Computer vision: models, learning and inference

Chapter 17

Models for shape

Structure

- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

Motivation: fitting shape model

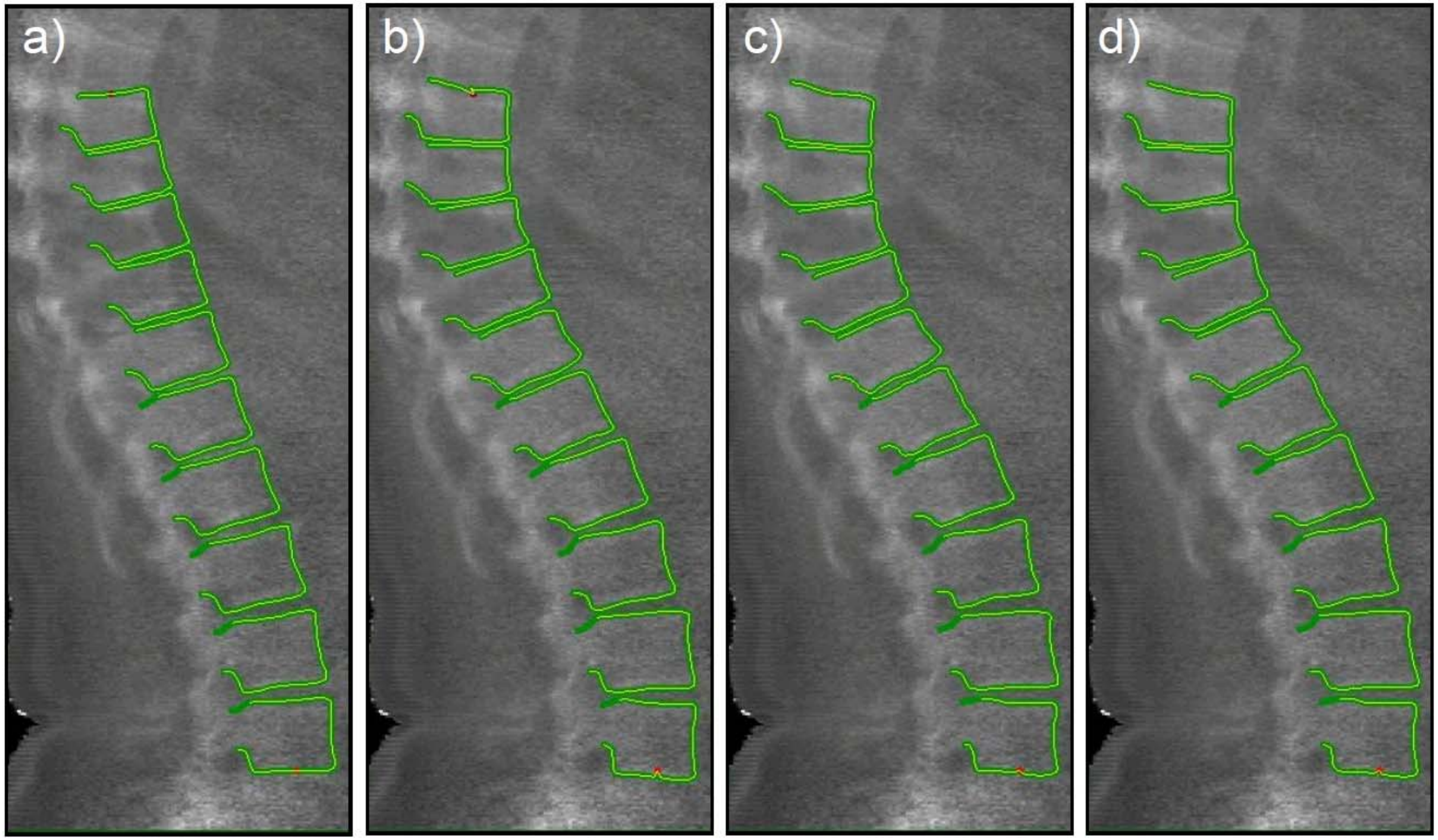


Figure provided by Tim Cootes

What is shape?

- Kendall (1984) – Shape "is all the geometrical information that remains when location scale and rotational effects are filtered out from an object"
- In other words, it is whatever is invariant to a similarity transformation

Representing Shape

- Algebraic modelling
  - Line:

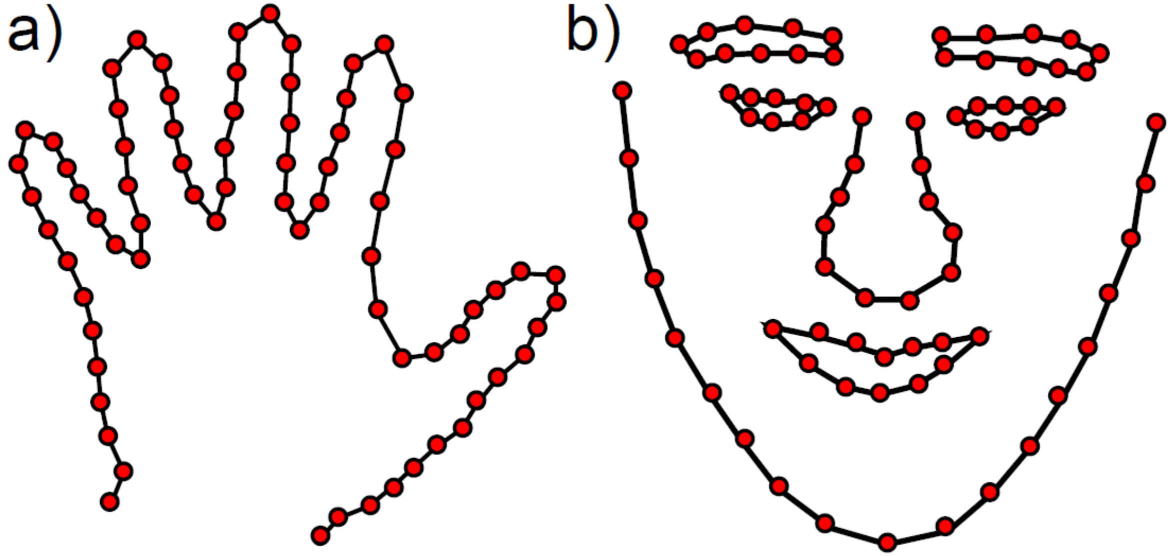
$$ax + by + c = 0$$

  - Conic:

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \alpha & \beta & \gamma \\ \beta & \delta & \epsilon \\ \gamma & \epsilon & \zeta \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

  - More complex objects? Not practical for spine.

Landmark Points



- Landmark points can be thought of as discrete samples from underlying contour
  - Ordered (single continuous contour)
  - Ordered with wrapping (closed contour)
  - More complex organisation (collection of closed and open)

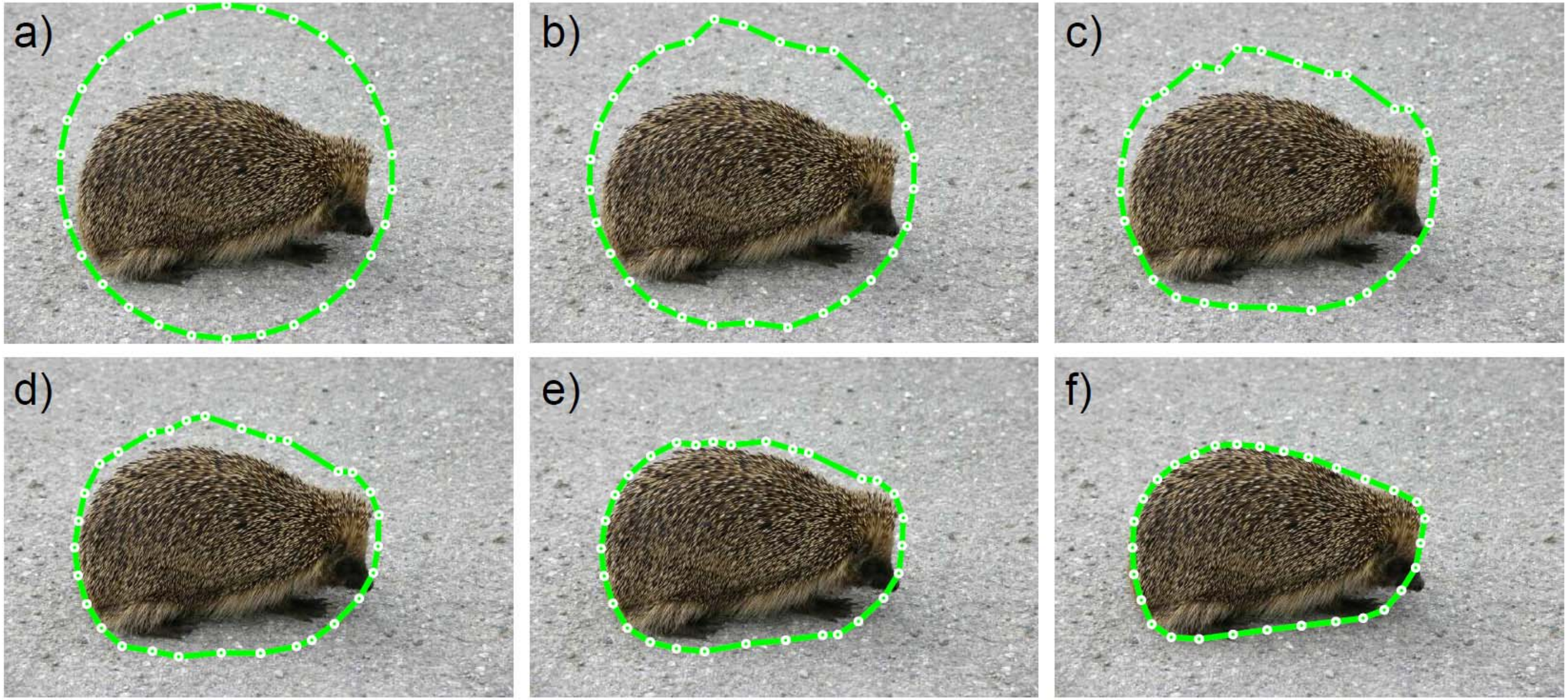
Snakes

- Provide only weak information: contour is smooth
- Represent contour as N 2D landmark points

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N]$$

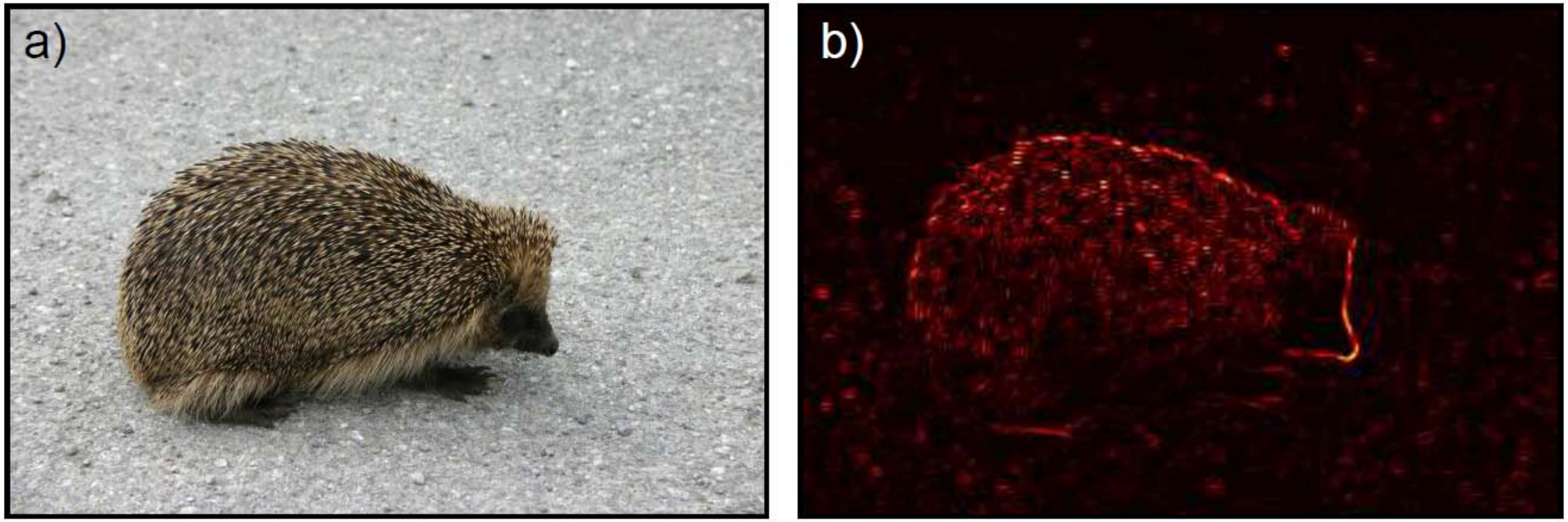
- We will construct terms for
  - The likelihood $Pr(\mathbf{x}|\mathbf{W})$ of observing an image $\mathbf{x}$ given landmark points $\mathbf{W}$. Encourages landmark points to lie on border in the image
  - The prior $Pr(\mathbf{W})$ of the landmark point. Encourages the contours to be smooth.

Snakes



Initialise contour and let it evolve until it grabs onto an object
Crawls across the image – hence called snake or active contour

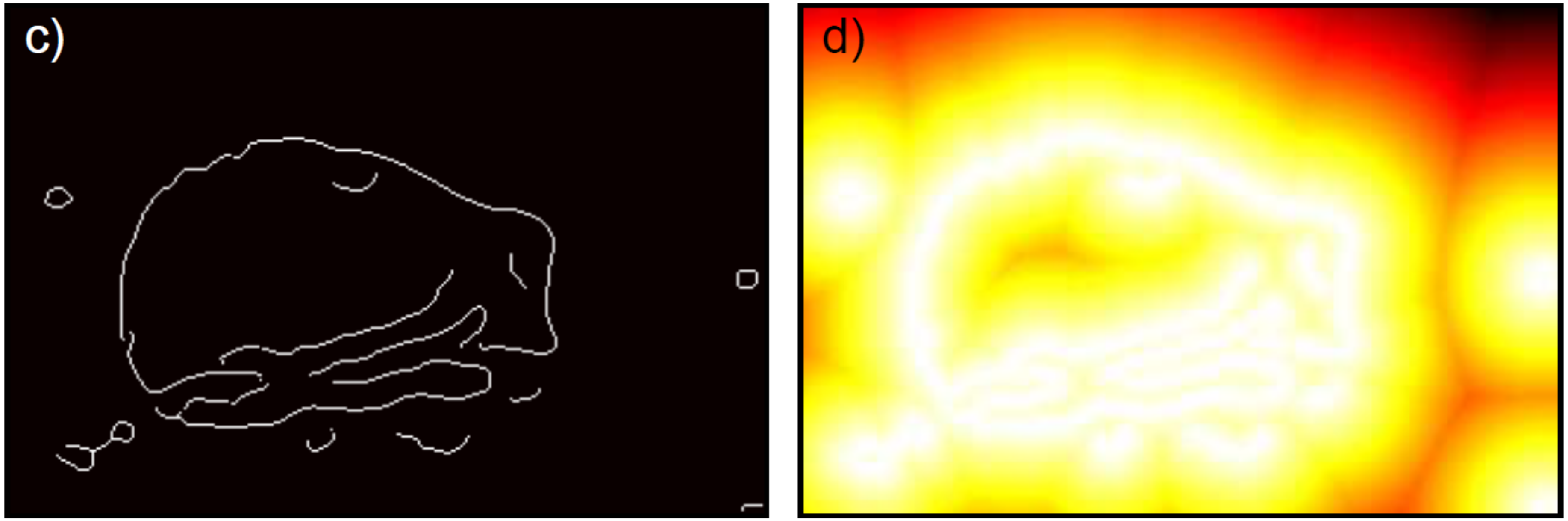
Snake likelihood



$$Pr(\mathbf{x}|\mathbf{W}) \propto \prod_{n=1}^{N} \exp\left[\text{sobel}\left[\mathbf{x}, \mathbf{w}_n\right]\right]$$

Has correct properties (probability high at edges), but flat in regions distant from the contour. Not good for optimisation.

Snake likelihood (2)



Compute edges (here using Canny) and then compute distance image – this varies smoothly with distance from the image

$$Pr(\mathbf{x}|\mathbf{W}) \propto \prod_{n=1}^{N} \exp\left[-(\text{dist}\left[\mathbf{x}, \mathbf{w}_n\right])^2\right]$$

Prior

- Encourages smoothness

$$Pr(\mathbf{W}) \propto \prod_{n=1}^{N} \exp\left[\alpha\, \text{space}[\mathbf{w}, n] + \beta\, \text{curve}[\mathbf{w}, n]\right]$$

  – Encourages equal spacing

$$\text{space}[\mathbf{w}, n] = -\left(\frac{\sum_{n=1}^{N}\sqrt{(\mathbf{w}_n - \mathbf{w}_{n-1})^T(\mathbf{w}_n - \mathbf{w}_{n-1})}}{N} - \sqrt{(\mathbf{w}_n - \mathbf{w}_{n-1})^T(\mathbf{w}_n - \mathbf{w}_{n-1})}\right)^2$$

  – Encourages low curvature

$$\text{curve}[\mathbf{w}, n] = -(\mathbf{w}_{n-1} - 2\mathbf{w}_n + \mathbf{w}_{n+1})^T(\mathbf{w}_{n-1} - 2\mathbf{w}_n + \mathbf{w}_{n+1})$$

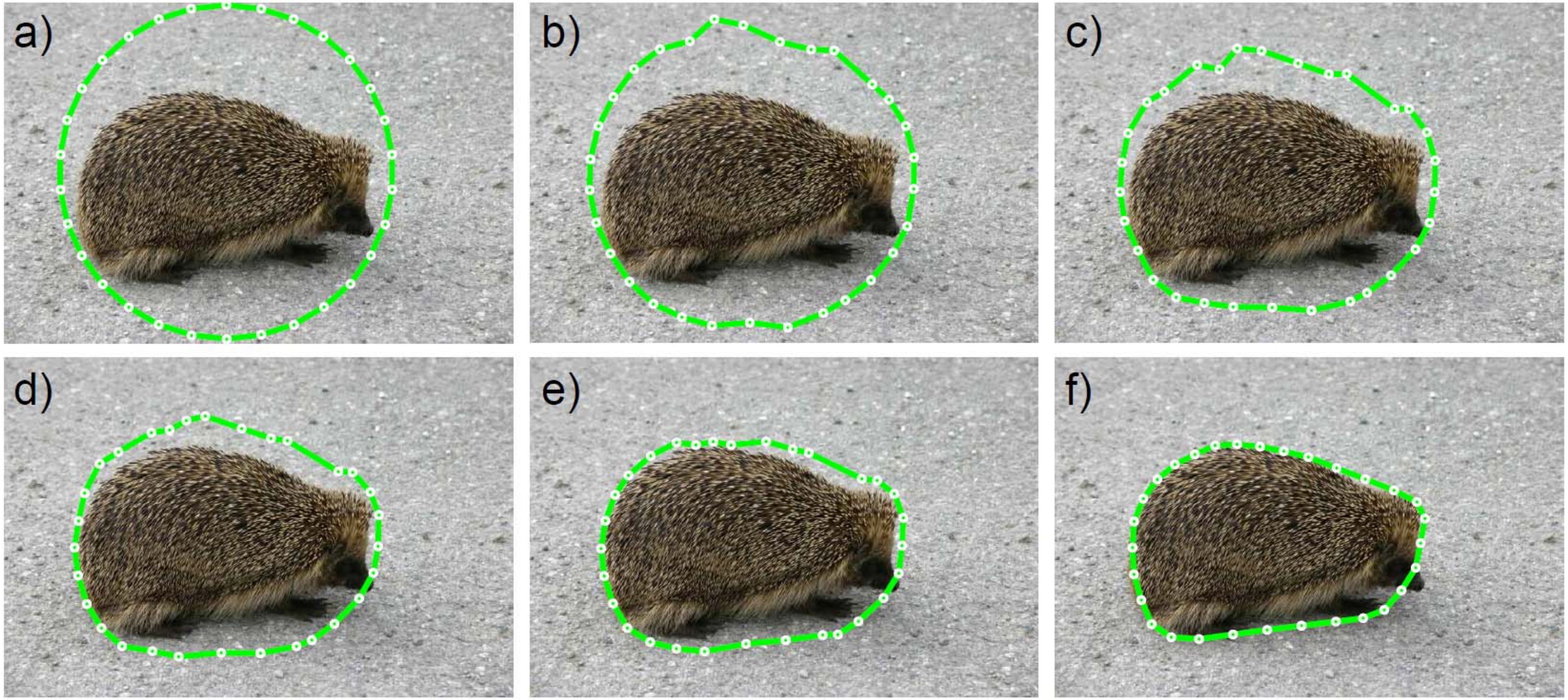
Inference

- Maximise posterior probability

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\left[Pr(\mathbf{W}|\mathbf{x})\right] \quad = \quad \underset{\mathbf{W}}{\operatorname{argmax}}\left[Pr(\mathbf{x}|\mathbf{W})Pr(\mathbf{W})\right]$$
$$= \quad \underset{\mathbf{W}}{\operatorname{argmax}}\left[\log[Pr(\mathbf{x}|\mathbf{W})] + \log[Pr(\mathbf{W})]\right]$$

- No closed form solution
- Must use non-linear optimisation method
- Number of unknowns = $2N$

Snakes



Notice failure at nose – falls between points.
A better model would sample image between landmark points

Inference

- Maximise posterior probability

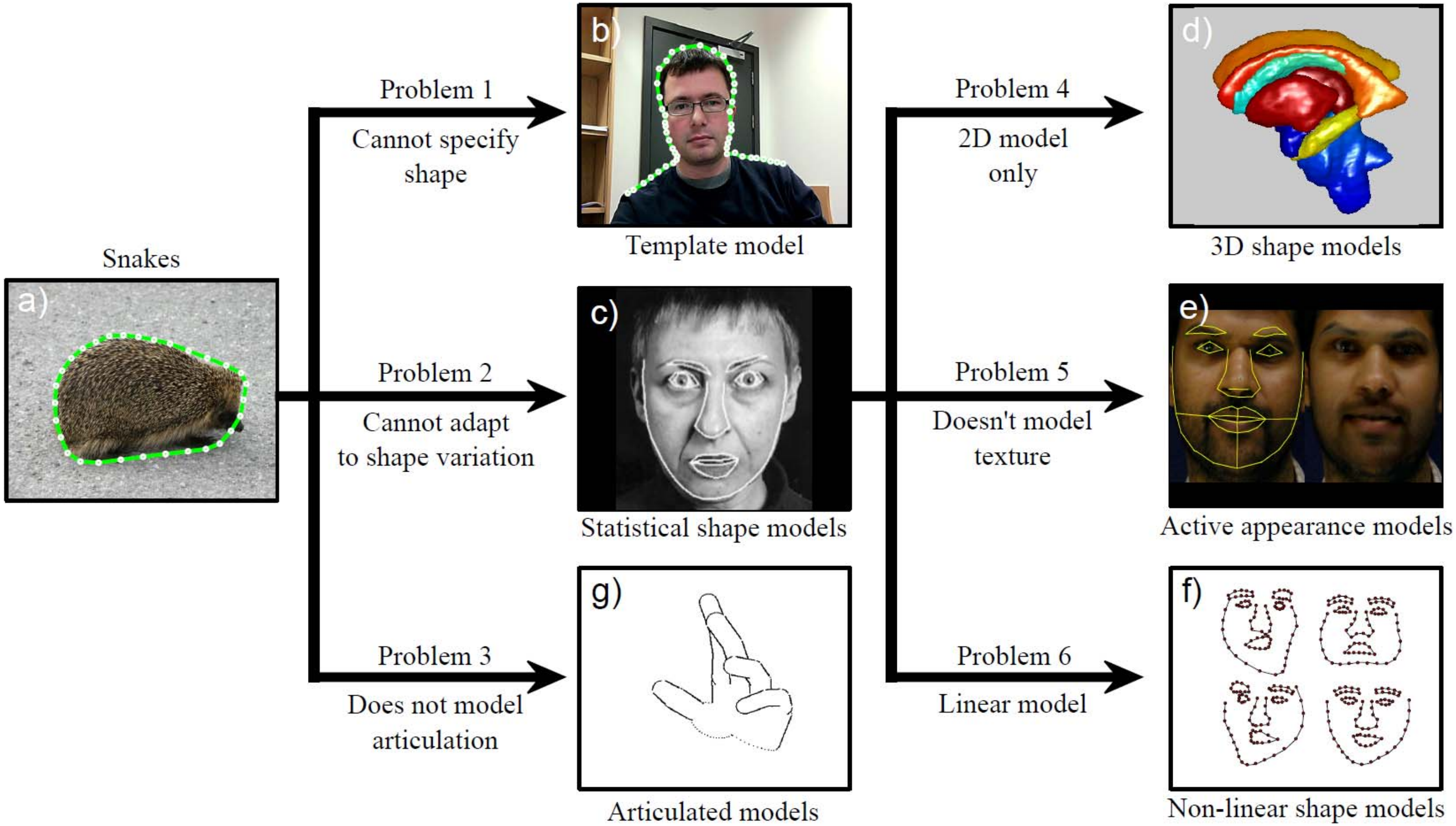
$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\left[Pr(\mathbf{W}|\mathbf{x})\right] = \underset{\mathbf{W}}{\operatorname{argmax}}\left[Pr(\mathbf{x}|\mathbf{W})Pr(\mathbf{W})\right]$$
$$= \underset{\mathbf{W}}{\operatorname{argmax}}\left[\log[Pr(\mathbf{x}|\mathbf{W})] + \log[Pr(\mathbf{W})]\right]$$

- Very slow. Can potentially speed it up by changing spacing element of prior:

$$\text{space}[\mathbf{w}, n] = -\left(\mu_s - \sqrt{(\mathbf{w}_n - \mathbf{w}_{n-1})^T(\mathbf{w}_n - \mathbf{w}_{n-1})}\right)^2$$

- Take advantage of limited connectivity of associated graphical model

Relationships between models



Structure

- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

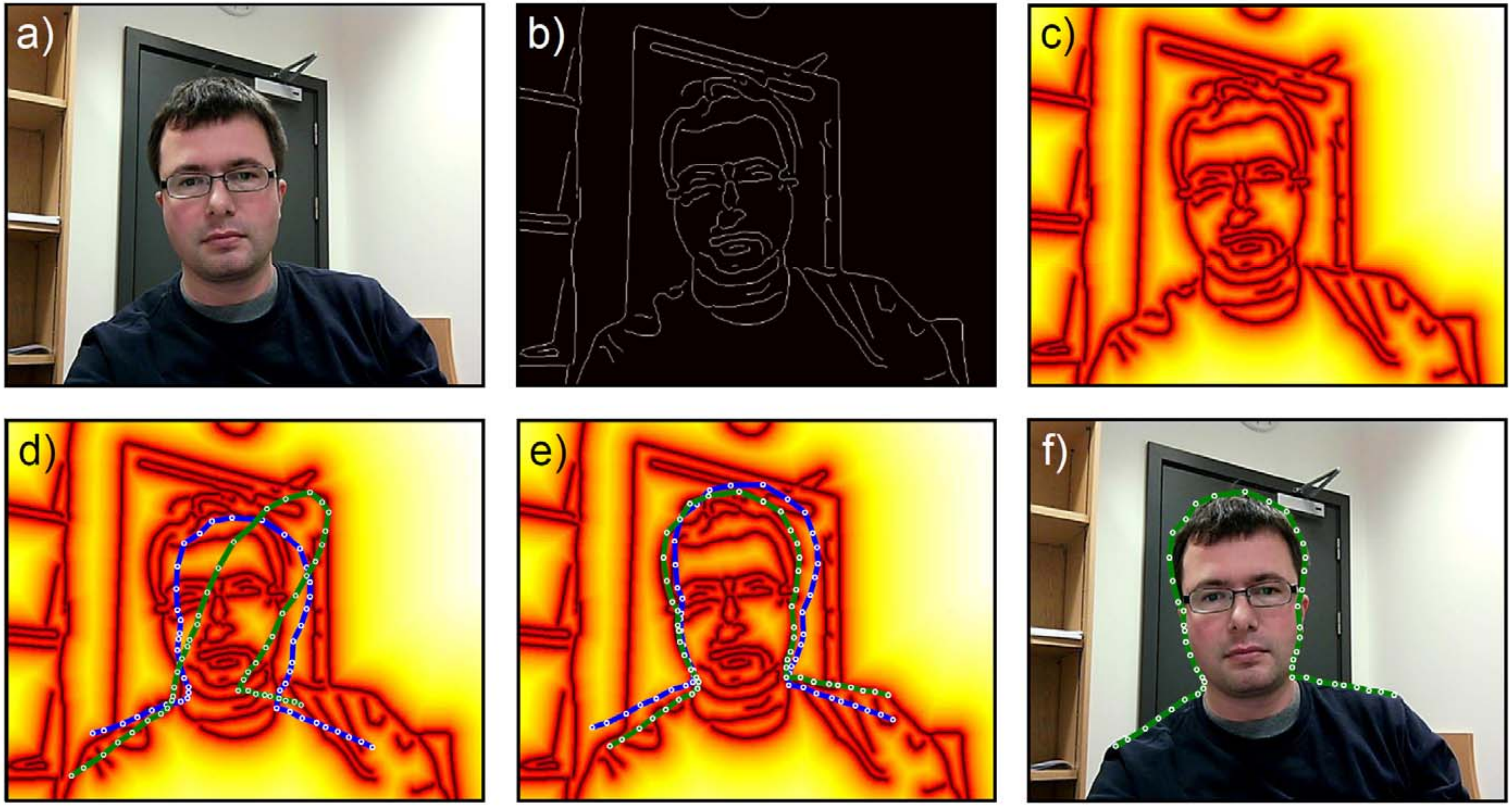
Shape template model

- Shape based on landmark points $\mathbf{W} = \{\mathbf{w}_n\}_{n=1}^{N}$
- These points are assumed known
- Mapped into the image by transformation $\textbf{trans}[\mathbf{w}, \boldsymbol{\Psi}]$
- What is left is to find parameters of transformation $\boldsymbol{\Psi}$
- Likelihood is based on distance transform:

$$Pr(\mathbf{x}|\mathbf{W}, \boldsymbol{\Psi}) \propto \prod_{n=1}^{N} \exp\left[-\left(\text{dist}\left[\mathbf{x}, \textbf{trans}[\mathbf{w}_n, \boldsymbol{\Psi}]\right)^2\right]\right.$$

- No prior on parameters (but could have one)

Shape template model



Inference

- Use maximum likelihood approach

$$\hat{\boldsymbol\Psi} = \underset{\boldsymbol\Psi}{\operatorname{argmax}}[L] = \underset{\boldsymbol\Psi}{\operatorname{argmax}}\left[\log\left[Pr(\mathbf{x}|\mathbf{W},\boldsymbol\Psi)\right]\right]$$

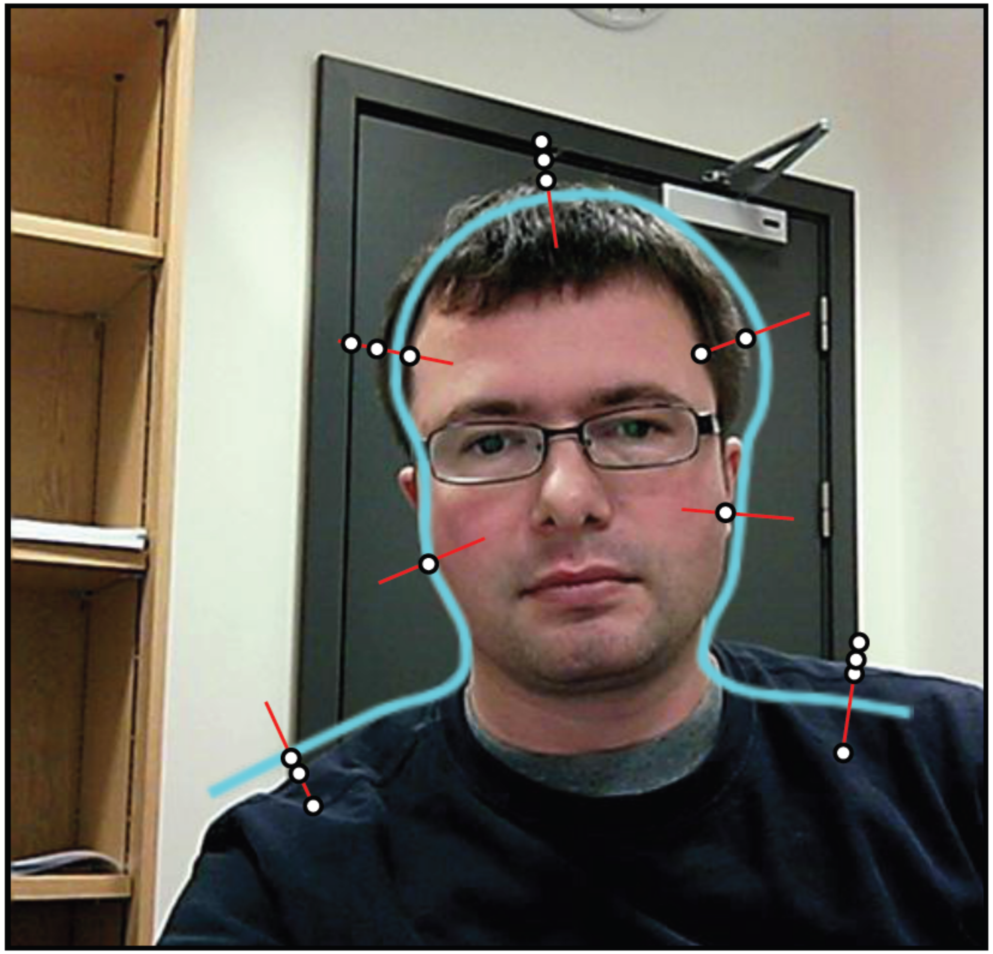
$$= \underset{\boldsymbol\Psi}{\operatorname{argmax}}\left[\sum_{n=1}^{N} -\left(\text{dist}\left[\mathbf{x},\mathbf{trans}[\mathbf{w}_n,\boldsymbol\Psi]\right]\right)^2\right]$$

- No closed form solution
- Must use non-linear optimization
- Use chain rule to compute derivatives

$$\frac{\partial L}{\partial \boldsymbol\Psi} = -\sum_{n=1}^{N}\sum_{j=1}^{2}\frac{\partial(\text{dist}[\mathbf{x},\mathbf{w}'_n])^2}{\partial w'_{jn}}\frac{\partial w'_{jn}}{\partial \boldsymbol\Psi}$$

where $\mathbf{w}'_n = \mathbf{trans}[\mathbf{w}_n,\boldsymbol\Psi]$

Iterative closest points



- Find nearest edge point to each landmark point
- Compute transformation in closed form
- Repeat

Structure

- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

Statistical shape models

- Also called
  - Point distribution models
  - Active shape models (as they adapt to the image)
- Likelihood:

$$Pr(\mathbf{x}_i|\mathbf{w}_i) \propto \prod_{n=1}^{N} \exp\left[-(\text{dist}\left[\mathbf{x}_i, \mathbf{trans}[\mathbf{w}_{in}, \mathbf{\Psi}_i]\right])^2\right]$$

- Prior:

$$Pr(\mathbf{w}_i) = \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}, \mathbf{\Sigma}]$$

Learning

- Usually, we are given the examples after they have been transformed

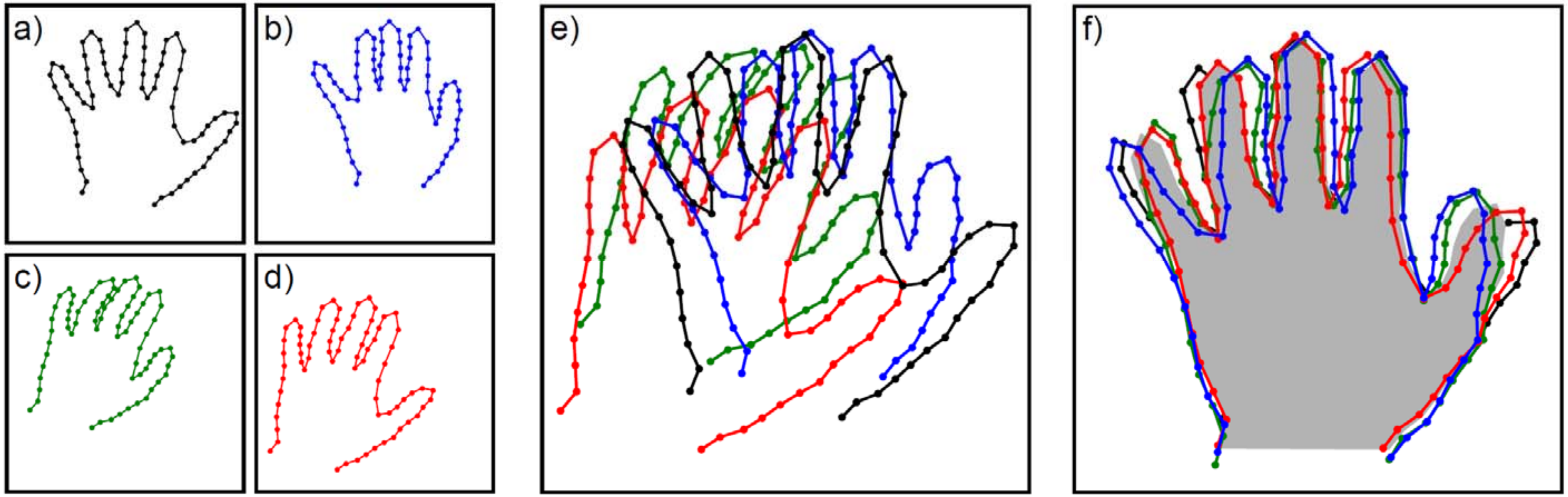
$$\mathbf{w}'_{in} = \mathbf{trans}[\mathbf{w}_{in}, \boldsymbol{\Psi}_i]$$

- Before we can learn the normal distribution, we must compute the inverse transformation

$$\mathbf{w}_{in} = \mathbf{trans}[\mathbf{w}'_{in}, \boldsymbol{\Psi}_i^{-}]$$

- Procedure is called generalized Procrustes analysis

Generalized Procrustes analysis



Training data Before alignment After alignment

Generalized Procrustes analysis

Alternately

- Update all transformations to map landmark points to current mean

$$\hat{\boldsymbol{\Psi}}_i^- = \underset{\boldsymbol{\Psi}_i^-}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left| \mathbf{trans}[\mathbf{w}'_{in}, \boldsymbol{\Psi}_i^-] - \boldsymbol{\mu}_n \right|^2 \right]$$

- Update mean to be average of transformed values,

$$\hat{\boldsymbol{\mu}} = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left| \mathbf{trans}[\mathbf{w}'_{in}, \boldsymbol{\Psi}_i^-] - \boldsymbol{\mu}_n \right|^2 \right]$$

Then learn mean and variance parameters.

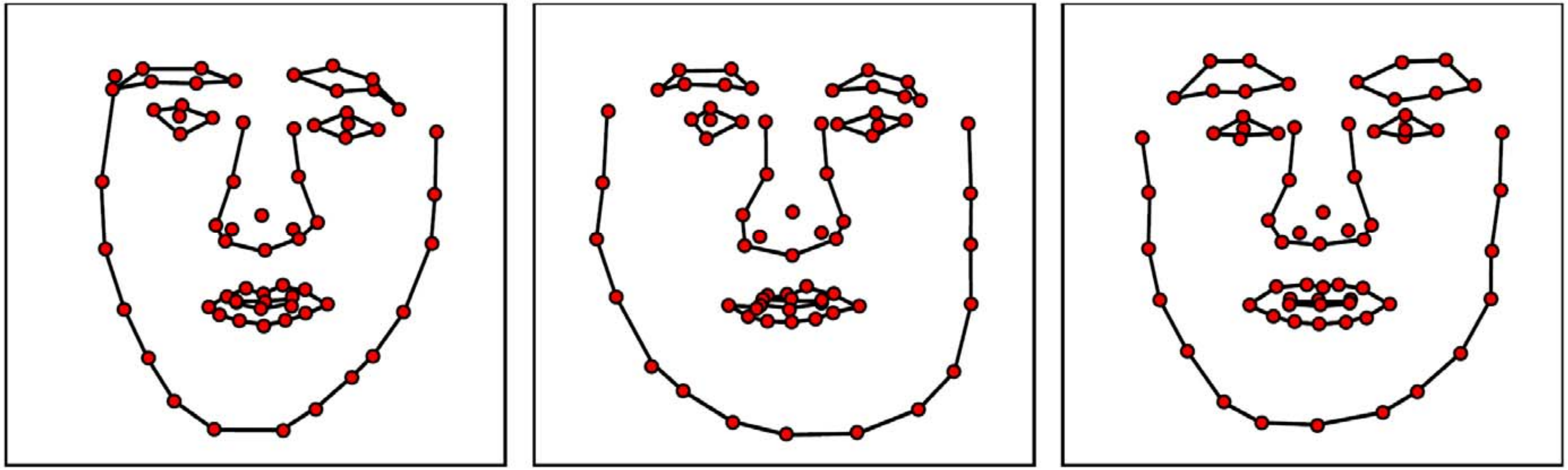
Inference

- Map inference:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \left[\max_{\boldsymbol{\Psi}} \left[\sum_{n=1}^{N} -(\operatorname{dist}\left[\mathbf{x}_i, \mathbf{trans}[\mathbf{w}_n, \boldsymbol{\Psi}]\right])^2 + \log[\operatorname{Norm}_{\mathbf{w}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}]] \right] \right]$$

- No closed form solution
- Use non-linear optimisation
- Or use ICP approach
- However, many parameters, and not clear they are all needed
- more efficient to use subspace model

Face model



Three samples from learned model for faces

Subspace shape model

- Generate data from model:

$$\mathbf{w}_i \quad = \quad \boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i + \epsilon_i$$

  – $\boldsymbol{\mu}$ is the mean shape
  – the matrix $\boldsymbol{\Phi} = [\phi_1, \phi_2, \ldots, \phi_K]$ contains K basis functions in it columns
  – ϵ_i is normal noise with covariance $\sigma^2\mathbf{I}$
- Can alternatively write

$$\mathbf{w}_i \quad = \quad \boldsymbol{\mu} + \sum_{k=1}^{K} \phi_k h_{ik} + \epsilon_i$$

Approximating with subspace

Subspace model

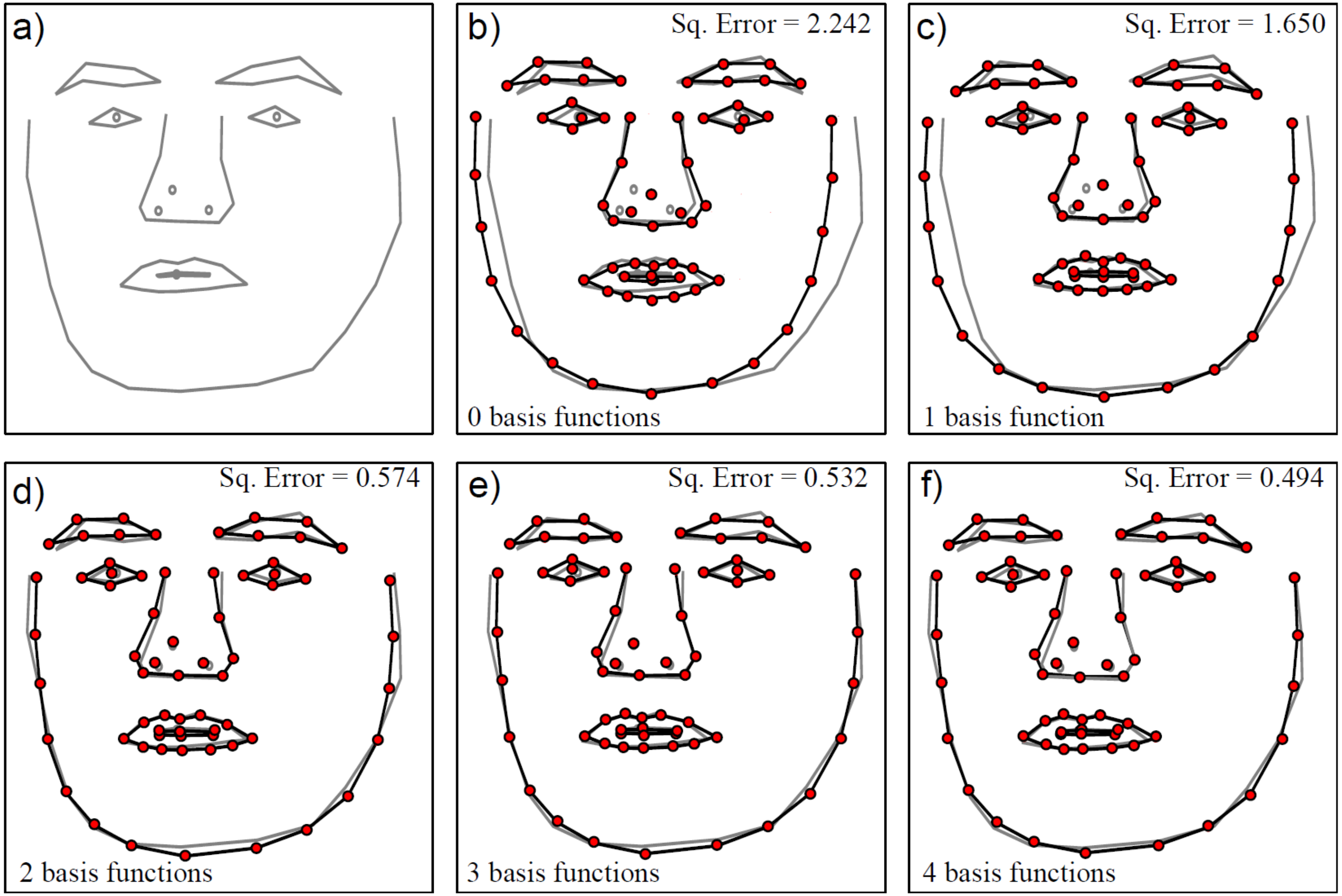
$$\mathbf{w}_i = \boldsymbol{\mu} + \sum_{k=1}^{K} \boldsymbol{\phi}_k h_{ik} + \boldsymbol{\epsilon}_i$$

Can approximate an vector w with a weighted sum of the basis functions

$$\mathbf{w}_i \approx \boldsymbol{\mu} + \sum_{k=1}^{K} \phi_k h_{ik}$$

Surprising how well this works even with a small number of basis functions

Subspace shape model



Probabilistic PCA

Generative eq: $$\mathbf{w}_i = \boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i + \epsilon_i$$

Probabilistic version:

$$Pr(\mathbf{w}_i|\mathbf{h}_i, \boldsymbol{\mu}, \boldsymbol{\Phi}, \sigma^2) = \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i, \sigma^2\mathbf{I}]$$

Add prior: $$Pr(\mathbf{h}_i) = \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}]$$

Density:
$$\begin{aligned} Pr(\mathbf{w}_i) &= \int Pr(\mathbf{w}_i|\mathbf{h}_i) Pr(\mathbf{h}_i) d\mathbf{h}_i \\ &= \int \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i, \sigma^2\mathbf{I}] \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] d\mathbf{h}_i \\ &= \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}, \boldsymbol{\Phi}\boldsymbol{\Phi}^T + \sigma^2\mathbf{I}]. \end{aligned}$$

17.5.1 Probabilistic principal component analysis

The particular subspace model that we will apply here is known as *probabilistic principal component analysis* or PPCA for short. To define the model, we re-express equation 17.20 in probabilistic terms:

$$Pr(\mathbf{w}_i|\mathbf{h}_i, \boldsymbol{\mu}, \boldsymbol{\Phi}, \sigma^2) = \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i, \sigma^2\mathbf{I}], \tag{17.23}$$

where $\boldsymbol{\mu}$ is a $2N \times 1$ mean vector, $\boldsymbol{\Phi}$ is a $2N \times K$ matrix containing K basis functions in its columns, and σ^2 controls the degree of additive noise. In the context of this model, the basis functions are known as *principal components*. The $K \times 1$ hidden variable $\mathbf{h}_i$ weights the basis functions and determines the final positions on the subspace, before the additive noise component is added.

To complete the model, we also define a prior over the hidden variable $\mathbf{h}_i$, and we choose a spherical normal distribution for this:

$$Pr(\mathbf{h}_i) = \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}]. \tag{17.24}$$

By marginalizing the joint distribution $Pr(\mathbf{w}_i, \mathbf{h}_i)$ with respect to the hidden variable $\mathbf{h}_i$, we can retrieve the prior density $Pr(\mathbf{w}_i)$, and this is given by

$$\begin{aligned} Pr(\mathbf{w}_i) &= \int Pr(\mathbf{w}_i|\mathbf{h}_i)Pr(\mathbf{h}_i)d\mathbf{h}_i \\ &= \int \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i, \sigma^2\mathbf{I}]\text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}]d\mathbf{h}_i \\ &= \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}, \boldsymbol{\Phi}\boldsymbol{\Phi}^T + \sigma^2\mathbf{I}]. \end{aligned} \tag{17.25}$$

This algebraic result is not obvious; however, it has a simple interpretation. The prior over the landmark points $\mathbf{w}_i$ is once more normally distributed, but now the covariance is divided into two parts: the term $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$, which explains the variation in the subspace (due to shape changes), and the term $\sigma^2\mathbf{I}$, which explains any remaining variation in the data (mainly noise in the training points).

Learning Probabilistic PCA

Learn parameters $\boldsymbol{\mu}$, $\boldsymbol{\Phi}$ and σ^2 from data $\{\mathbf{w}_i\}_{i=1}^{I}$, where $\mathbf{w}_i = [\mathbf{w}_{i1}^T, \mathbf{w}_{i2}^T \ldots, \mathbf{w}_{iN}^T]^T$.

Learn mean:
$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{I} \mathbf{w}_i}{I}$$

Then set $\mathbf{W} = [\mathbf{w}_1 - \boldsymbol{\mu}, \mathbf{w}_2 - \boldsymbol{\mu}, \ldots, \mathbf{w}_I - \boldsymbol{\mu}]$ and compute eigen-decomposition

$$\mathbf{W}\mathbf{W}^T = \mathbf{U}\mathbf{L}^2\mathbf{U}^T$$

Choose parameters

$$\hat{\sigma}^2 = \frac{1}{D-K} \sum_{j=K+1}^{D} L_{jj}^2$$

$$\hat{\boldsymbol{\Phi}} = \mathbf{U}_K(\mathbf{L}_K^2 - \hat{\sigma}^2\mathbf{I})^{1/2}$$

Properties of basis functions

Learning of parameters based on eigen-decomposition:

$$\mathbf{W}\mathbf{W}^T = \mathbf{U}\mathbf{L}^2\mathbf{U}^T$$

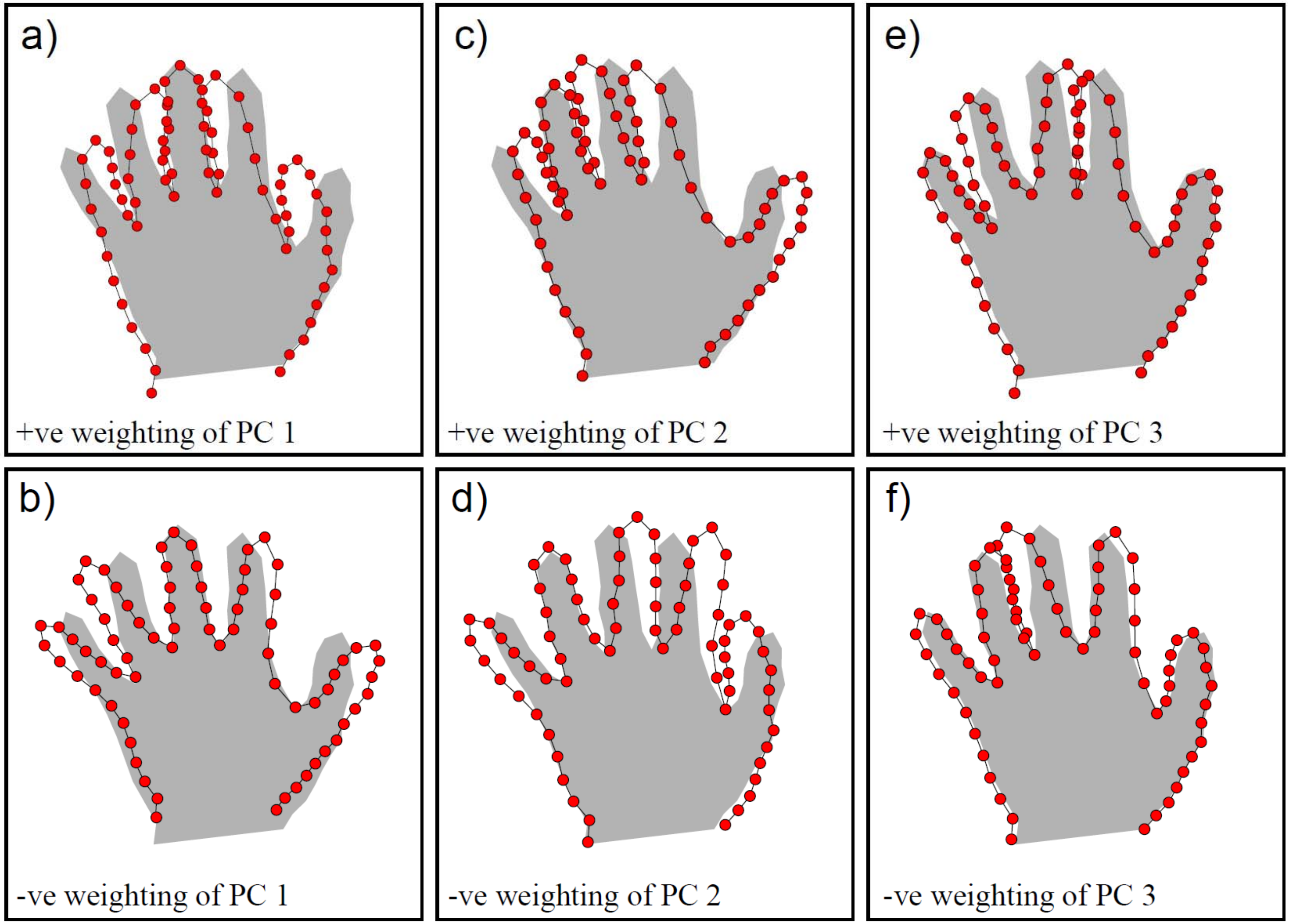
Parameters

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{D-K}\sum_{j=K+1}^{D} L_{jj}^2 \\ \hat{\mathbf{\Phi}} &= \mathbf{U}_K(\mathbf{L}_K^2 - \hat{\sigma}^2\mathbf{I})^{1/2} \end{aligned}$$

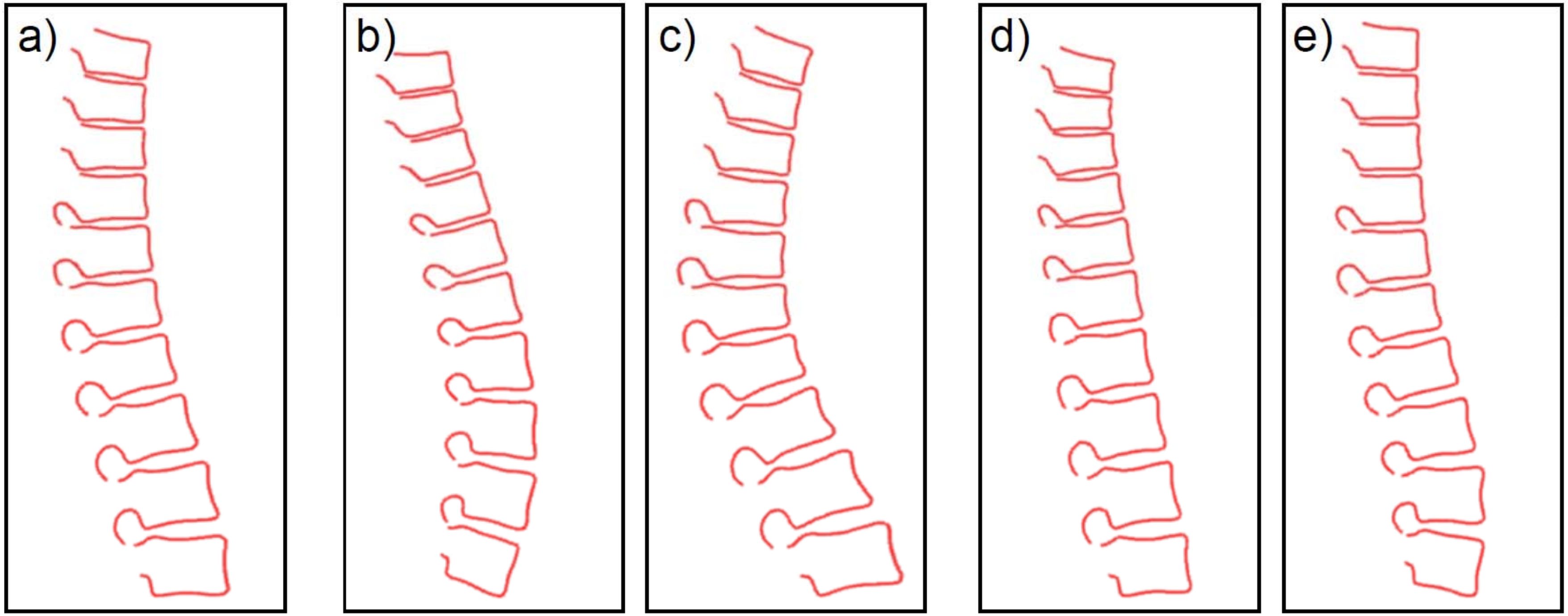
Notice that:

- Basis functions in $\mathbf{\Phi}$ are orthogonal
- Basis functions in $\mathbf{\Phi}$ are ordered

Learned hand model



Learned spine model



Mean

Manipulating first principal component

Manipulating second principal component

Inference

To fit model to an image:

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmax}} \left[\max_{\boldsymbol{\Psi}} \left[\sum_{n=1}^{N} \left(-\frac{(\text{dist}\left[\mathbf{x}_i, \mathbf{trans}[\boldsymbol{\mu}_n + \boldsymbol{\Phi}_n \mathbf{h}, \boldsymbol{\Psi}]\right])^2}{\sigma^2} \right) + \log[\text{Norm}_{\mathbf{h}}[\mathbf{0}, \mathbf{I}]] \right] \right]$$

likelihood

prior

ICP Approach:

- Find closest points to current prediction
- Update weightings $\mathbf{h}$
- Find closest points to current prediction
- Update transformation parameters ψ

Inference

1. Update weightings $\mathbf{h}$

$$\begin{aligned}\hat{\mathbf{h}} &= \underset{\mathbf{h}}{\operatorname{argmax}} \left[\sum_{n=1}^{N} \log[Pr(\mathbf{y}_n|\mathbf{h}), \mathbf{\Psi}] + \log[Pr(\mathbf{h})] \right] \\ &= \underset{\mathbf{h}}{\operatorname{argmax}} \left[\sum_{n=1}^{N} - (\mathbf{y}_n - \text{trans}[\boldsymbol{\mu}_n + \mathbf{\Phi}_n \mathbf{h}, \mathbf{\Psi}])^2 / \sigma^2 - \log[\mathbf{h}^T \mathbf{h}] \right]\end{aligned}$$

If transformation parameters can be represented as a matrix $\mathbf{A}$

$$\hat{\mathbf{h}} = \left(\sigma^2 \mathbf{I} + \sum_{n=1}^{N} \mathbf{\Phi}_n^T \mathbf{A}^T \mathbf{A} \mathbf{\Phi}_n \right)^{-1} \sum_{n=1}^{N} \mathbf{A} \mathbf{\Phi}_n (\mathbf{y}_n - \mathbf{A}\boldsymbol{\mu} - \mathbf{b})$$

2. Update transformation parameters ψ

- Using one of the closed form solutions

Fitting model

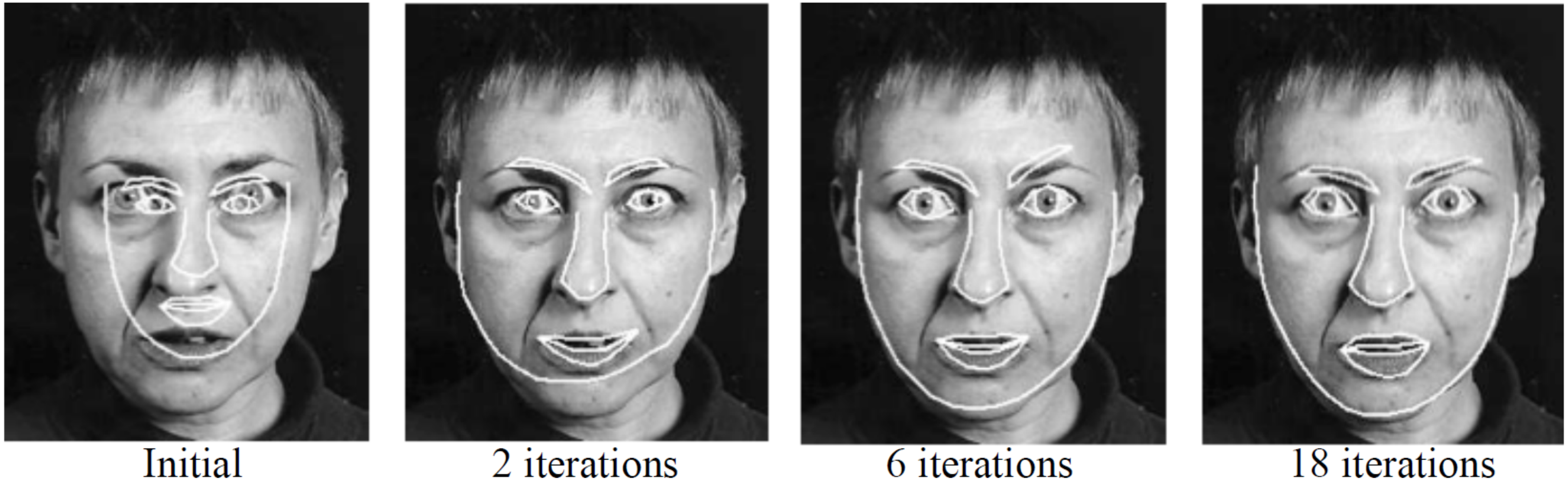


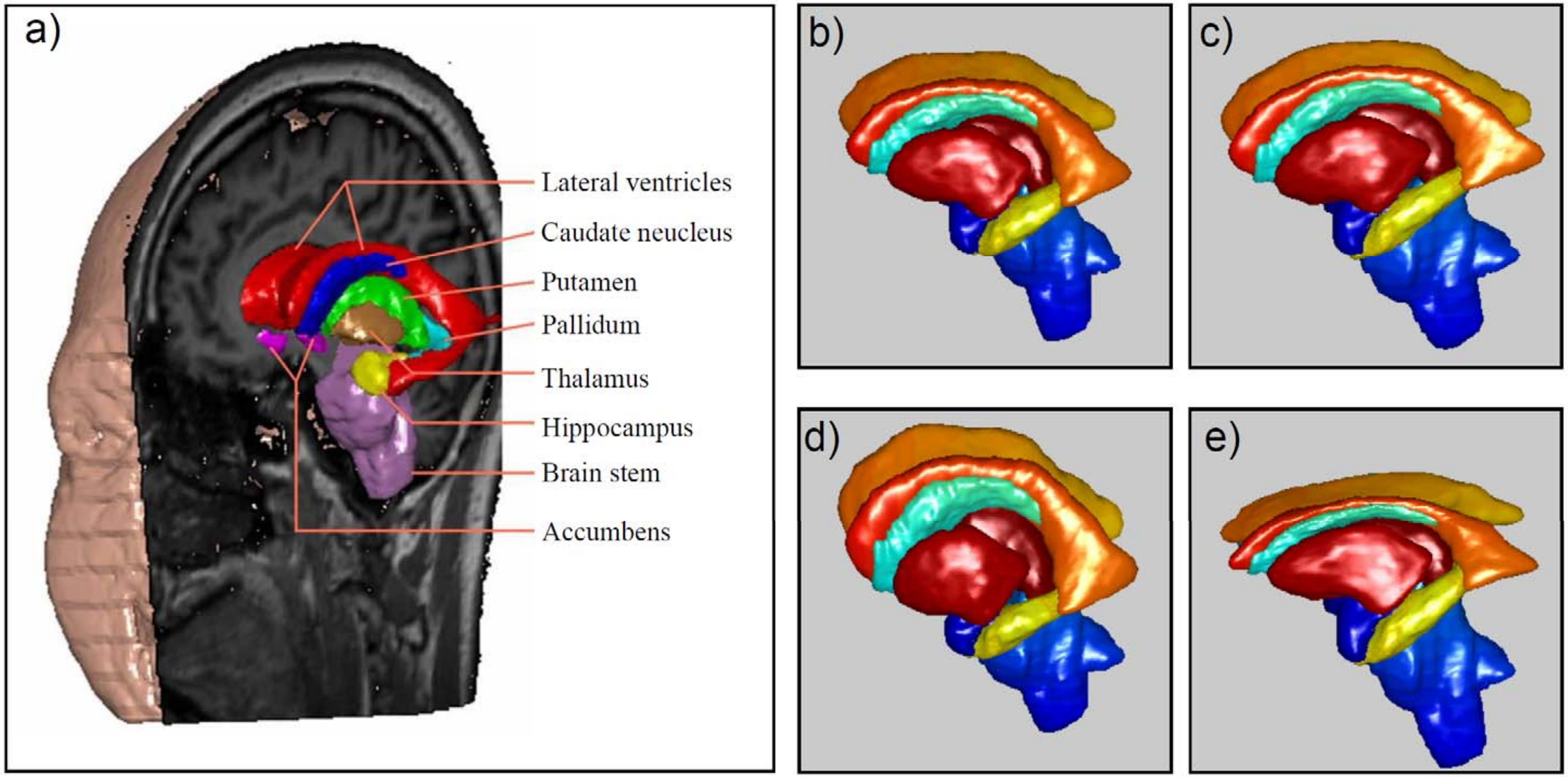
Figure provided by Tim Cootes

Much better to use statistical classifier instead of just distance from edges

Structure

- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

3D shape models



Babalola *et al.* (2008)

Structure

- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

Statistical models for shape and appearance

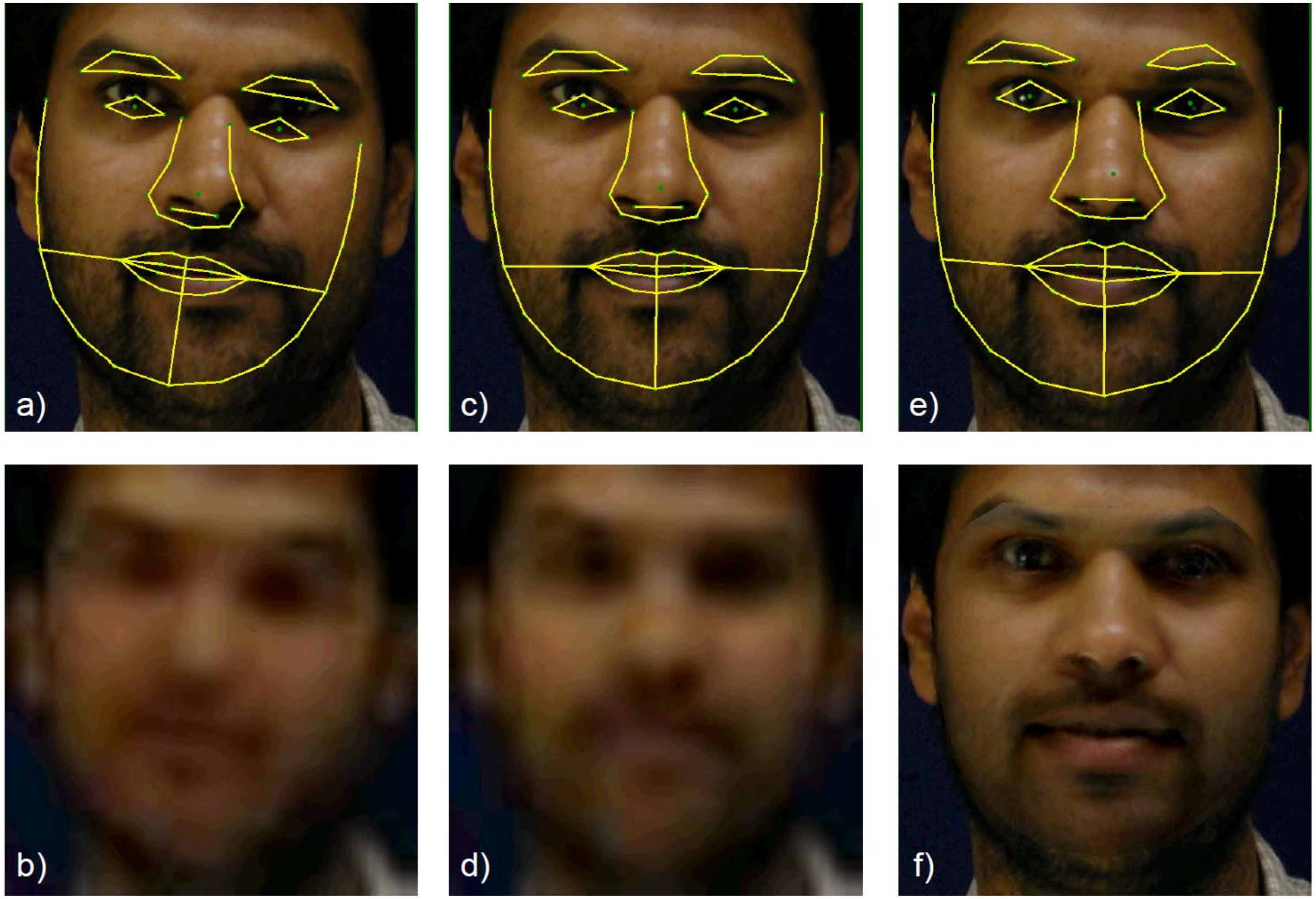


Figure provided by Tim Cootes

Statistical models for shape and appearance

$$
\begin{aligned}
Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\
Pr(\mathbf{w}_i|\mathbf{h}_i) &= \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}_w + \boldsymbol{\Phi}_w \mathbf{h}_i, \sigma_w^2 \mathbf{I}] \\
Pr(\mathbf{x}_i|\mathbf{w}_i, \mathbf{h}_i) &= \text{Norm}_{\mathbf{x}_i}[\mathbf{warp}[\boldsymbol{\mu}_x + \boldsymbol{\Phi}_x \mathbf{h}_i, \mathbf{w}_i, \boldsymbol{\Psi}_i], \sigma_x^2 \mathbf{I}]
\end{aligned}
$$

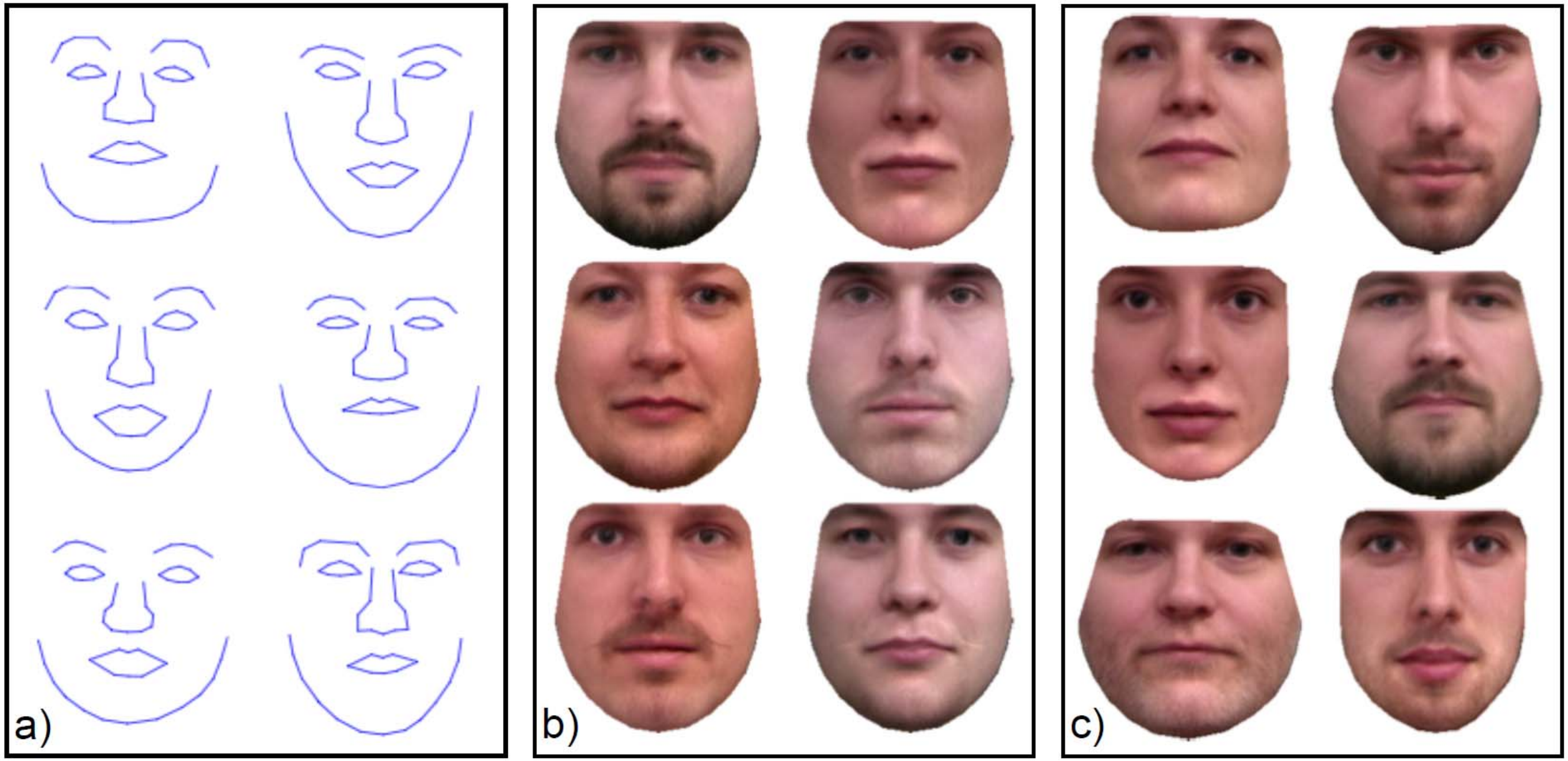
1. We draw a hidden variable $\mathbf{h}$ from a prior

Statistical models for shape and appearance

$$
\begin{aligned}
Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\
Pr(\mathbf{w}_i|\mathbf{h}_i) &= \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}_w + \boldsymbol{\Phi}_w \mathbf{h}_i, \sigma_w^2 \mathbf{I}] \\
Pr(\mathbf{x}_i|\mathbf{w}_i, \mathbf{h}_i) &= \text{Norm}_{\mathbf{x}_i}[\mathbf{warp}[\boldsymbol{\mu}_x + \boldsymbol{\Phi}_x \mathbf{h}_i, \mathbf{w}_i, \boldsymbol{\Psi}_i], \sigma_x^2 \mathbf{I}]
\end{aligned}
$$

1. We draw a hidden variable $\mathbf{h}$ from a prior
2. We draw landmark points $\mathbf{w}$ from a subspace model
3. We draw image intensities $\mathbf{x}$.
    - Generate image intensities in standard template shape
    - Transform the landmark points (parameters ψ)
    - Transform the image to landmark points
    - Add noise

Shape and appearance model



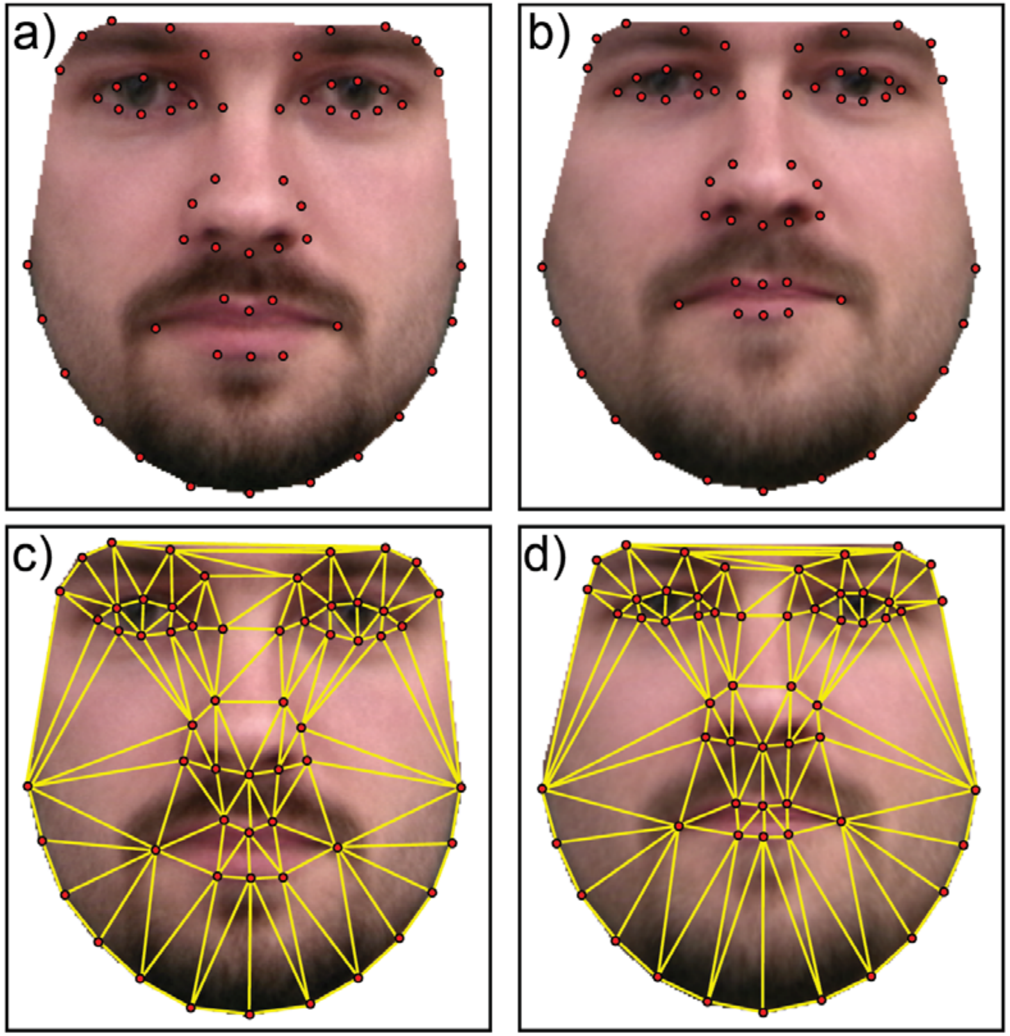
Shape model

Intensity model

Shape and intensity

Adapted from Stegmann (2002)

Warping images



Adapted from Stegmann (2002)

Piecewise affine transformation

Triangulate image points using Delaunay triangulation.

Image in each triangle is warped by an affine transformation.

Learning

$$
\begin{aligned}
Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0},\mathbf{I}] \\
Pr(\mathbf{w}_i|\mathbf{h}_i) &= \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}_w + \boldsymbol{\Phi}_w\mathbf{h}_i, \sigma_w^2\mathbf{I}] \\
Pr(\mathbf{x}_i|\mathbf{w}_i,\mathbf{h}_i) &= \text{Norm}_{\mathbf{x}_i}[\mathbf{warp}[\boldsymbol{\mu}_x + \boldsymbol{\Phi}_x\mathbf{h}_i, \mathbf{w}_i, \boldsymbol{\Psi}_i], \sigma_x^2\mathbf{I}]
\end{aligned}
$$

Goal is to learn parameters :

$$\{\boldsymbol{\mu}_w, \boldsymbol{\Phi}_w, \sigma_w^2, \boldsymbol{\mu}_x, \boldsymbol{\Phi}_x, \sigma_x^2\}$$

Problem

- We are given the transformed landmark points
- We are given the warped and transformed images

Solution

- Use Procrustes analysis to un-transform landmark points
- Warp observed images to template shape

Learning

Now have aligned landmark points $\mathbf{w}$, and aligned images $\mathbf{x}$, we can learn the simpler model:

$$
\begin{aligned}
Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\
Pr(\mathbf{w}_i|\mathbf{h}_i) &= \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}_w + \boldsymbol{\Phi}_w\mathbf{h}_i, \sigma_w^2\mathbf{I}] \\
Pr(\mathbf{x}_i|\mathbf{h}_i) &= \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}_x + \boldsymbol{\Phi}_x\mathbf{h}_i, \sigma_x^2\mathbf{I}].
\end{aligned}
$$

Can write generative equation as:

$$
\begin{bmatrix} \mathbf{w}_i \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_w \\ \boldsymbol{\mu}_x \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_w \\ \boldsymbol{\Phi}_x \end{bmatrix} \mathbf{h}_i + \begin{bmatrix} \boldsymbol{\epsilon}_{wi} \\ \boldsymbol{\epsilon}_{xi} \end{bmatrix}
$$

Has the form of a factor analyzer $\mathbf{x}' = \boldsymbol{\mu}' + \boldsymbol{\Phi}'\mathbf{h} + \boldsymbol{\epsilon}'$

Inference

Likelihood of observed intensities

$$Pr(\mathbf{x}|\mathbf{h}) \quad = \quad \text{Norm}_{\mathbf{x}}[\mathbf{warp}[\boldsymbol{\mu}_x + \boldsymbol{\Phi}_x\mathbf{h}, \boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}, \boldsymbol{\Psi}], \sigma_x^2\mathbf{I}]$$

To fit the model use maximum likelihood

$$\begin{aligned}\hat{\mathbf{h}}, \hat{\boldsymbol{\Psi}} &= \underset{\mathbf{h}, \boldsymbol{\Psi}}{\text{argmax}}\left[\log\left[Pr(\mathbf{x}|\mathbf{h})\right]\right] \\ &= \underset{\mathbf{h}, \boldsymbol{\Psi}}{\text{argmin}}\left[(\mathbf{x} - \mathbf{warp}[\boldsymbol{\mu}_x + \boldsymbol{\Phi}_x\mathbf{h}, \boldsymbol{\mu}_w + \boldsymbol{\Phi}_w\mathbf{h}, \boldsymbol{\Psi}])^T \right. \\ &\qquad\qquad \left. (\mathbf{x} - \mathbf{warp}[\boldsymbol{\mu}_x + \boldsymbol{\Phi}_x\mathbf{h}, \boldsymbol{\mu}_w + \boldsymbol{\Phi}_w\mathbf{h}, \boldsymbol{\Psi}])\right]\end{aligned}$$

This has the least squares form

$$f[\boldsymbol{\theta}] = \mathbf{z}[\boldsymbol{\theta}]^T\mathbf{z}[\boldsymbol{\theta}]$$

Inference

This has the least squares form

$$f[\boldsymbol{\theta}] = \mathbf{z}[\boldsymbol{\theta}]^T\mathbf{z}[\boldsymbol{\theta}]$$

Use Gauss-Newton method or similar

$$\boldsymbol{\theta}^{[t]} = \boldsymbol{\theta}^{[t-1]} + \lambda(\mathbf{J}^T\mathbf{J})^{-1}\frac{\partial f}{\partial \boldsymbol{\theta}}$$

Where the Jacobian $\mathbf{J}$ is a matrix with elements

$$J_{mn} = \frac{\partial z_m}{\partial \theta_n}$$

Statistical models for shape and appearance

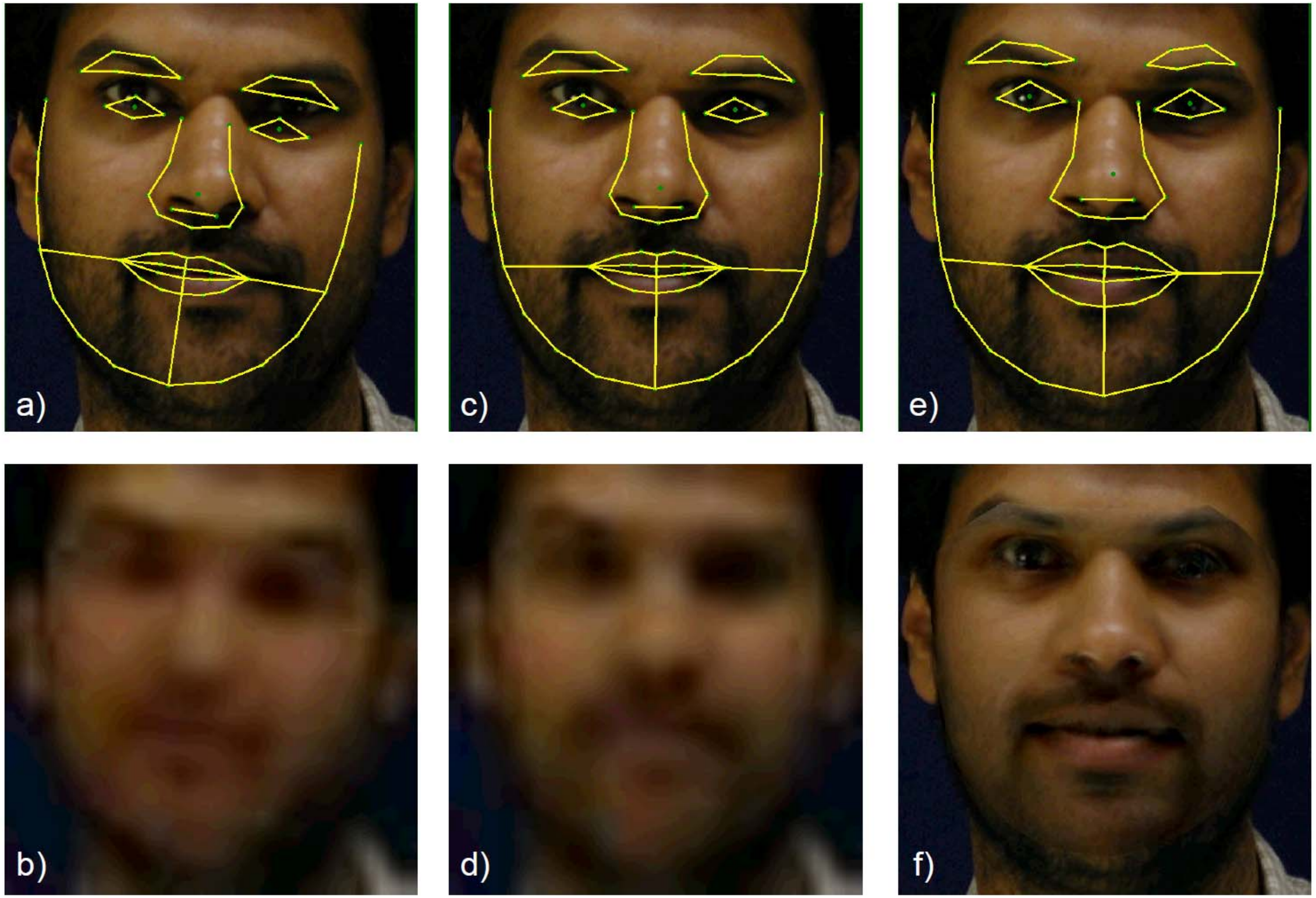


Figure provided by Tim Cootes

Structure

- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

Non-linear models

- The shape and appearance models that we have studied so far are based on the normal distribution
- But more complex shapes might need more complex distributions
  - Could use mixture of PPCAs or similar
  - Or use a non-linear subspace model
- We will investigate the Gaussian process latent variable model (GPLVM)
- To understand the GPLVM, first think about PPCA in terms of regression.

PPCA as regression

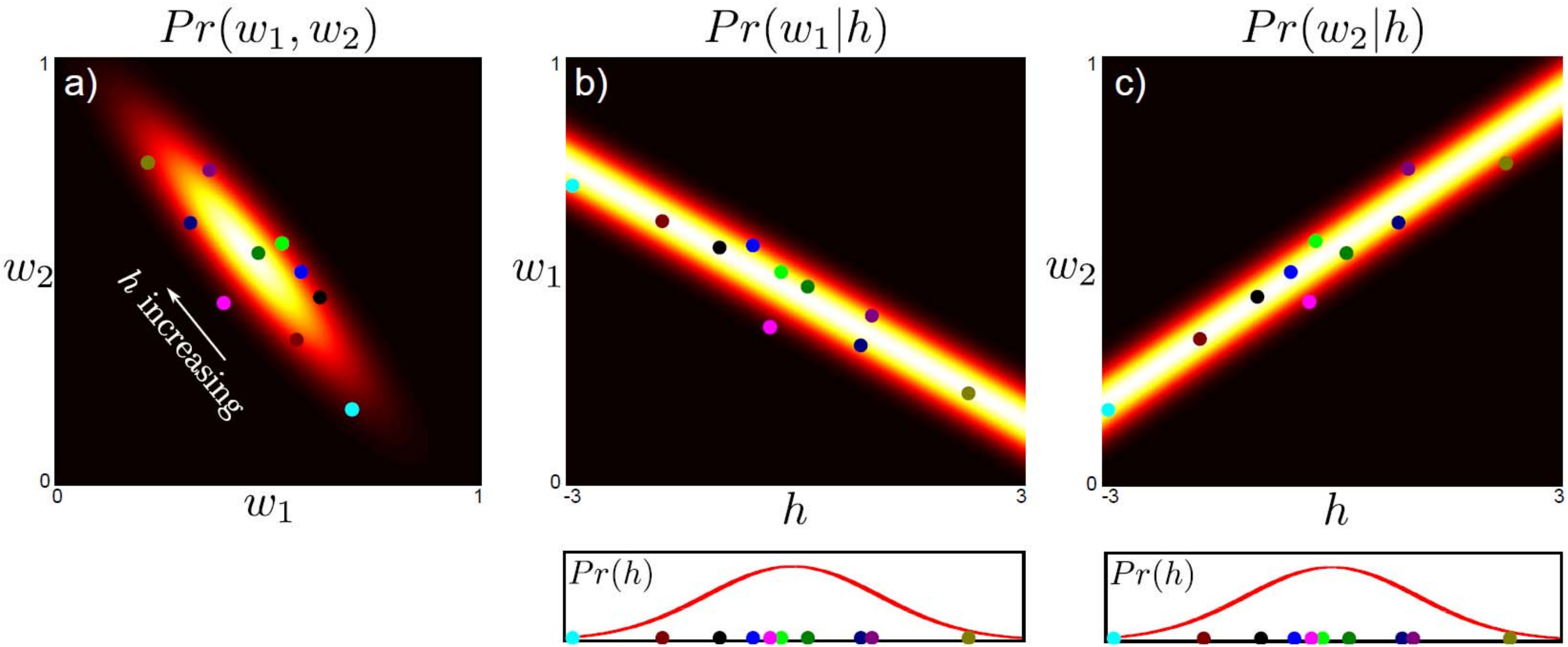
PPCA model:

$$
\begin{aligned}
Pr(\mathbf{w}|\boldsymbol{\mu},\boldsymbol{\Phi},\sigma^2) &= \int Pr(\mathbf{w},\mathbf{h}|\boldsymbol{\mu},\boldsymbol{\Phi},\sigma^2)d\mathbf{h}\\
&= \int Pr(\mathbf{w}|\mathbf{h},\boldsymbol{\mu},\boldsymbol{\Phi},\sigma^2)Pr(\mathbf{h})d\mathbf{h}\\
&= \int \text{Norm}_{\mathbf{w}}[\boldsymbol{\mu}+\boldsymbol{\Phi}\mathbf{h},\sigma^2\mathbf{I}]\text{Norm}_{\mathbf{h}}[\mathbf{0},\mathbf{I}]d\mathbf{h}.
\end{aligned}
$$

- First term in last equation looks like regression
- Predicts w for a given h
- Considering each dimension separately, get linear regression

$$Pr(w_d|\mathbf{h},\boldsymbol{\mu},\boldsymbol{\Phi},\sigma^2) = \text{Norm}_{\mathbf{w}_d}[\mu_d+\boldsymbol{\phi}_d^T\mathbf{h},\sigma^2]$$

PPCA as regression



Joint probability distribution

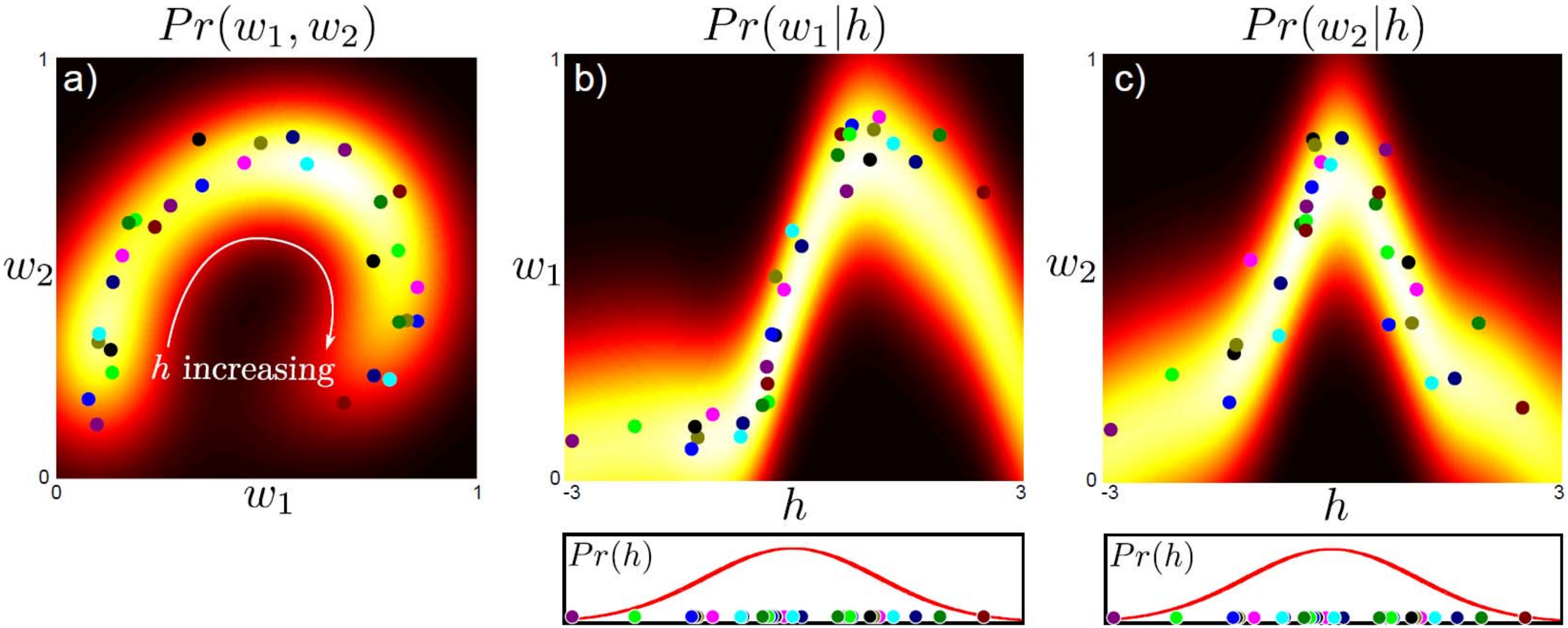
Regress 1st dimension against hidden variable

Regress 2nd dimension against hidden variable

Gaussian process latent variable model

- Idea: replace the linear regression model with a non-linear regression model
- As name suggests, use Gaussian process regression
- Implications
  - Can now marginalize over parameters μ and Φ
  - Can no longer marginalize over variable $\mathbf{h}$

GPLVM as regression



Joint probability distribution

Regress 1st dimension against hidden variable

Regress 2nd dimension against hidden variable

Learning

- In learning the Gaussian process regression model, we optimized the marginal likelihood of the data with respect to the parameter σ^2.

$$\begin{aligned}\hat{\sigma^2} &= \underset{\sigma^2}{\operatorname{argmax}}\left[Pr(\mathbf{w}|\mathbf{X},\sigma^2)\right]\\ &= \underset{\sigma^2}{\operatorname{argmax}}\left[\int Pr(\mathbf{w}|\mathbf{X},\boldsymbol{\Phi},\sigma^2)Pr(\boldsymbol{\Phi})d\boldsymbol{\Phi}\right]\\ &= \underset{\sigma^2}{\operatorname{argmax}}\left[\text{Norm}_{\mathbf{w}}[\mathbf{0},\sigma_p^2\mathbf{K}[\mathbf{X},\mathbf{X}]+\sigma^2\mathbf{I}]\right]\end{aligned}$$

Learning

- In learning the GPLVM, we still optimized the marginal likelihood of the data with respect to the parameter σ^2, but must also find the values of the hidden variables that we regress against .

$$\begin{aligned}
\hat{\mathbf{H}}, \hat{\sigma^2} &= \underset{\mathbf{H},\sigma^2}{\operatorname{argmax}} \left[Pr(\mathbf{W}, \mathbf{H}, \sigma^2) \right] \\
&= \underset{\mathbf{H},\sigma^2}{\operatorname{argmax}} \left[\int Pr(\mathbf{W}|\mathbf{X}, \mathbf{\Phi}, \sigma^2) Pr(\mathbf{\Phi}) Pr(\mathbf{H}) d\mathbf{\Phi} \right] \\
&= \underset{\mathbf{H},\sigma^2}{\operatorname{argmax}} \left[\prod_{d=1}^{D} \text{Norm}_{\mathbf{w}_{\bullet d}}[\mathbf{0}, \sigma_p^2 \mathbf{K}[\mathbf{H}, \mathbf{H}] + \sigma^2 \mathbf{I}] \prod_{i=1}^{I} \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \right]
\end{aligned}$$

- Use non-linear optimization technique

Inference

- To predict a new value of the data using a hidden variable

$$Pr(w_d^*|\mathbf{h}^*,\mathbf{H},\mathbf{W}) =$$

$$\text{Norm}_{w_d^*}\left[\frac{\sigma_p^2}{\sigma^2}\mathbf{K}[\mathbf{h}^*,\mathbf{H}]\mathbf{w}_{\bullet\mathbf{d}} - \frac{\sigma_p^2}{\sigma^2}\mathbf{K}[\mathbf{h}^*,\mathbf{H}]\left(\mathbf{K}[\mathbf{H},\mathbf{H}]+\frac{\sigma_p^2}{\sigma^2}\mathbf{I}\right)^{-1}\mathbf{K}[\mathbf{H},\mathbf{H}]\mathbf{w}_{\bullet d},\right.$$

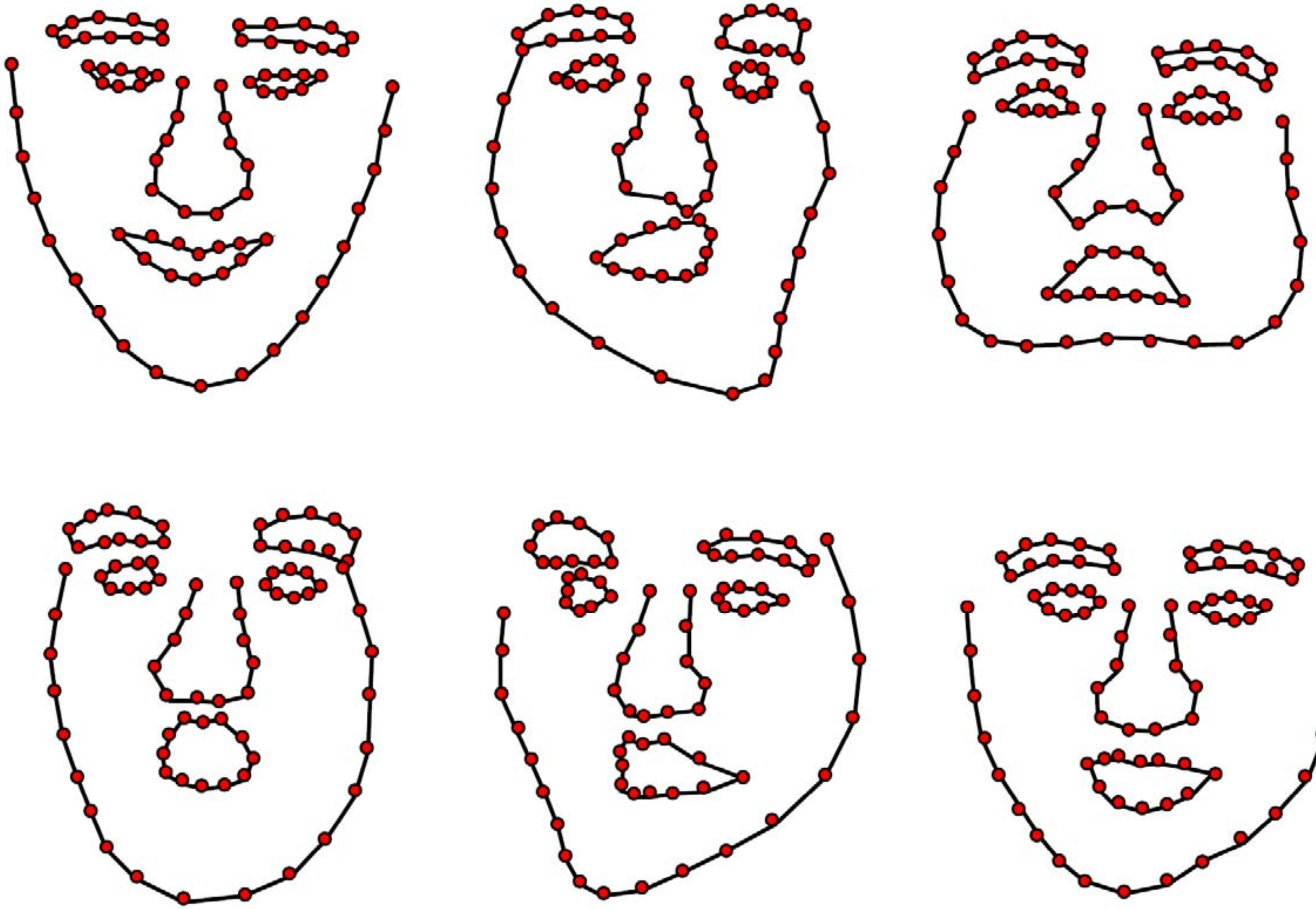
$$\left.\sigma_p^2\mathbf{K}[\mathbf{h}^*,\mathbf{h}^*] - \sigma_p^2\mathbf{K}[\mathbf{h}^*,\mathbf{H}]\left(\mathbf{K}[\mathbf{H},\mathbf{H}]+\frac{\sigma_p^2}{\sigma^2}\mathbf{I}\right)^{-1}\mathbf{K}[\mathbf{H},\mathbf{h}^*]+\sigma^2\right].$$

- To compute density

$$Pr(\mathbf{w}) = \prod_{d=1}^{D}\int Pr(w_d^*|\mathbf{h}^*,\mathbf{H},\mathbf{W})Pr(\mathbf{h}^*)d\mathbf{h}^*$$

- Cannot be computed in closed from

GPLVM Shape models



Huang *et al.* (2011). ©2011 IEEE.

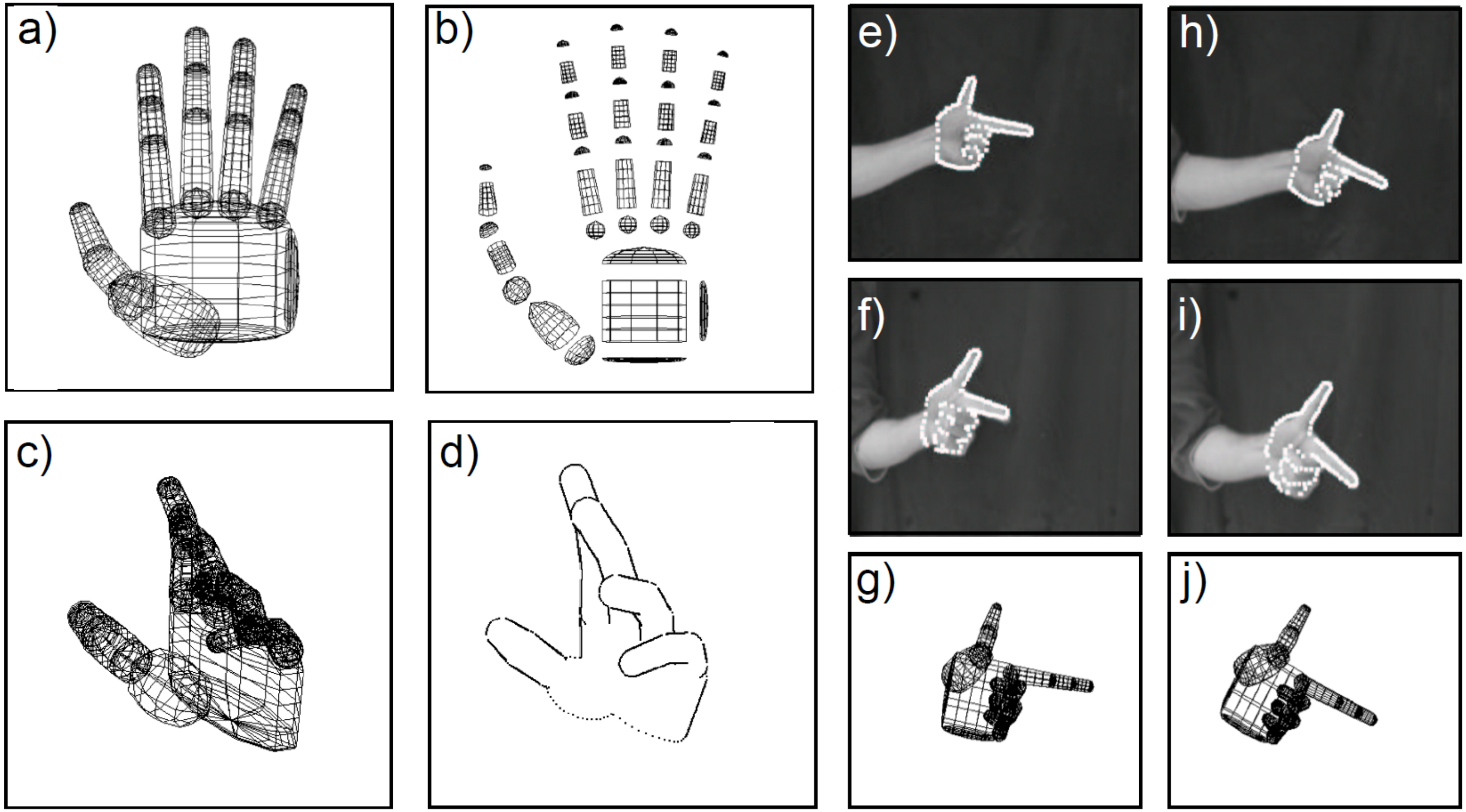
Structure

- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

Articulated Models

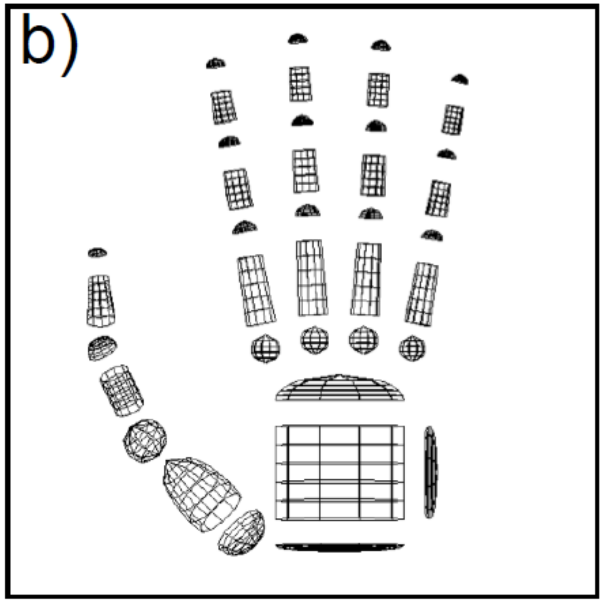
- Transformations of parts applied one after each other
- Known as a kinematic chain
- e.g. Foot transform is relative to lower leg, which is relative to upper leg etc.
- One root transformation that describes the position of model relative to camera

Articulated Models



Adapted from Stenger *et al.* (2001a). ©2001 IEEE.

Articulated Models



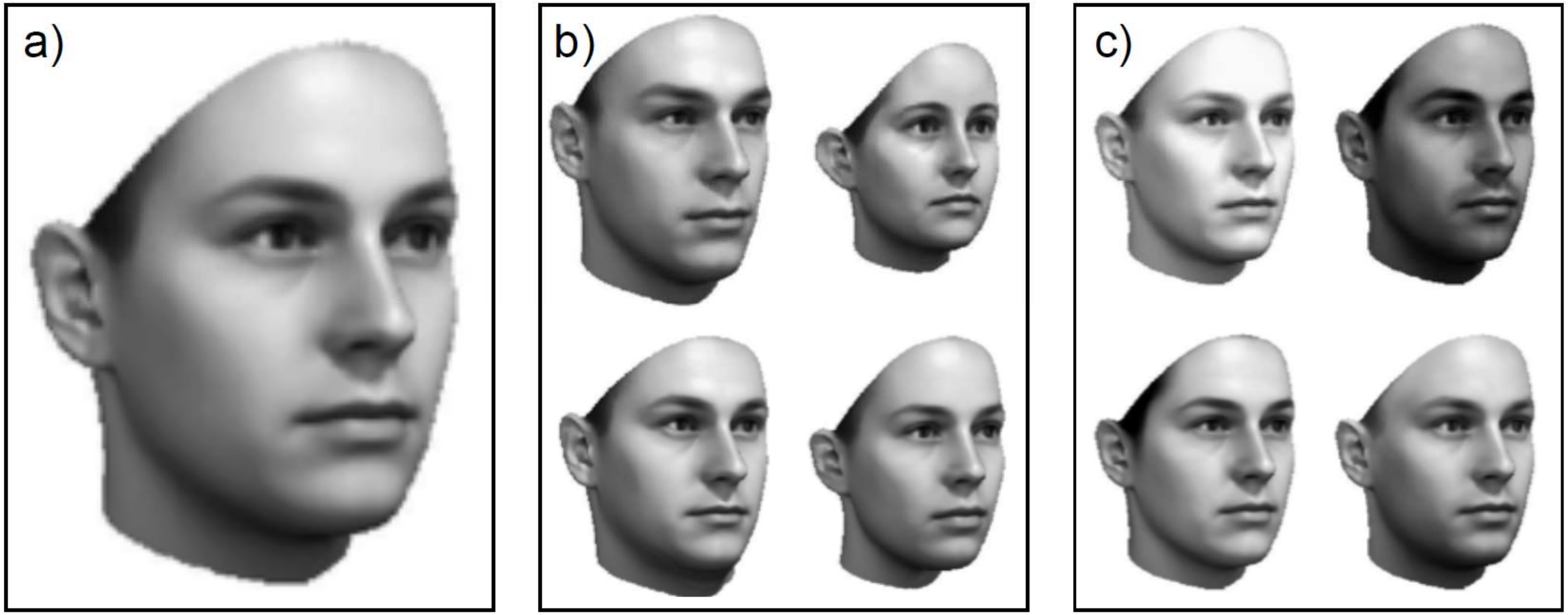
- One possible model for an object part is a quadric
- Represents spheres, ellipsoids, cylinders, pairs of planes and others
- Make truncated cylinders by clipping with cylinder with pair of planes
- Projects to conic in the image

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 & \psi_4 \\ \psi_2 & \psi_5 & \psi_6 & \psi_7 \\ \psi_3 & \psi_6 & \psi_8 & \psi_9 \\ \psi_4 & \psi_7 & \psi_9 & \psi_{10} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$$

Structure

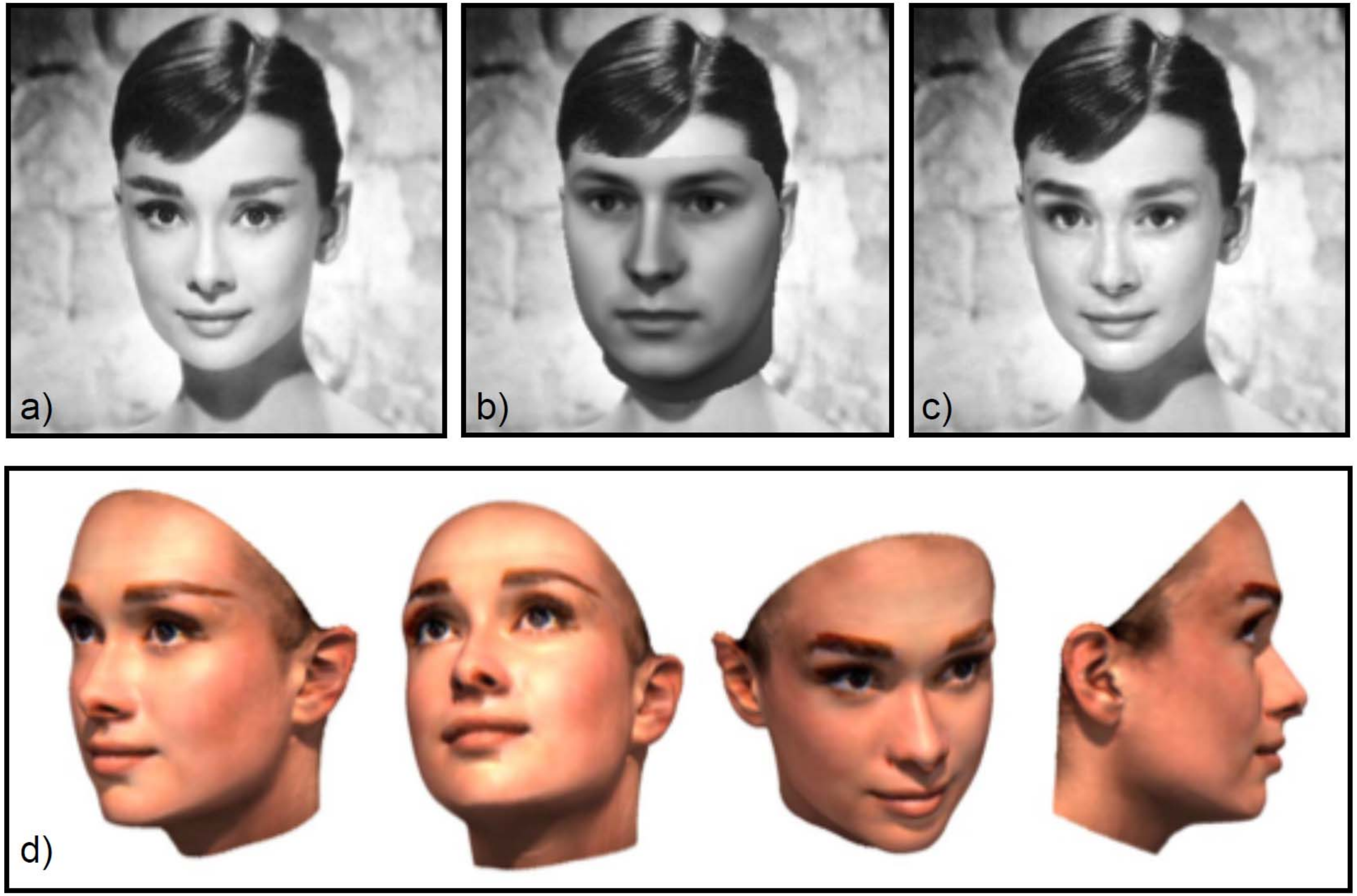
- Snakes
- Template models
- Statistical shape models
  - 3D shape models
  - Models for shape and appearance
  - Non-linear models
- Articulated models
- Applications

3D morphable models



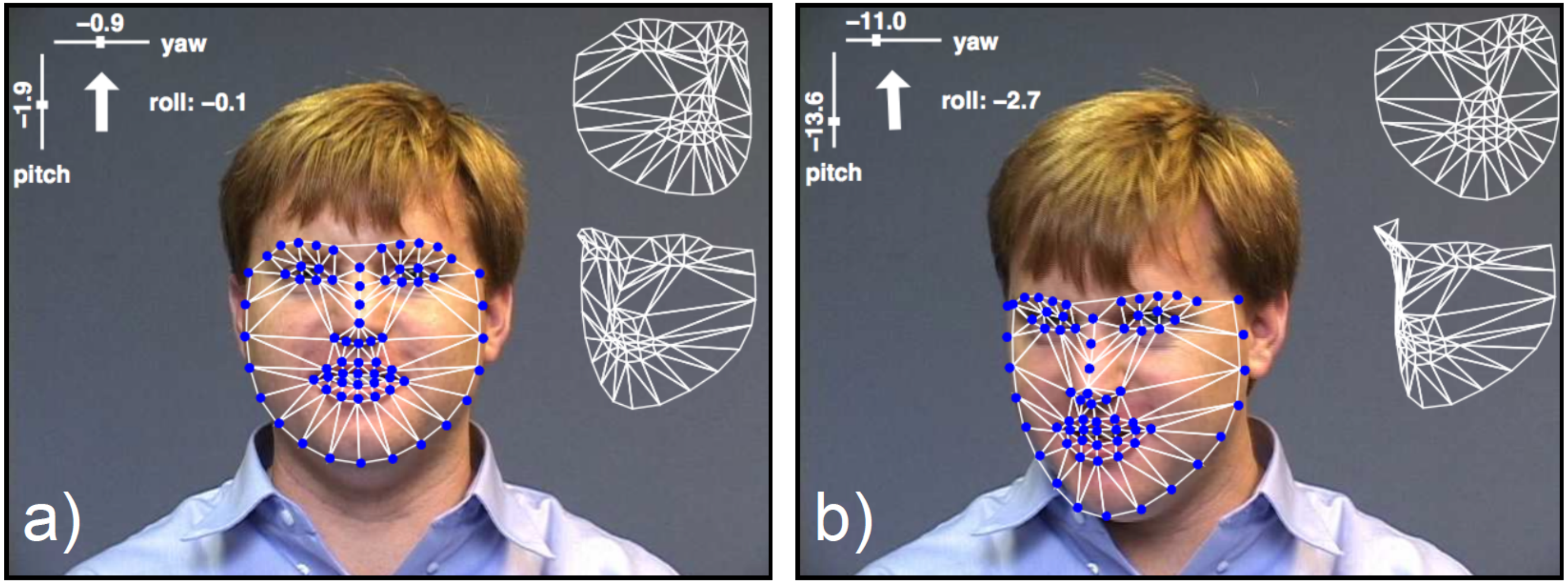
Adapted from Blanz & Vetter (2003). ©2003 IEEE

3D morphable models



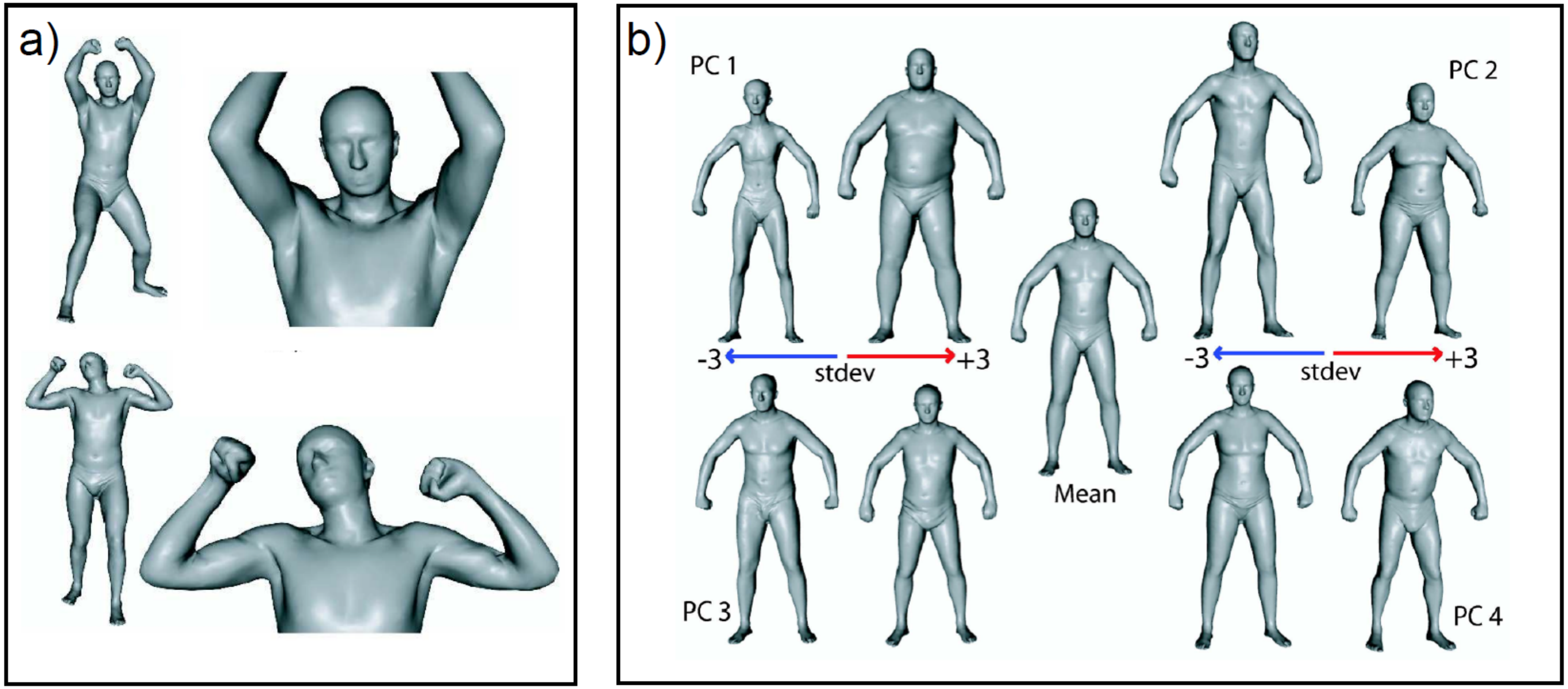
Adapted from Blanz & Vetter (1999).

3D morphable models



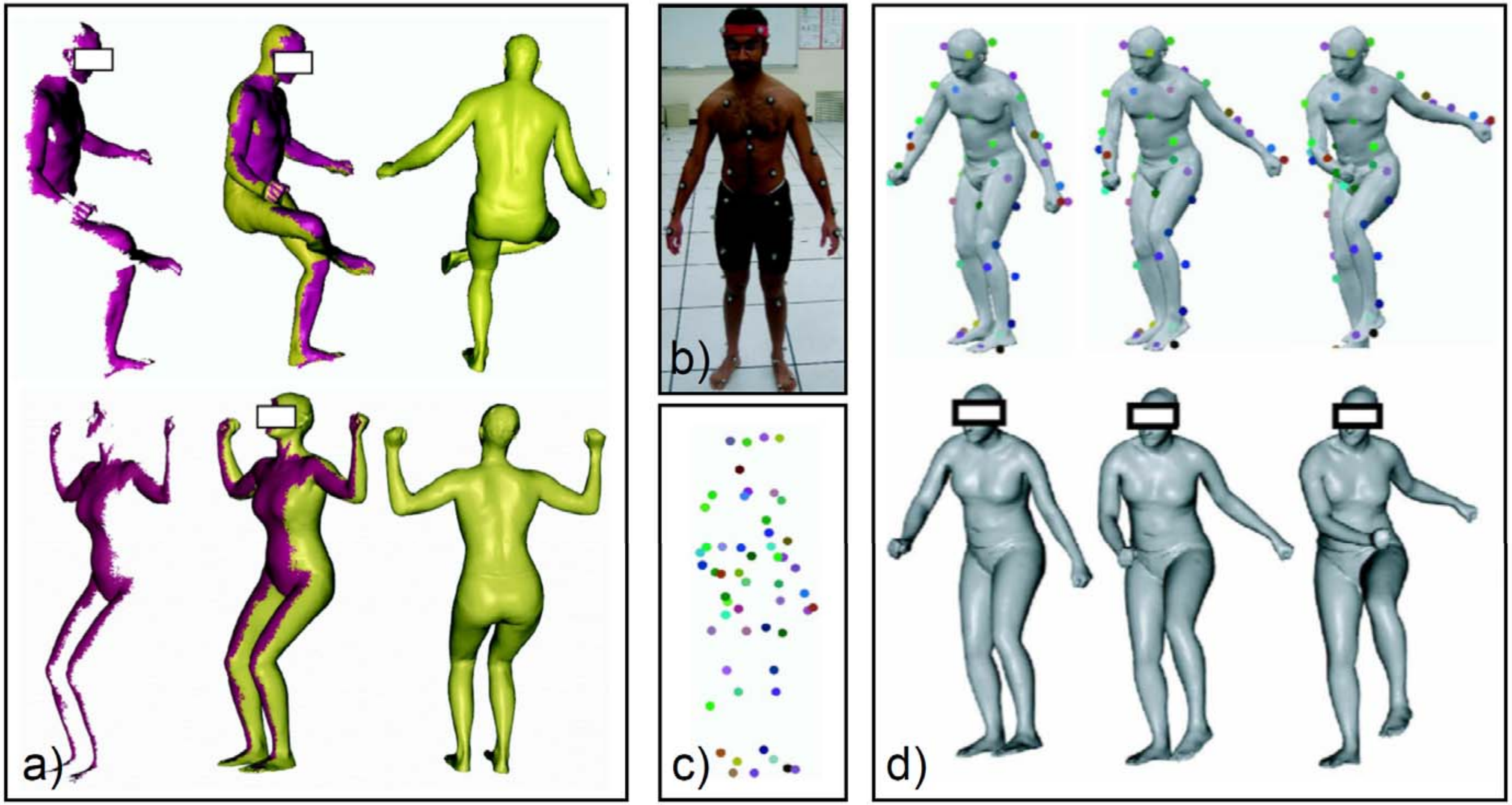
Matthews *et al.* (2007). ©2007 Springer.

3D body model



Adapted from Anguelov *et al.* (2005). ©2005 ACM

3D body model applications



Adapted from Anguelov *et al.* (2005). ©2005 ACM

Conclusions

- Introduced a series of models for shape
- Assume different forms of prior knowledge
  - Contour is smooth (snakes)
  - Shape is known, but not position (template)
  - Shape class is known (statistical models)
  - Structure of shape known (articulated model)
- Relates to other models
  - Based on subspace models
  - Tracked using temporal models