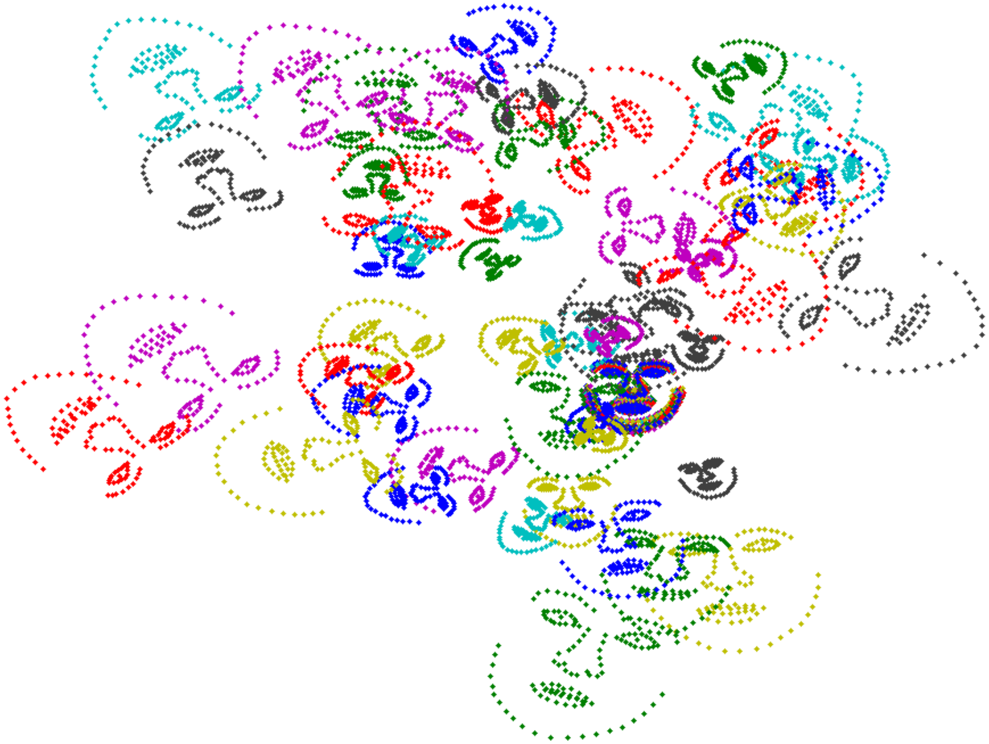
Point-Based Registration

- Procrustes Alignment – 3 steps
  - Center both sets of points
  - Determine rotation -> SVD
  - Determine translation

- See previous Lecture 7 for details!

Aligning a Set of Shapes

- Now we know how to align two shapes (Procrustes A.)



Aligning a Set of Shapes

- Generalized Procrustes Analysis
  - Find the N transformations T_i which minimize

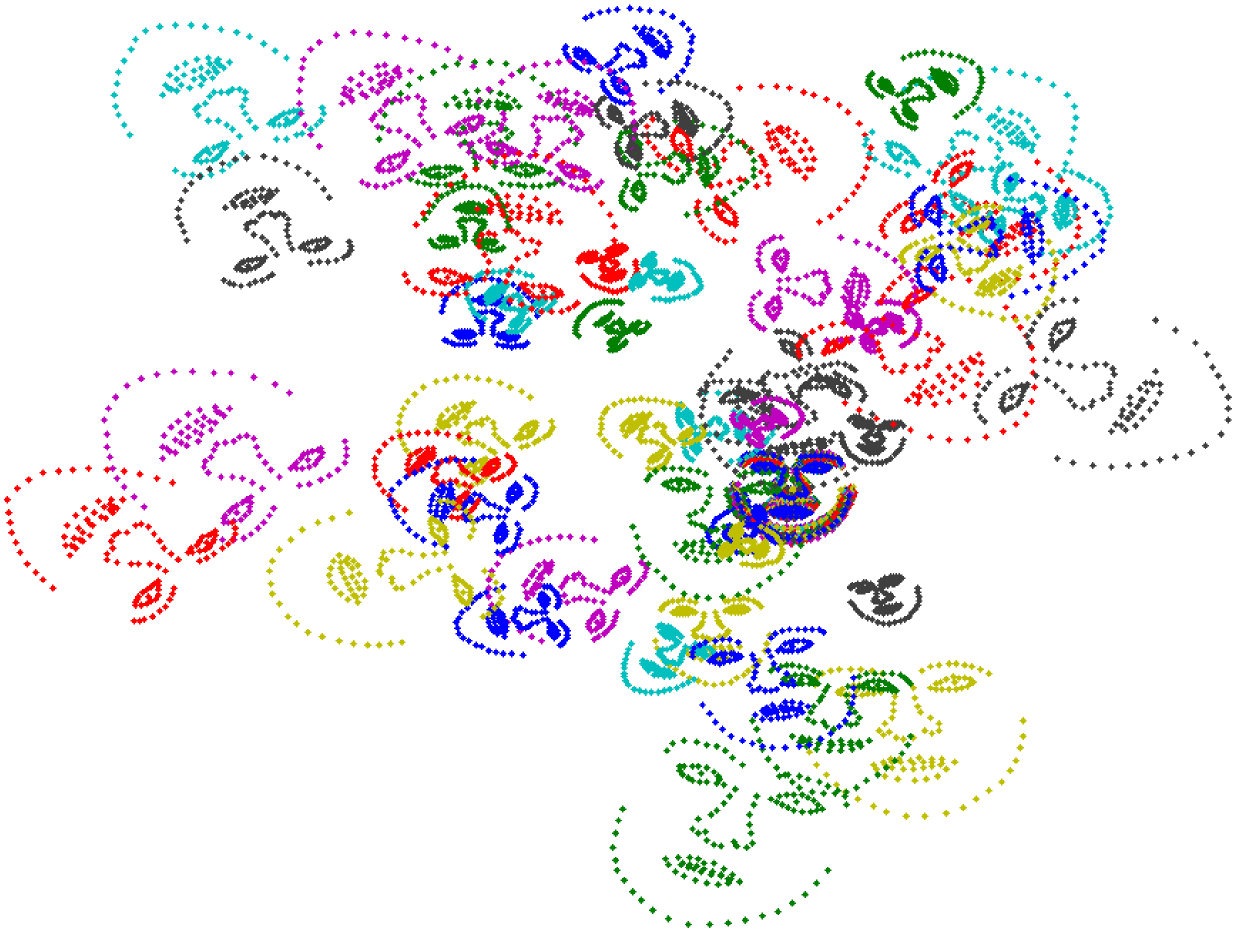
$$\sum \left| \mathbf{m} - T_i(\mathbf{x}_i) \right|^2$$

  - Where $\mathbf{m} = \frac{1}{N} \sum T_i(\mathbf{x}_i)$ is a mean shape estimate

Aligning Shapes: Algorithm

- Normalize all shapes to CoG at origin
- Let $\mathbf{m}=\mathbf{x}_\mathrm{i}$ (i is a randomly chosen shape)
- Align each shape with $\mathbf{m}$ (use Procrustes A.)
- Re-calculate $$\mathbf{m} = \frac{1}{N}\sum T_i(\mathbf{x}_i)$$
- Normalize $\mathbf{m}$ to default size, align with previous estimation of $\mathbf{m}$
- Repeat until convergence

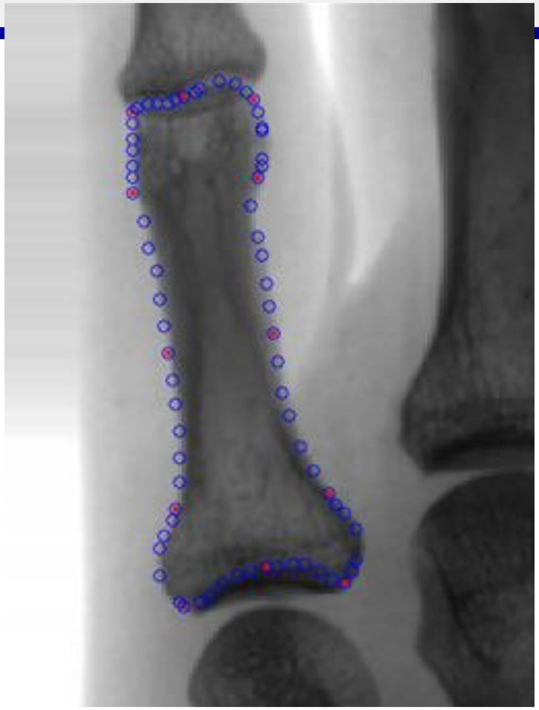
Aligning Shapes - Example



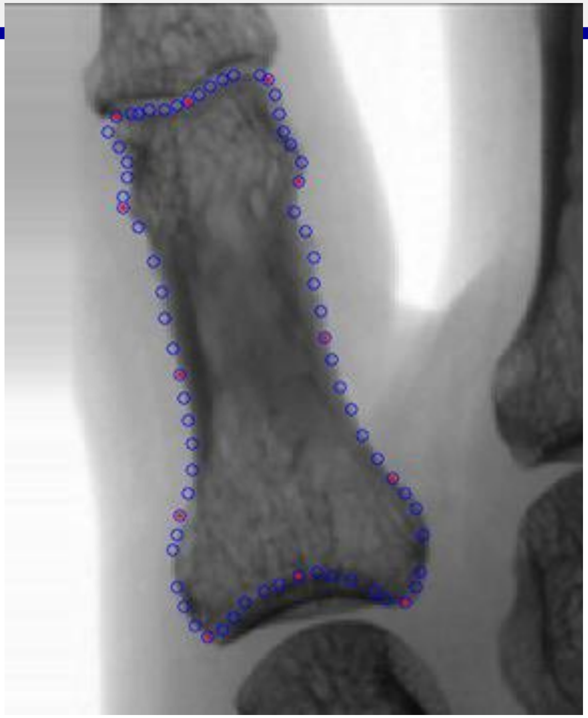
Aligning Shapes - Example

Building Shape Models

$\mathbf{x}^1$



$\mathbf{x}^j$



$\mathbf{x}^s$

Define shape vector of aligned data, e.g.:

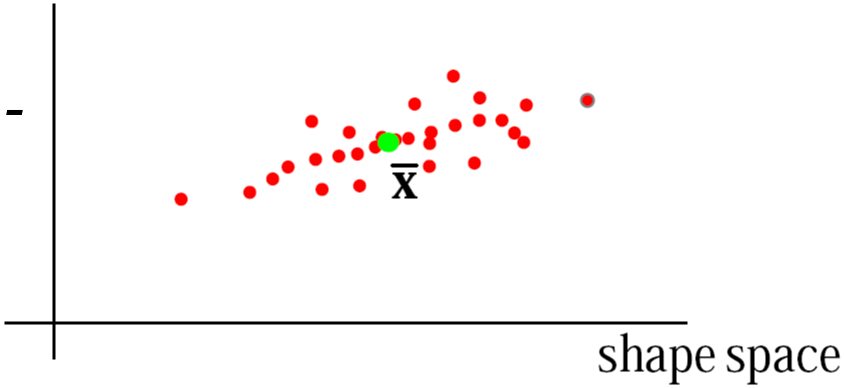
$$\mathbf{x}_j = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^T$$

Building Shape Models

$$\mathbf{x}_j = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^T$$

An instance x_j of the training set is a 2n column vector with x and y of all (aligned) points stacked together!

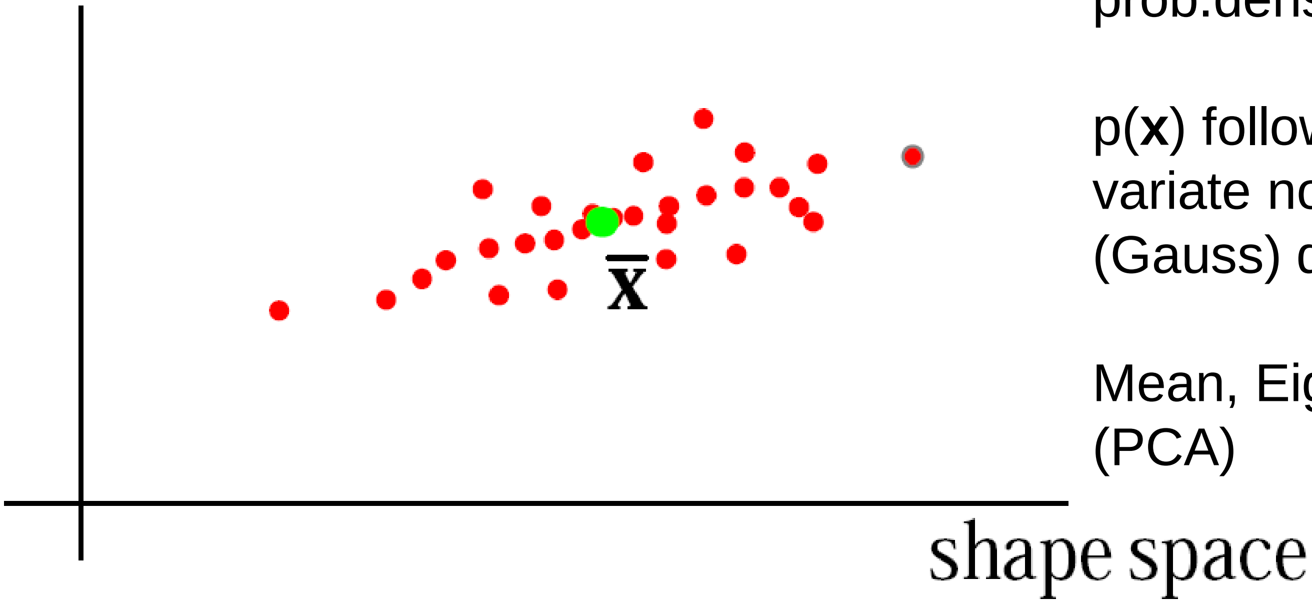
This gives us a high-dimensional feature space similar to the Eigen-Face approach!



Note: Stacked points define correspondence!!!

Aligned Shapes

- Need to model the s aligned shapes
- Shapes form a distribution in 2n-dim. space (shape space)



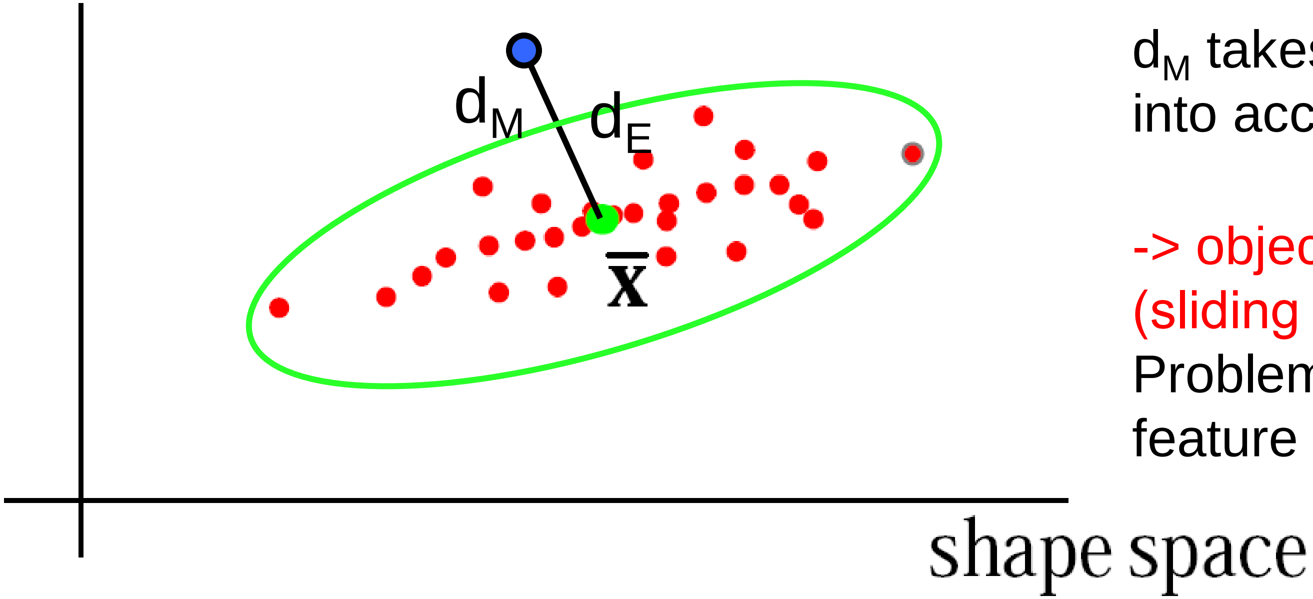
Aligned shapes re-presented by their prob.dens.func. p(**x**)

p(**x**) follows multi-variate normal (Gauss) distribution

Mean, Eigenvectors (PCA)

Statistical Shape Models

- To compare a given shape to the model, knowing p(**x**) would be sufficient (matching)
  - Use Euclidean d_E or better Mahalanobis distance d_M to mean



d_M takes covariance into account

-> object detection (sliding "window")
Problem: high-dim. feature space

Statistical Shape Models

- For shape synthesis & model fitting
  - Parameterized model preferable

$$\mathbf{x} = f_{shape}(\mathbf{b})$$

  - A PCA model is simple and compact, therefore it is frequently used

$$\underbrace{\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb}}_{f_{shape}}$$

Building Shape Models

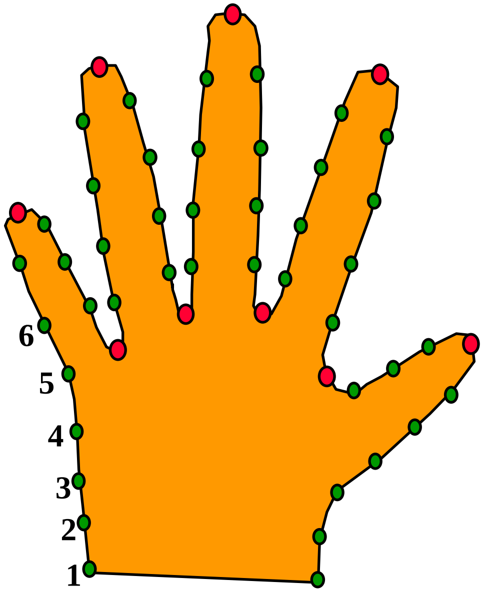
- Given aligned shapes, $\{\mathbf{x}_i\}$, $i=1,\ldots,s$
- Compute Mean, apply PCA, remove eigen-vectors corresponding to small eigenvalues

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

- $\mathbf{P}_\mathbf{t}$ – First t eigenvectors of covariance matrix
  – We want Dimensionality Reduction!
- $\mathbf{b}_\mathbf{t}$ – Shape model parameters
  – Defines a set of deformable model parameters!

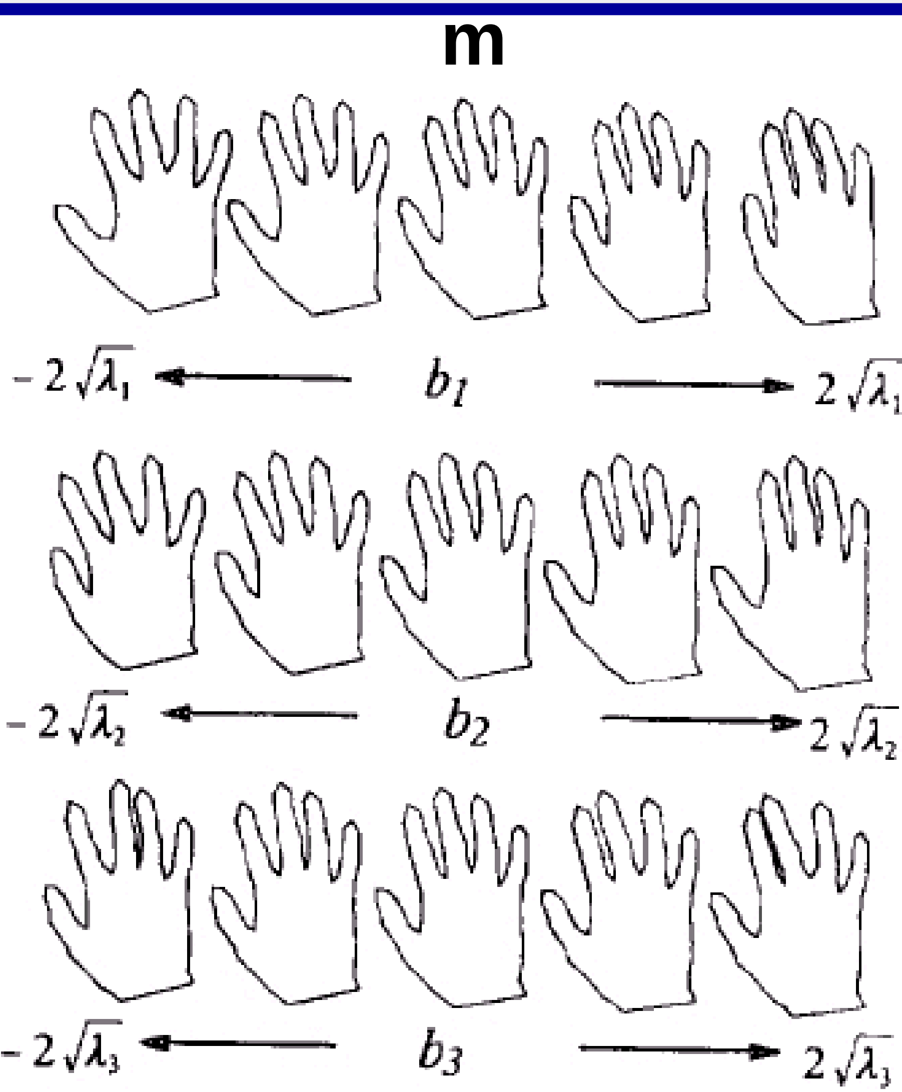
Hand Shape Model

- 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 over outline

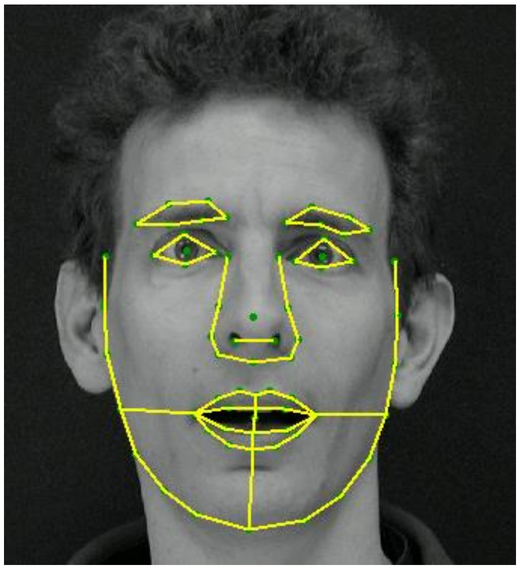


Hand Shape Model

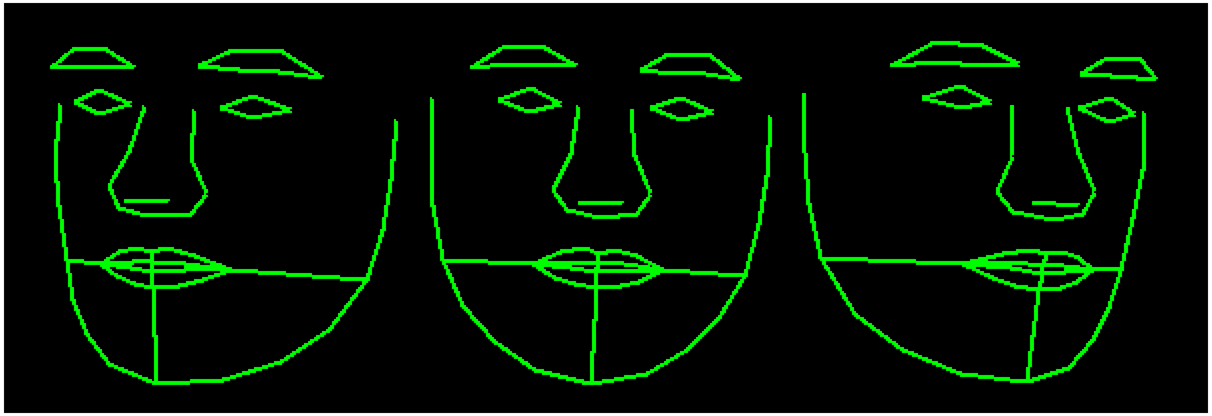
- 96% of variability due to first 6 modes
- First 3 modes vary finger movements



Face Shape Model

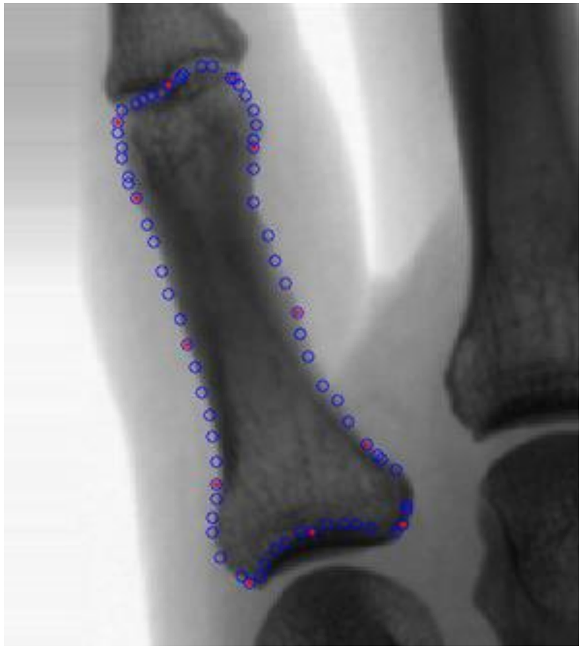


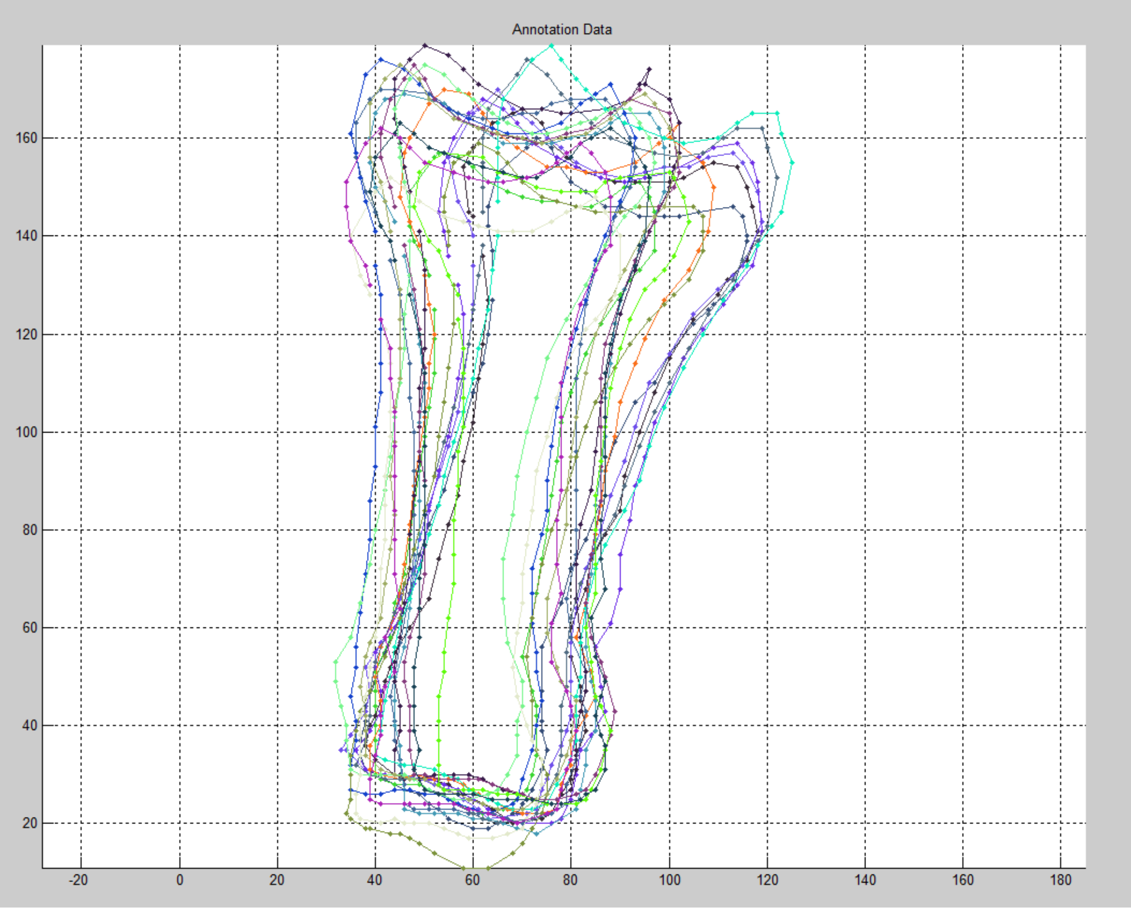
Shape of the facial structures with 68 points



Bone Shape Model

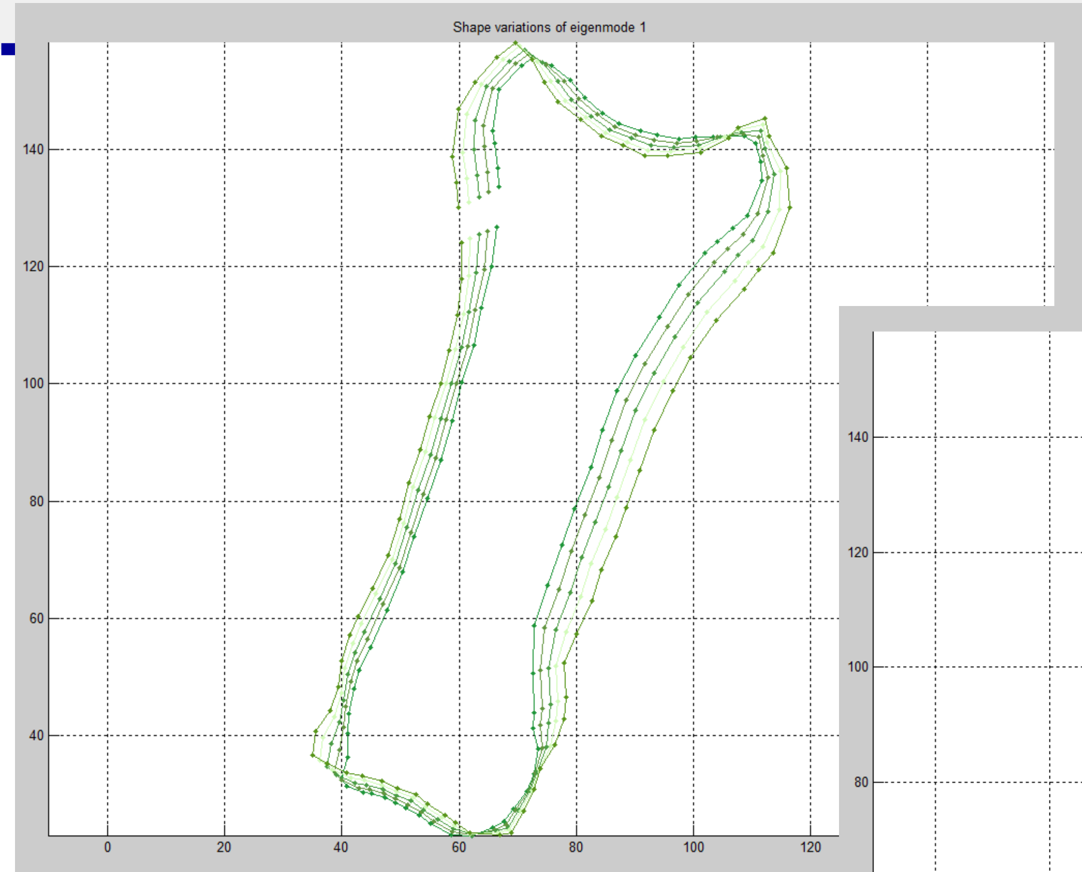
- Input: 20 training images with annotated bone shapes





After Generalized Procrustes

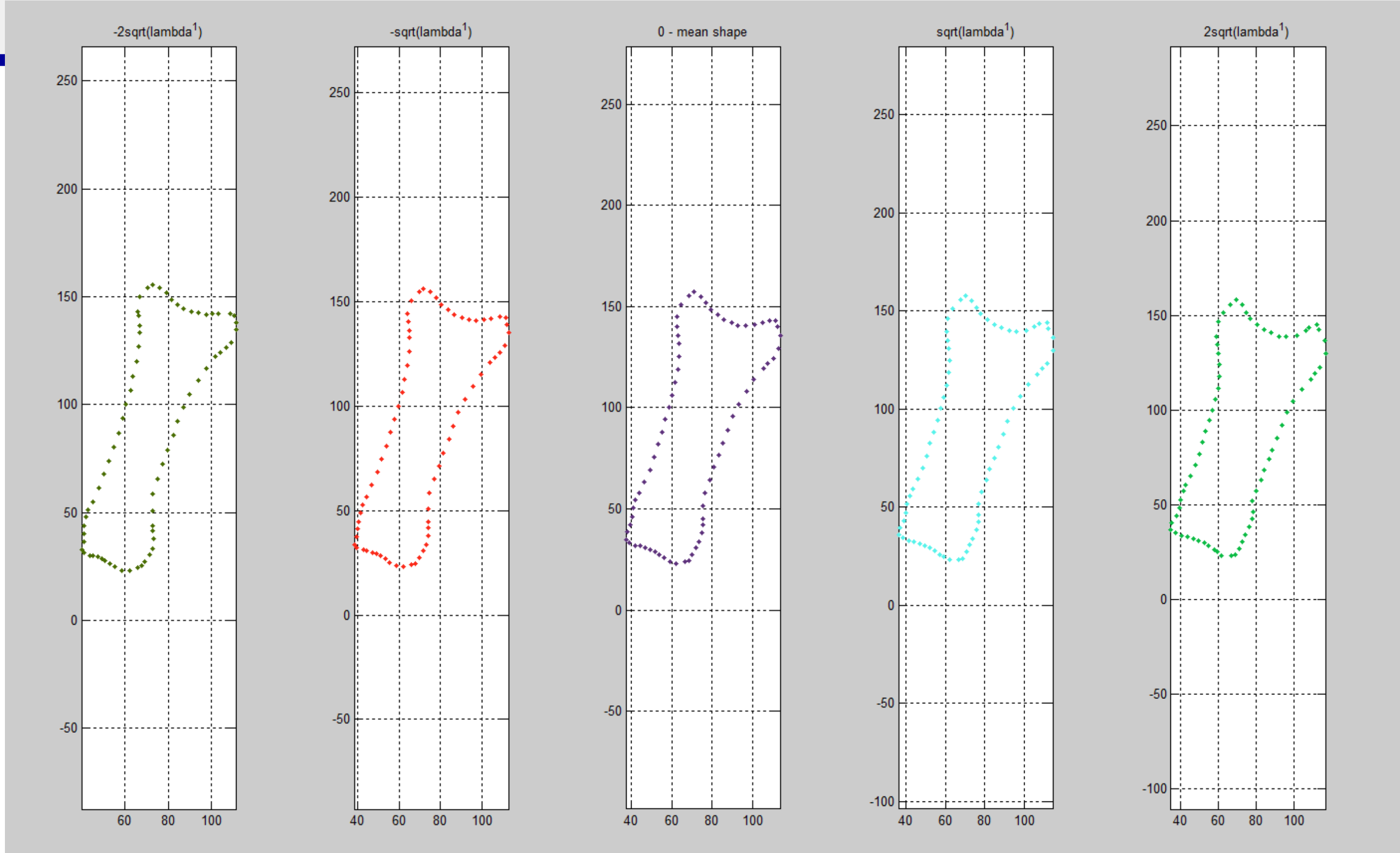
Shape Variations of Eigen-Modes



Shown After PCA

Variation Against
Mean Shape

Deformable Model w/ Shape Prior



Model:

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

By modifying model parameters b, we can now create synthetic object instances restricted to training shapes!

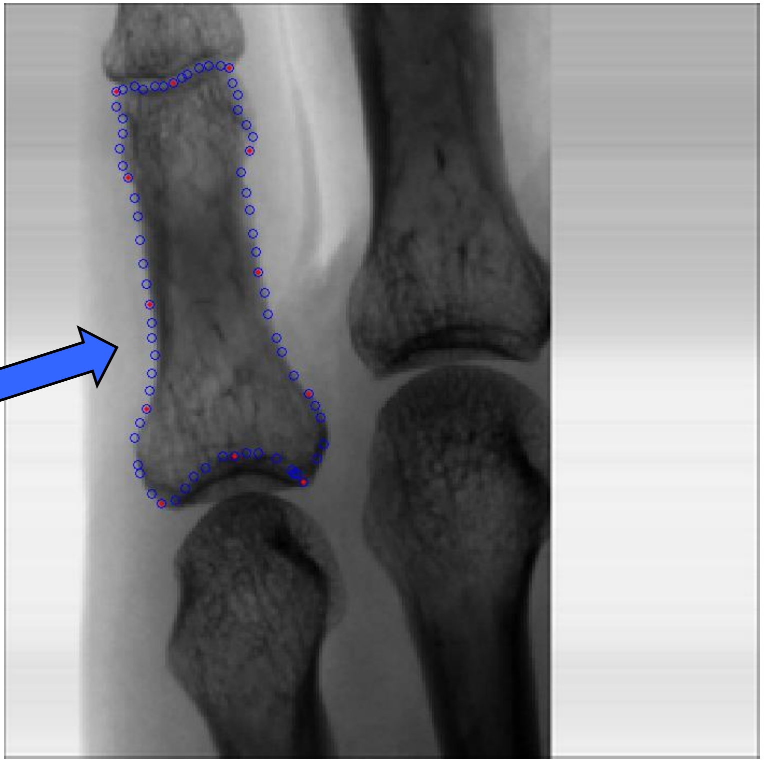
Deformable Model Fitting

$$\mathbf{x} = \overline{\mathbf{g}} + \mathbf{P}\mathbf{b}$$

Modify Model Parameters

Synthetic Object

Fit Synthetic Object to Unseen Image



Summary

- We can build statistical models of shape change
- Require correspondences across training set
- Get compact model (few parameters)

- Next: Fitting models to images