High-level shape models

Julien Carretero 'Mutating Bench'

Iasonas Kokkinos
Center for Visual Computing
Ecole Centrale Paris / INRIA Saclay

Core problem: correspondence

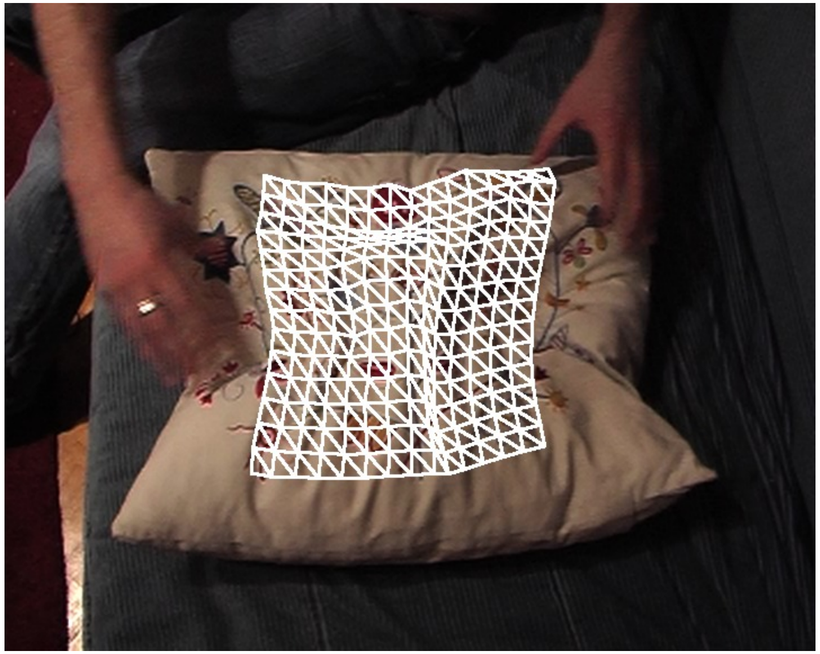
‘what is here comes from here’

Simple case: global rigid transformation (4-9 DoF)

Core problem: correspondence

‘what is here comes from here’

Complications: non-rigid transformation



Core problem: correspondence

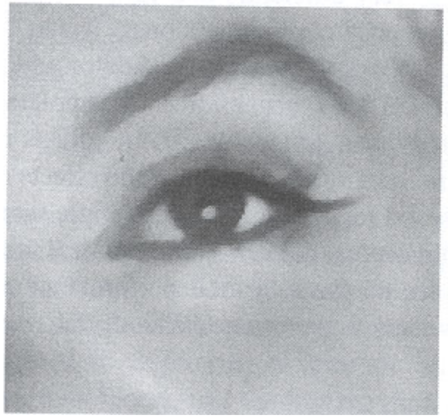
‘what is here comes from here’

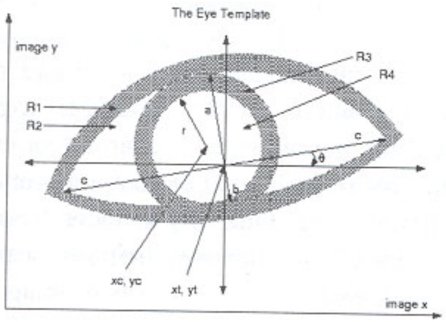
Complications: background changes

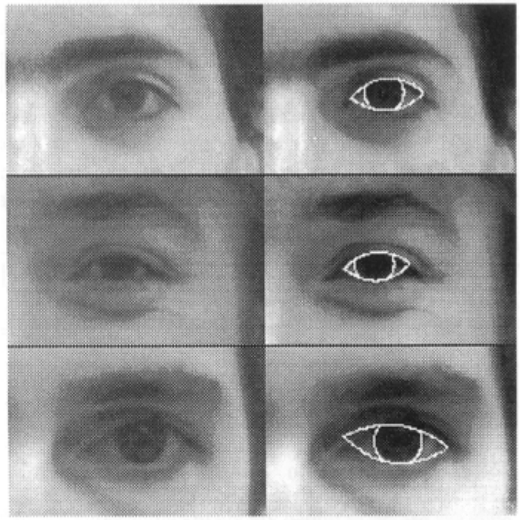
Core problem: correspondence

‘what is here comes from there’

Complications: intra-category variation







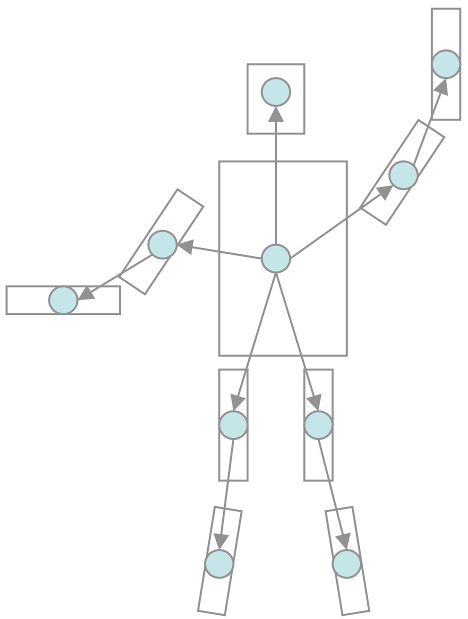
‘One shape fits all’

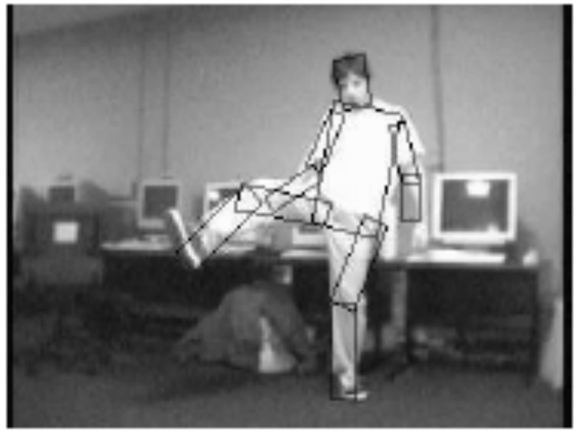
A.L. Yuille, D.S. Cohen and P.W. Hallinan. Feature extraction from faces using deformable templates. CVPR 1989.

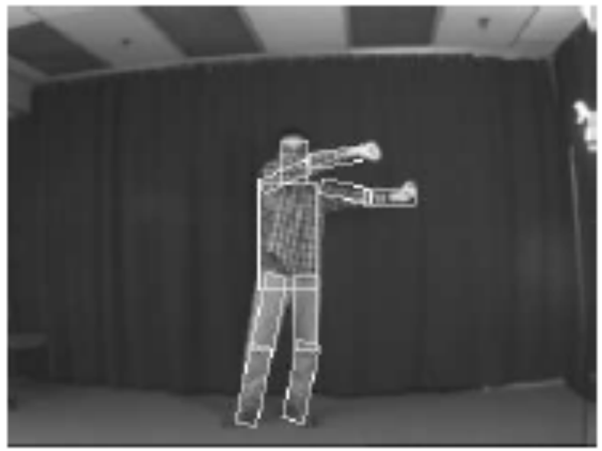
Core problem: correspondence

‘what is here comes from there’

Complications: computation







Is there a shape? Where?

Core problem: correspondence

‘what is here comes from there’

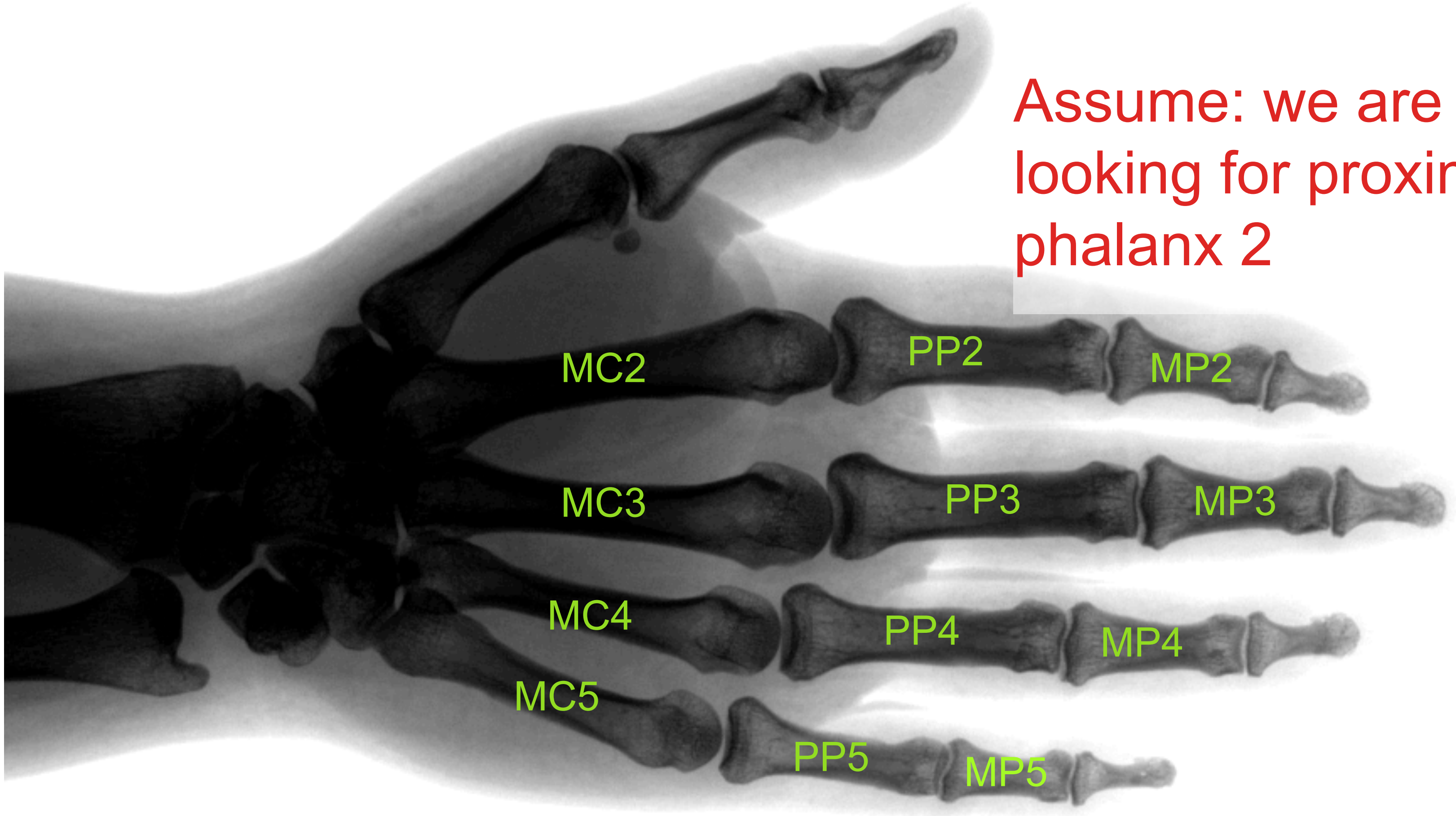
Complications: 3D, pose variability

T. J. Cashman, A. W. Fitzgibbon: What Shape Are Dolphins? Building 3D Morphable Models from 2D Images, 2013
http://research.microsoft.com/en-us/um/people/awf/dolphins/

Motivation: Analyze a hand radiograph

Slide credit: G. Langs

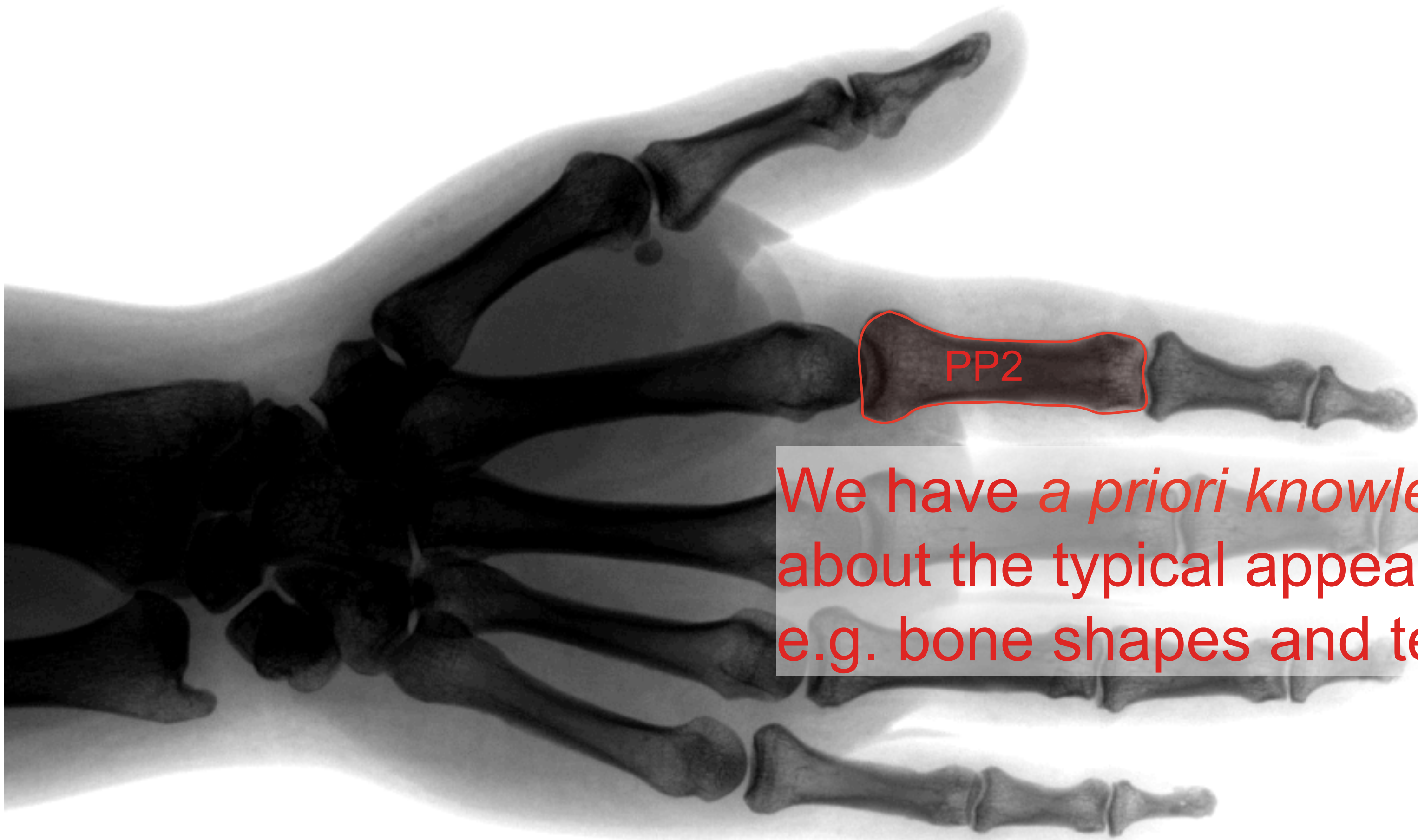
Motivation: Analyze a hand radiograph



Assume: we are looking for proximal phalanx 2

Slide credit: G. Langs

Motivation: Analyze a hand radiograph



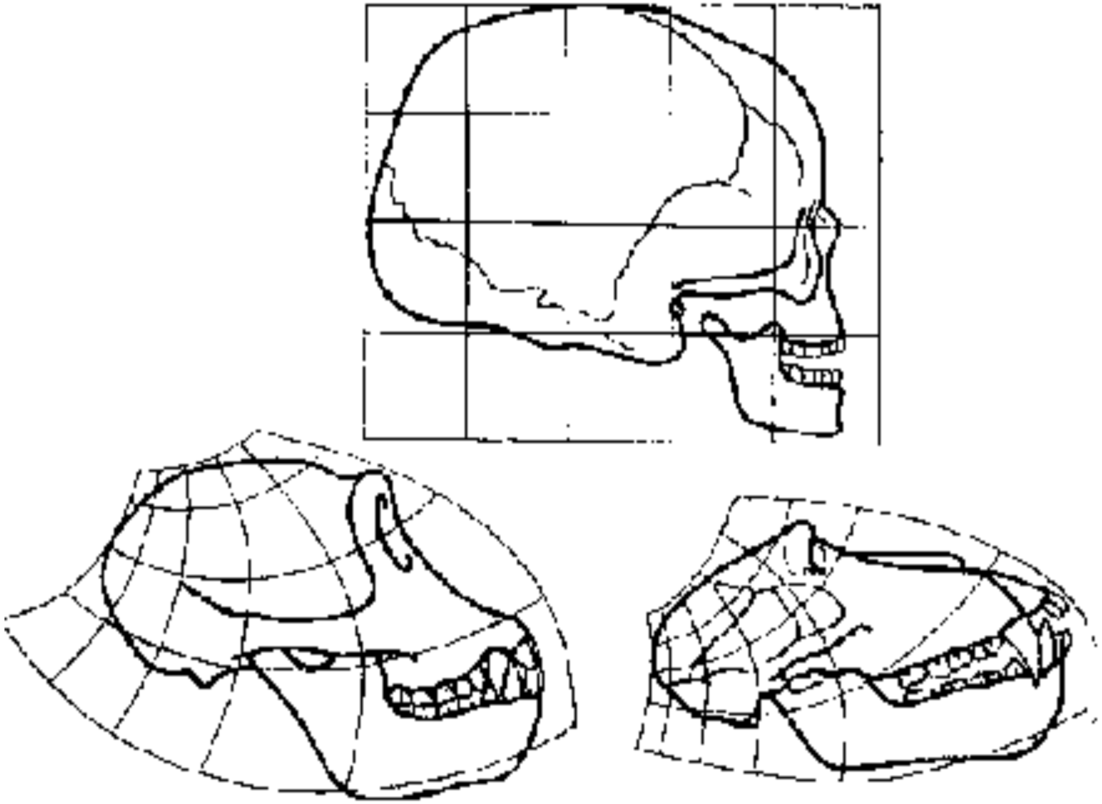
We have *a priori knowledge* about the typical appearance: e.g. bone shapes and texture

How can we represent this knowledge?

How can we exploit it?

Slide credit: G. Langs

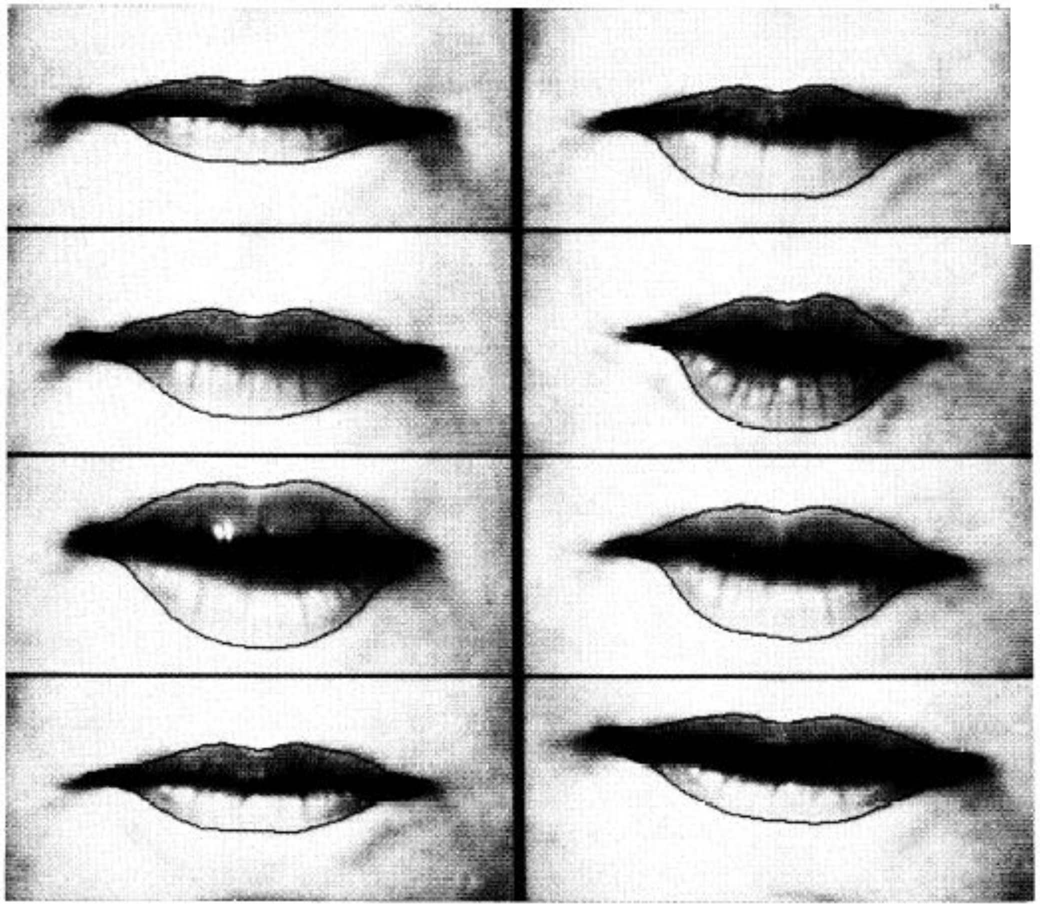
On growth and form (1917)



Skulls of a human, a chimpanzee and a baboon
and transformations between them

D’Arcy Thompson, “On Growth and Form” (1917)

Snakes-Active Contour Models (1987)



$$E^*_{\text{snake}} = \int_0^1 E_{\text{snake}}(\mathbf{v}(s))\, ds$$
$$= \int_0^1 E_{\text{int}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s))$$
$$+ E_{\text{con}}(\mathbf{v}(s))\, ds$$

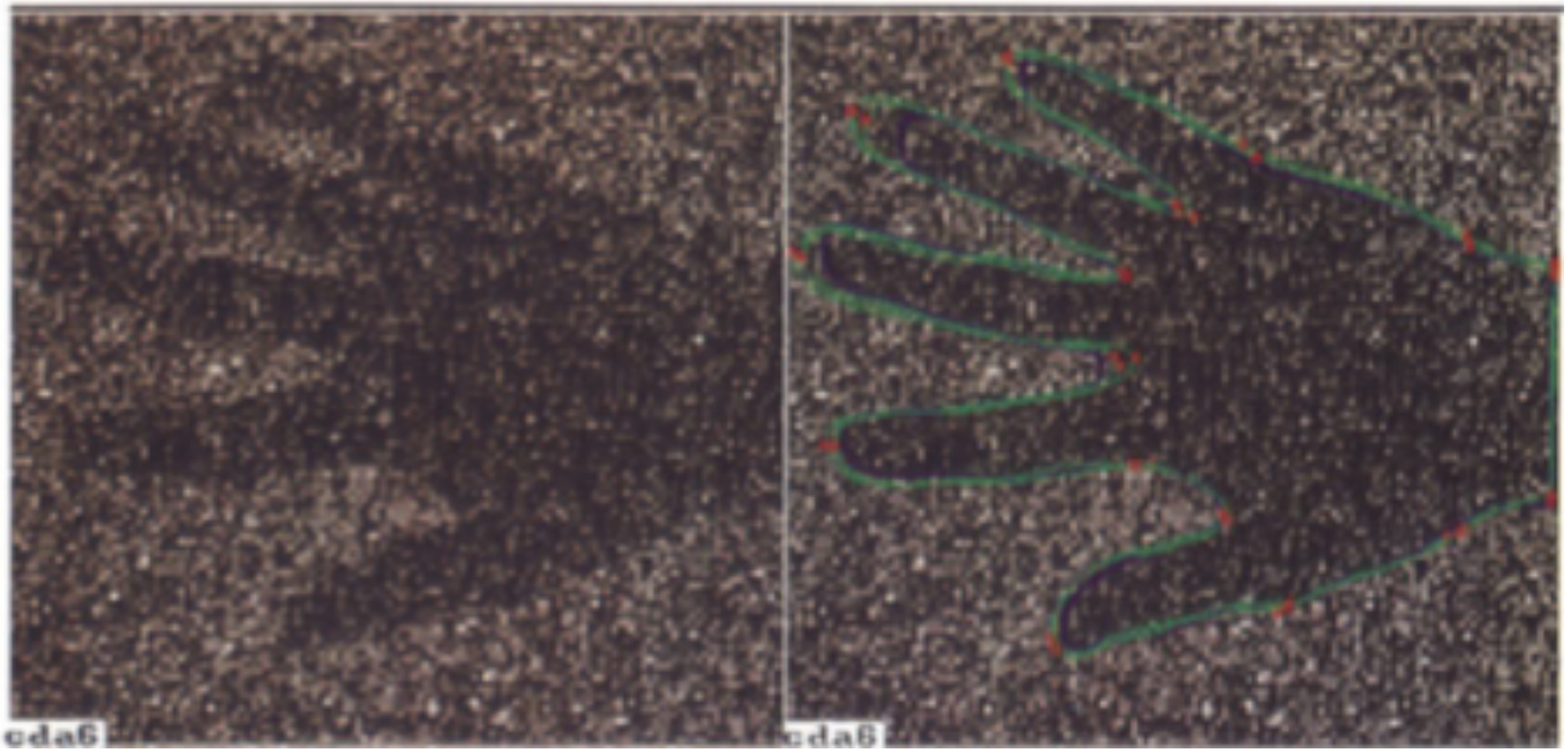
M. Kass, A. Witkin and D. Terzopoulos, `Snakes', 1987

Pattern theory - Hands (1990)

Samples from the prior distribution

U. Grenander, Hands: a pattern theoretic study of biological shapes, 1990

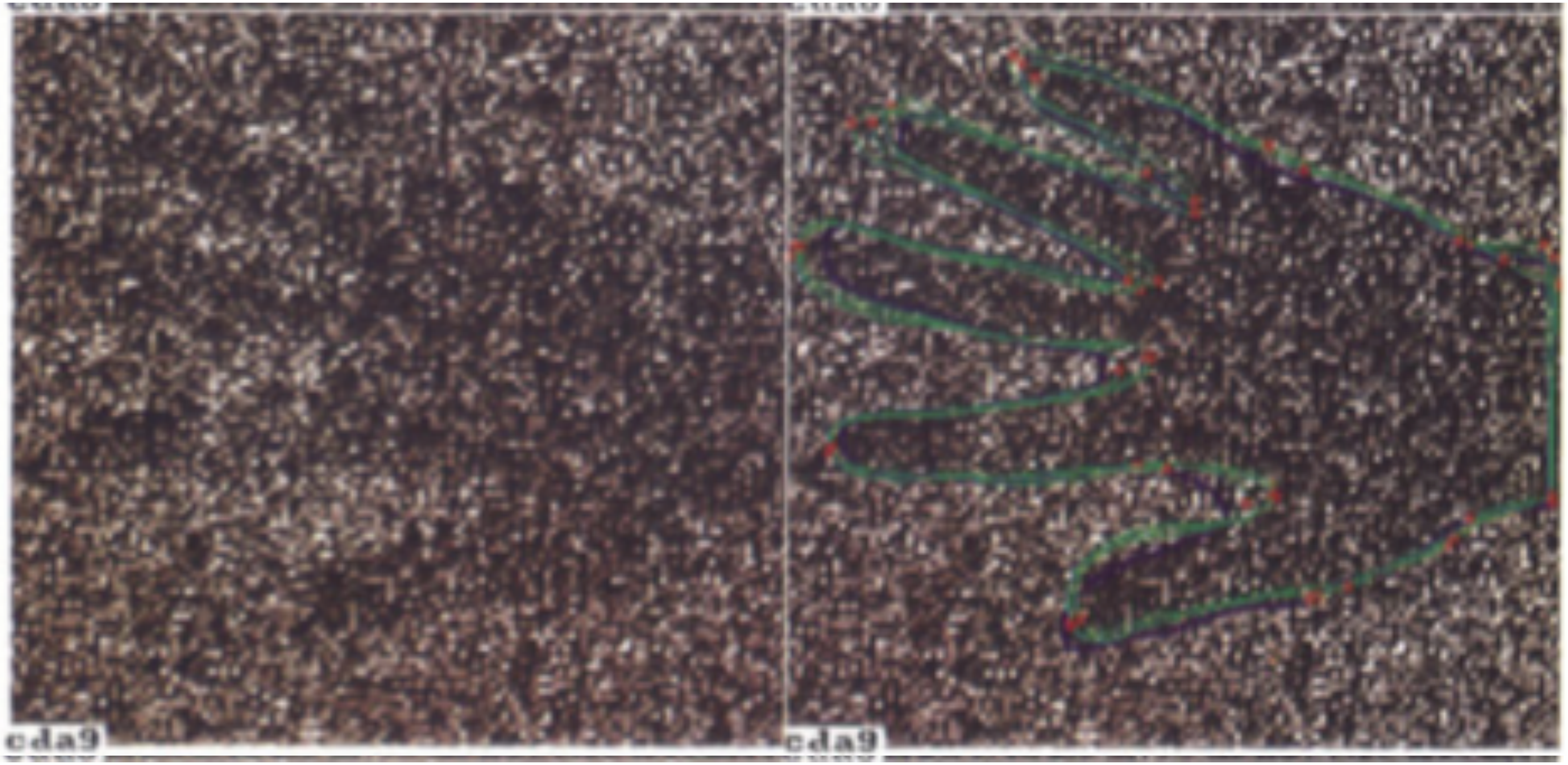
Pattern theory - Hands (1990)



Input Samples from posterior

U. Grenander, Hands: a pattern theoretic study of biological shapes, 1990

Pattern theory - Hands (1990)

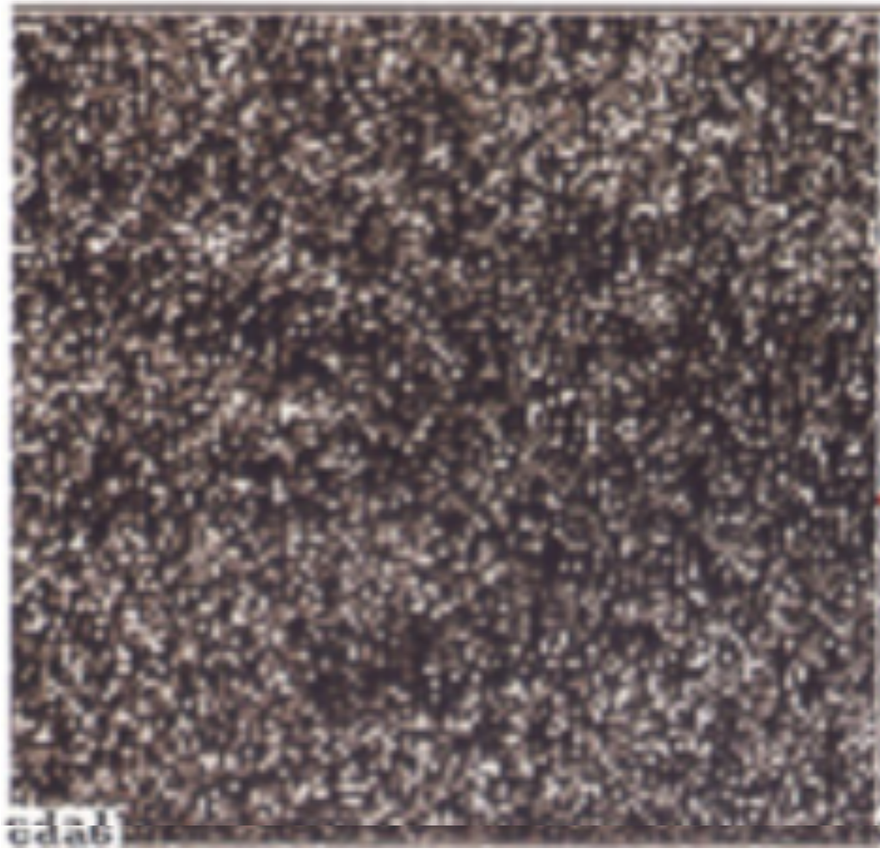


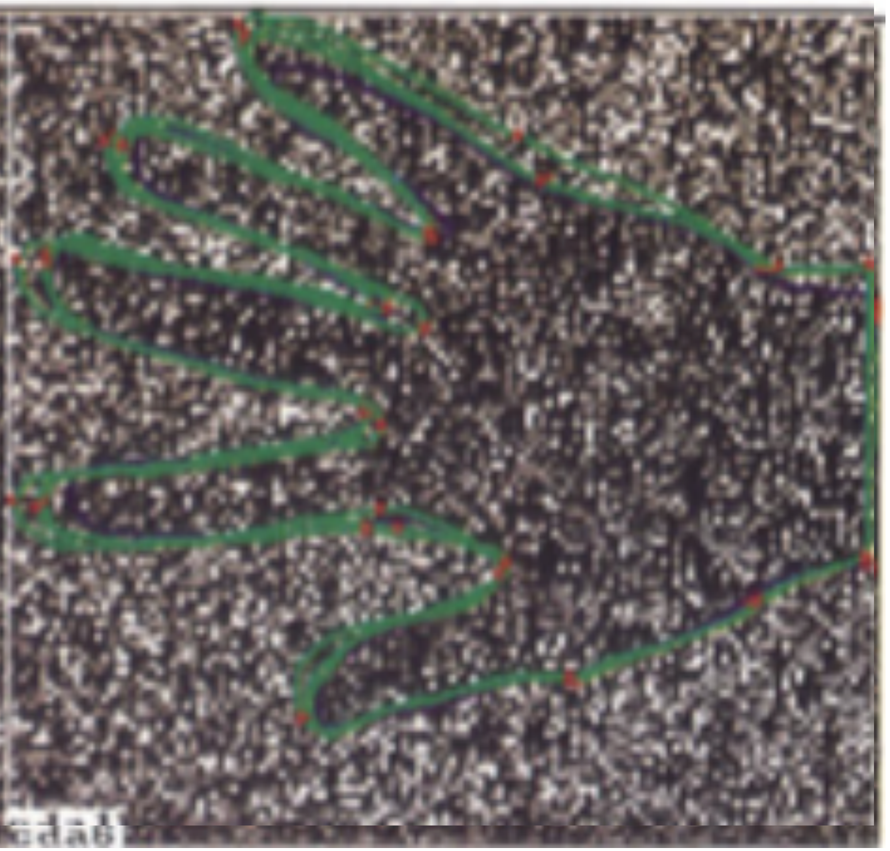
Input

Samples from posterior

U. Grenander, Hands: a pattern theoretic study of biological shapes, 1990

Pattern theory - Hands (1990)



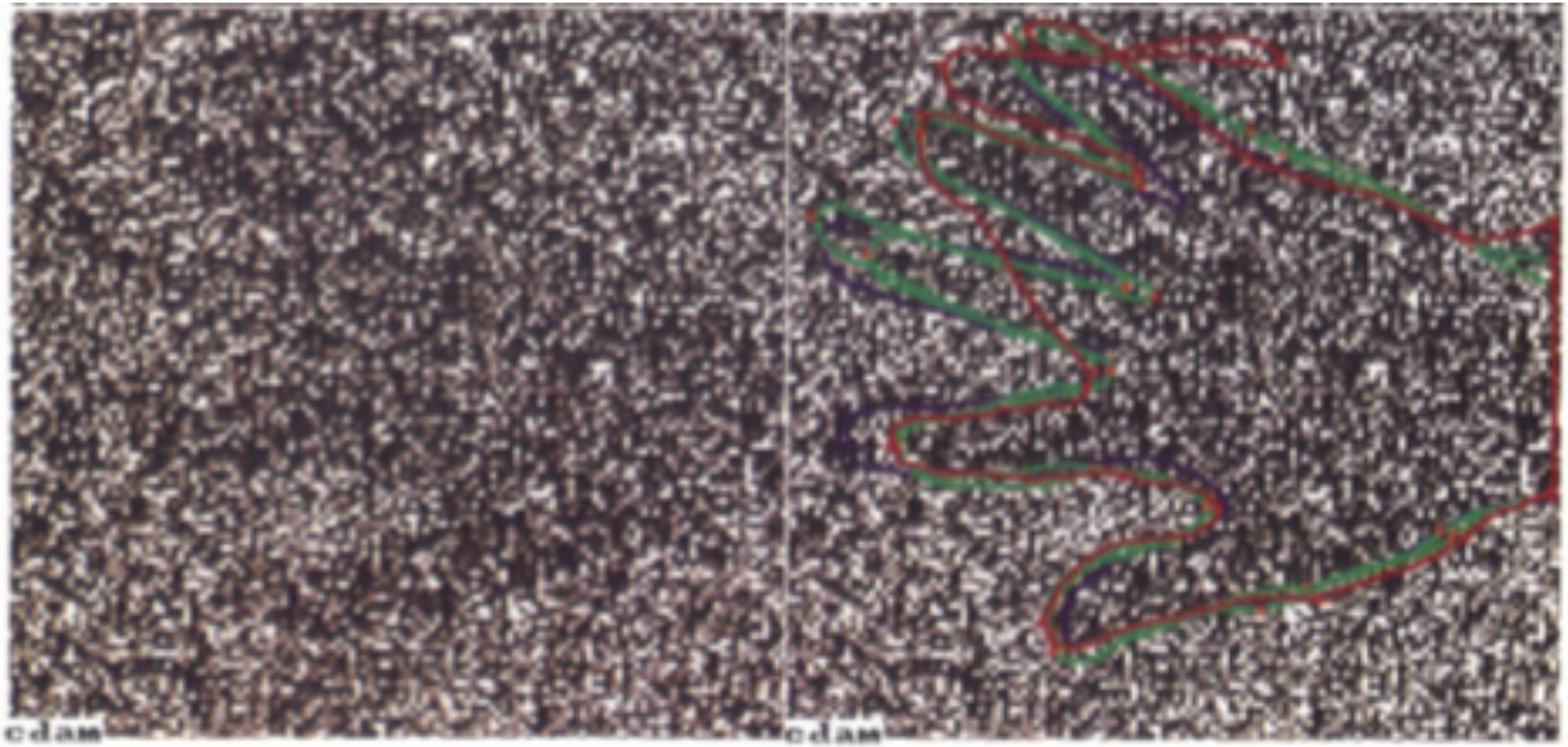


Input

Samples from posterior

U. Grenander, Hands: a pattern theoretic study of biological shapes, 1990

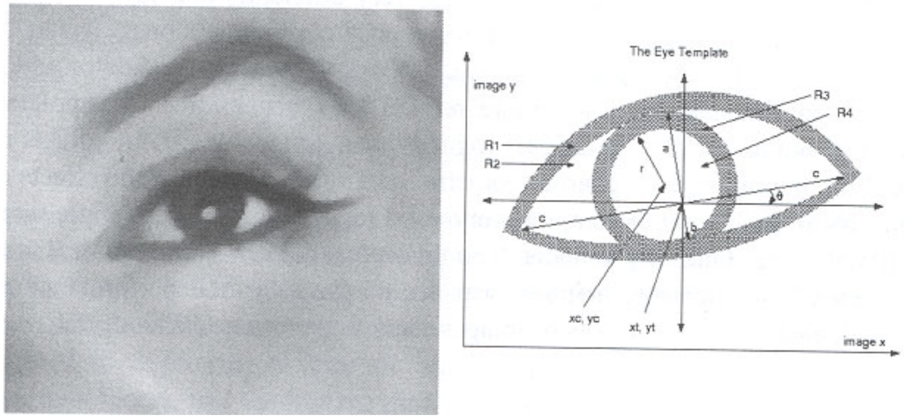
Pattern theory - Hands (1990)

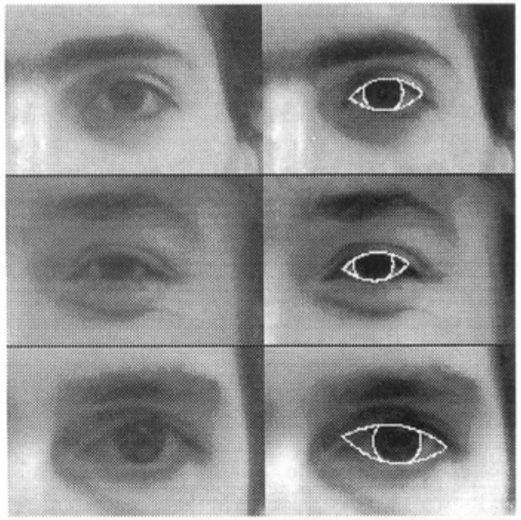


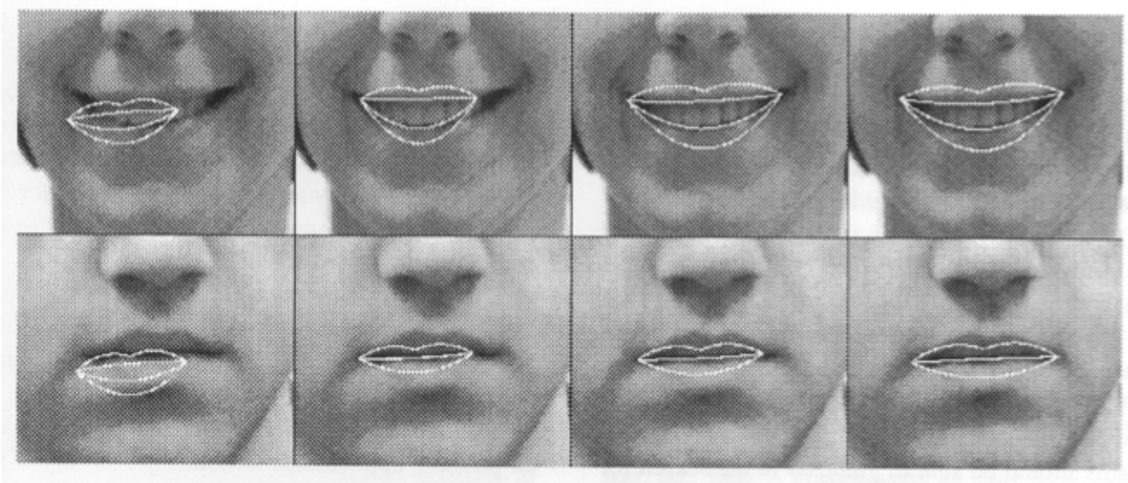
Input Samples from posterior

U. Grenander, Hands: a pattern theoretic study of biological shapes, 1990

Deformable templates - 1989

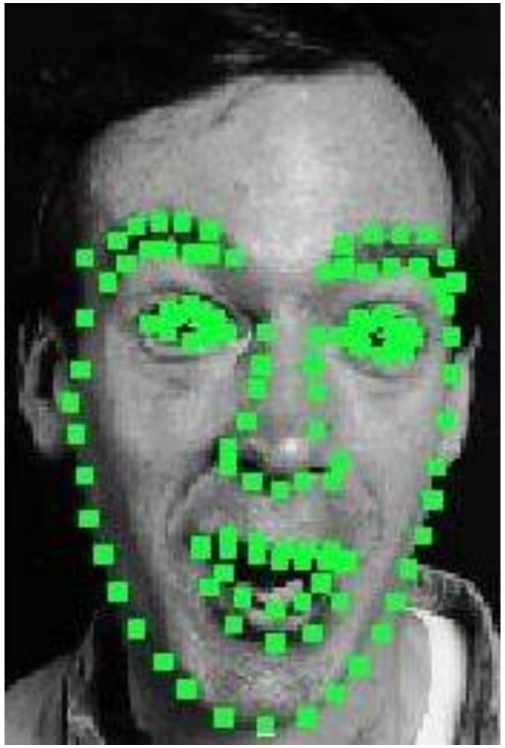






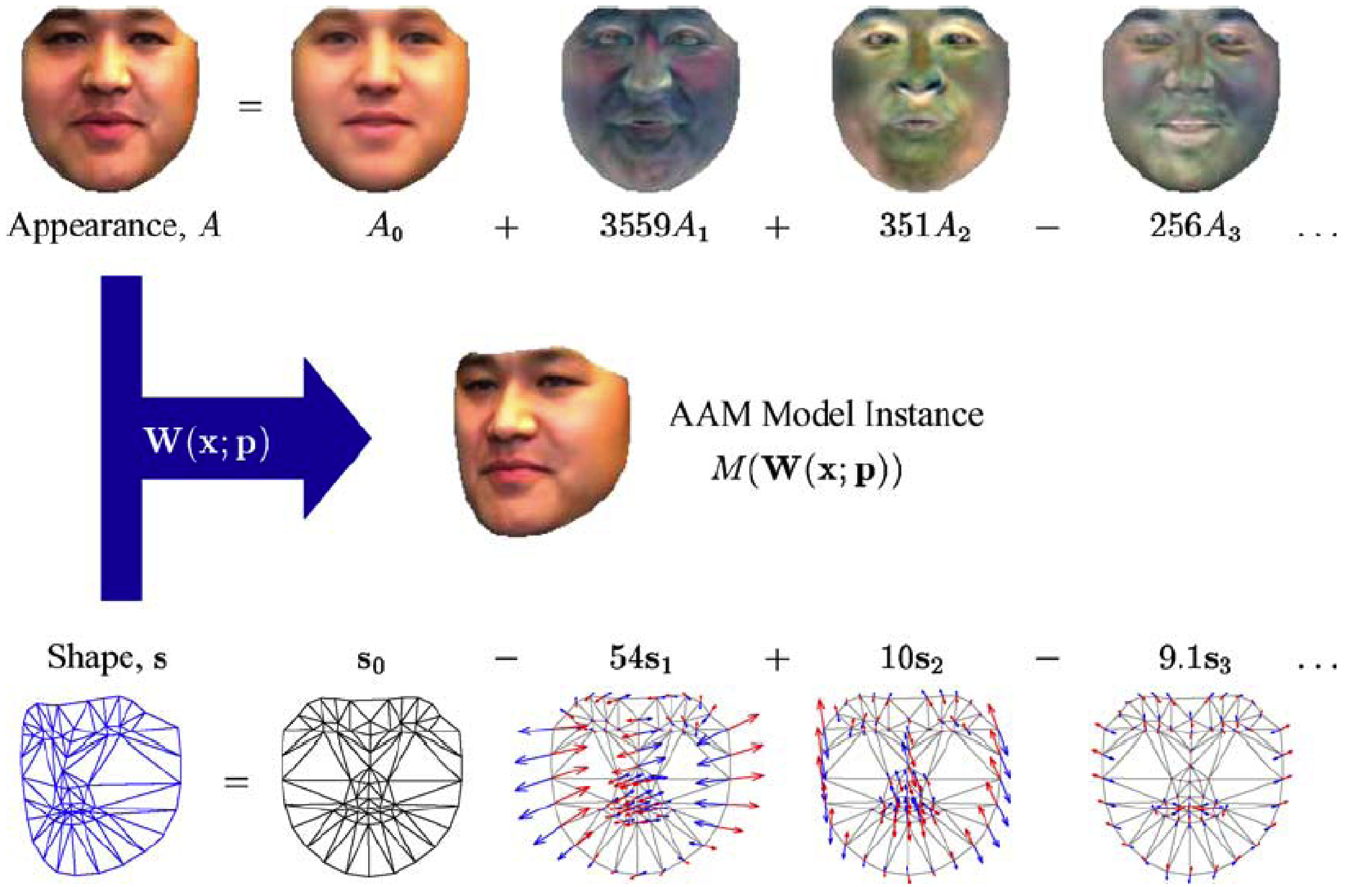
A.L. Yuille, D.S. Cohen and P.W. Hallinan. Feature extraction from faces using deformable templates. CVPR 1989.

Active Shape Models - 1992



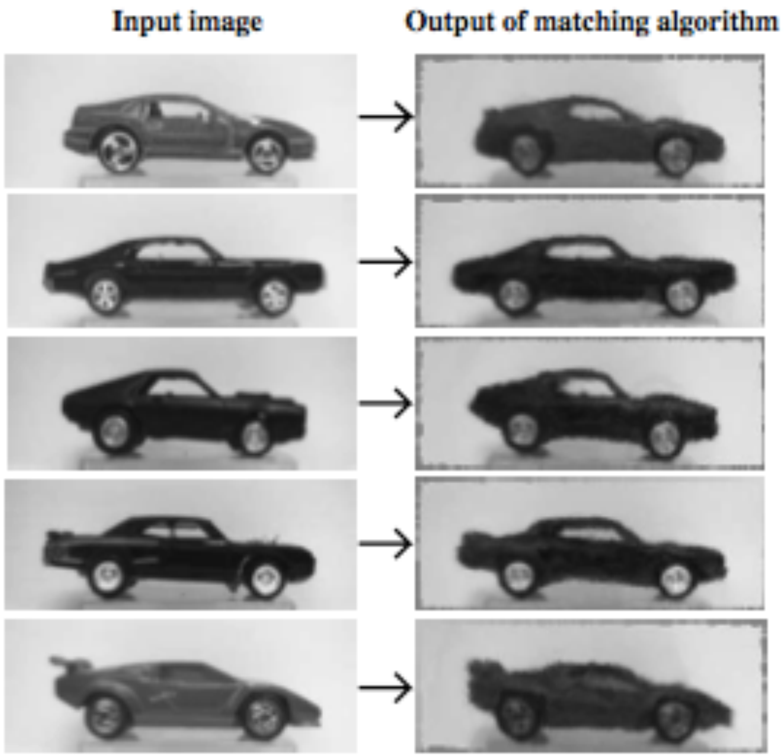
T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape **from sets of examples.** BMVC 1992

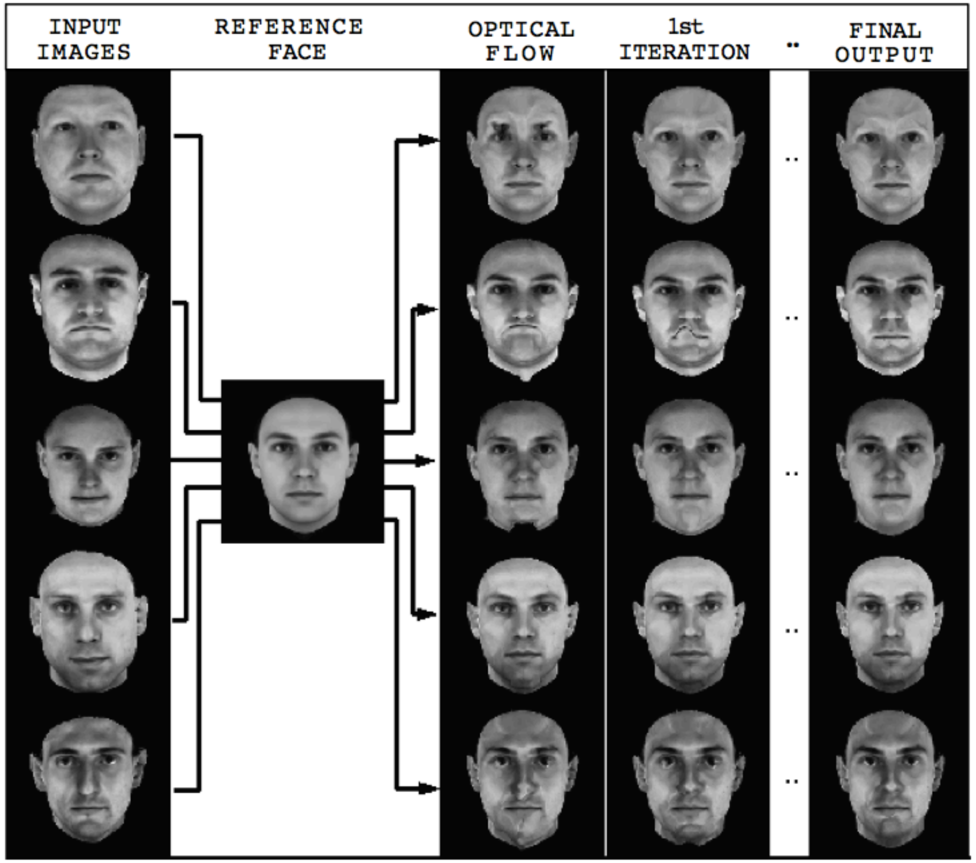
Active Appearance Models - 1998



T.F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models, 1998
T. F. Cootes, G. V. Wheeler, K. N. Walker, C. J. Taylor: View-based active appearance models.2002,
I. Matthews and S. Baker, "Active Appearance Models Revisited," 2004.

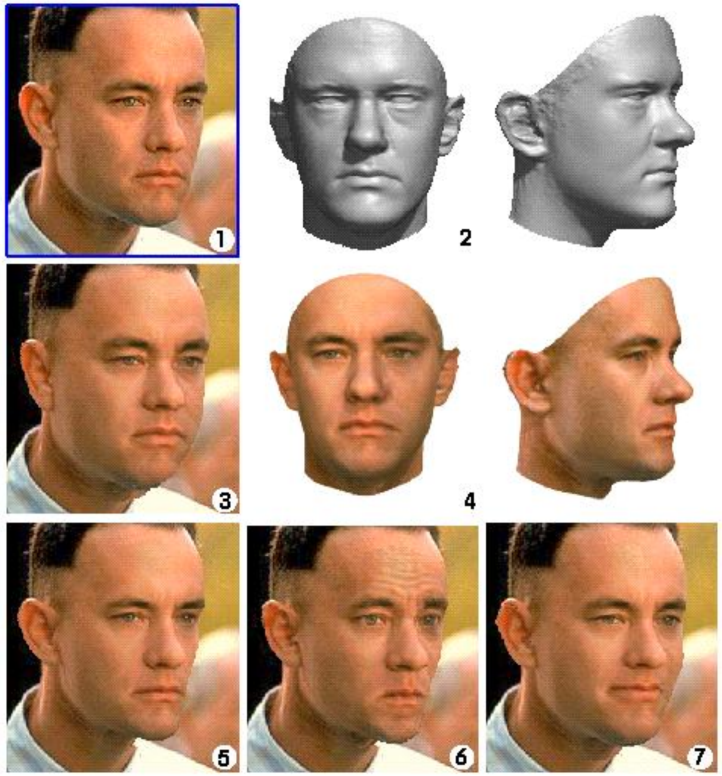
Morphable Models - 1996





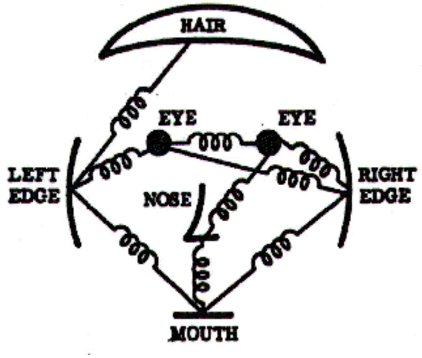
T. Vetter, T. Poggio: Linear Object Classes and Image Synthesis From a Single Example Image., 1997.
T. Vetter, M. J. Jones, T. Poggio: A bootstrapping algorithm for learning linear models of object classes. CVPR 1997
M. J. Jones, Y. Poggio: Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes. IJCV 1998

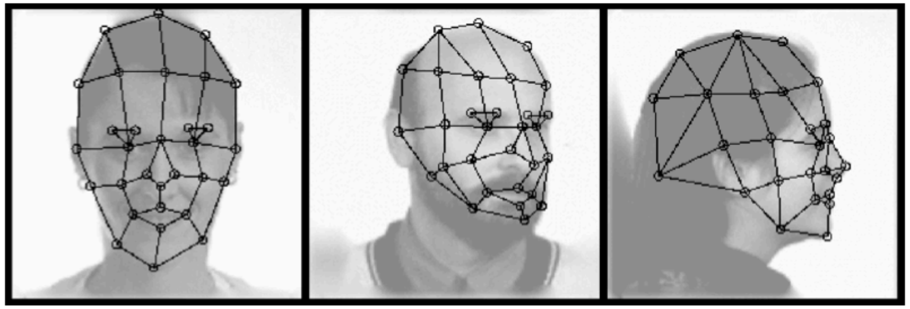
3D Morphable Models - 1999



V. Blanz, T. Vetter: A Morphable Model for the Synthesis of 3D Faces. SIGGRAPH 1999
V. Blanz, T. Vetter: Face Recognition Based on Fitting a 3D Morphable Model. IEEE PAMI, 2003
T. J. Cashman, A. W. Fitzgibbon: What Shape Are Dolphins? Building 3D Morphable Models from 2D Images, 2013

Deformable Part Models (DPMs)

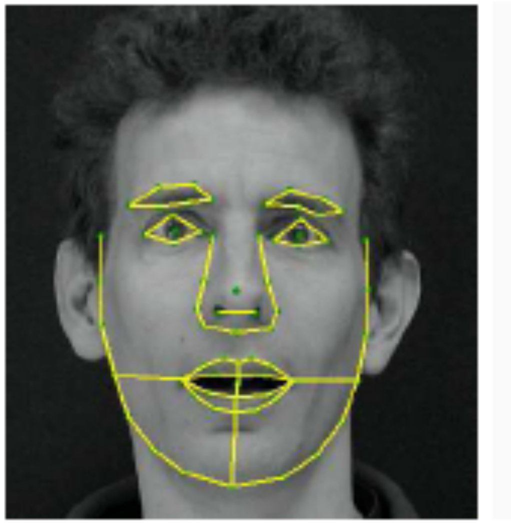




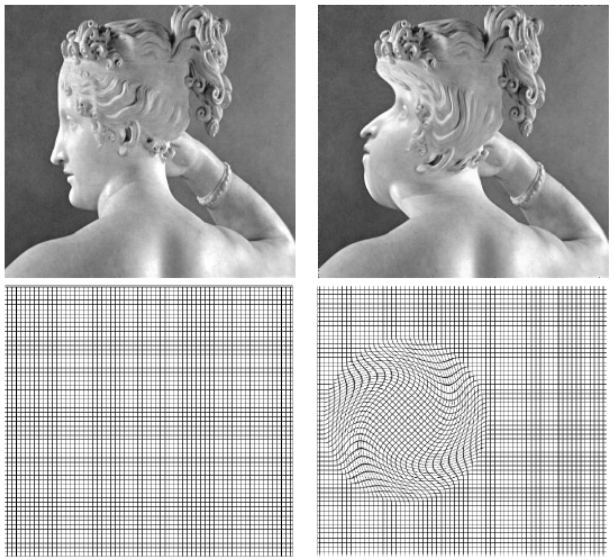
M. Fischler and R. Erschlanger. The Representation and Matching of Pictorial Structures '73.
M. Lades, et al: Distortion Invariant Object Recognition in the Dynamic Link Architecture. '93
Y. Amit, A. Kong: Graphical Templates for Model Registration. '96
A. L. Yuille, J. M. Coughlan: An A perspective on deterministic optimization for deformable templates. '00*
M. C. Burl, P. Perona: Recognition of Planar Object Classes. '96
M. C. Burl, M. Weber, P. Perona: A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry. '98
M. Weber, M. Welling, P. Perona: Unsupervised Learning of Models for Recognition. '00
P. Felzenszwalb, and D. Huttenlocher, Pictorial Structures for Object Recognition, IJCV '05
P. Felzenszwalb, et. al., Object Detection with Discriminatively Trained DPMs,'10

Three classes of deformable models

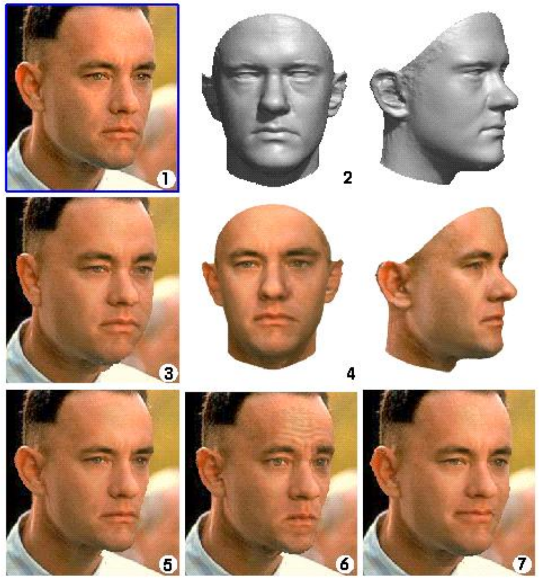
ASM



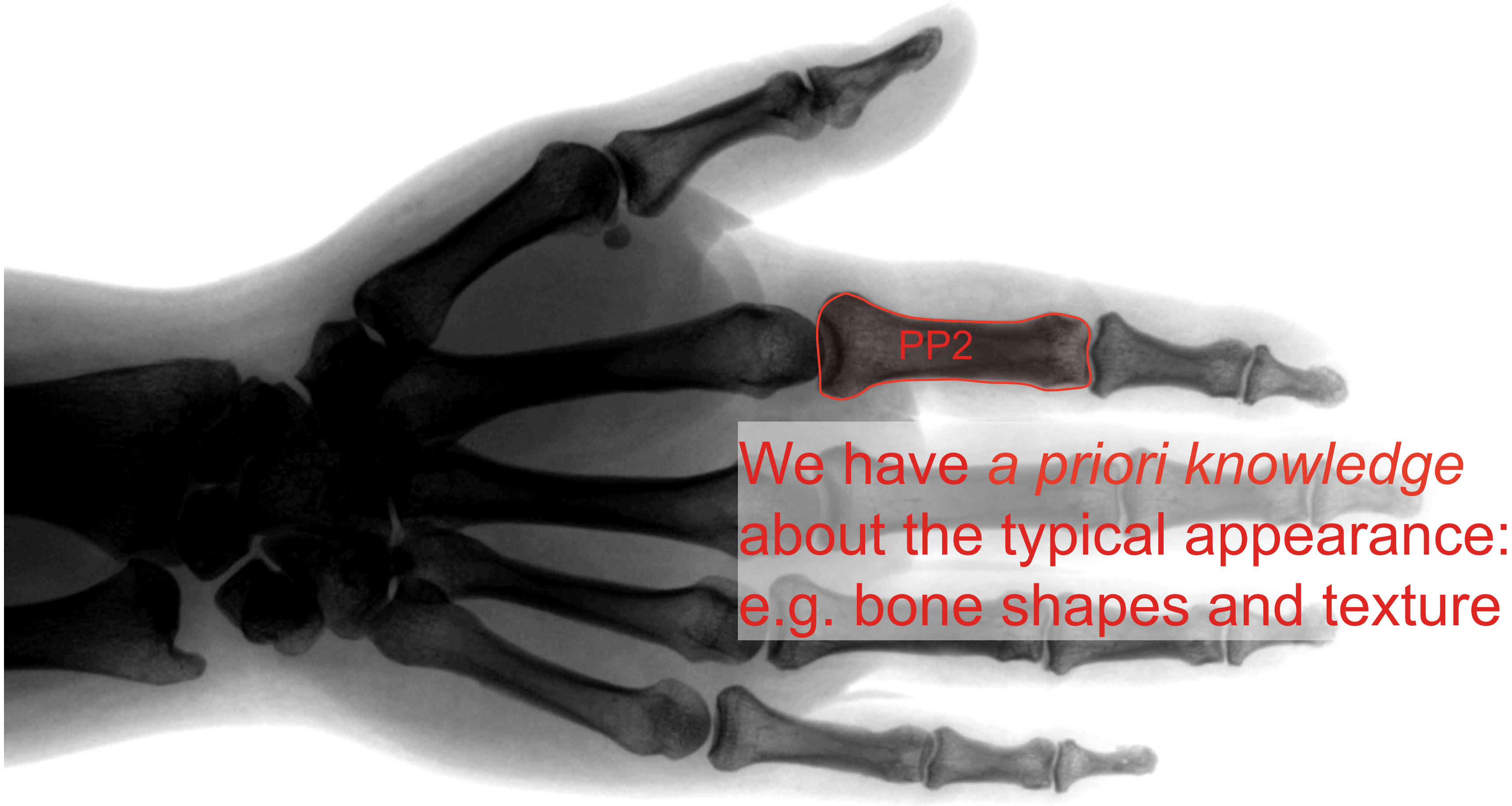
AAM



3D Morphable

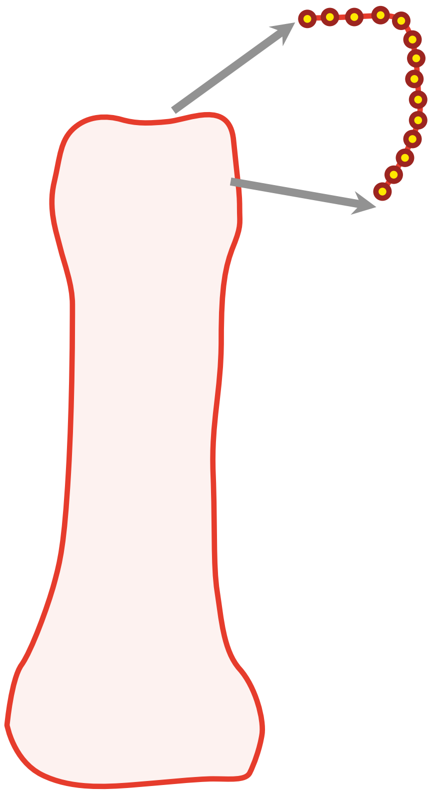


Analyzing a hand radiograph



How can we represent this knowledge?
How can we exploit it?

How to capture *a priori knowledge*?



$$\mathbf{x}_i = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_m \\ y_m \end{pmatrix}$$

Each example is represented by a vector containing the coordinates of the landmarks.

Learning: Model Acquisition
Inference: Model Fitting

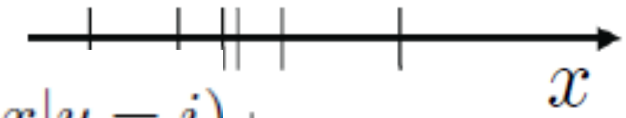
Goal:

- capture common properties of the bone
- find a representation that is restricted to plausible bones.

Slide credits: G. Langs

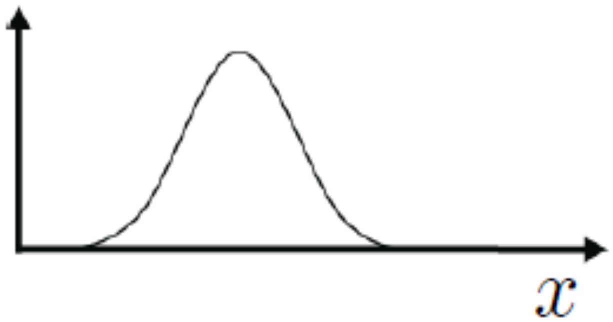
Density Estimation

- Training set: $\{(X^i, y^i)\}, \quad i = 1 \ldots N$
- Examples corresponding to class k: $s^k = \{i : y^i = k\}$
- Training data for class k: $\{X^j, j \in s^k\}$



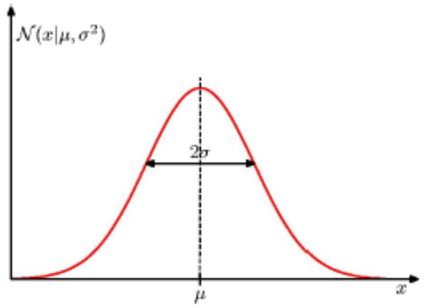
- One density estimate per class: $P(X = x | y = i)$

  for short: $P(x)$



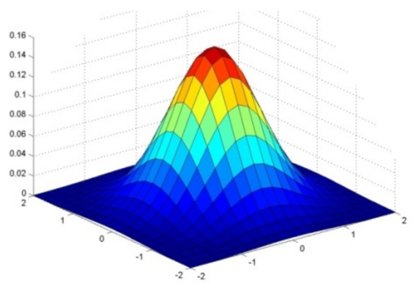
Parametric Distributions: Gaussian

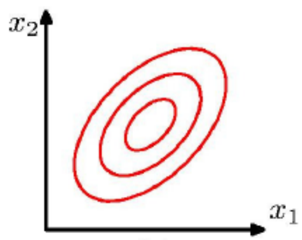
– 1 D $$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$



– N D $$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

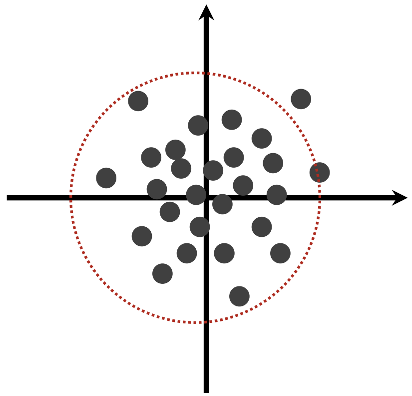
- e.g. 2D:



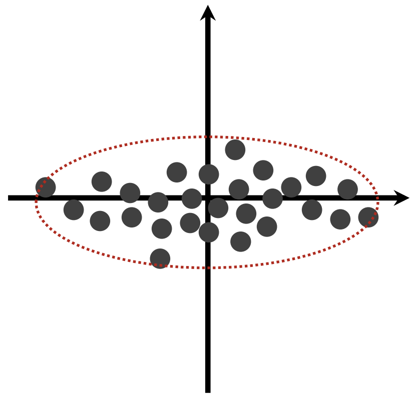


Covariance matrix reminder

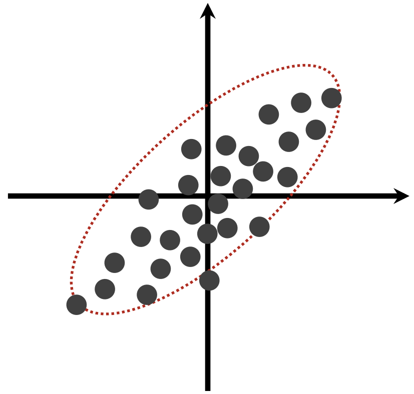
Covariance matrix: $$\Sigma_{i,j} = E\left((x_i - E(x_i))(x_j - E(x_j))\right)$$



$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 0.5 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 2.45 & 1.2 \\ 1.2 & 2.1 \end{pmatrix}$$

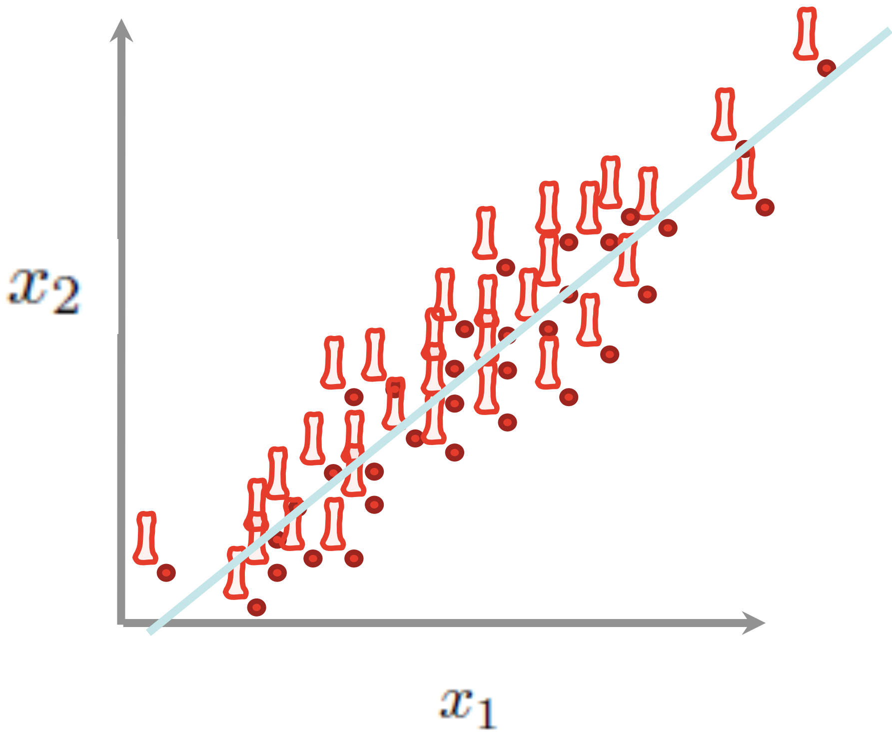
Height, Income

Height, Weight

Uncorrelated coordinates: diagonal covariance

The space of all bone shapes

- Bone shapes: vectors in R^{2m}

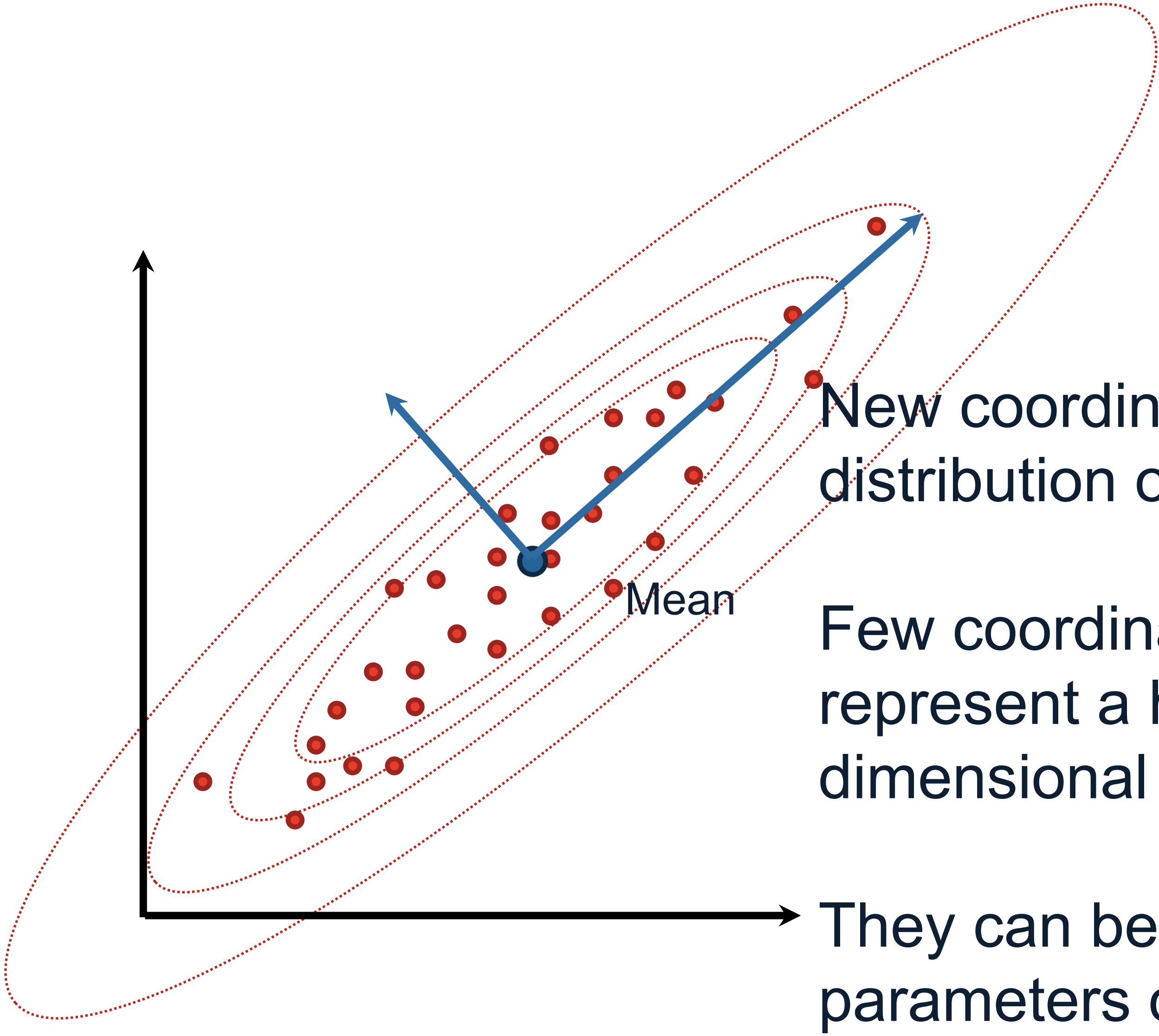


$$\mathbf{x}_i = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_m \\ y_m \end{pmatrix}$$

- Goal: project data onto a low-dimensional linear subspace that best explains their variation.

Slide credits: G. Langs

New subspace: `better' coordinate system



New coordinates reflect the distribution of the data.

Few coordinates suffice to represent a high dimensional vector

They can be viewed as parameters of a model

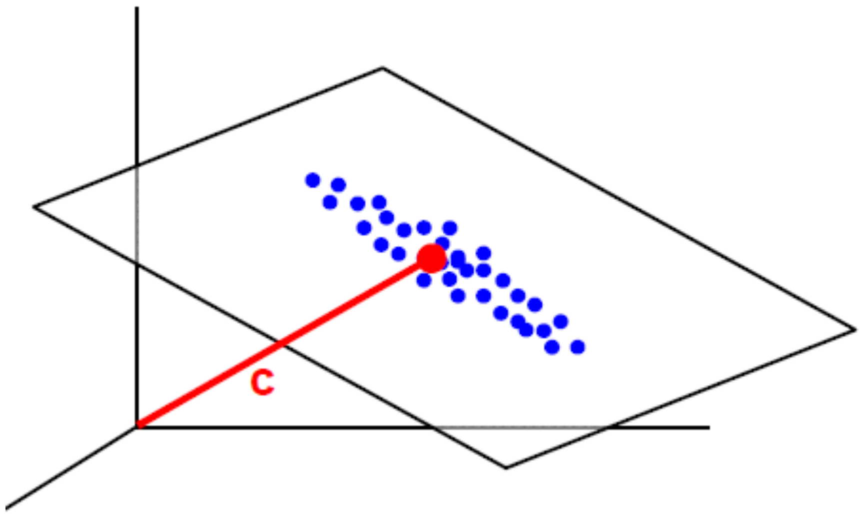
Slide credits: G. Langs

Principal Component Analysis (PCA)

- Find a low-dimensional subspace to reconstruct high-dimensional data
- Reconstruction on orthogonal basis Approximation with K terms

$$
\begin{aligned}
x^i &= \sum_{j=1}^{M} h_j^i b_j \\
&= \sum_{j=1}^{M} ({h^i}^T b_j) b_j
\end{aligned}
$$

$$\bar{x}^i = \sum_{j=1}^{K} ({h^i}^T b_j) b_j$$

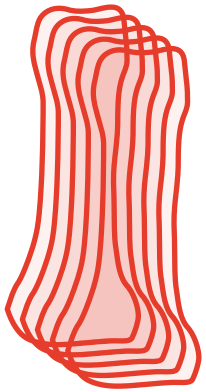


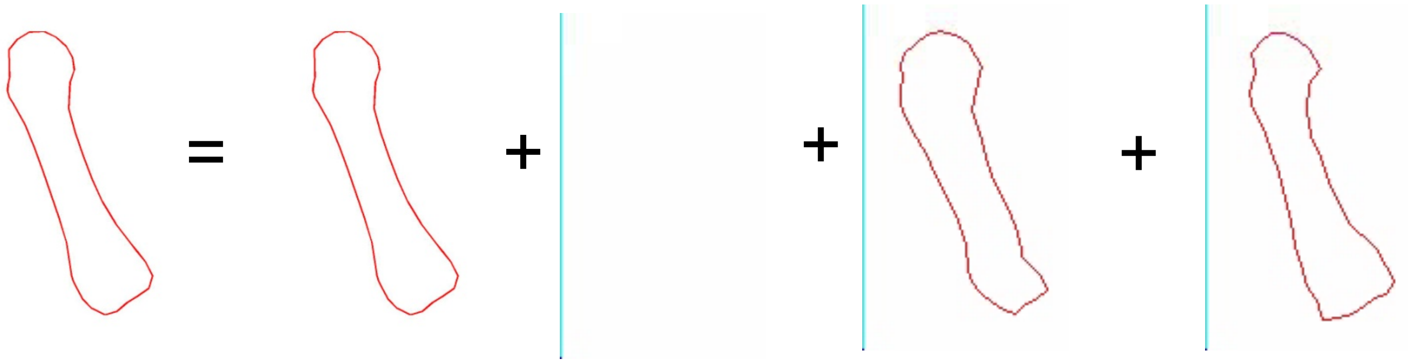
Principal component analysis

- The *k* orthogonal directions that capture most of the data variance are the *k* leading (largest-eigenvalue) covariance eigenvectors

Factor Analysis	PCA
Λ matrix	Leading K eigenvectors of covariance
Hidden variables	Inner product of data with eigenvectors

Shape Eigenbasis



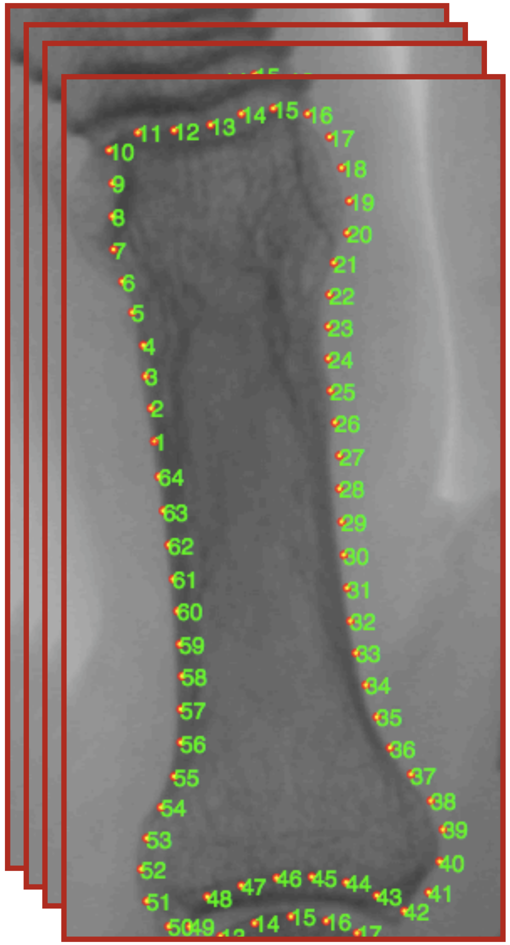


$$\mathbf{x}_{new} = \hat{\mathbf{m}} + b_1\mathbf{e}_1 + b_2\mathbf{e}_2 + b_3\mathbf{e}_3$$

Slide credits: G. Langs

Active shape models (ASM)

- A set of training examples (images)
- A set of landmarks, that are present on all images
- Build a statistical model of shape variation (PCA)
- Build a statistical model of the local texture (PCA)
- Use the model for the search in a new image

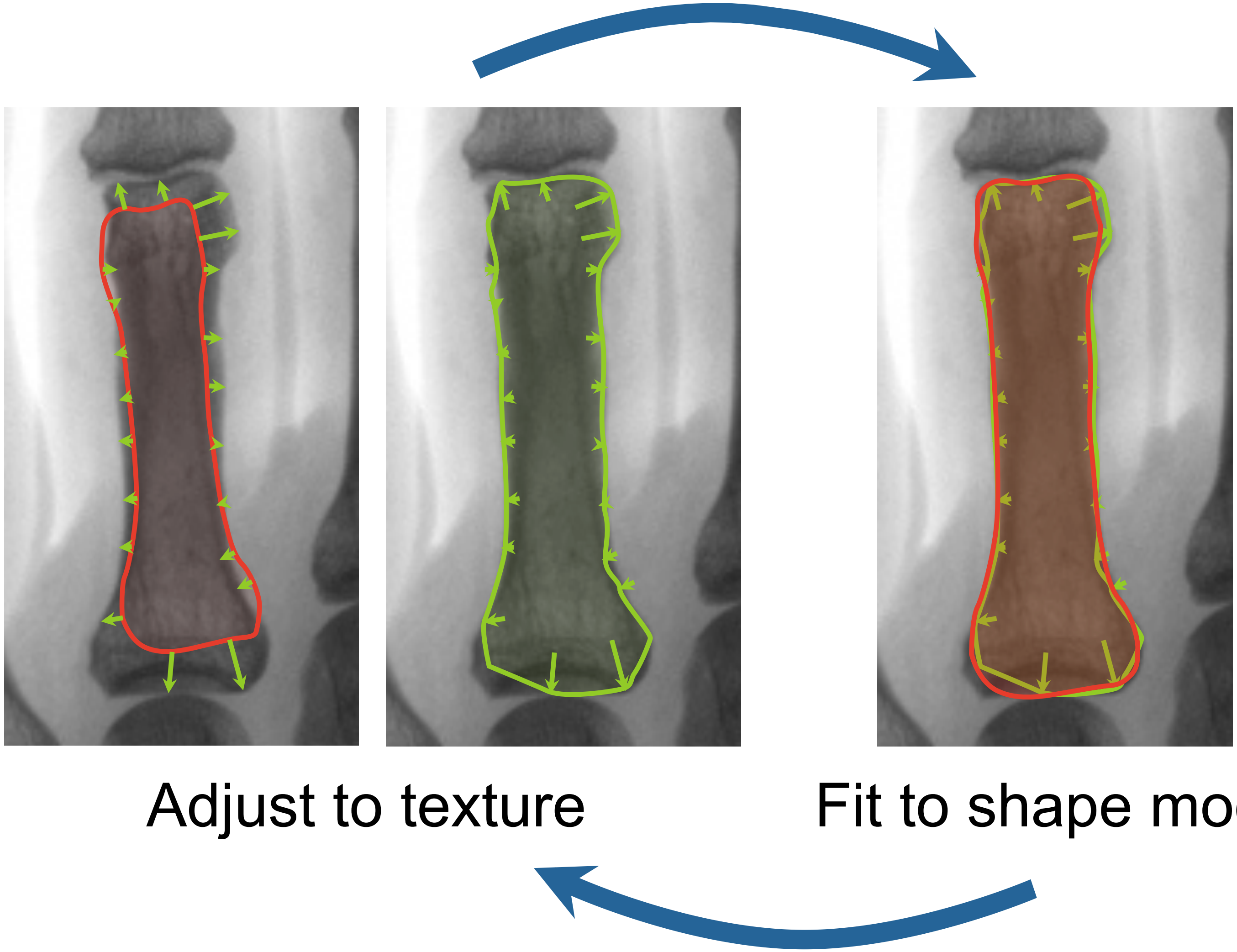


T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape **from sets of examples.** BMVC 1992

Slide credits: G. Langs

ASM search

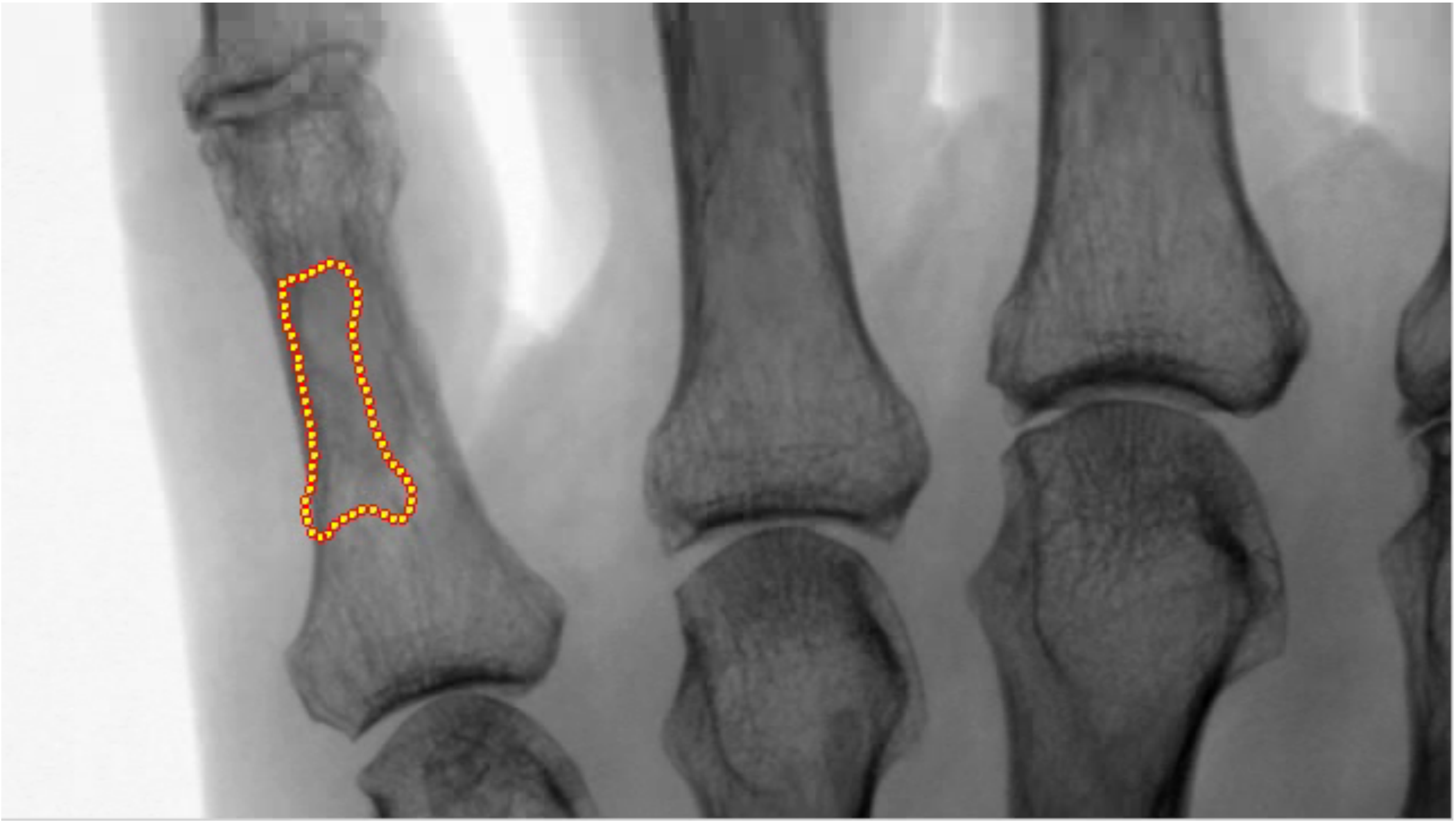
Initialize



Adjust to texture

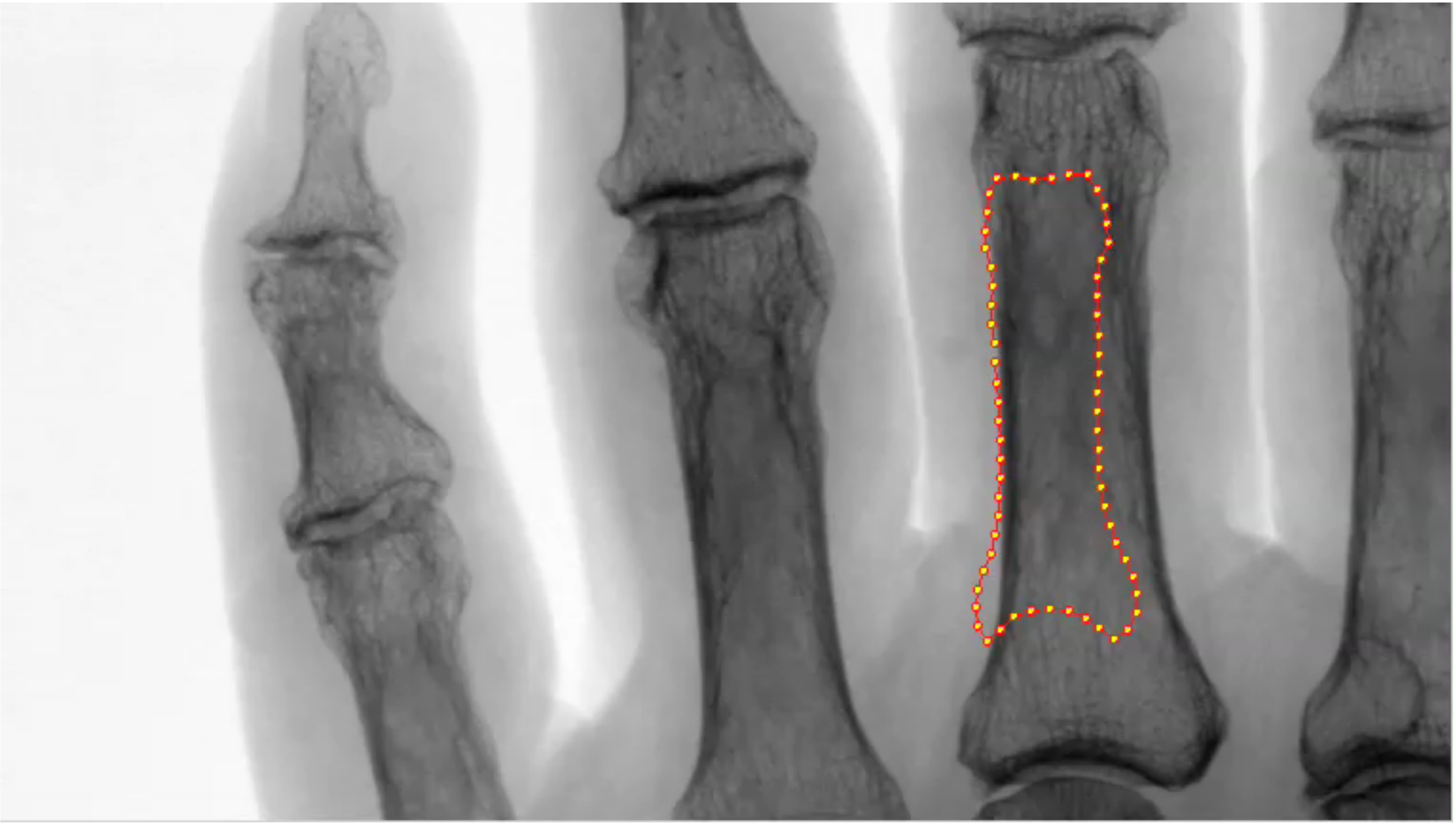
Fit to shape model

ASM search



Slide credit: G. Langs

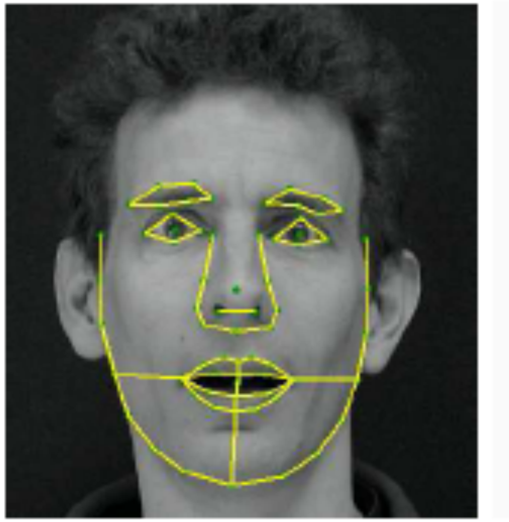
ASM search



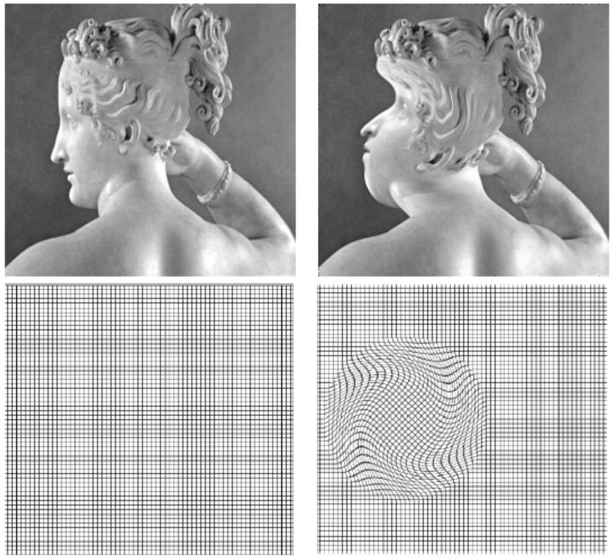
Slide credit: G. Langs

Three classes of deformable models

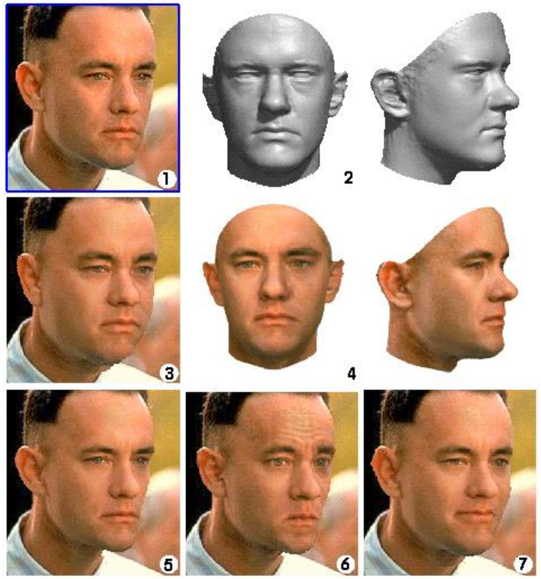
ASM



AAM



3D Morphable



Eigenfaces (Sirovich & Kirby 87, Turk & Pentland 91)

- Very few 100x100 vectors correspond to valid face images

- model the subspace (`manifold’) of face images

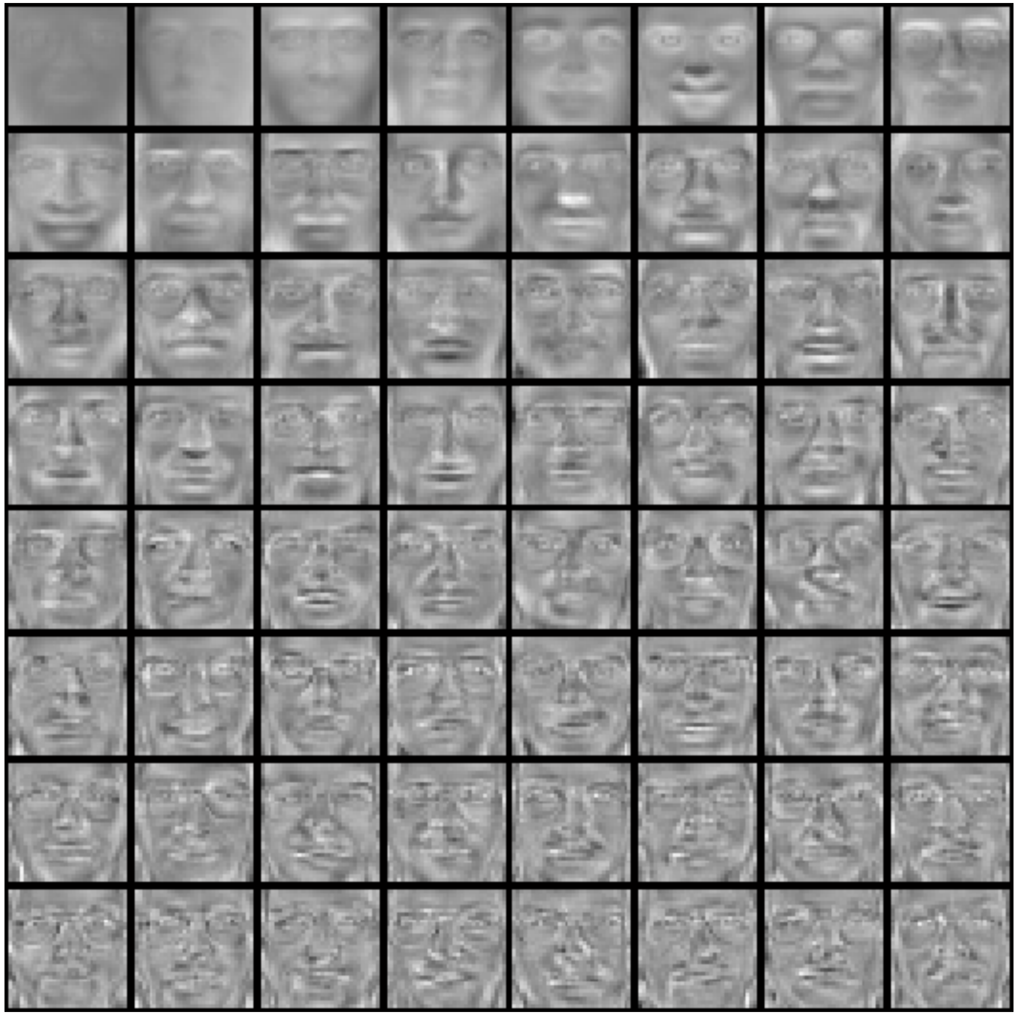
Eigenfaces

- Training images
- $x_1, \ldots, x_N$

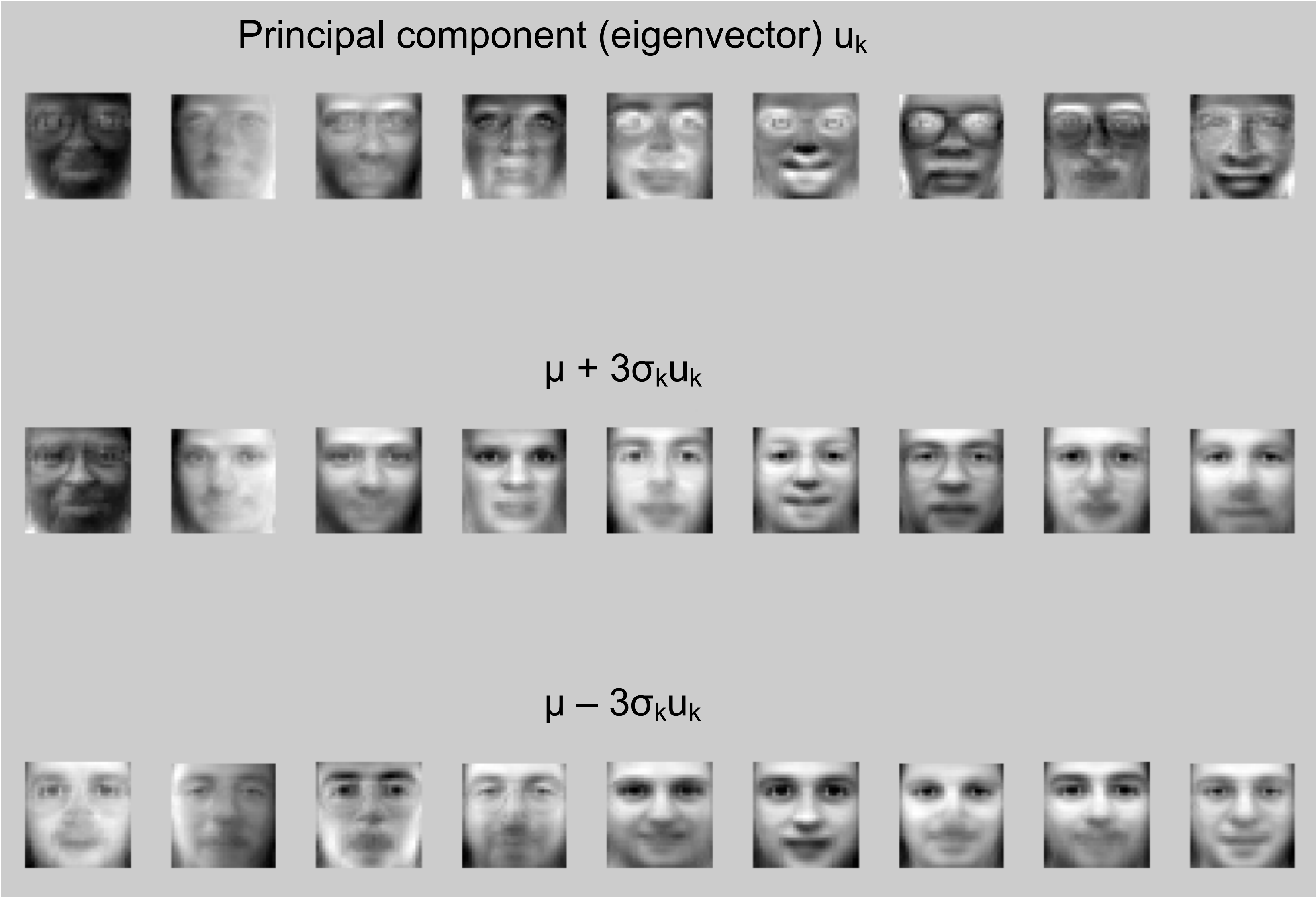
Eigenfaces

Mean: μ

Top eigenvectors: $u_1, \ldots u_k$



Eigenfaces


Principal component (eigenvector) u_k
$\mu + 3\sigma_k u_k$
$\mu - 3\sigma_k u_k$

Eigenfaces example

- Face x in “face space” coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x} - \mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x} - \mu)]$$
$$= \quad w_1, \ldots, w_k$$

- Reconstruction:

 =  + 

$\hat{x}$ = μ + $w_1u_1+w_2u_2+w_3u_3+w_4u_4+ \ldots$

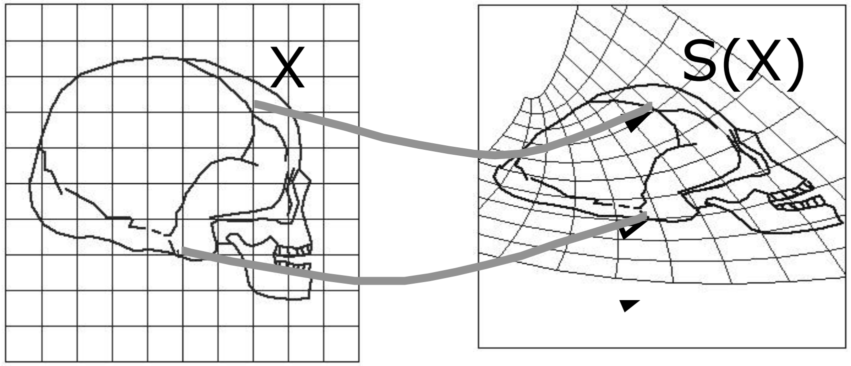
Limitations

- Global appearance method: not robust to misalignment, background variation

Appearance and Shape interpolation

Active Appearance Models (AAMs)

$$T(\mathbf{x}) \quad = \quad I(\mathcal{S}(\mathbf{x}))$$

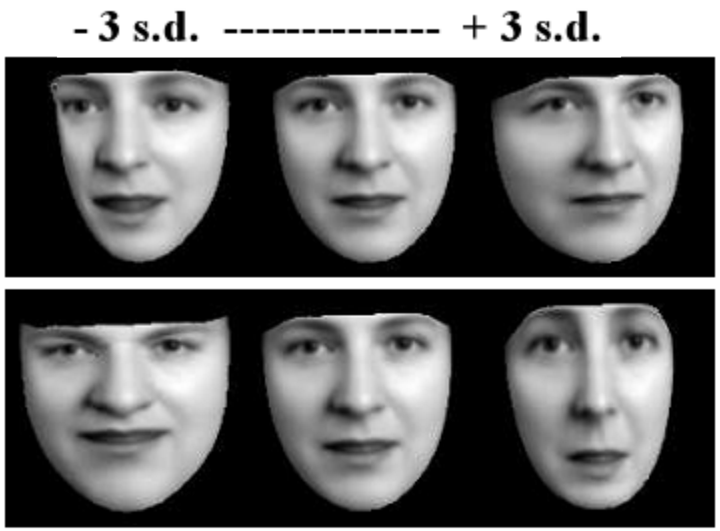


Shape: $$\mathcal{S}(\mathbf{x};\mathbf{s}) \quad = \quad \sum_i \mathbf{s}_i S_i(\mathbf{x}),$$

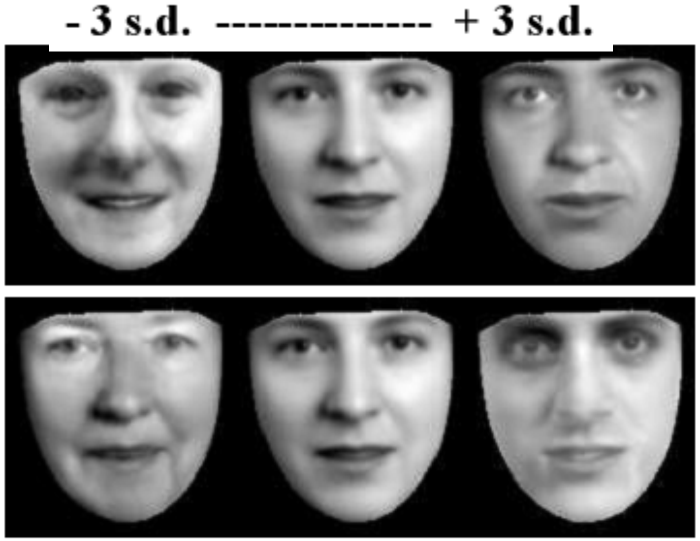
Appearance: $$\mathcal{T}(\mathbf{x};\mathbf{t}) \quad = \quad \sum_i \mathbf{t}_i T_i(\mathbf{x})$$

Synthesis: $$I(\mathcal{S}(\mathbf{x};\mathbf{s})) \quad \simeq \quad \mathcal{T}(\mathbf{x};\mathbf{t})$$

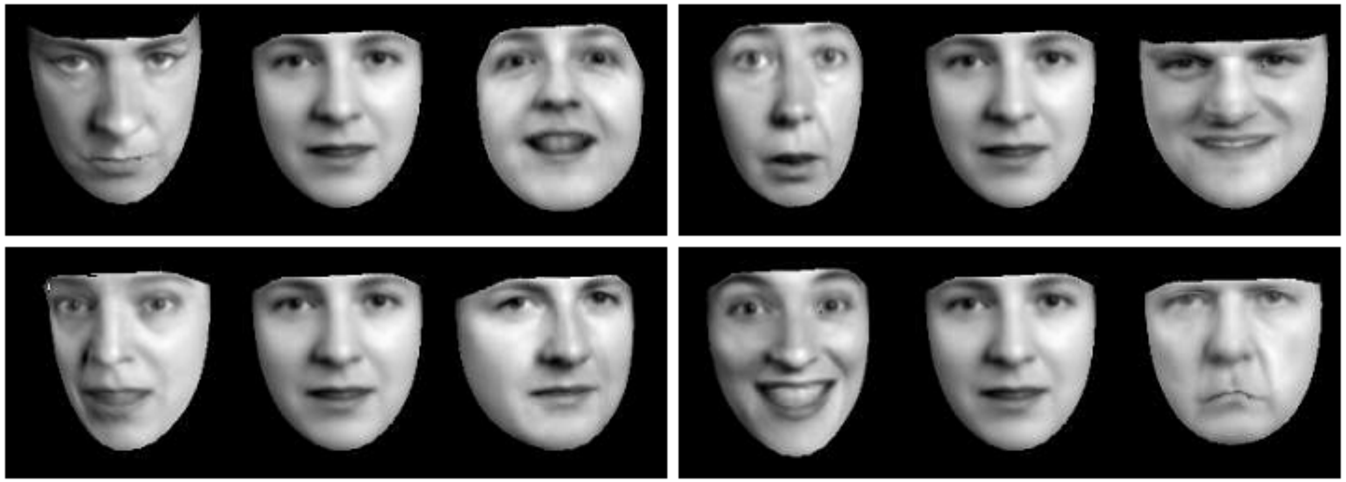
Active Appearance Models (AAMs)



First two modes of shape variation



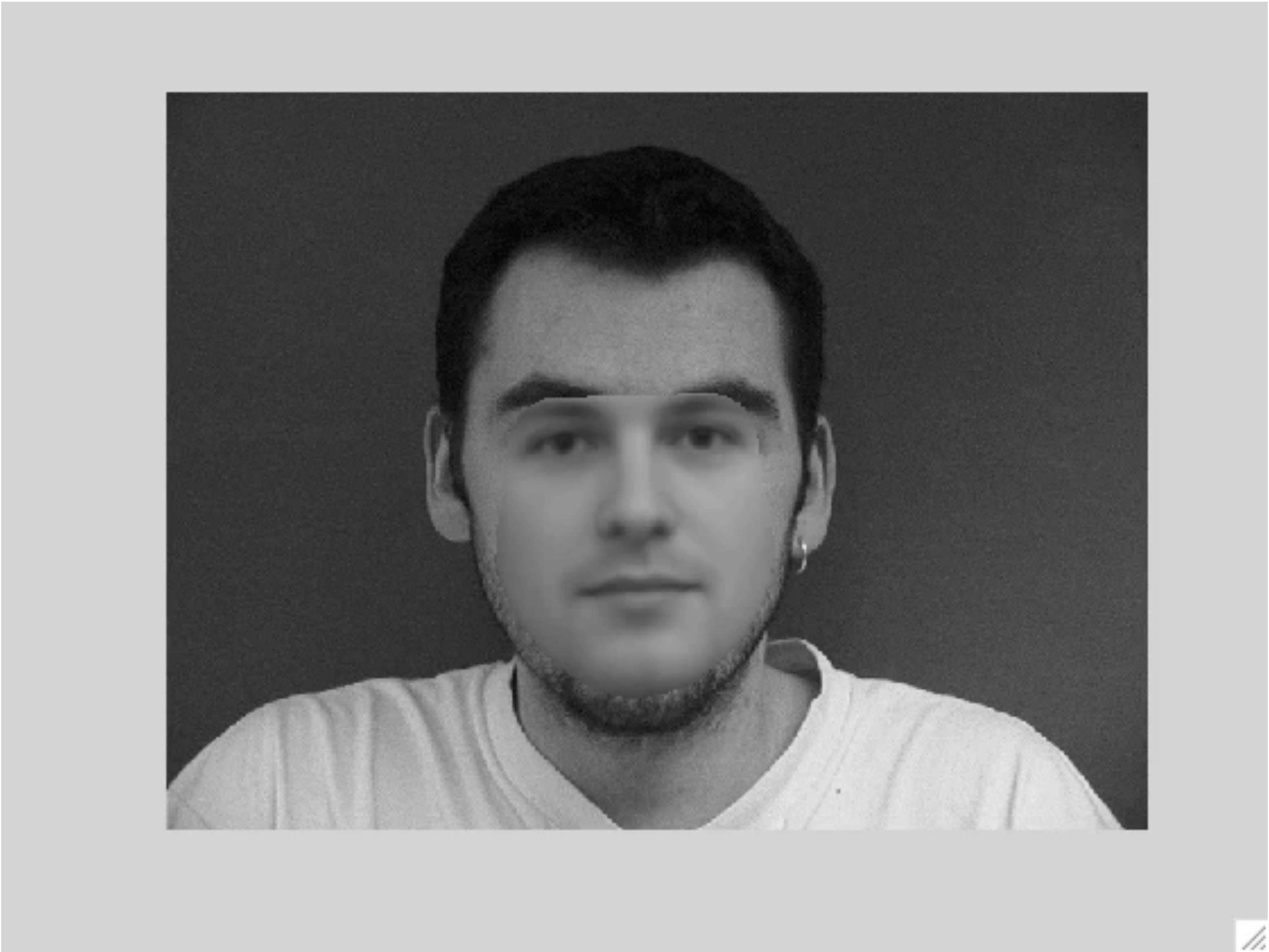
First two modes of gray-level variation



First four modes of appearance variation

Slide credits: Tim Cootes

AAM Search



Slide credits: Tim Cootes

Active Appearance Model Search (Results)

Initial 2 its 8 its 14 its 20 its converged

Slide credits: Tim Cootes

Active Appearance Model fitting

Given:
1) an appearance model,
2) a new image,
3) a starting approximation

Find:
the best matching synthetic image

- Minimize reconstruction error

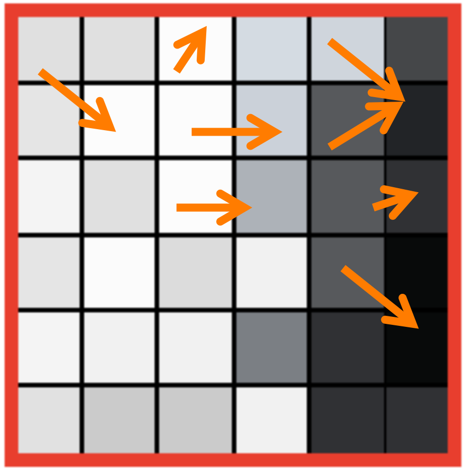
$$E(\mathbf{s},\mathbf{t}) = \sum_{\mathbf{x}} \left[I(\mathcal{S}(\mathbf{x};\mathbf{s})) - \mathcal{T}(\mathbf{x};\mathbf{t}) \right]^2$$

  - Alternate between estimating s and t
- For t: projection of deformed image $I(S(\mathbf{x};\mathbf{s}))$ onto PCA basis
- For s?

Reminder: Lucas-Kanade method

Brightness constancy constraint

$$I(x_i + u_i, y + v_i, t) = I(x_i, y_i, t + 1)$$



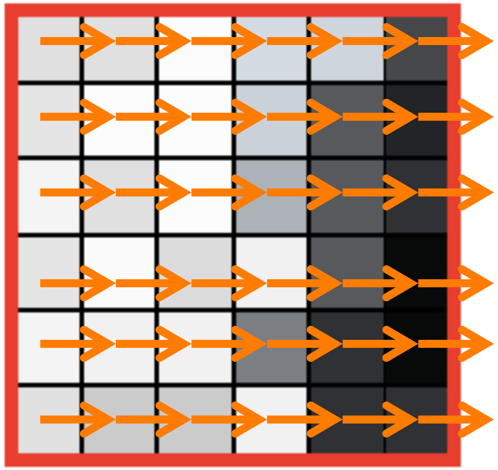
Linearization:

$$I_x(p_i)u_i + I_y(p_i)v_i + I_t(p_i) = 0$$

Reminder: Lucas-Kanade method

Brightness constancy constraint

$$I(x_i + u, y + v, t) = I(x_i, y_i, t+1)$$



Linearization:

$$I_x(p_i)u + I_y(p_i)v + I_t(p_i) = 0$$

From Lucas-Kanade to AAMs

Brightness constancy constraint

$$I(x_i + u, y + v, t) = I(x_i, y_i, t+1)$$

$$I(\mathbf{x}_i + u \cdot (\mathbf{1}, \mathbf{0}) + v \cdot (\mathbf{0}, \mathbf{1}), t) = I(\mathbf{x}_i, t+1)$$

AAM synthesis equation:

$$I_1(\mathbf{x}_i + \sum_k a_k \mathbf{b}_k(\mathbf{x}_i)) = I_2(\mathbf{x}_i)$$

I. Matthews and S. Baker, "Active Appearance Models Revisited," 2004.

AAM parameter estimation: shape

- Iterative scheme

$$E(\mathbf{s},\mathbf{t}) = \sum_{\mathbf{x}} \left[I(\mathcal{S}(\mathbf{x};\mathbf{s})) - \mathcal{T}(\mathbf{x};\mathbf{t})\right]^2$$

$$E(\mathbf{s}+\boldsymbol{\Delta}\mathbf{s},\mathbf{t}) = E(\mathbf{s},\mathbf{t}) + \mathcal{J}\boldsymbol{\Delta}\mathbf{s} + \frac{1}{2}\boldsymbol{\Delta}\mathbf{s}^T\mathcal{H}\boldsymbol{\Delta}\mathbf{s}$$

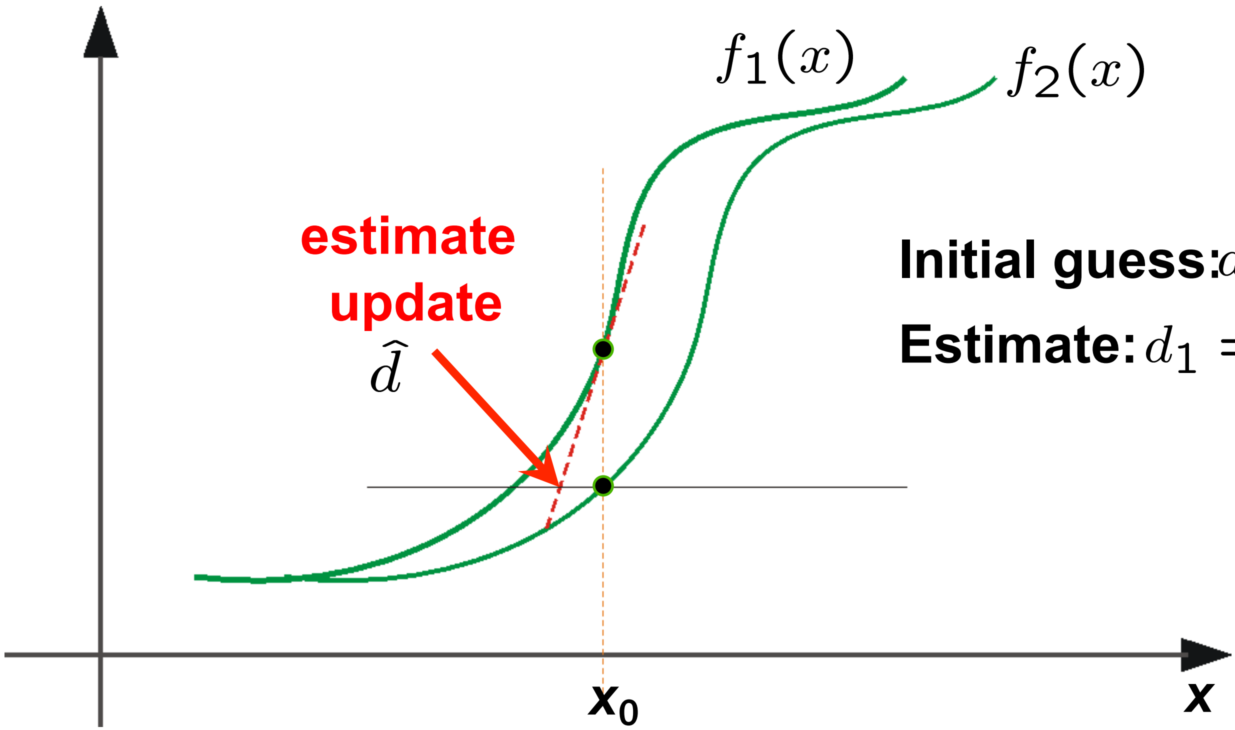
$$\Delta\mathbf{s}^* = -\mathcal{J}\mathcal{H}^{-1} \qquad \mathbf{s}' = \mathbf{s} - \mathcal{J}\mathcal{H}^{-1}$$

$$I(S(\mathbf{x};\mathbf{s}+\Delta\mathbf{s})) \simeq I(S(\mathbf{x};\mathbf{s})) + \sum_{i=1}^{N_S} \frac{dI}{d\mathbf{s}}_i(\mathbf{x};\mathbf{s})\Delta\mathbf{s}_i$$

$$\frac{dI}{d\mathbf{s}}_i(\mathbf{x};\mathbf{s}) = \frac{\partial I(S(\mathbf{x};\mathbf{s}))}{\partial x}\frac{\partial S_x}{\partial \mathbf{s}_i} + \frac{\partial I(S(\mathbf{x};\mathbf{s}))}{\partial y}\frac{\partial S_y}{\partial \mathbf{s}_i},$$

$$\mathcal{J}_i = \sum_x \left[I(S(\mathbf{x},\mathbf{s})) - T(\mathbf{x},\mathbf{t})\right]\frac{\mathbf{dI}}{\mathbf{ds_i}}(x) \qquad \mathcal{H}_{i,j} = \sum_x \frac{\mathbf{dI}}{\mathbf{ds_i}}(x)\frac{\mathbf{dI}}{\mathbf{ds_j}}(x)$$

Optical Flow: Iterative Estimation

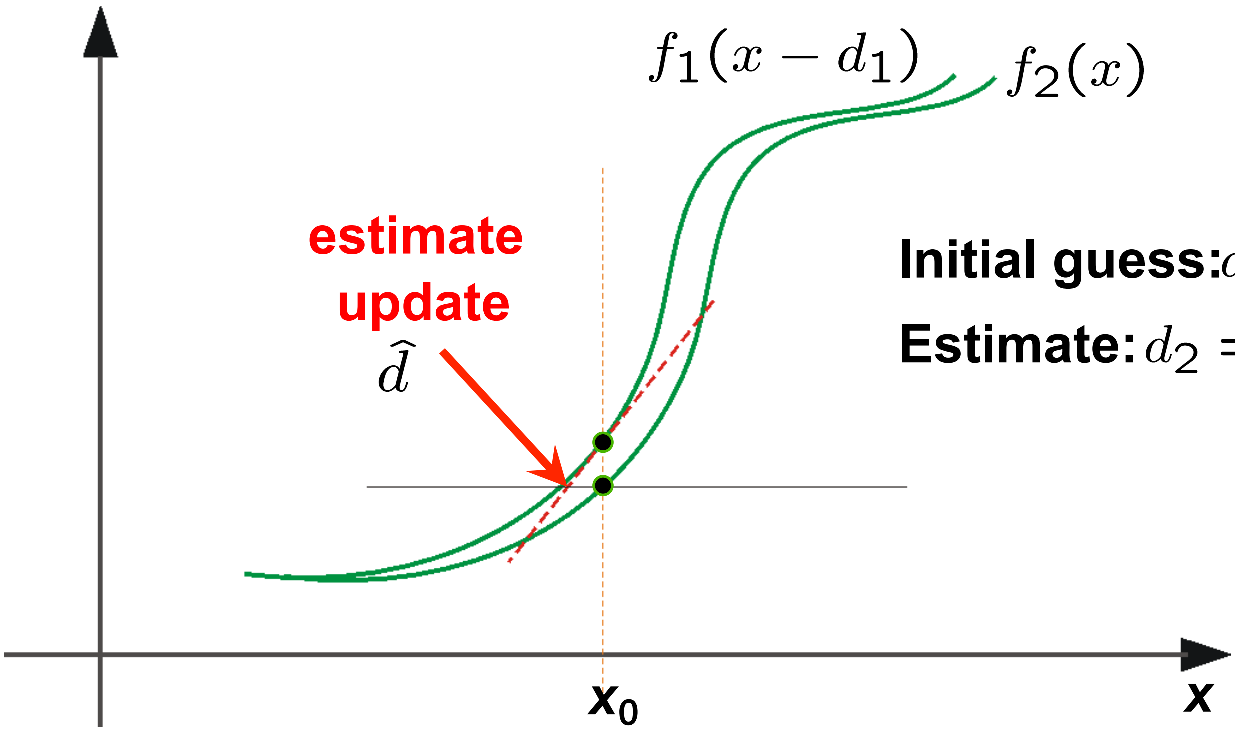


Initial guess: $d_0 = 0$

Estimate: $d_1 = d_0 + \hat{d}$

(using *d* for *displacement* here instead of *u*)

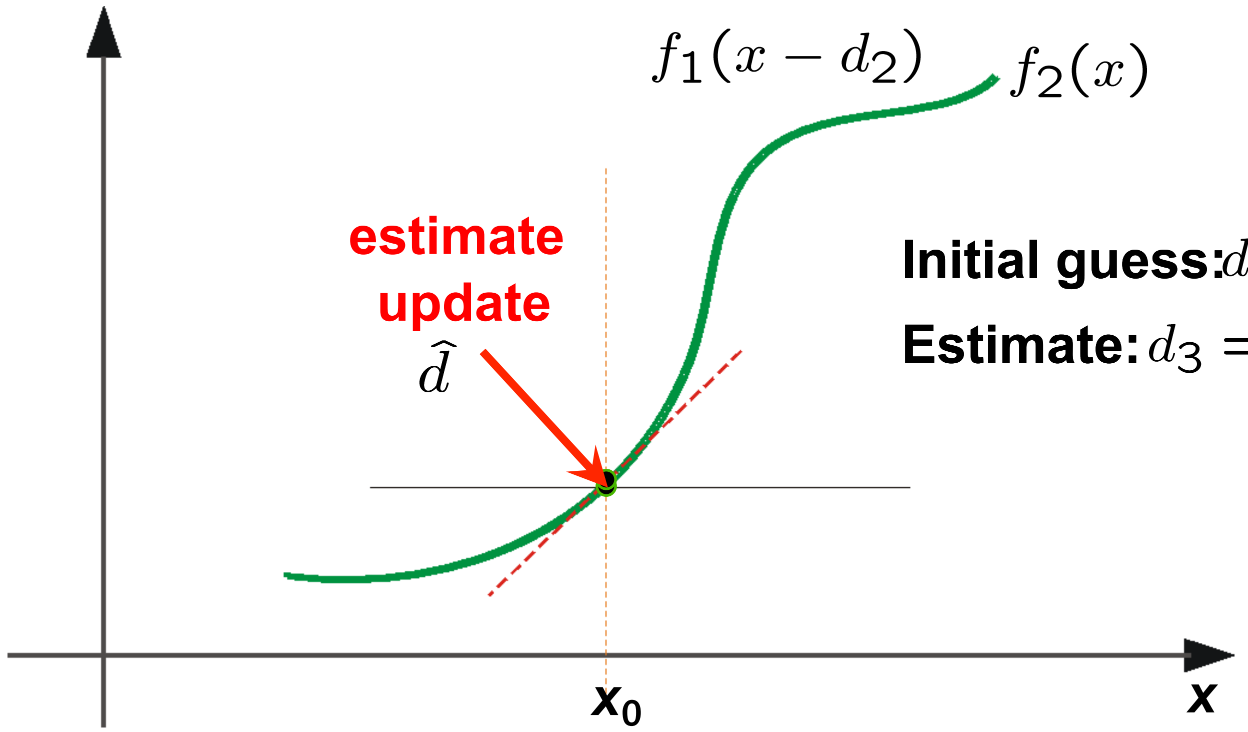
Optical Flow: Iterative Estimation



Initial guess: d_1

Estimate: $d_2 = d_1 + \hat{d}$

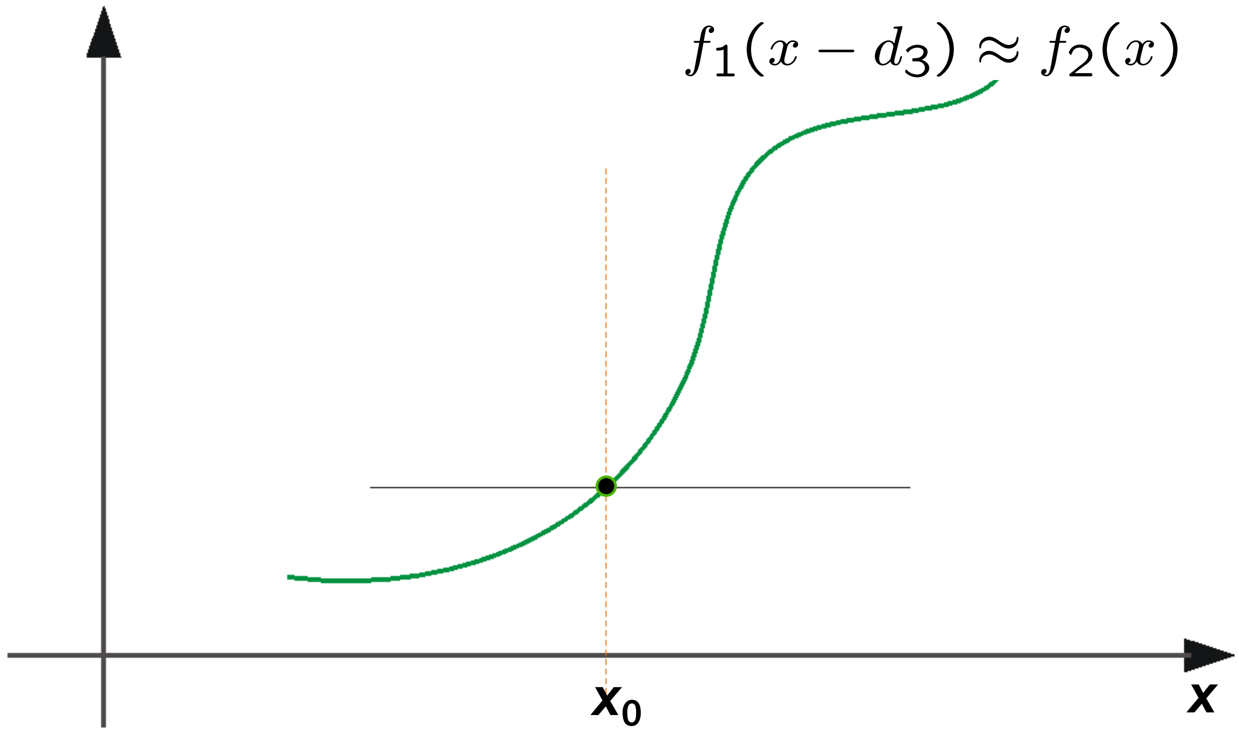
Optical Flow: Iterative Estimation



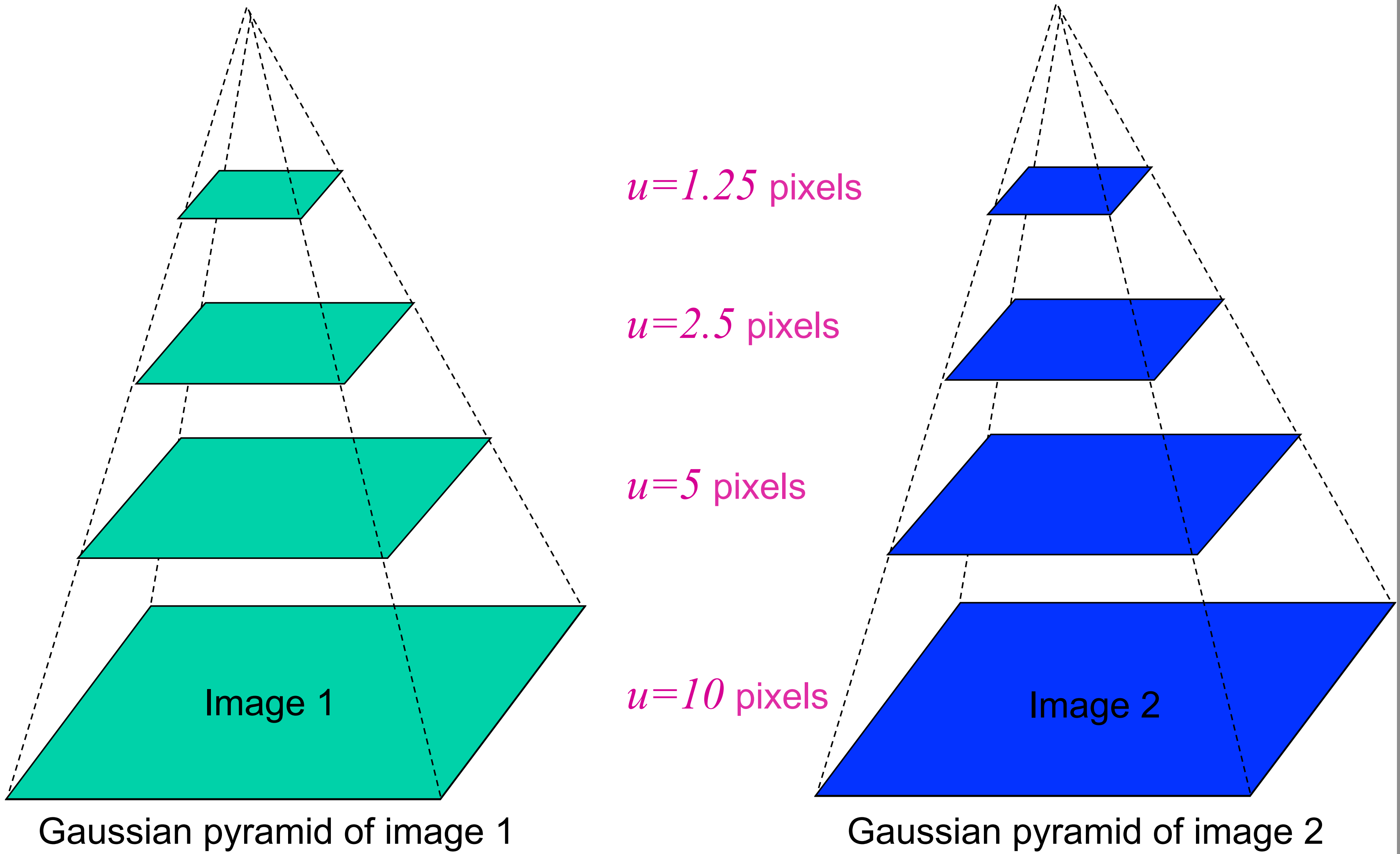
Initial guess: d_2

Estimate: $d_3 = d_2 + \hat{d}$

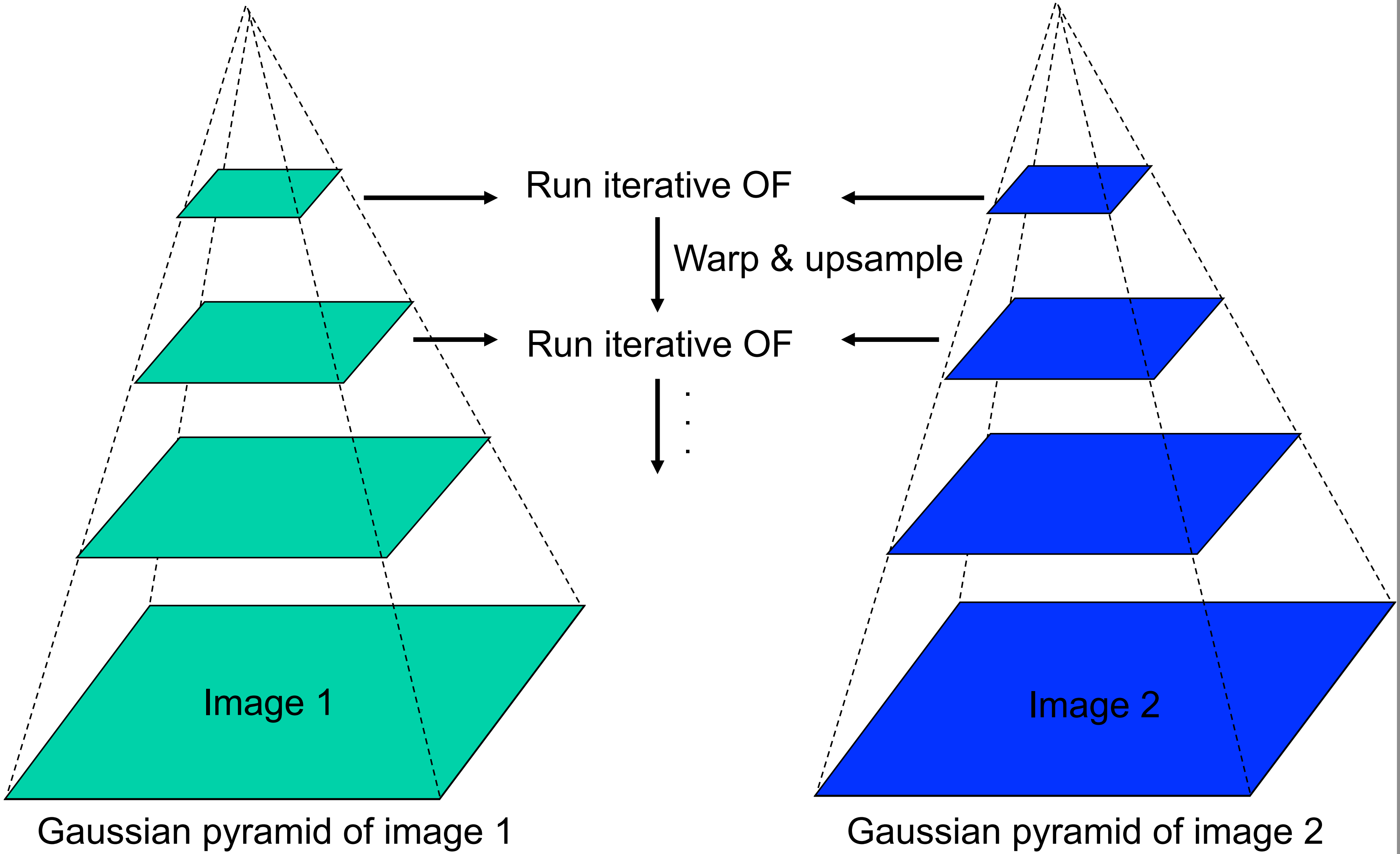
Optical Flow: Iterative Estimation



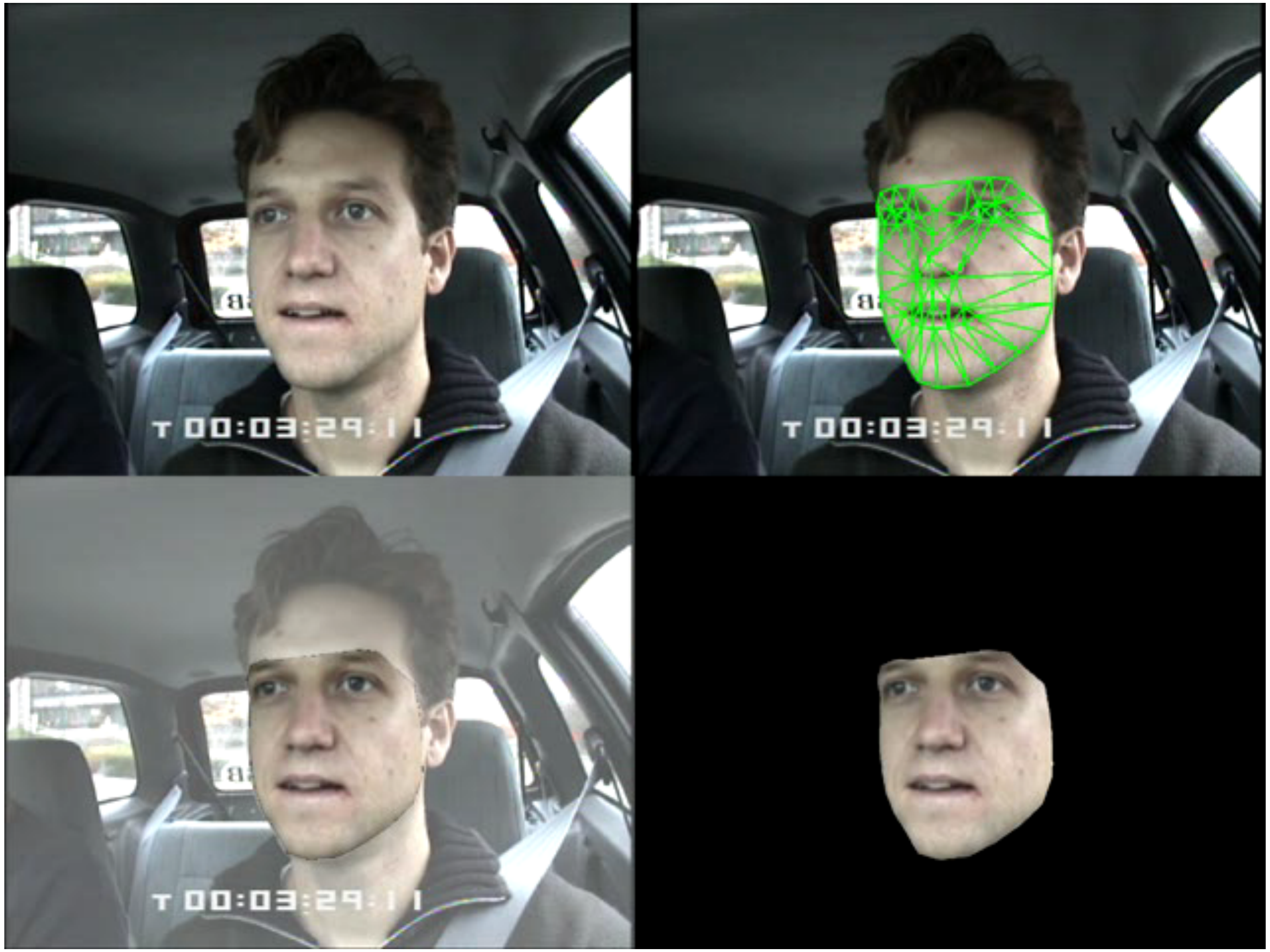
Coarse-to-fine Optical Flow Estimation



Coarse-to-fine Optical Flow Estimation

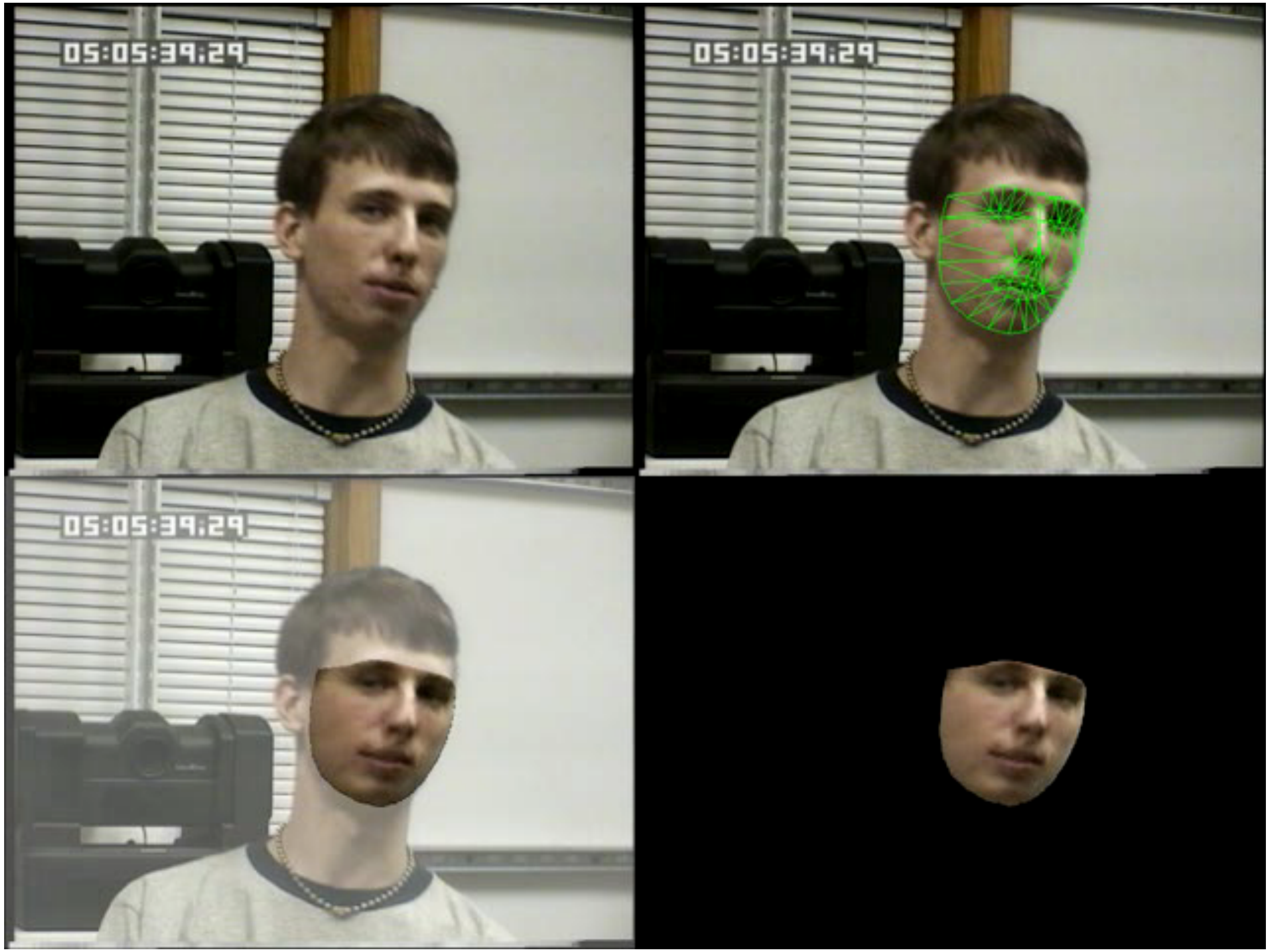


AAM for face tracking



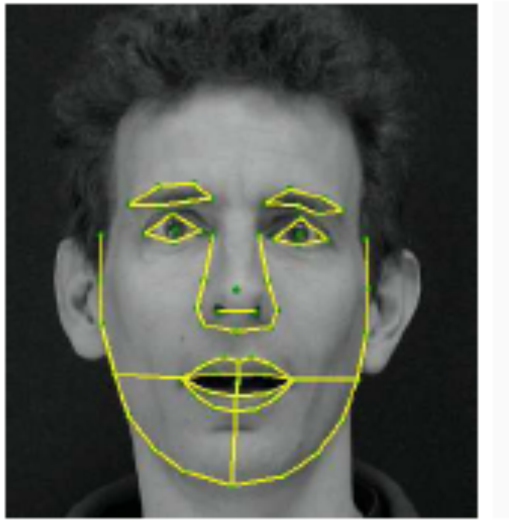
CMU group: I. Matthews, S. Baker, R. Gross
(230 Frames per second, 2004)

AAM for face tracking

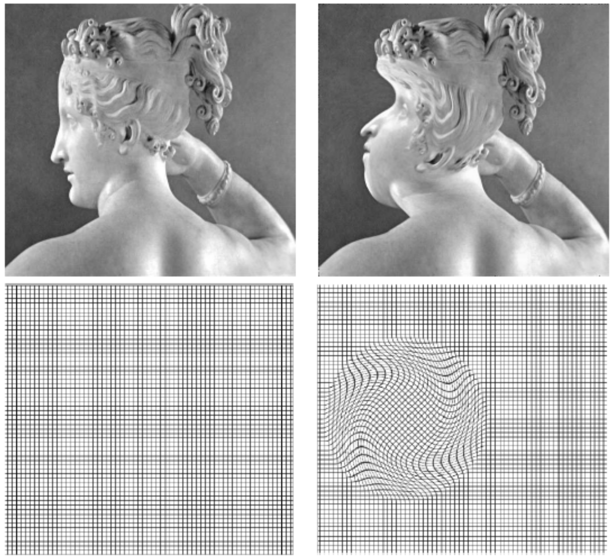


Three classes of deformable models

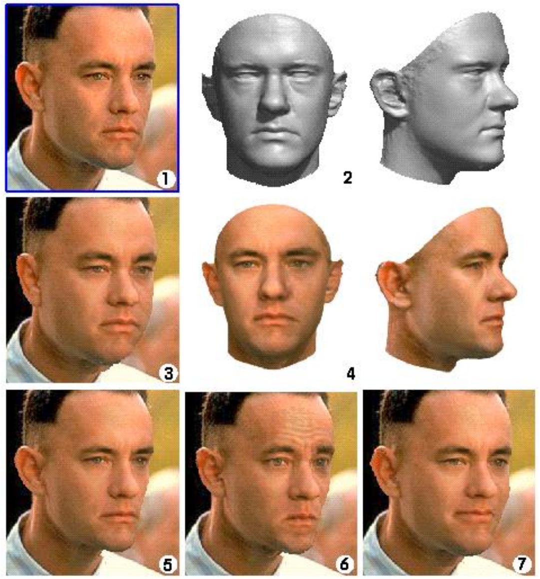
ASM



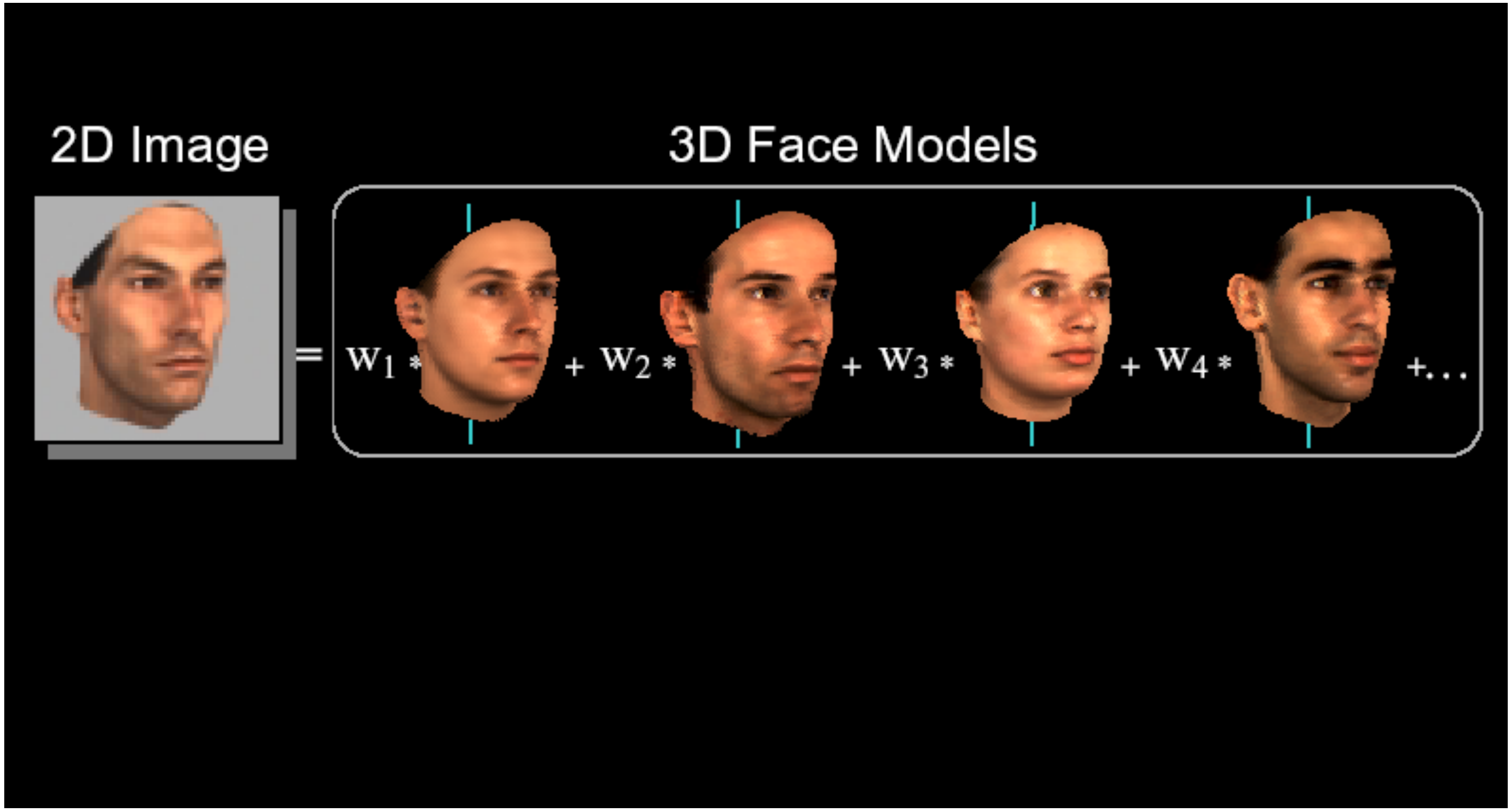
AAM



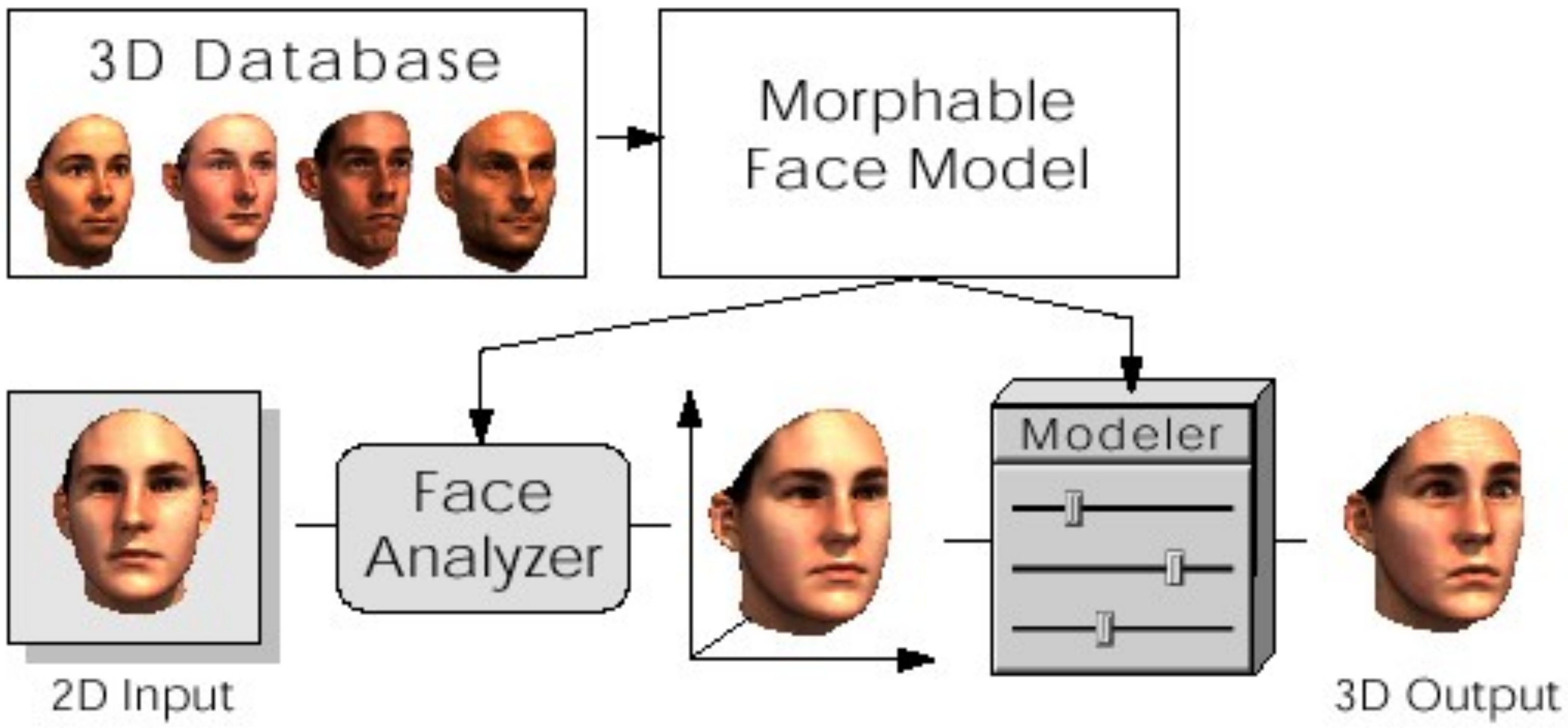
3D Morphable



Morphable Models: Blanz and Vetter



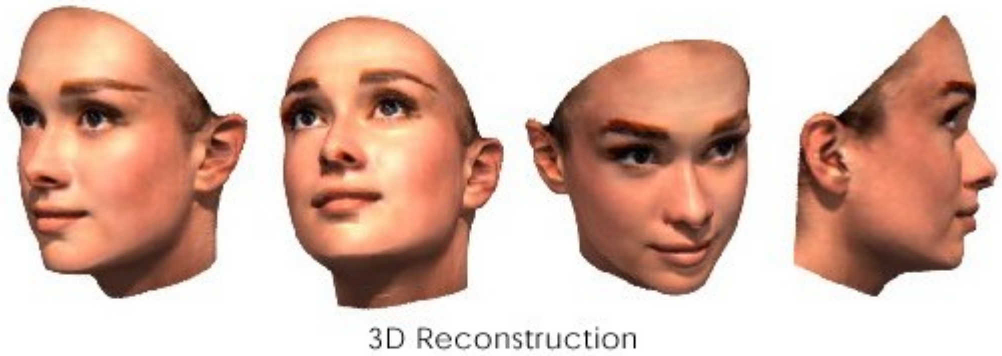
3-D Morphable Models



3-D Morphable Model fitting

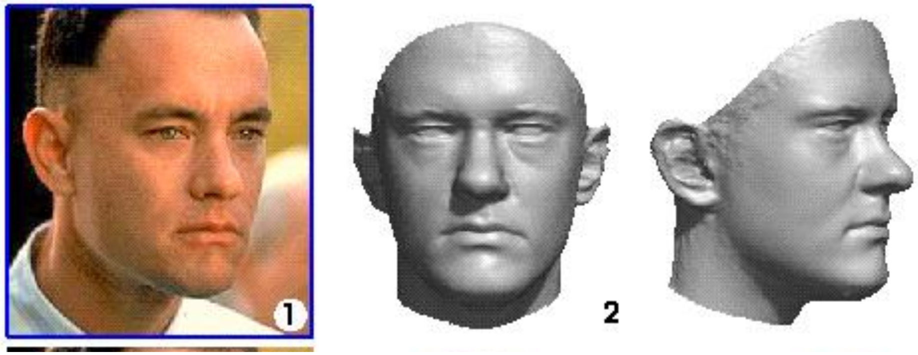
- Rough manual initialization
- Gradient descent to minimize reconstruction error functional

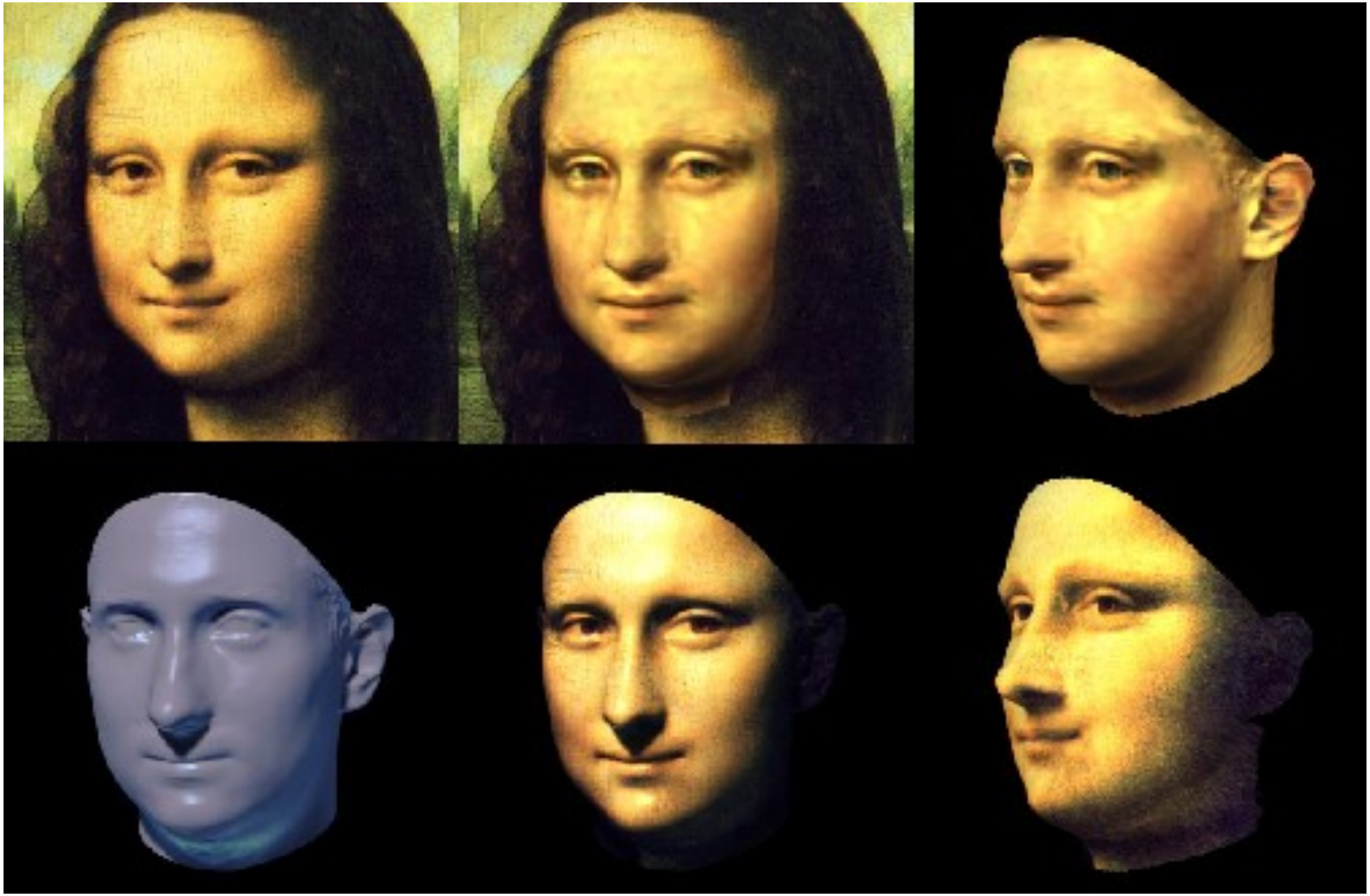
- And then

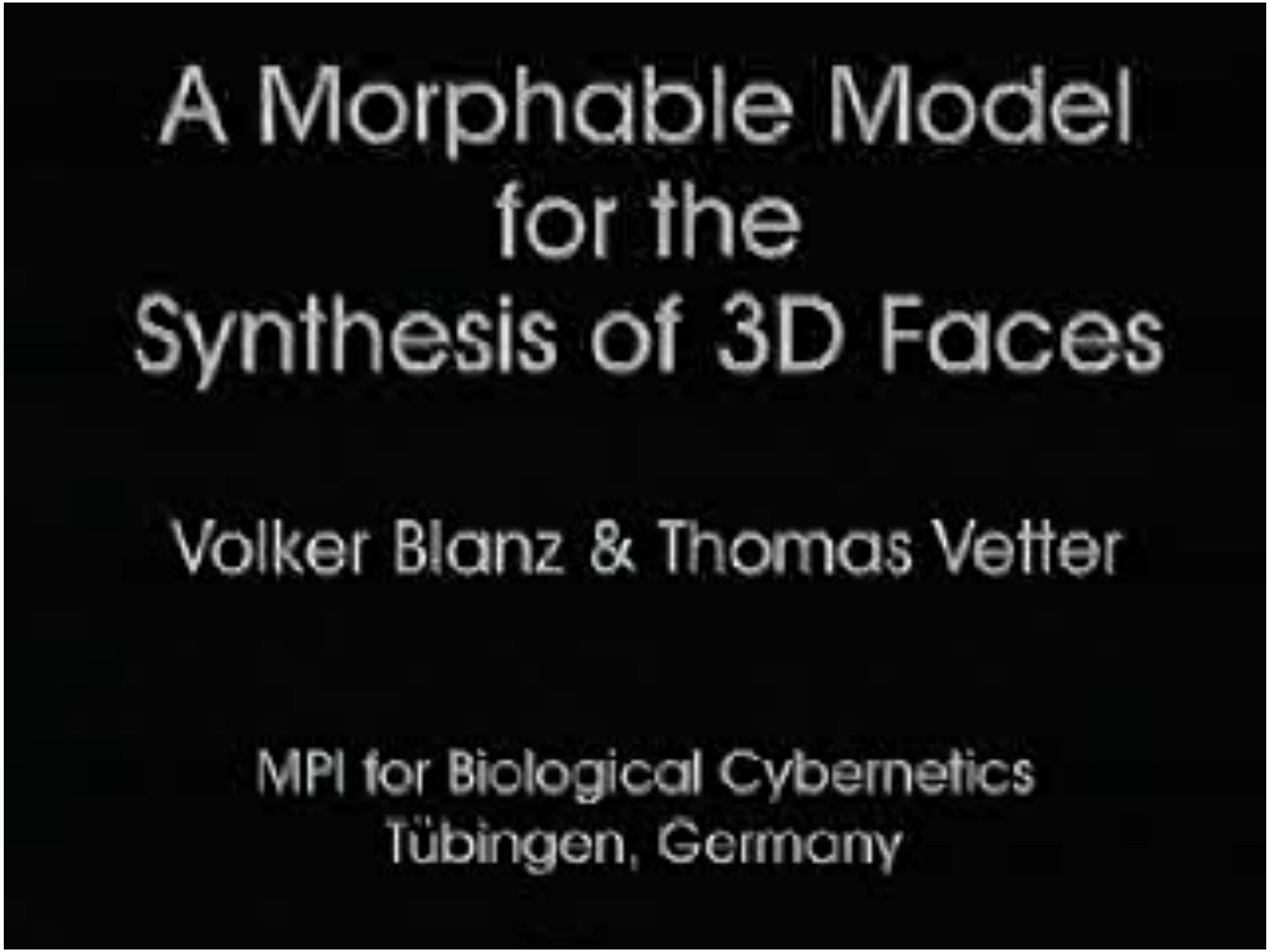


3D Morphable models

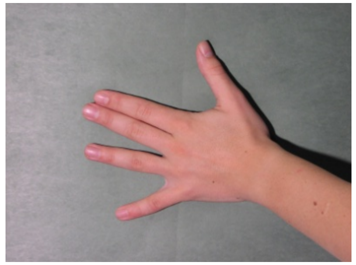
Recover Shape






A Morphable Model
for the
Synthesis of 3D Faces
Volker Blanz & Thomas Vetter
MPI for Biological Cybernetics
Tübingen, Germany

Unsupervised learning of deformable models

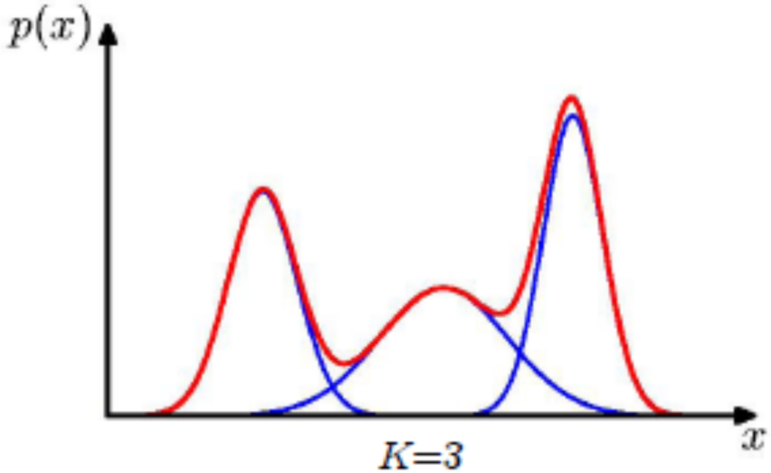
   

Mixture of Gaussians model

Combine simple models into a complex model:

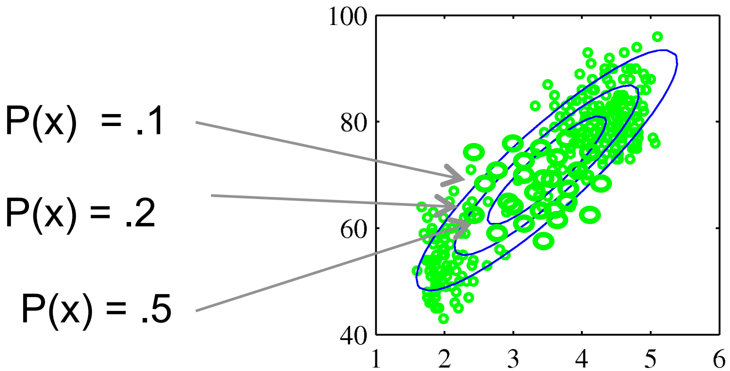
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \underbrace{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Component}}$$

Mixing coefficient

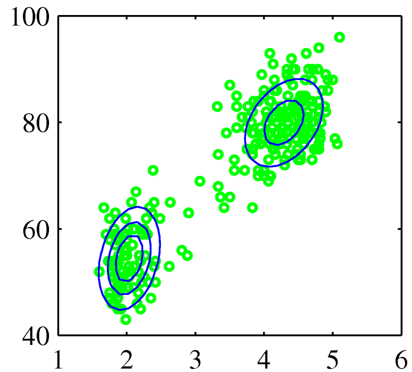


$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

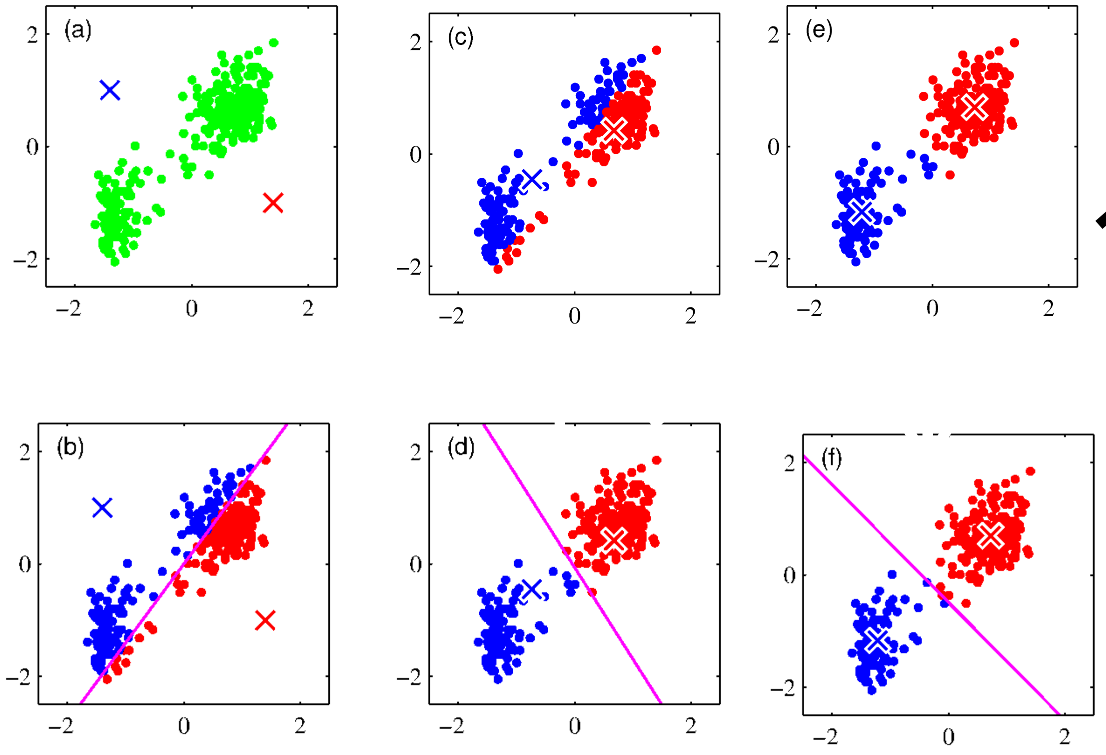
Main challenge: parameter estimation



Which points go with which cluster?

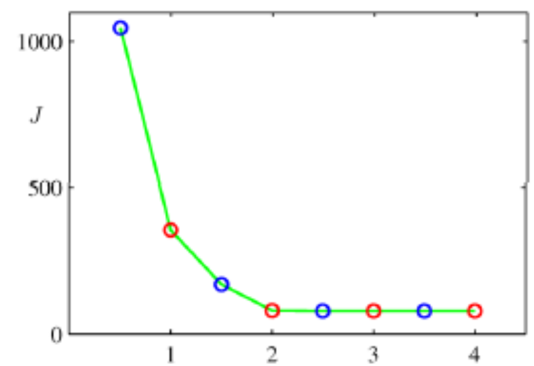


K-Means algorithm



$$c^j = \frac{\sum_{i=1}^{N} (m(i) = j) x^i}{\sum_{i=1}^{N} (m(i) = j)}$$

$$m(i) = \operatorname{argmin}_j |x^i - c^j|$$

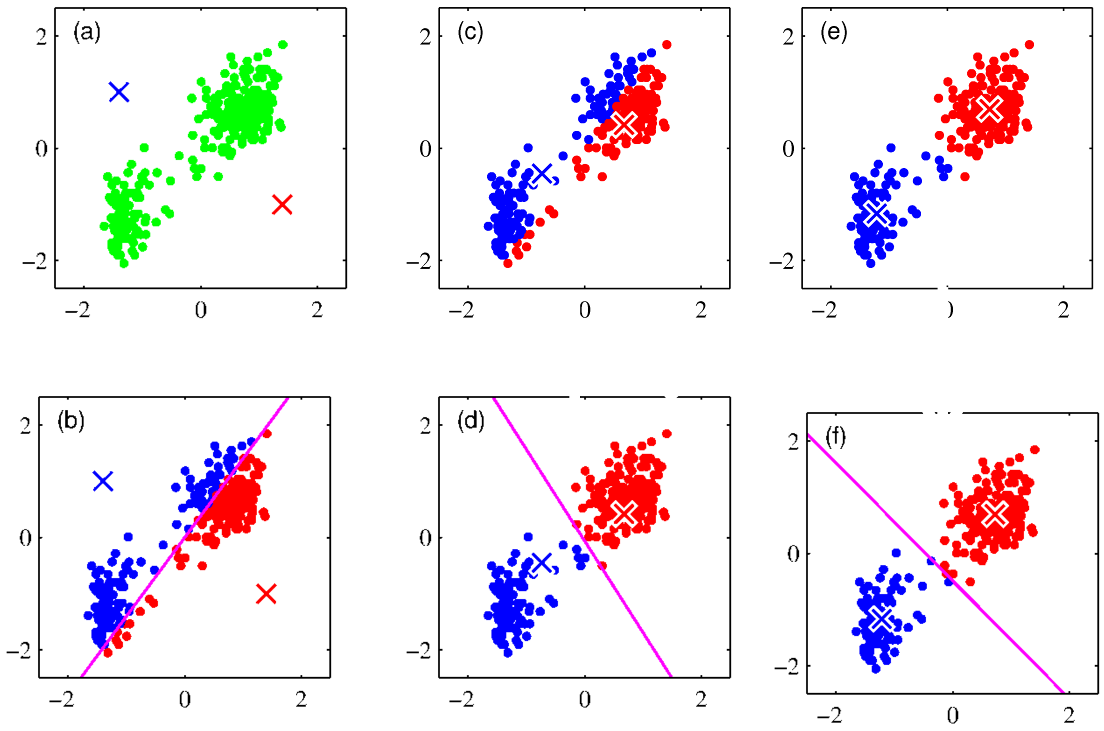


– Coordinate descent on distortion cost:

$$F(m,c) = \sum_{i=1}^{N} |x^i - c^{m(i)}|^2$$

– Local minima (multiple initializations to find better solution)

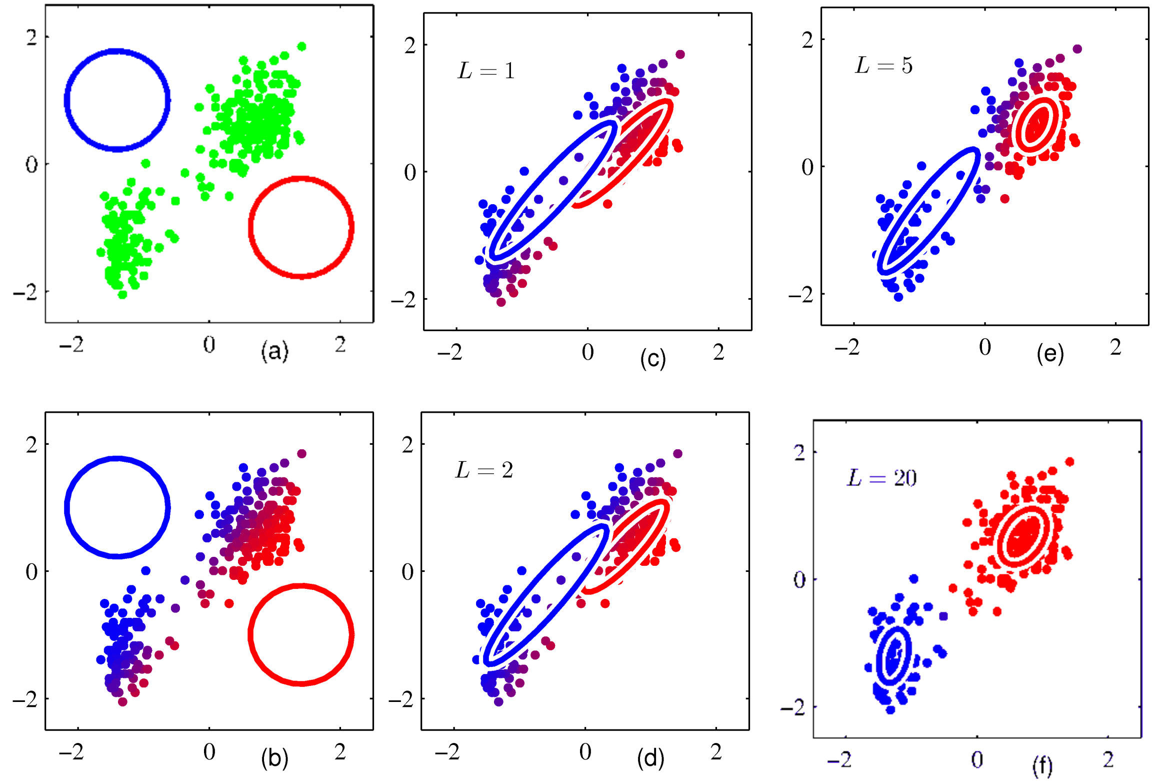
K-Means algorithm



$$c^j = \frac{\sum_{i=1}^{N}(m(i)=j)x^i}{\sum_{i=1}^{N}(m(i)=j)}$$

$$m(i) = \mathrm{argmin}_j |x^i - c^j|$$

Adaptation for Gaussian distributions

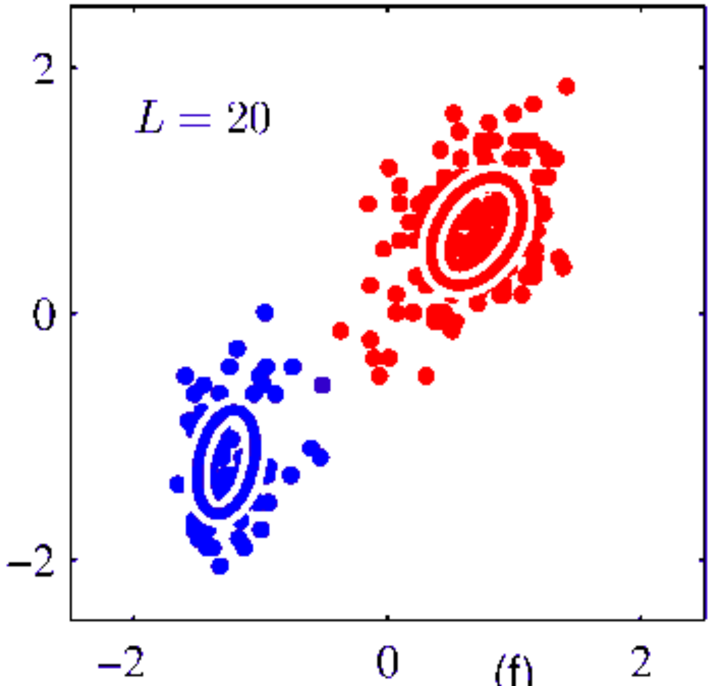


$$\mu^j = \frac{\sum_{i=1}^N R_{i,j} x^i}{\sum_{i=1}^N R_{i,j}}$$

$$\Sigma^j = \frac{\sum_{i=1}^N R_{i,j} (x^i - \mu^j)^T (x^i - \mu^j)}{\sum_{i=1}^N R_{i,j}}$$

$$R_{i,j} = P(z^i = j|\theta) = \frac{\pi_j P(x^i|\mu_j, \Sigma_j)}{\sum_{k=1}^K P(x^i|\mu_k, \Sigma_k)\pi_k}$$

Expectation Maximization algorithm



(f)

E-step

$$R_{i,j} = P(z^i = j|\theta)$$

$$= \frac{\pi_j P(x^i|\mu_j, \Sigma_j)}{\sum_{k=1}^{K} P(x^i|\mu_k, \Sigma_k)\pi_k}$$

M-step

$$\mu^j = \frac{\sum_{i=1}^{N} R_{i,j} x^i}{\sum_{i=1}^{N} R_{i,j}}$$

$$\pi^j = \frac{\sum_{i=1}^{N} R_{i,j}}{N}$$

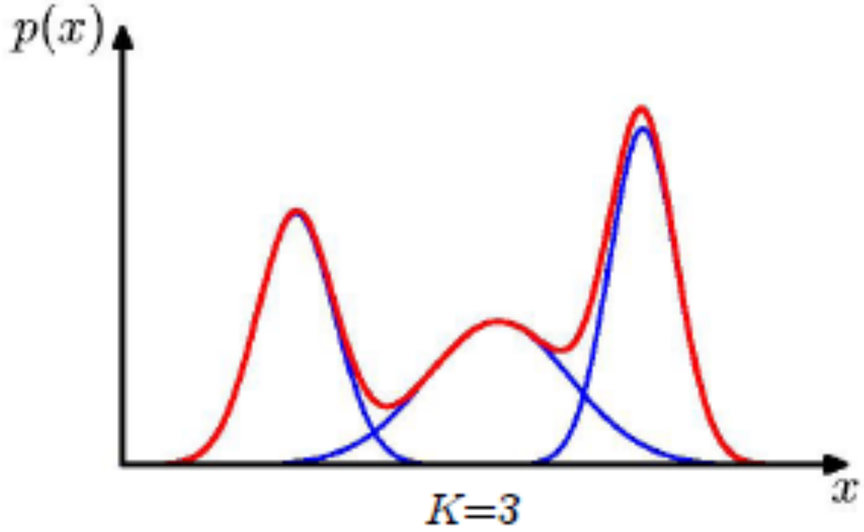
$$\Sigma^j = \frac{\sum_{i=1}^{N} R_{i,j} (x^i - \mu^j)^T (x^i - \mu^j)}{\sum_{i=1}^{N} R_{i,j}}$$

Mixture of Gaussians

- Combine simple models into a complex model:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \underbrace{\pi_k}_{\text{Mixing coefficient}} \underbrace{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Component}}$$

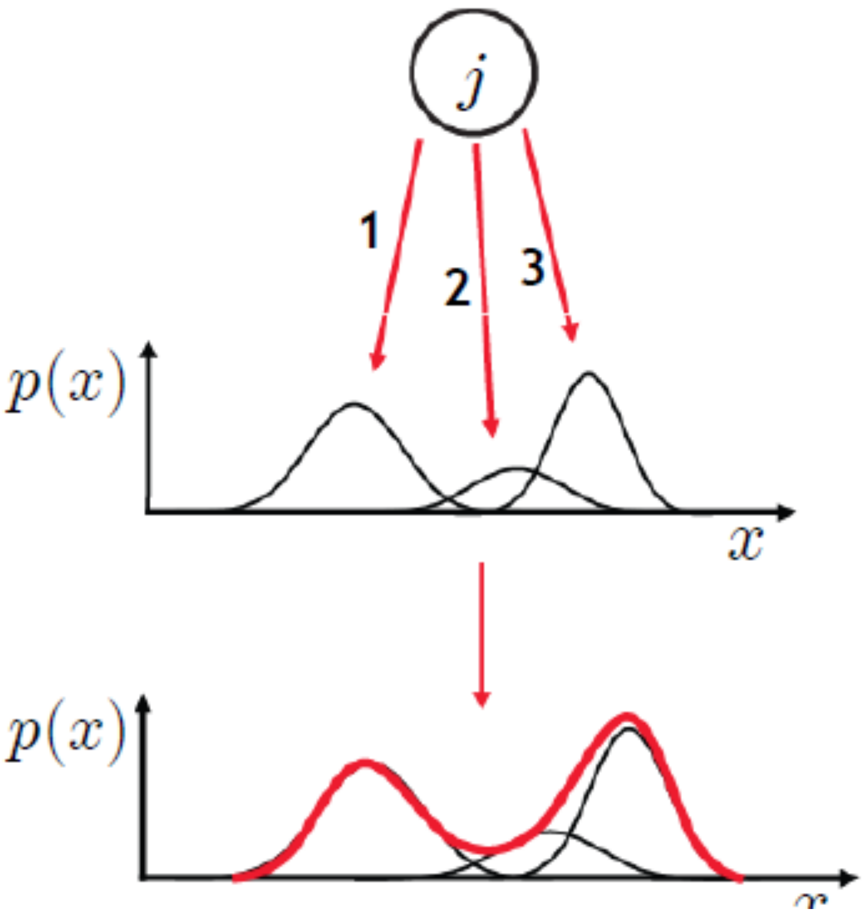
$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$



- Maximum Likelihood Estimation: $P(x|\theta) = \prod_{i=1}^{N} P(x^i|\theta)$

$$l(\theta; x) = \log P(x|\theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} P(x^i|\mu_k, \Sigma_k)\pi_k$$

- "Generative model"



$$p(j) = \pi_j$$

"Weight" of mixture component

$$p(x|\theta_j)$$

Mixture component

Mixture density

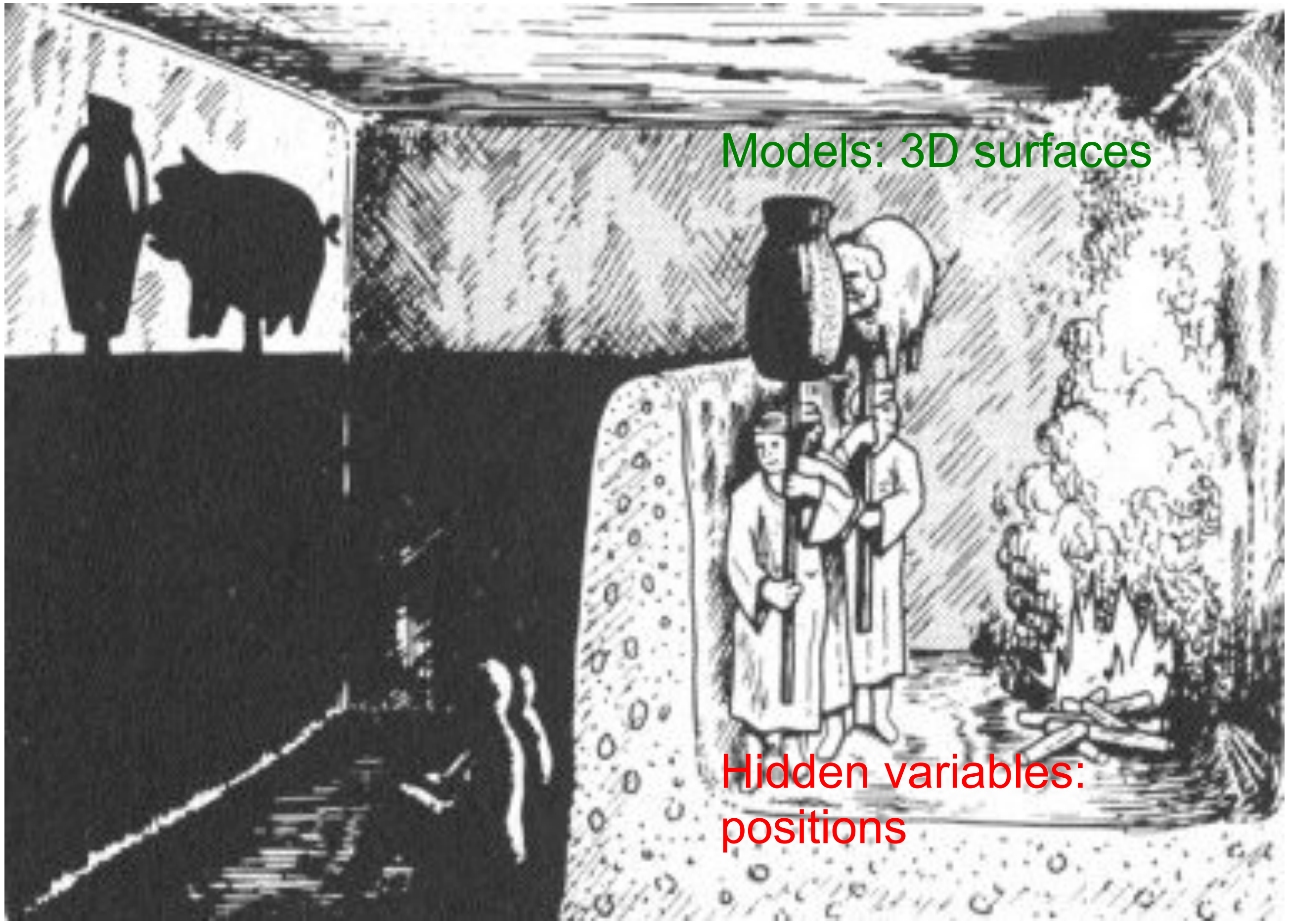
$$p(x|\theta) = \sum_{j=1}^{M} p(x|\theta_j)p(j)$$

Hidden Variables:

- Criterion: $l(\theta; x) = \log P(x|\theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} P(x^i|\mu_k, \Sigma_k)\pi_k$
  - Problem: Summation inside logarithm
  - We do not know which component generated each point
  - What if we knew?

Plato’s cave

Observations:
B&W Images

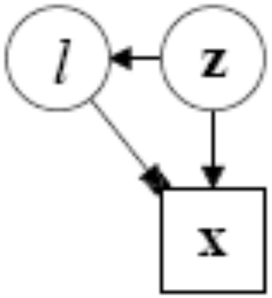


Transformation-invariant image averaging

Consider shift as a hidden variable, l

Estimate model with EM

Shift $p(\mathbf{z})$ Deformation-free image

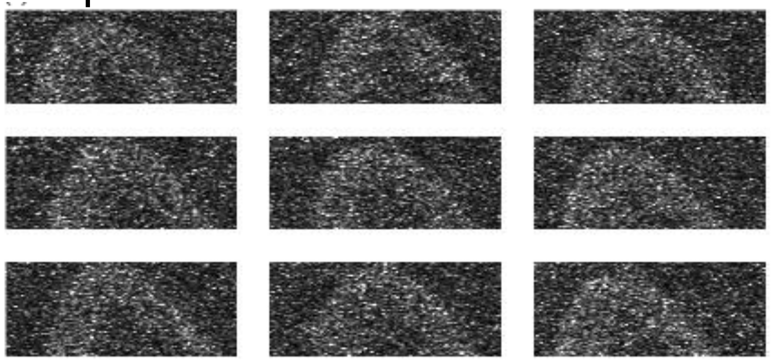


$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Phi}),$$

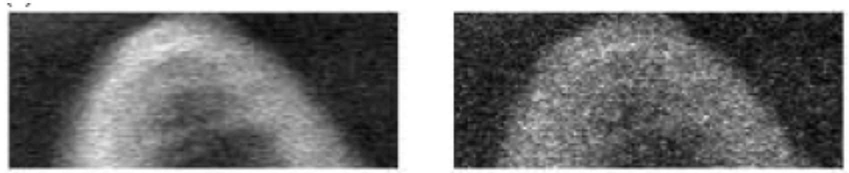
Observed Image

$$p(\mathbf{x}, \ell, \mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{G}_\ell \mathbf{z}, \boldsymbol{\Psi}) \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Phi}) \rho_\ell$$

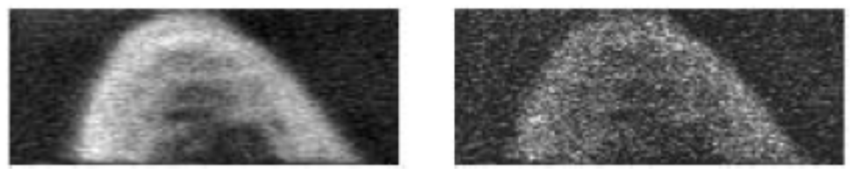
Input



Plain mean & std

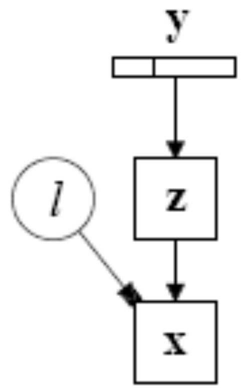


With transformation & EM



Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, 2003

Transformed Components Analysis



Latent variables for synthesis (continuous)

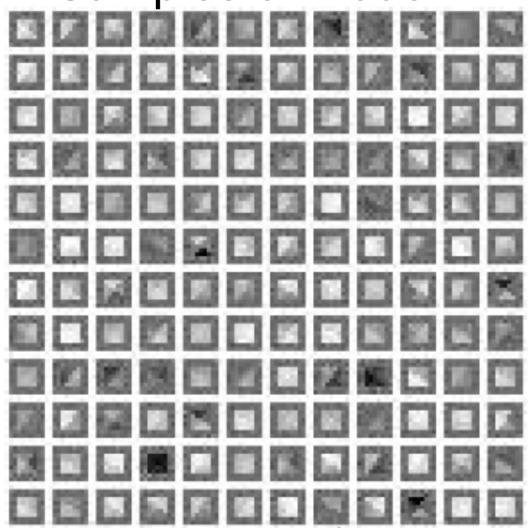
Latent variables for shift (discrete)

Estimate mean basis using EM

Plain mean & PCA

With offset

Samples of model

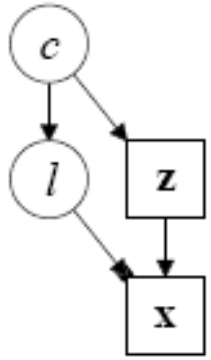


Input

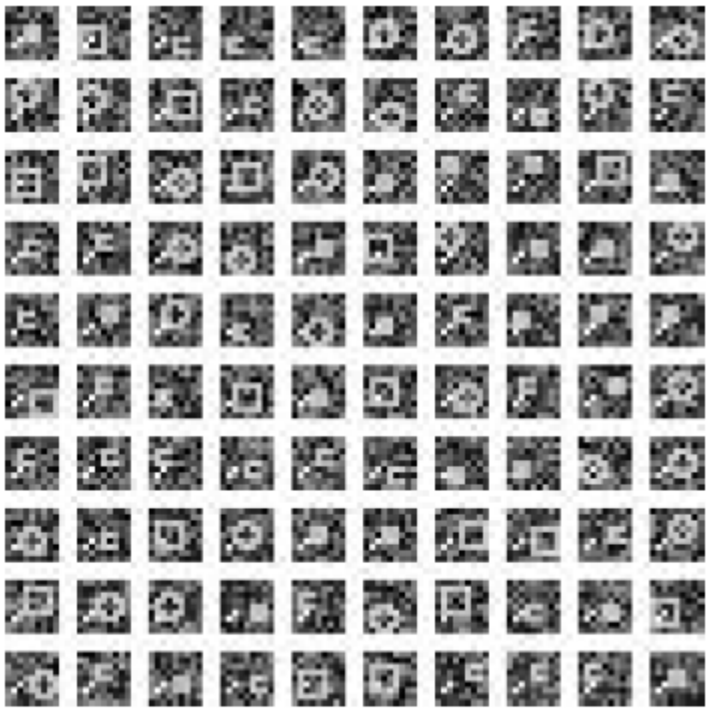
Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, PAMI 2003

Transformed Mixture of Gaussians

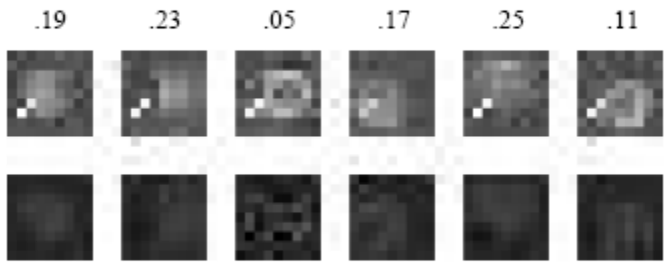
Latent variables for cluster (discrete)
Latent variables for shift (discrete)



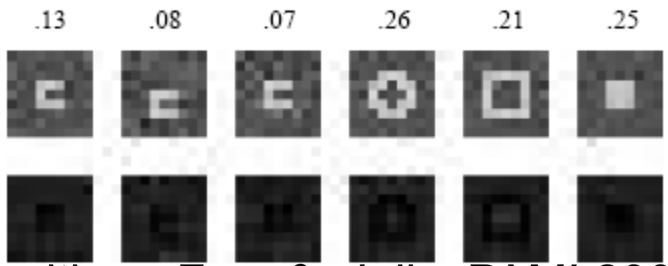
Input



Plain Mixture-of-Gaussians

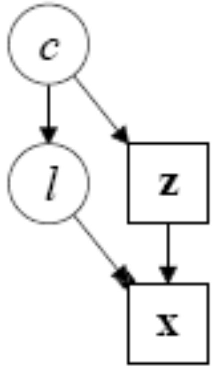


With offset



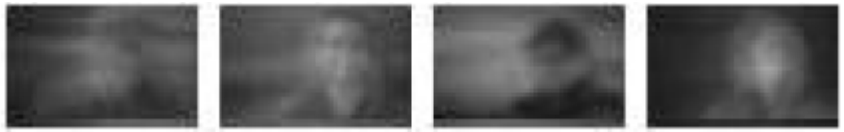
Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, PAMI 2003

Transformed Mixture of Gaussians



Input

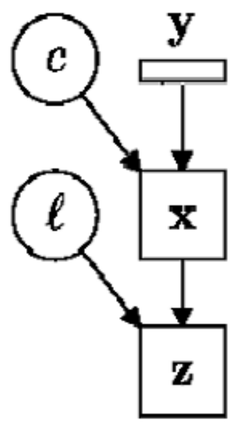
Plain Mixture-of-Gaussians



With offset

Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, PAMI 2003

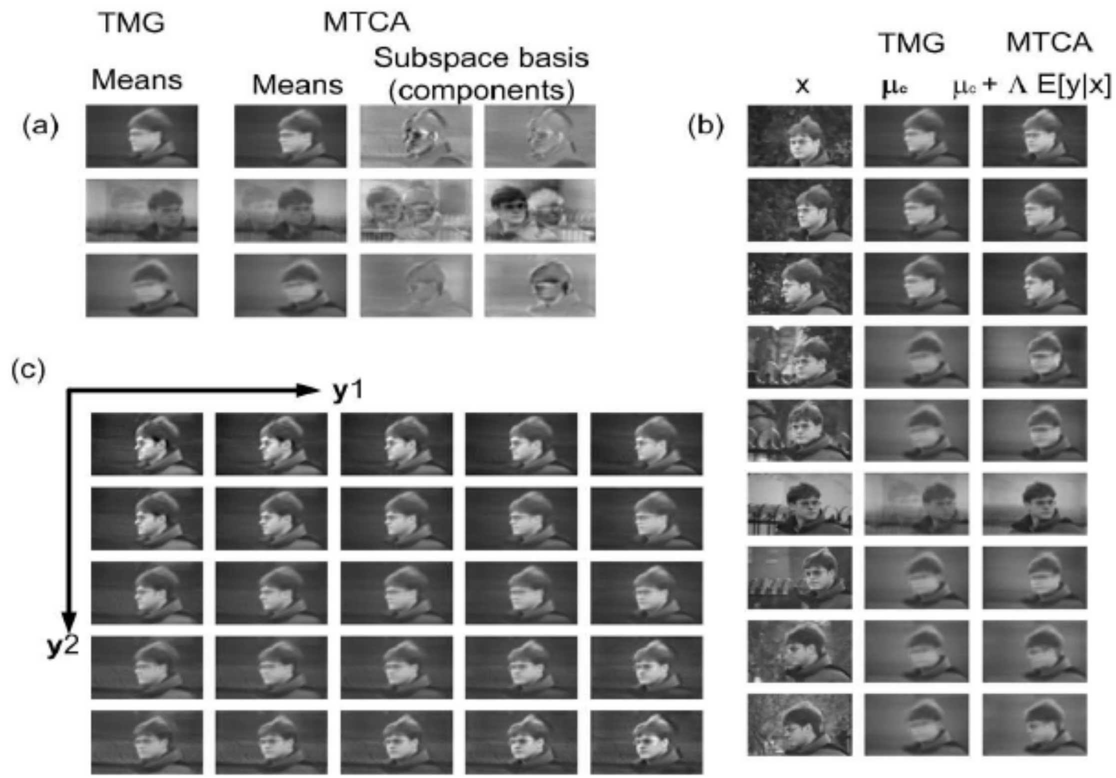
Mixture of Transformed Components



Latent variables for cluster

Latent variables for components

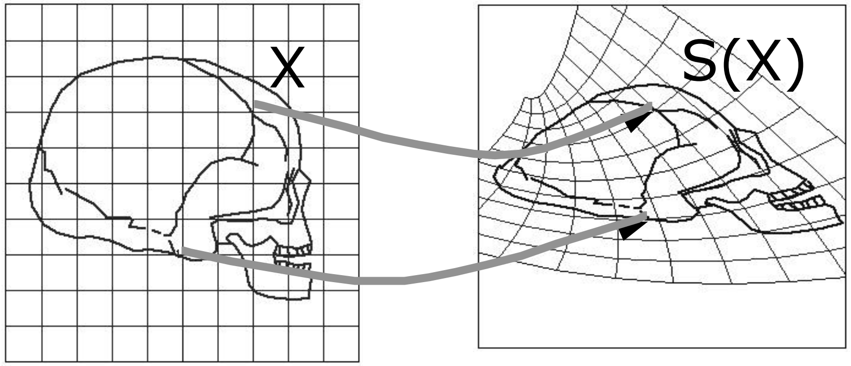
Latent variables for shift



Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, PAMI 2003

Nonrigid deformations: AAMs

$$T(\mathbf{x}) \quad = \quad I(\mathcal{S}(\mathbf{x}))$$



- Active Appearance Models

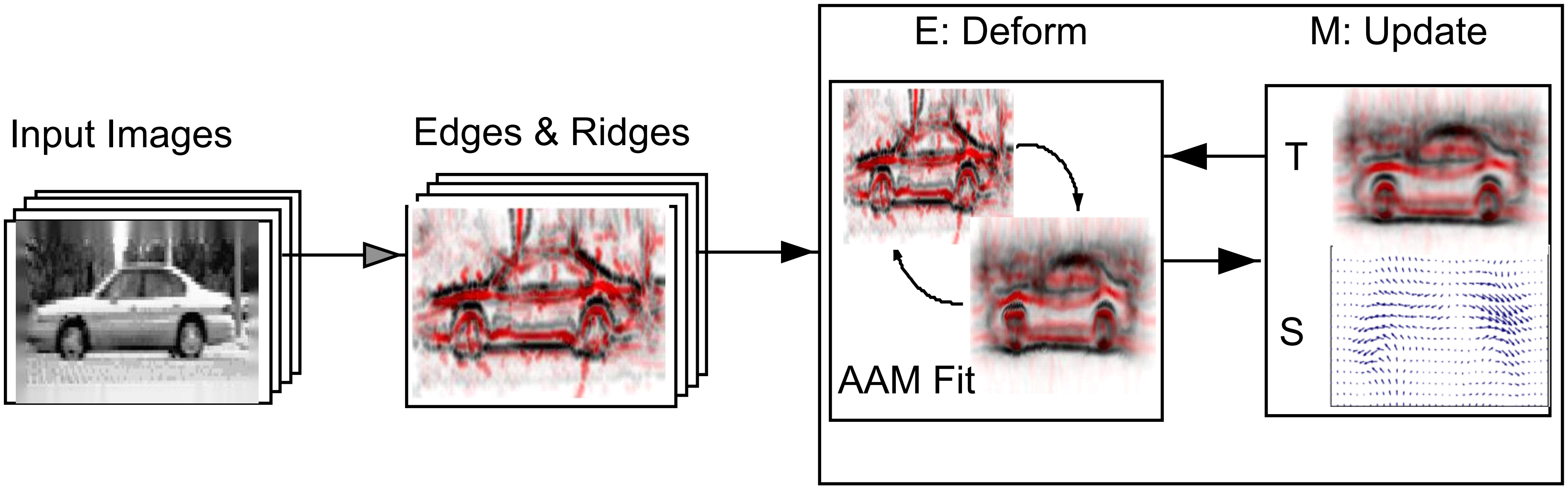
$$\mathcal{S}(\mathbf{x};\mathbf{s}) = \sum_i \mathbf{s}_i S_i(\mathbf{x}) \qquad \mathcal{T}(\mathbf{x};\mathbf{t}) = \sum_i \mathbf{t}_i T_i(\mathbf{x})$$

EM-based AAM learning

Training criterion:

$$\min_{\mathbf{s}_k, S_i, \mathcal{T}} \sum_{k=1}^{K} \sum_{\mathbf{x}} \left[I_k(\mathcal{S}(\mathbf{x};\mathbf{s}_k)) - \mathcal{T}(\mathbf{x})\right]^2$$

$$\mathcal{S}(\mathbf{x};\mathbf{s}_k) = \sum_i \mathbf{s}_{i,k} S_i(\mathbf{x})$$



I. Kokkinos and A. Yuille, Unsupervised learning of object deformation models, ICCV 2007

Bottle models

Observations

Template

1st basis element

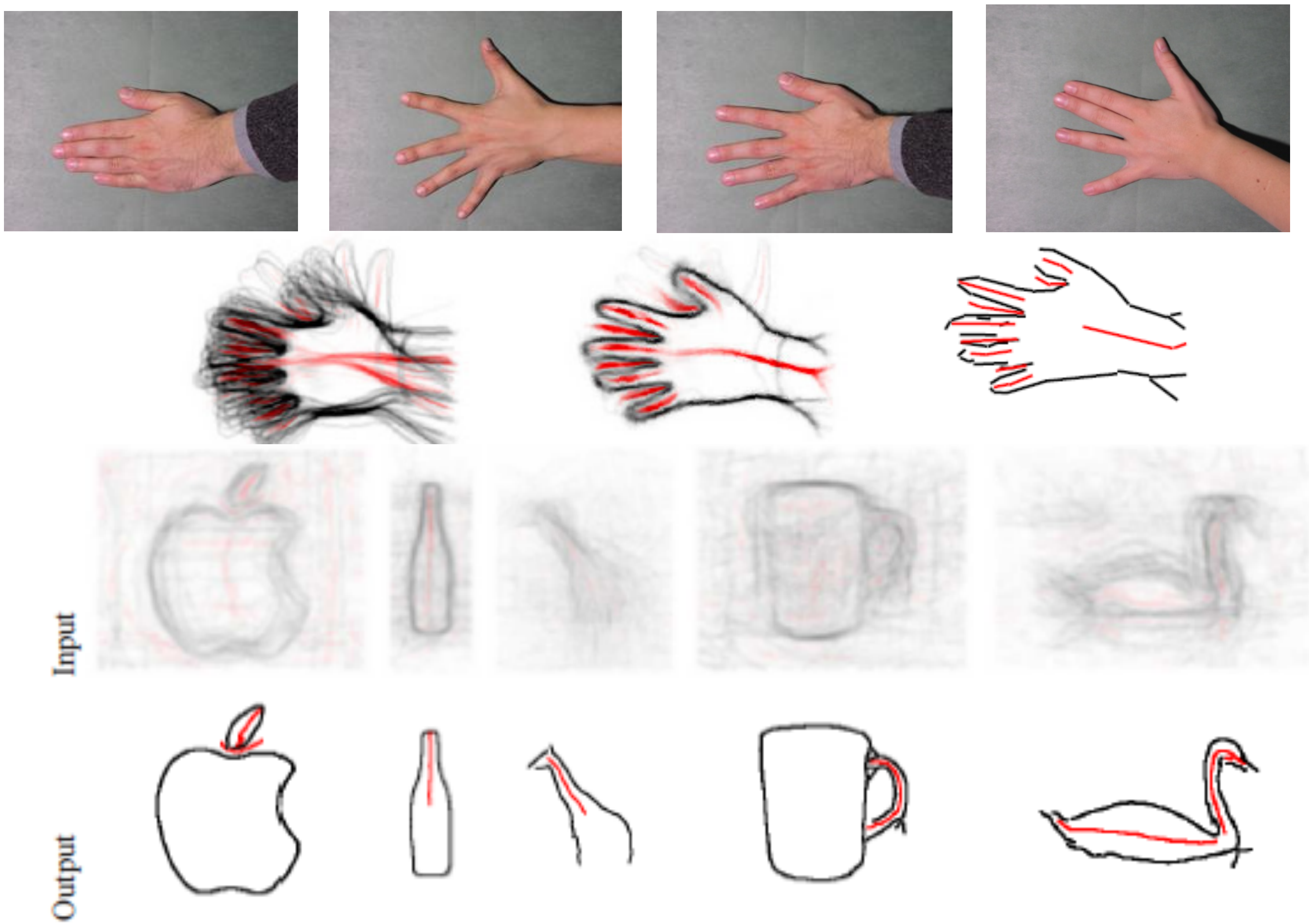
2nd basis element

I. Kokkinos and A. Yuille, Unsupervised learning of object deformation models, ICCV 2007

Recovering Object Contours (2007)

I. Kokkinos and A. Yuille, Unsupervised learning of object deformation models, ICCV 2007

Hand, apple, giraffe, mug, swan models (2011)



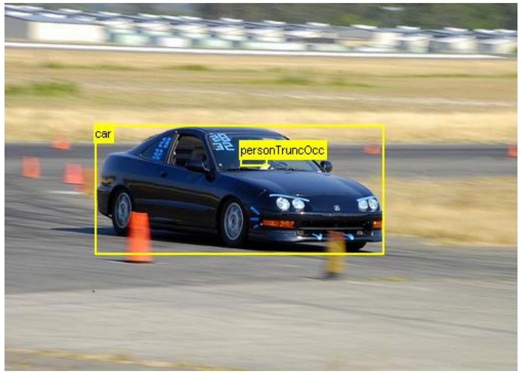
I. Kokkinos and A. Yuille, Inference and Learning for Hierarchical Shape Models, IJCV 2011

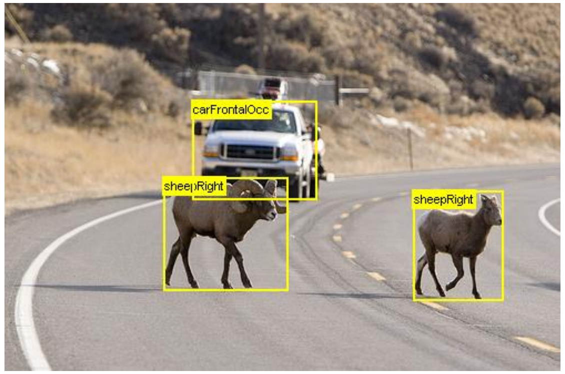
Pascal Dataset

20 Categories, 25000 images

`Bus’

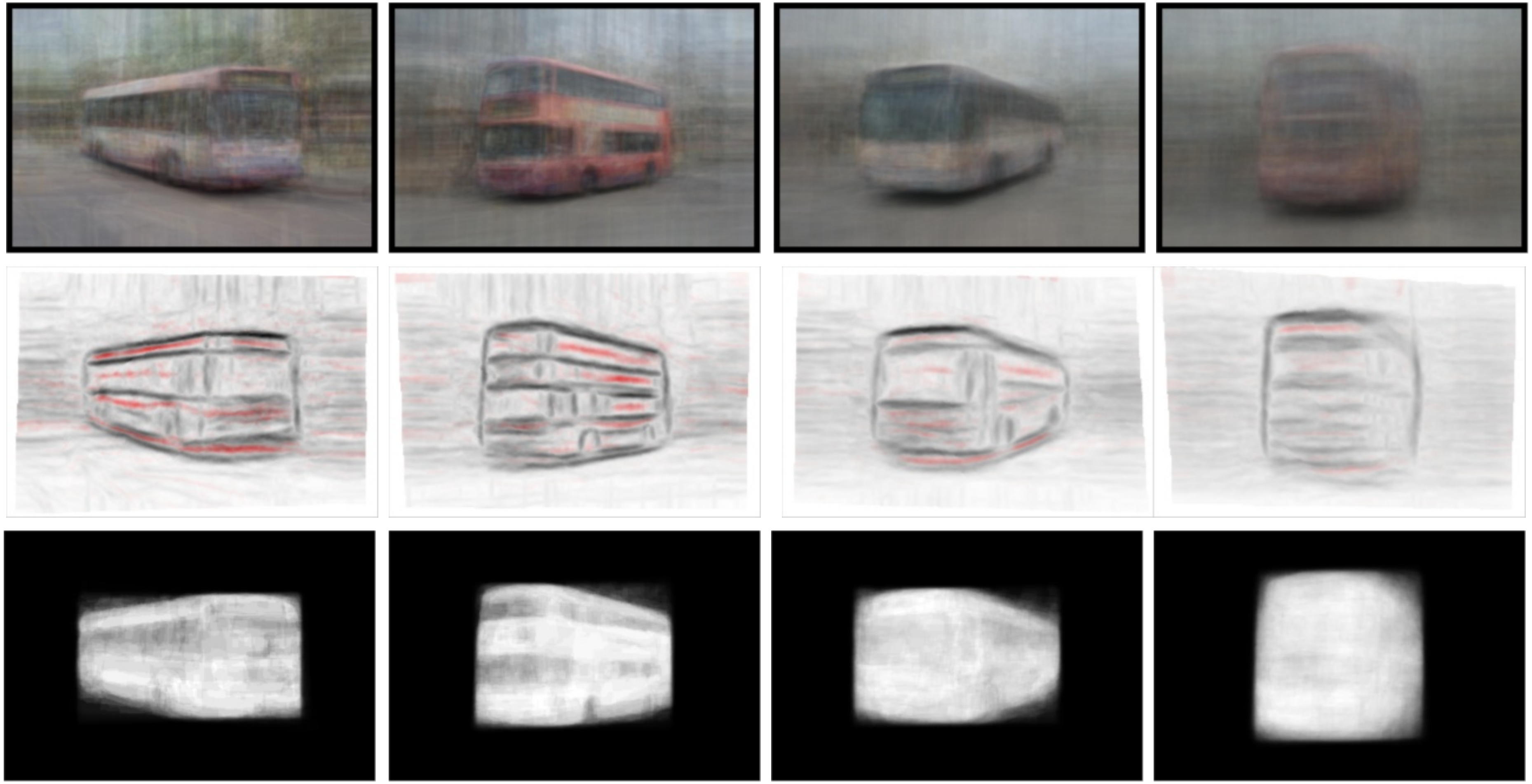
`Car’



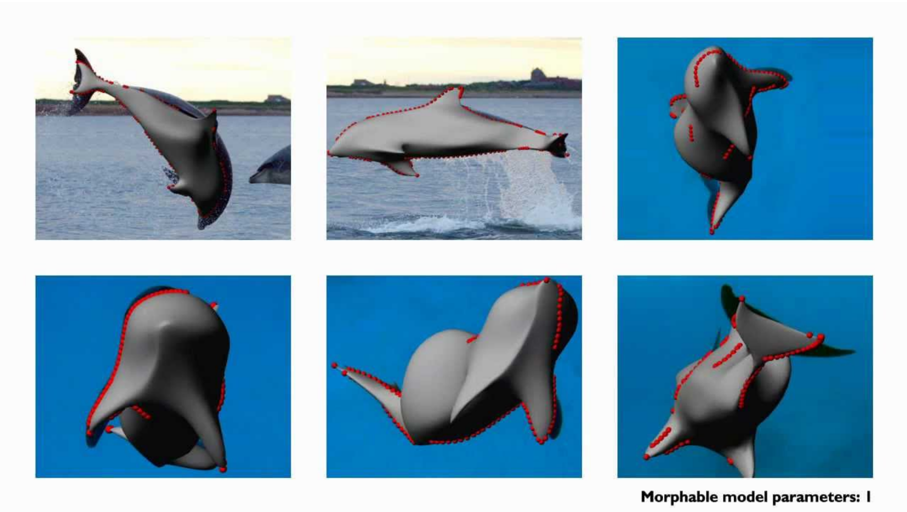


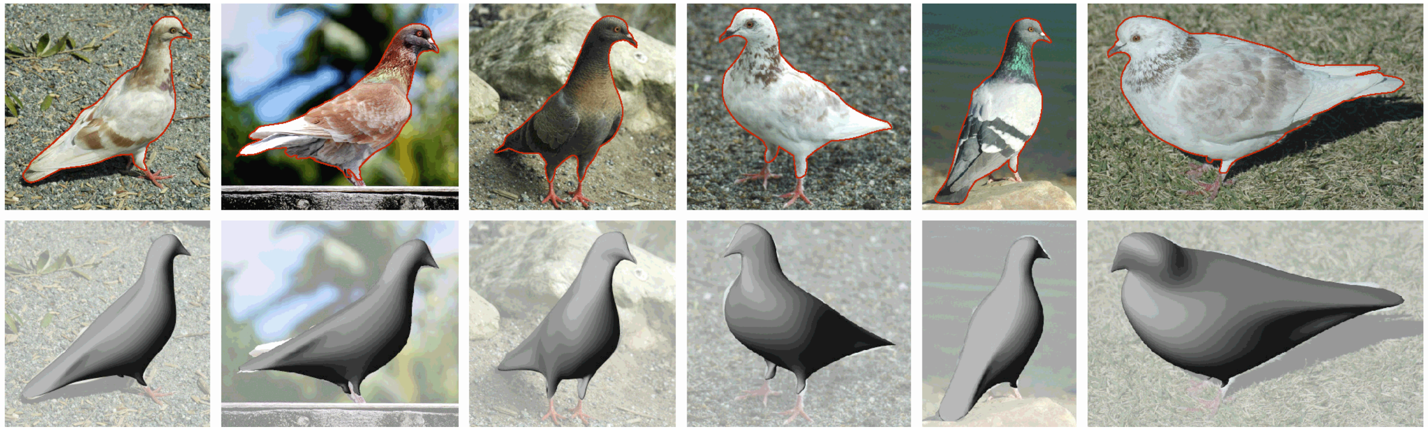
Multi-view models (20??) – work in progress

Multi-view models + segmentation



Semi-automated learning of 3D morphable models





T. J. Cashman, A. W. Fitzgibbon: What Shape Are Dolphins? Building 3D Morphable Models from 2D Images, 2013
http://research.microsoft.com/en-us/um/people/awf/dolphins/