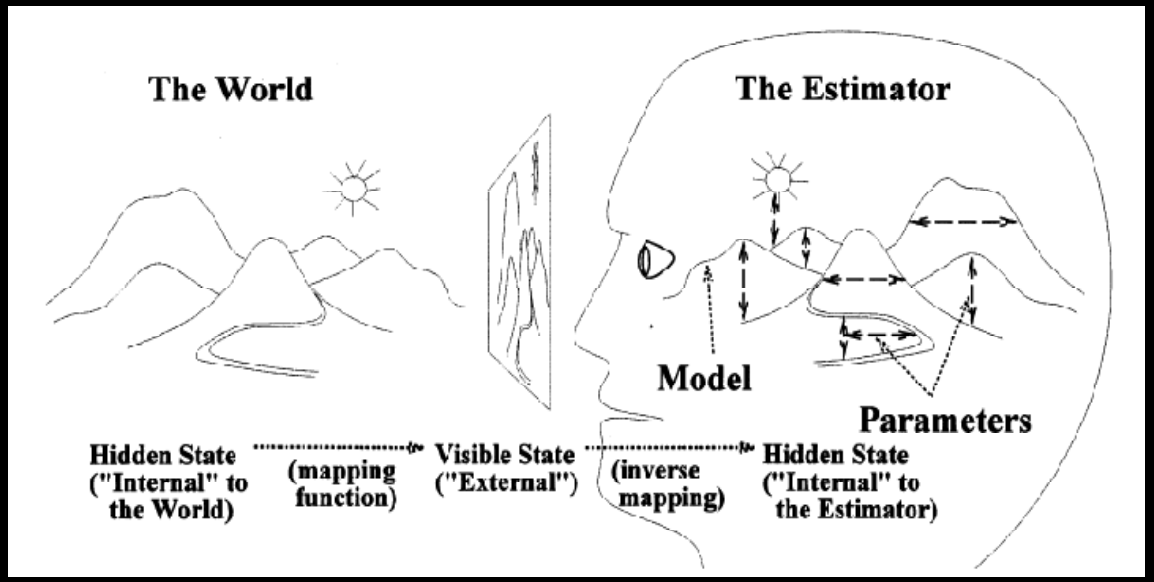
Machine Learning for Computer Vision

14 October, 2013
MVA – ENS Cachan



Lecture 5: Introduction to generative models

Iasonas Kokkinos

Iasonas.kokkinos@ecp.fr

Center for Visual Computing
Ecole Centrale Paris

Galen Group
INRIA-Saclay

Lecture outline

Bayes' rule and generative models

Density estimation

Parametric deformable models

Decision Theory

- What is an optimal decision rule?
- Consider loss matrix

$$L_{kj} = loss\ for\ decision\ \mathcal{C}_j\ if\ \ truth\ \ is\ \mathcal{C}_k$$

- Consider underlying joint distribution of data-label pairs

$$P(X, y), \quad X \in R^K, y \in Z$$

- Find decision rule f that minimizes expected loss:

$$E[L] = \sum_k \sum_j \int_{R_j} L_{k,j} P(X, C_k) dX$$

$$R_j : \{X, \ \text{s.t.} \ \ f(X) = j\}$$

Optimal Classifier

- Consider zero-one loss function: $L(y, f(x)) = 1 - [y = f(x)]$
- Form `Expected Prediction Error':

$$EPE(f) = \int_x \sum_y L(y, f(x)) P(x, y) dx = \int_x \left[\sum_y L(y, f(x)) P(y|x) \right] P(x) dx$$

- Optimal decision at any x :

$$\hat{f}(x) = \arg\min_g \sum_y L(y, g) P(y|x)$$

$$\overset{\sum_y P(y|x)=1}{=} \arg\min_g 1 - P(g|x) = \arg\max_g P(g|x)$$

  - `Bayes-optimal classifier'

Bayes' theorem

Sum Rule	$p(X) = \sum_Y p(X, Y)$
Product Rule	$p(X, Y) = p(Y\|X)p(X)$

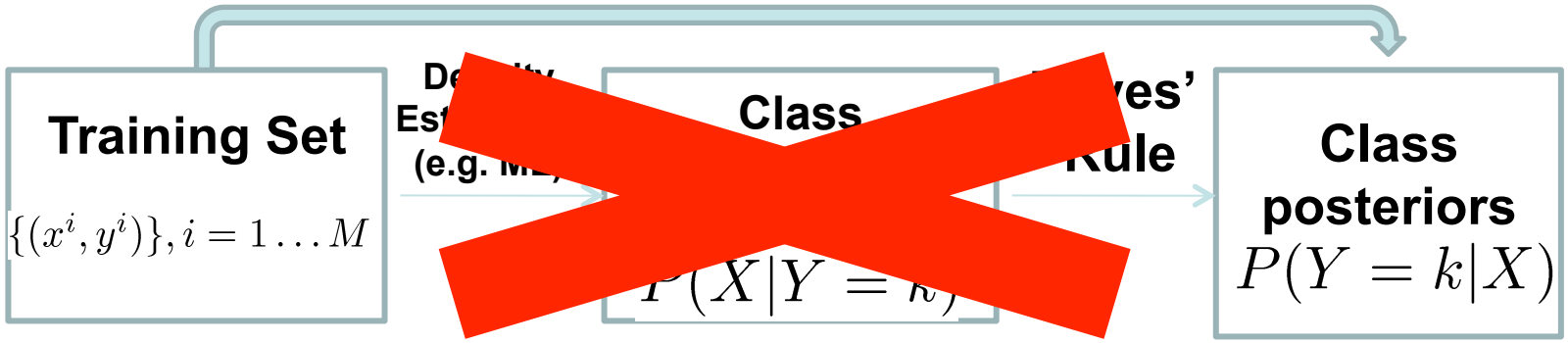
$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$$= \frac{P(X|Y)P(Y)}{\sum_{Y'} P(X, Y')} = \frac{P(X|Y)P(Y)}{\sum_{Y'} P(X|Y')P(Y')}$$

- P(X|Y): **likelihood** of observations X, given class Y.
- P(Y): **Prior** probability of class Y
- P(Y|X): **Posterior** probability of class Y, given observations Y.

Why is this identity important?

Generative or disciminative?

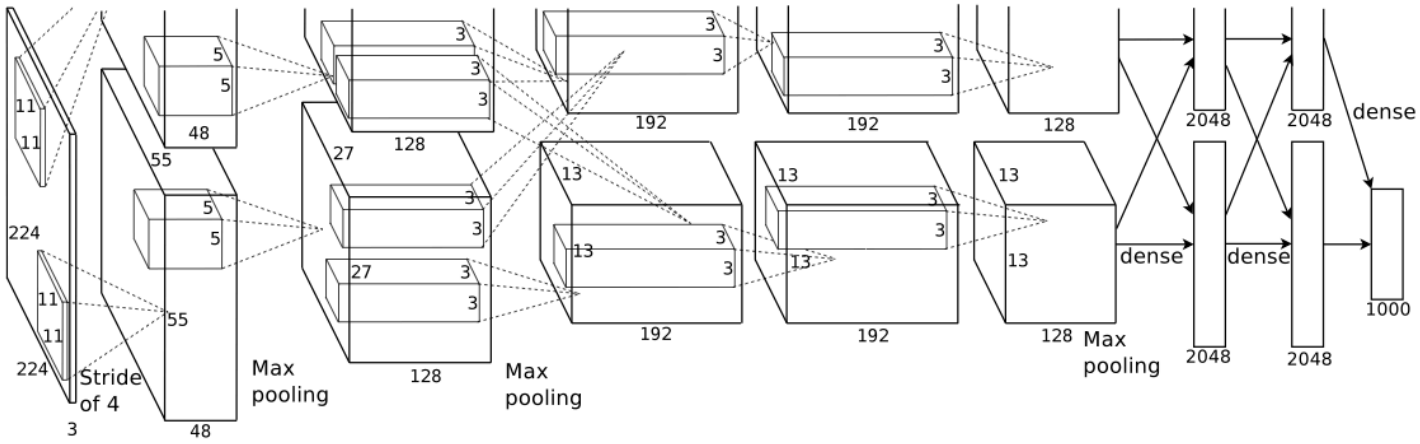


– Discriminative Models (lectures 1-4) : **Skip density estimation**
  - More robust to wrong distribution assumptions (e.g. outliers)
  - V. Vapnik: `one should solve the classification problem directly and never solve a more general problem as an intermediate step'

– Generative Models (Lectures 5-7) : **Core task: density estimation**
  - If we know the distributions, requires smaller training sets
  - Dealing with missing/corrupt data
  - Explicit modelling of sources of variation (e.g. translation)
  - Conceptual clarity, ability for `visual debugging'

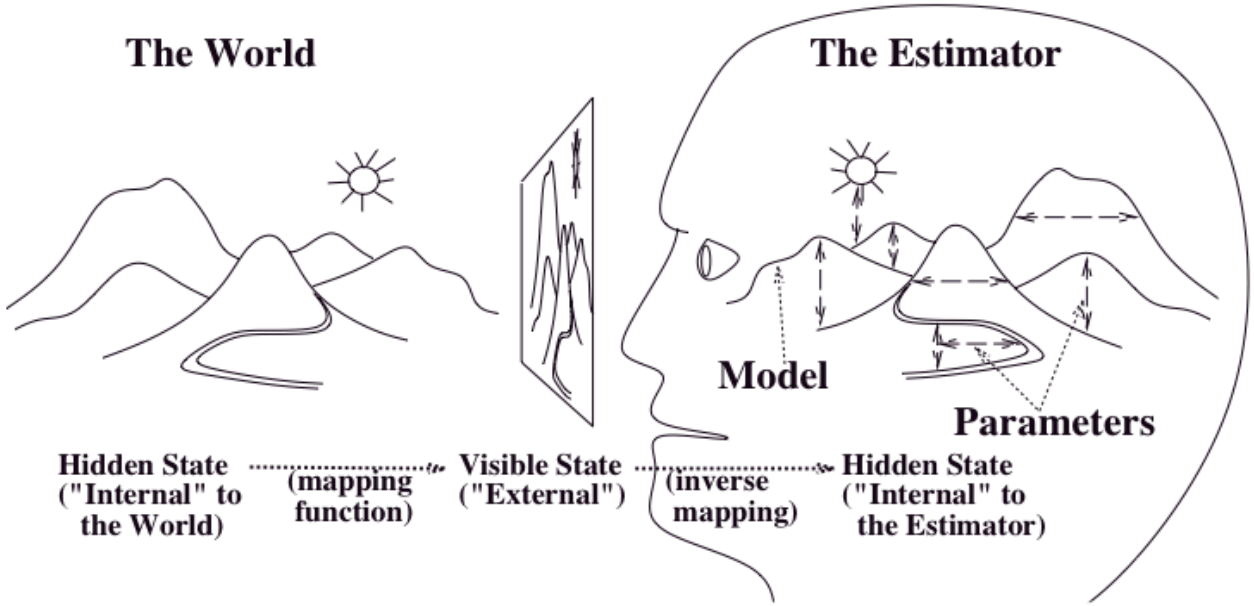
Two Main Approaches

- Discriminative



Two Main Approaches

- Generative

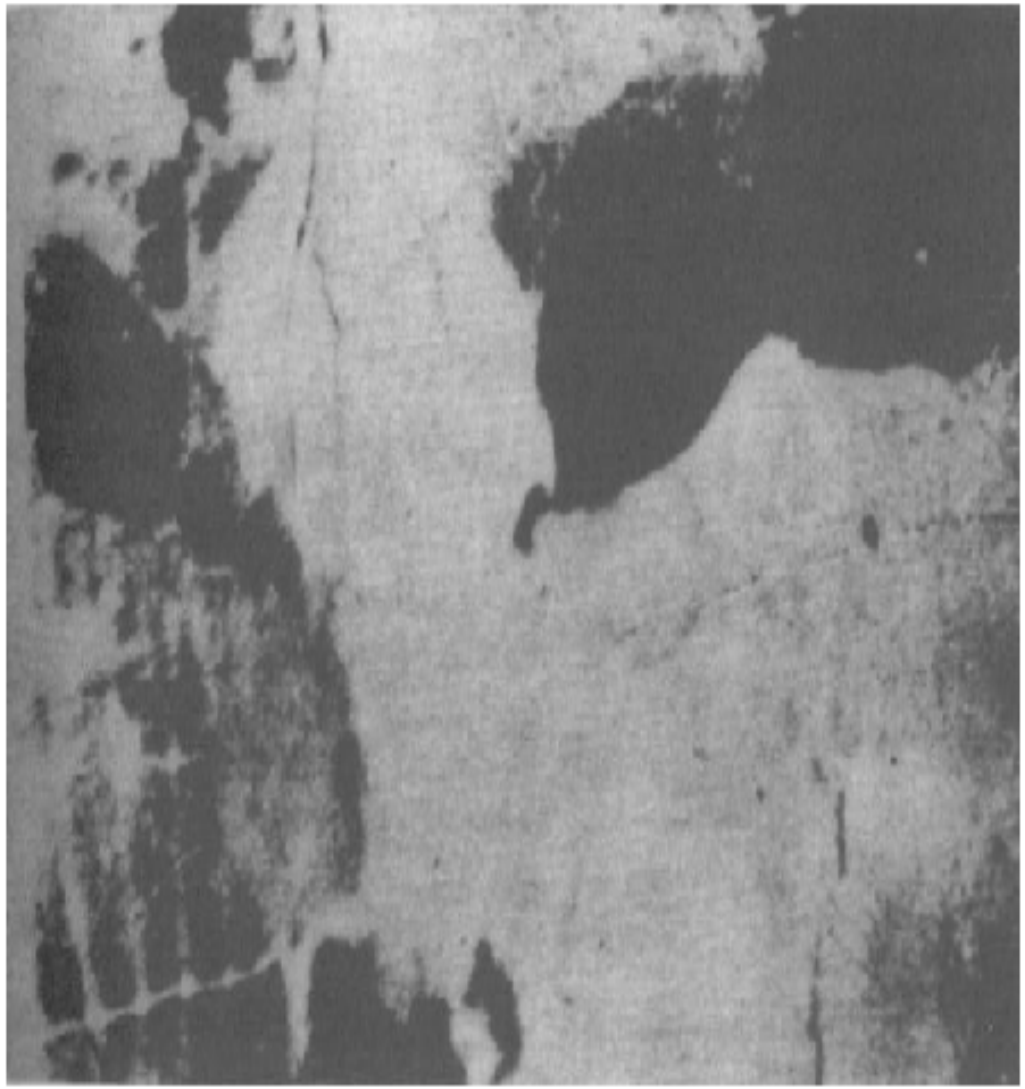


Likelihood Prior

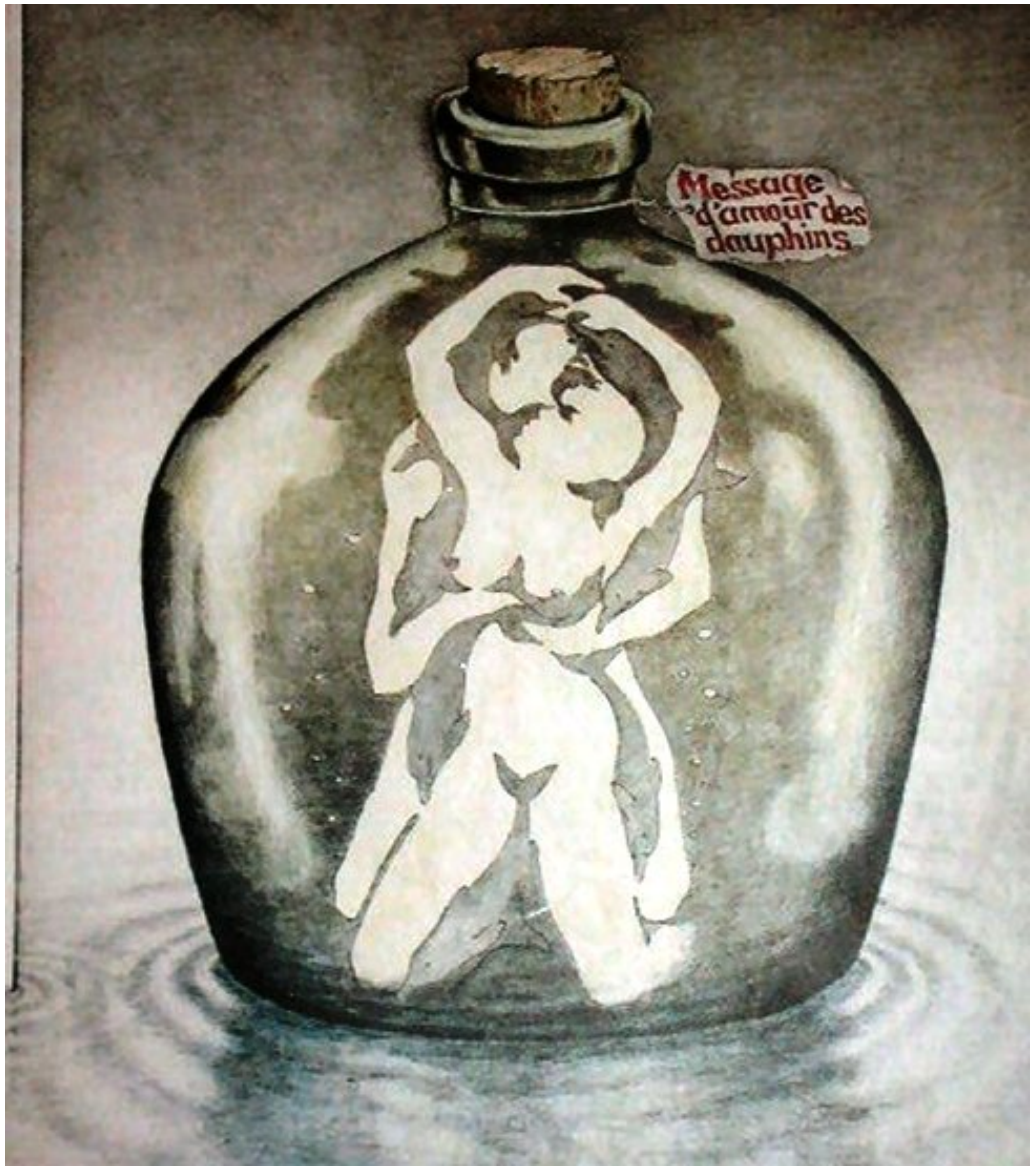
Posterior

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Vision & priors



Vision & priors



Proof by eminent authority

Socrates: *..which is more correct, to say that we see with the eyes, or through the eyes?*
Theaeteus: *I should say "through," Socrates, rather than "with".*
Socrates: *Yes son, because it would be woeful if all these senses were sitting in us like Trojan horses instead of converging to some idea, soul, or whatever, by which we use such organs to perceive what is perceivable.*
Plato, *Theaetetus*, 369 BC

the perception is neither a seeing, nor a touching, nor an imagining ... rather it is an inspection.. of the mind.
Descartes *Meditations*, 1641

..it is quite possible that our empirical knowledge is a compound of that which we receive through impressions, and that which the faculty of cognition supplies from itself (sensuous impressions giving merely the occasion)
Kant, *A critique of pure reason*, 1787

to read nature is to see it, as if through a veil, in terms of an interpretation.
Cézanne, 1839-1906

…the value of a simple stimulus … for conveying information depends not only on the information conveyed by the stimulus itself but on the whole nervous constitution of the receiver of the stimulus as well.
Wiener, *Cybernetics*, 1948

... The signal never makes it to our consciousness, but gets overlaid with a clearly and precisely patterned version whose computation demands extensive use of memories, expectations, and logic.
Mumford, *Pattern Theory – a unifying perspective*, 1995

Pattern Theory

The four transformations that I propose as the basic types occurring in natural perceptual signals are:

Shape

Domain warping

Segmentation

Interruptions

Scale

Multi-scale superposition

Noise and blur

What makes pattern theory hard is not that any of the above transformations are that hard to detect and decode in isolation, but rather that all of them tend to coexist, and then the decoding becomes hard.

D. Mumford, *Pattern Theory – a unifying perspective*, 1995

Lecture outline

Bayes' rule and generative models

Density estimation

Gaussian distributions

Mixture-of-Gaussian models

Hidden Variables and Expectation-Maximization algorithm

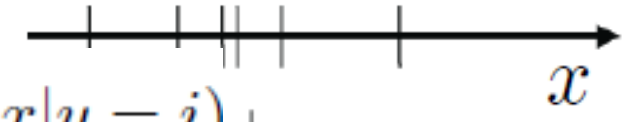
Factor Analysis & PCA

Mixed Discrete/Continuous hidden variable models

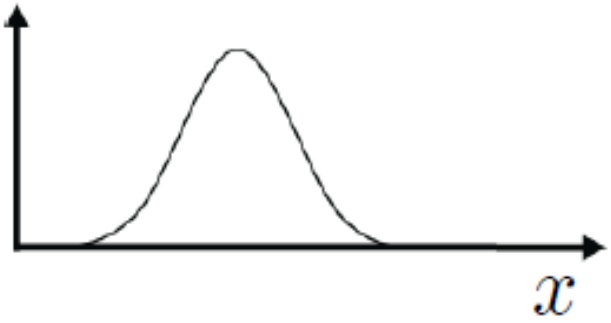
Parametric deformable models

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: $P(X = x|y = i)$

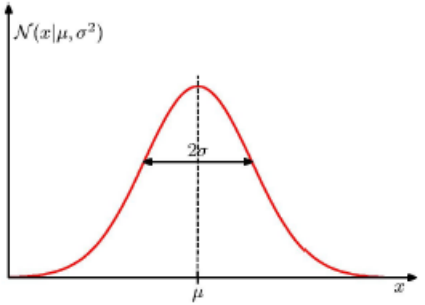


for short: $P(x)$



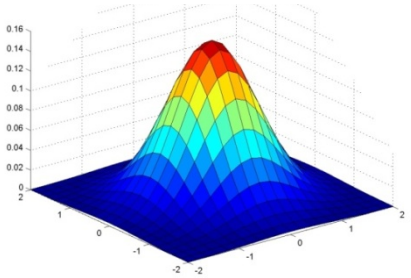
Parametric Distributions: Gaussian

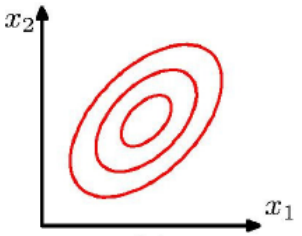
– 1 D $$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$



– 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:

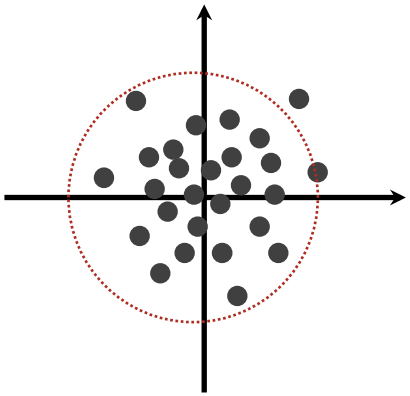




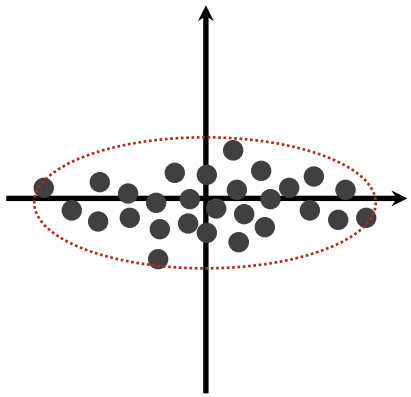
Covariance matrix reminder

Covariance matrix:

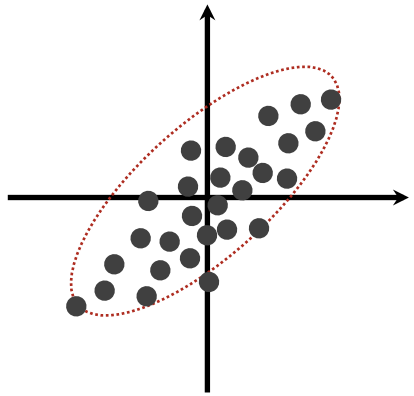
$$\Sigma_{i,j} = E\left((x_i - E(x_i))(x_j - E(x_j))\right)$$



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



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



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

Height, Income

Height, Weight

Uncorrelated coordinates: diagonal covariance

Density Estimation for a Gaussian distribution

- Given: $(x^i, y^i), \quad i = 1 \dots M, \quad x^i \in R^d \quad y^i \in \{0, 1, \dots K\}$

- Notation: $r_k^i = [y^i = k]$
- Maximum Likelihood Estimation for class k:

$$\left(\hat{\mu}_k, \hat{\Sigma}_k\right) = \arg\max \prod_{i=1}^{M} P(x_i|\mu_k, \Sigma_k)^{r_k^i} = \arg\max \sum_{i=1}^{M} r_k^i \log P(x_i|\mu_k, \Sigma_k)$$

$$= \arg\max \sum_{i=1}^{M} r_k^i \left[\frac{1}{2}\log|\Sigma_k^{-1}| - \frac{1}{2}(x^i - \mu_k)\Sigma_k^{-1}(x^i - \mu_k)\right]$$

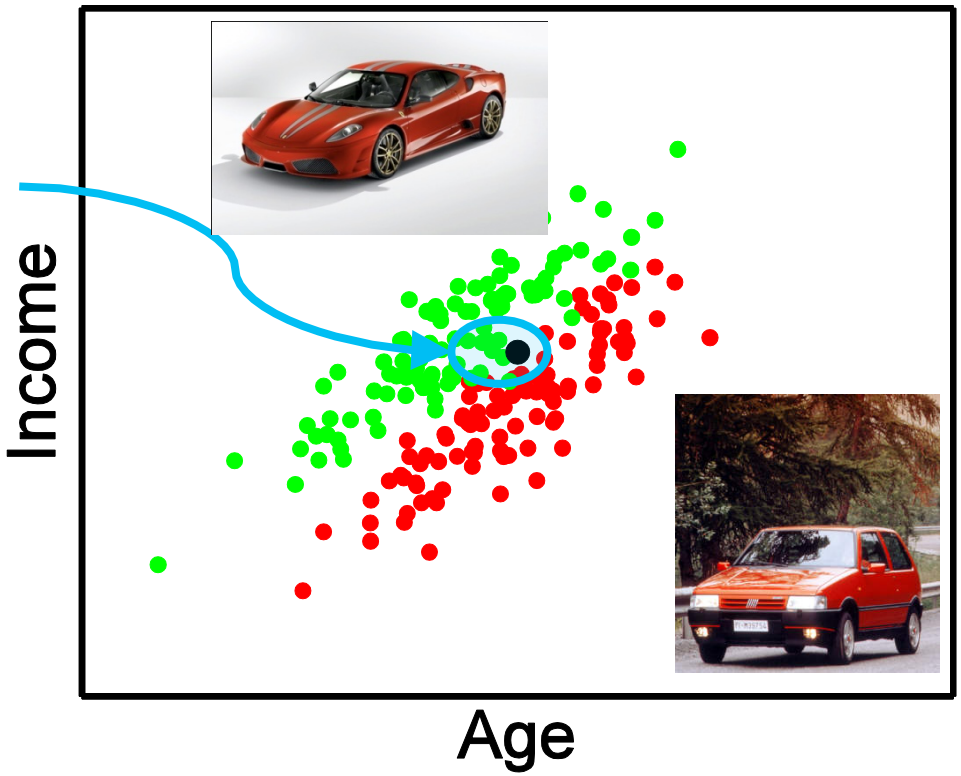
$$\overset{C=\Sigma^{-1}}{=} \arg\max \sum_{i=1}^{M} r_k^i \left[\frac{1}{2}\log|C| - \frac{1}{2}(x^i - \mu_k)C(x^i - \mu_k)\right]$$

$$0 = \frac{\partial L_k}{\partial \mu} = \sum_i r_k^i 2(x^i - \mu)^T C \rightarrow \hat{\mu}_k = \frac{\sum_i r_k^i x^i}{\sum_i r_k^i}$$

$$0 = \frac{\partial L_k}{\partial C} = \sum_i r_k^i C^{-1} - \sum_i (x^i - \hat{\mu})^T (x^i - \hat{\mu}) \rightarrow \hat{\Sigma}_k = \frac{\sum_i r_k^i (x^i - \hat{\mu})^T (x^i - \hat{\mu})}{\sum_i r_k^i}$$

Classification task: Ferrari or Fiat?

- Consider placing a personalized ad. Which car will you try?
- New client: x



- Classification problem: new client is likely to buy Fiat/Ferarri.
- Build class-specific probability distributions

Ferrari or Fiat, continued

Class-specific Gaussian Distributions

$$P(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{n/2}|\Sigma_k|^{(1/2)}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right), \quad k = 0, 1$$

Parameter estimation: Maximum Likelihood (ML)

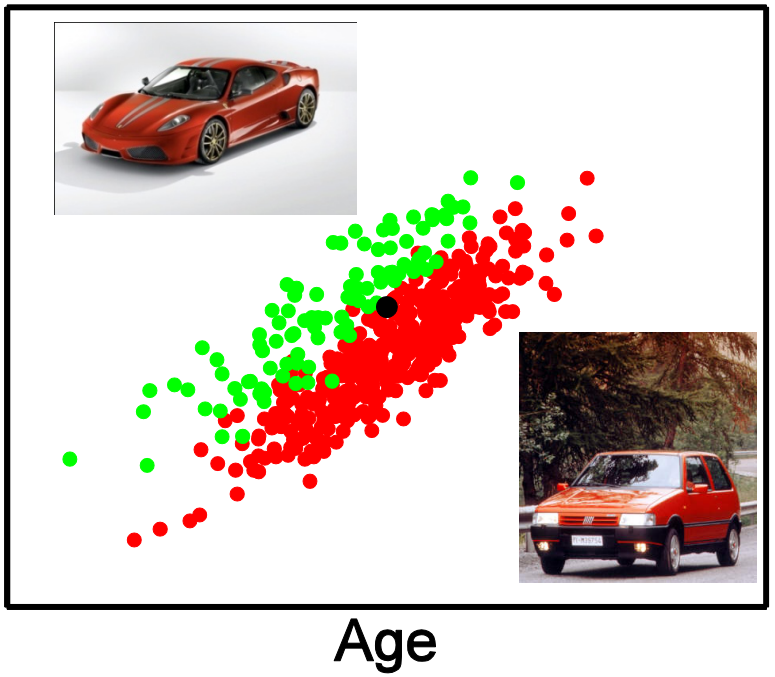
Can we proceed to classification?

Bayes' rule:

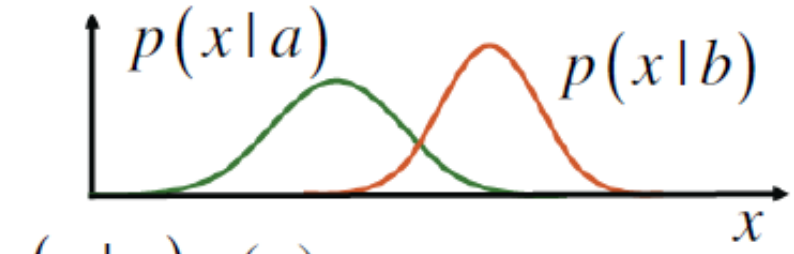
$$\hat{i} = \arg\max_i P(y=i|x) \propto P(x|y=i)P(y=i)$$

Need to estimate: $\pi_i = P(y=i)$

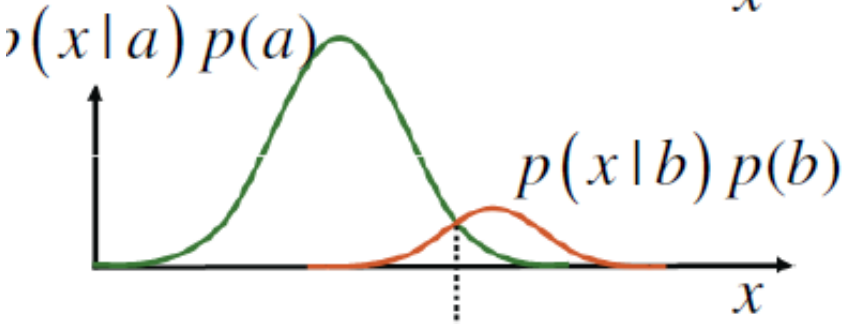
ML estimate: $$\pi_k = \sum_{i=1}^{N} \frac{[y_i = k]}{N}$$



Bayes rule, 1D

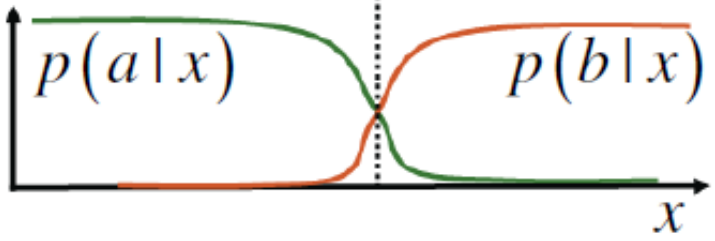


Likelihood



Likelihood $\times$ *Prior*

Decision boundary



$$Posterior = \frac{Likelihood \times Prior}{NormalizationFactor}$$

Classifier form for Gaussian Distributions

- Choose class $k = \arg\max_k P(y = k|x)$
- Decision boundary for the binary case:

$$P(y = 1|x) = \frac{1}{2} \rightarrow x^T Ax + x^T B + C = 0$$

Quadratic Decision Boundaries

- Special case: $\Sigma_0 = \Sigma_1$

$$A = 0, \quad B = \Sigma^{-1}(\mu_0 - \mu_1), \quad C = \frac{1}{2}(\mu_1^T \Sigma^{-1} \mu_1 - \mu_0^T \Sigma^{-1} \mu_0) + \log(\frac{1-\pi}{\pi})$$

$$P(y = 1|x) = \frac{1}{1 + \exp(x^T B + C)}$$

Linear Decision Boundaries

Lecture outline

Bayes' rule and generative models

Density estimation

Gaussian distributions

Mixture-of-Gaussian models

Expectation-Maximization algorithm and hidden variables

Factor Analysis & PCA

Mixed Discrete/Continuous hidden variable models

Parametric deformable models

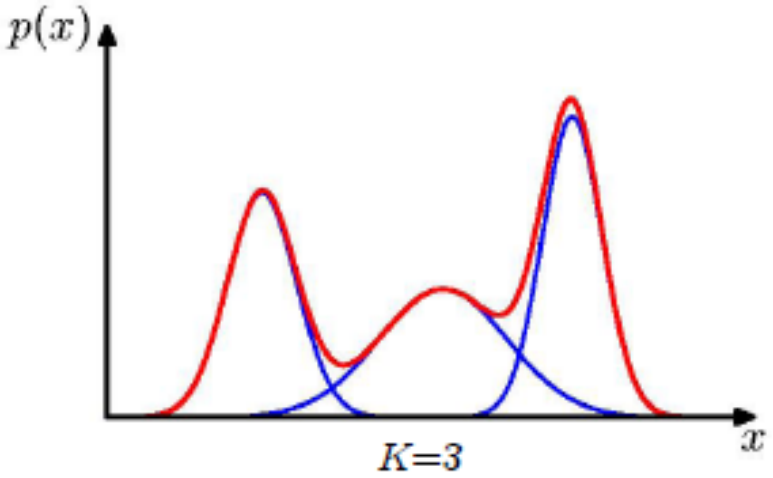
Mixture of Gaussians model

Combine simple models into a complex model:

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

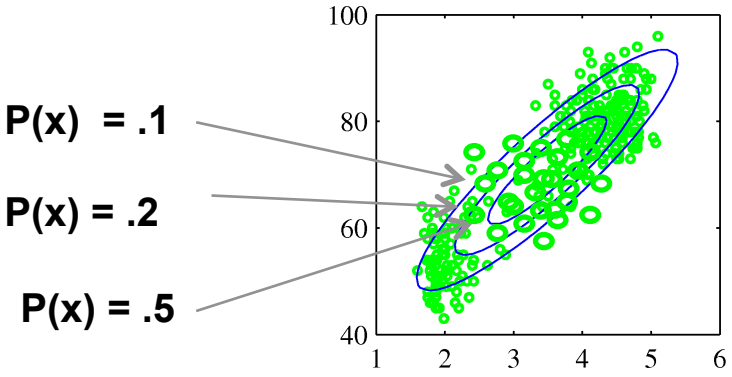
Component

Mixing coefficient

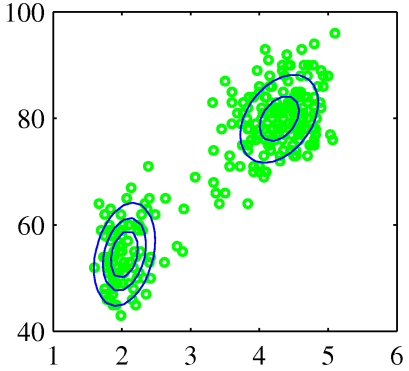


$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

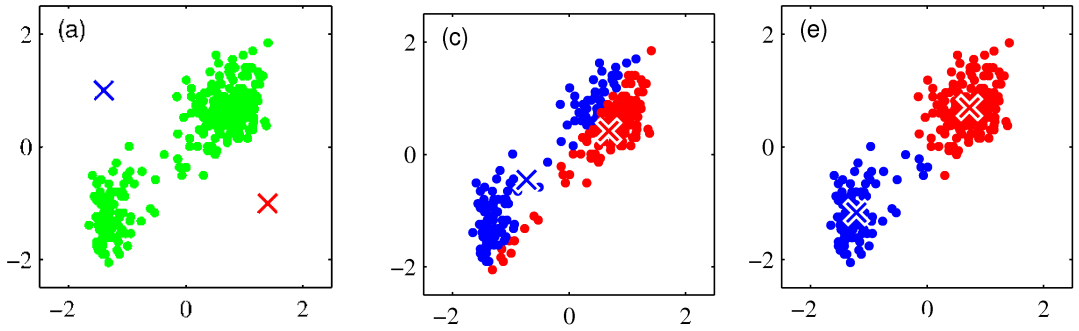
Main challenge: parameter estimation



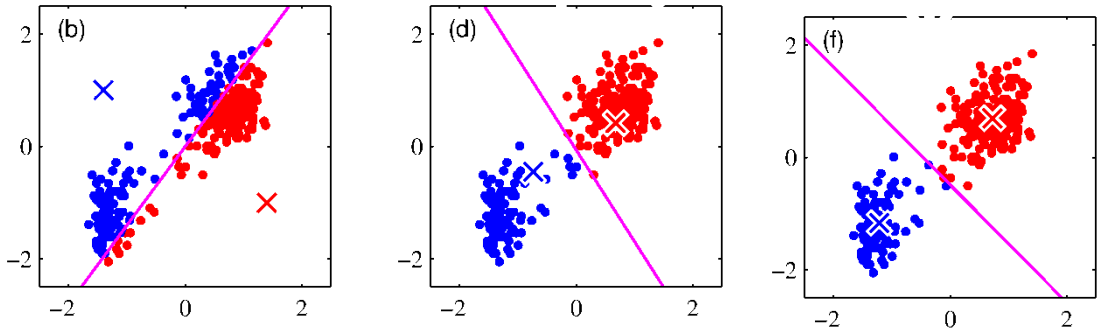
Which points go with which cluster?



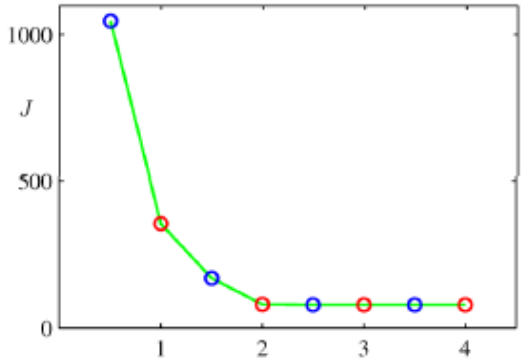
K-Means algorithm



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



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



– Coordinate descent on distortion cost:

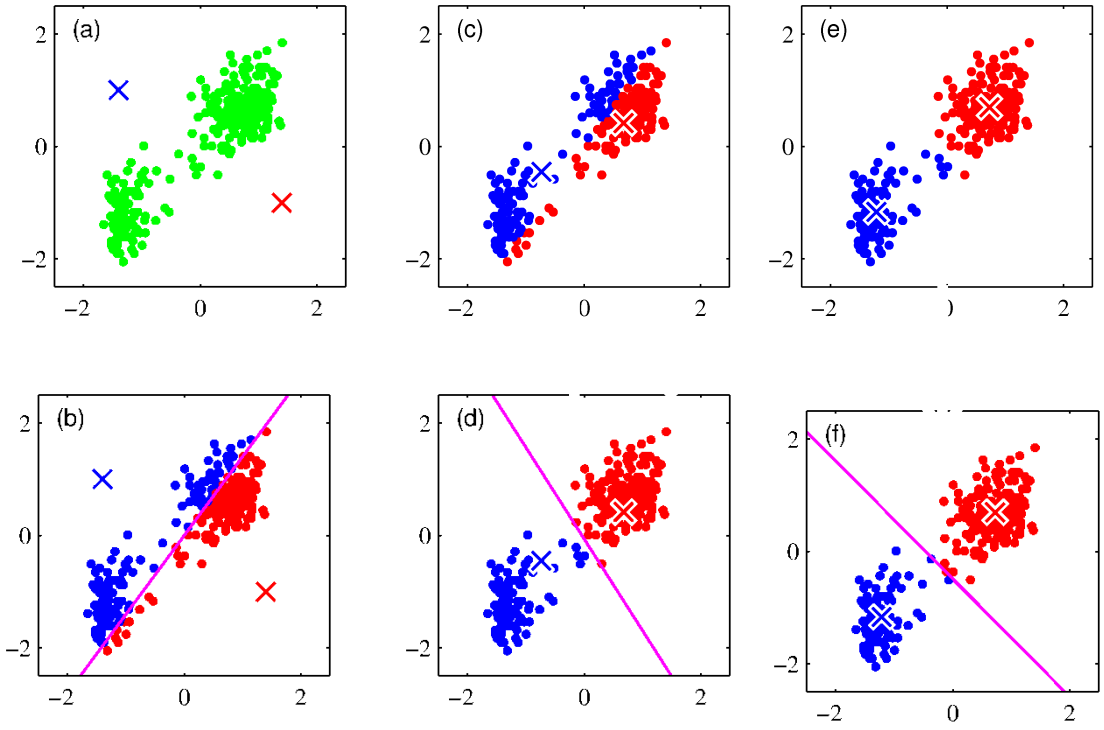
$$F(m,c) = \sum_{i=1}^{N} |x^i - c^{m(i)}|^2$$

– Local minima (multiple initializations to find better solution)

Lecture outline

- Bayes' rule and generative models
- Density estimation
  - Gaussian distributions
  - Mixture-of-Gaussian models
    - Expectation-Maximization algorithm and hidden variables
  - Factor Analysis & PCA
  - Mixed Discrete/Continuous hidden variable models
- Parametric deformable models

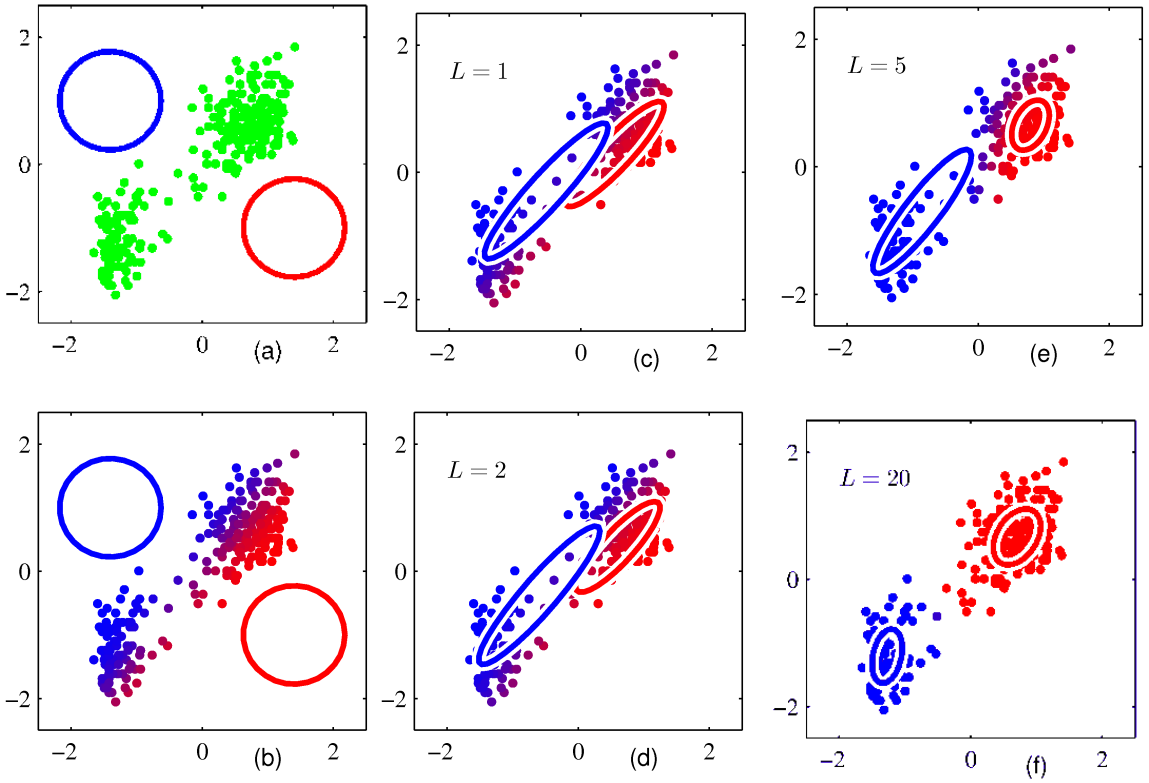
K-Means algorithm



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

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

Adaptation for Gaussian distributions

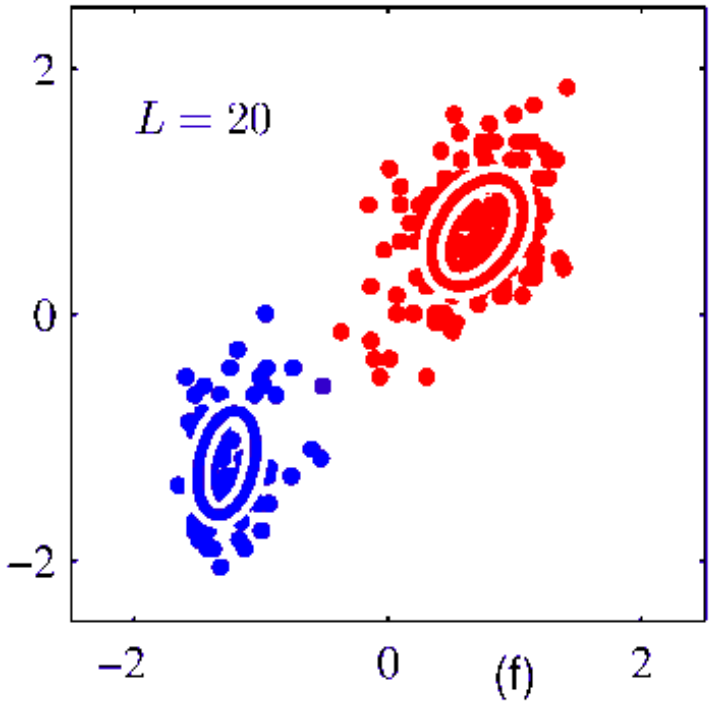


$$\mu^j = \frac{\sum_{i=1}^N R_{i,j} x^i}{\sum_{i=1}^N R_{i,j}}$$

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

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

Expectation Maximization algorithm



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

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

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

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

K-means vs. EM

k-means	EM
Closest center's index	Soft assignment, R
Isotropic Distance (Euclidean)	Anisotropic Likelihood (Covariance-based,`Mahalanobis')
Fast (e.g. kd-trees)	Accurate & more flexible
More robust to initalization	Prone to local minima

Typical usage: initialize EM with k-means results

Coordinate Descent on

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

Coordinate descent on?

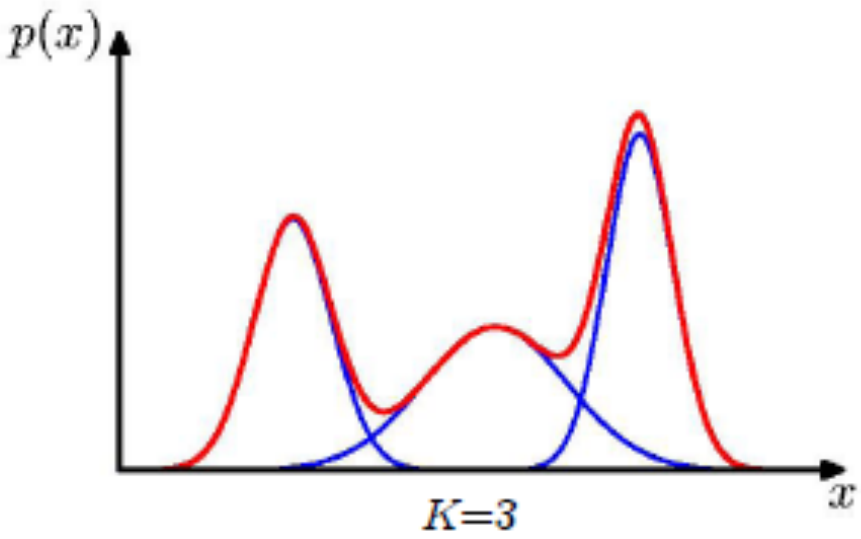
Mixture of Gaussians

- **Combine simple models into a complex model:**

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

Component

Mixing coefficient



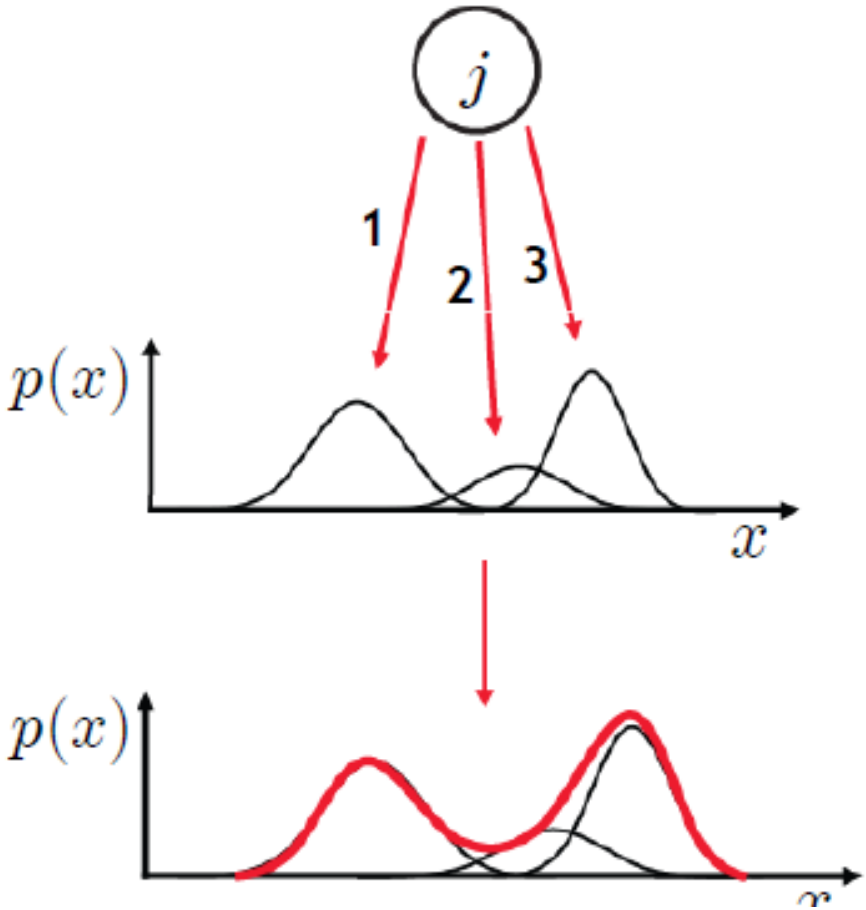
$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

- 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

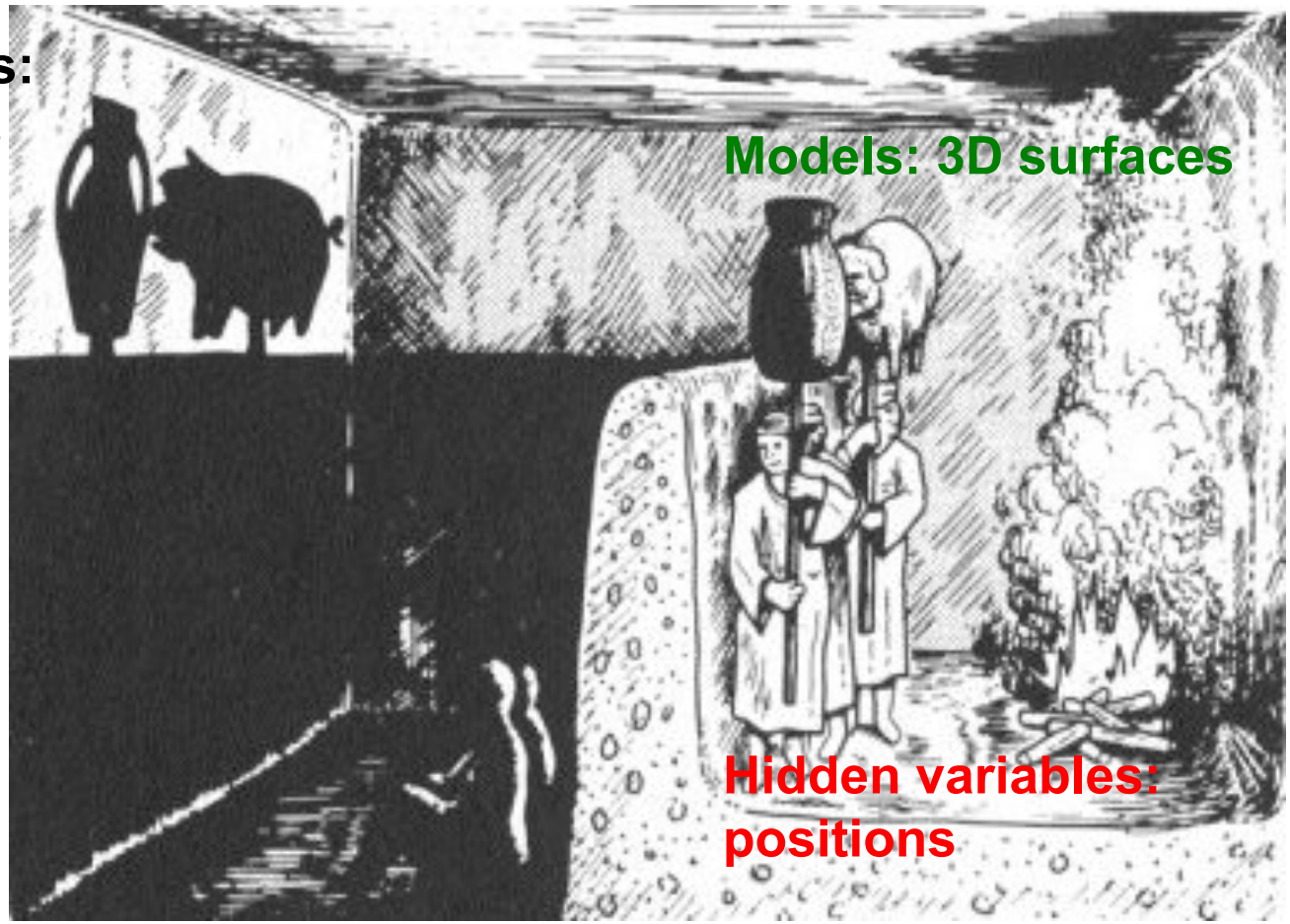
$$p(x|\theta) = \sum_{j=1}^{M} p(x|\theta_j)p(j)$$

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



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?

- Hidden variable h^i
  - Indicate which component is responsible for each point
  - Multinomially distributed variable

$$P(h = j) = \pi_j \qquad P(h|\pi) = \prod_{j=1}^{K} \pi_j^{h=j}$$

Rewriting the MoG distribution

- Marginalization $P(a|b) = \sum_c P(a, c|b)$
- Chain rule $P(a, c|b) = P(a|c, b)P(c|b)$
- We have

$$\begin{aligned} P(x^i|\theta) &= \sum_{h^i} P(x^i, h^i|\theta) \\ &= \sum_{h^i} P(x^i|h^i, \theta)P(h^i|\theta) \\ &= \sum_{k=1} P(x^i|\mu_k, \Sigma_k)\pi_k \end{aligned}$$

Complete Log-Likelihood

- Assume hidden variables are given
- Data+ hidden variables = *complete observations*
- *Complete log-likelihood*

$$
\begin{aligned}
l(\theta; x, h) = \log P(x, h|\theta) &= \sum_{i=1}^{N} \log P(x^i, h^i|\theta) \\
&= \sum_{i=1}^{N} \log P(x^i|h^i, \theta) P(h^i|\theta) \\
&= \sum_{i=1}^{N} \log \prod_{k=1}^{K} P(x^i|\mu_k, \theta_k)^{(h^i=k)} (\pi_k)^{(h^i=k)} \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} (h^i = k) \log P(x^i|\mu_k, \theta_k)(\pi_k)
\end{aligned}
$$

- Summation falls outside the logarithm!

Full Observation Log Likelihood

- Given: Hidden Variables

$$\log P(x, h|\theta) = \sum_{i=1}^{N} \log \prod_{k=1}^{K} P(x^i|\mu_k, \theta_k)^{(h^i=k)} (\pi_k)^{(h^i=k)}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} (h^i = k) \log \pi_k + (h^i = k) \left[-\frac{1}{2}(x^i - \mu_k)^T \Sigma^{-1} (x^i - \mu_k) - \frac{1}{2} \log |\Sigma| + c \right]$$

- Maximize w.r.t . parameters

$$\pi_k = \frac{\sum_{i=1}^{N} (h^i = k)}{N} \qquad \mu_k = \frac{\sum_{i=1}^{N} (h^i = k) x^i}{N}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} (h^i = k)(x^i - \mu)(x^i - \mu)^T}{N}$$

Expected Complete Log-Likelihood

- We do not know the hidden variables (`*missing data*')
- Complete log-likelihood is a random quantity.
- Form its expectation, using a distribution q(h) on hidden variables:

$$< l(\theta; x) >_q = \sum_h q(h) \log P(x, h|\theta)$$

- *Expected complete log-likelihood*

Full Observation Log-Likelihood

- Given: Hidden Variables

$$\log P(x, h|\theta) = \sum_{i=1}^{N} \log \prod_{k=1}^{K} P(x^i|\mu_k, \theta_k)^{(h^i=k)} (\pi_k)^{(h^i=k)}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} (h^i = k) \log \pi_k + (h^i = k) \left[-\frac{1}{2}(x^i - \mu_k)^T \Sigma^{-1} (x^i - \mu_k) - \frac{1}{2} \log |\Sigma| + c \right]$$

- Maximize w.r.t . parameters

$$\pi_k = \frac{\sum_{i=1}^{N} (h^i = k)}{N}$$

$$\mu_k = \frac{\sum_{i=1}^{N} (h^i = k) x^i}{N}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} (h^i = k)(x^i - \mu)(x^i - \mu)^T}{N}$$

Expected Log-Likelihood

- Given: Probability of assignment

$$< \log P(x, h|\theta) >_q \quad = \quad \sum_{i=1}^{N} \sum_{k=1}^{K} q(h^i = k) \log \pi_k + q(h^i = k) \left[-\frac{1}{2}(x^i - \mu_k)^T \Sigma^{-1} (x^i - \mu_k) - \frac{1}{2} \log |\Sigma| + c \right]$$

- Maximize w.r.t . parameters
  - M-step!

$$\pi_k \quad = \quad \frac{\sum_{i=1}^{N} q(h^i = k)}{N} = \frac{\sum_{i=1}^{N} R_{i,k}}{N}$$

$$\mu_k \quad = \quad \frac{\sum_{i=1}^{N} q(h^i = k) x^i}{N} = \frac{\sum_{i=1}^{N} R_{i,k} x^i}{N}$$

$$\Sigma_k \quad = \quad \frac{\sum_{i=1}^{N} q(h^i = k)(x^i - \mu)(x^i - \mu)^T}{N} = \frac{\sum_{i=1}^{N} R_{i,k}(x^i - \mu)(x^i - \mu)^T}{N}$$

Lecture outline

Bayes' rule and generative models

Density estimation

Gaussian distributions

Mixture-of-Gaussian models

Expectation-Maximization algorithm and hidden variables

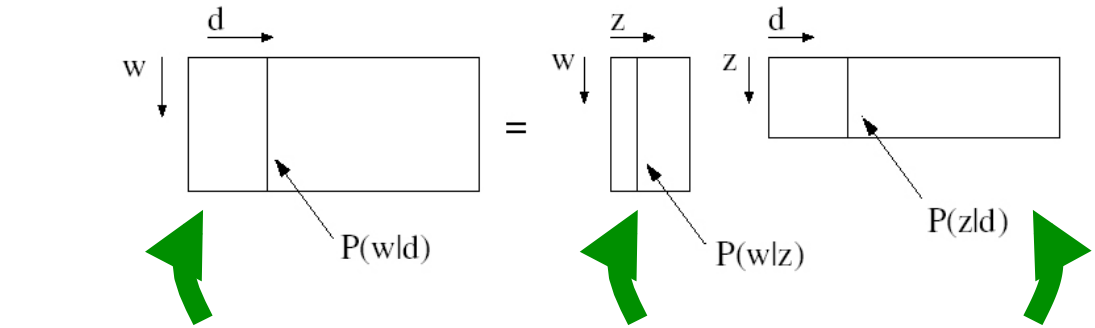
Factor Analysis & PCA

Mixed Discrete/Continuous hidden variable models

Parametric deformable models

P(Grades|MVA)

- 10 students, 20 courses
  - How can we model the distribution of the grades?
  - Consider a Gaussian distribution..
    - 20X19/2 Parameters in covariance, 10 measurements
  - Could we `summarize' performance in a more compact way?
- 3 `hidden' causes
  - Math skills, CS skills, Effort
  - Different skills per student
  - Different effects of skills on grade per course



Generative Model: Factor Analysis

- Hidden variables (skills) $h \propto N(0, I)$
- Observations
  - `factor loading' matrix Λ (course-specific effect of skills on grade)
  - noise covariance matrix Ψ (performance on exam)

- Linear model $x = \mu + \Lambda h + w, \quad w \propto N(0, \Psi)$

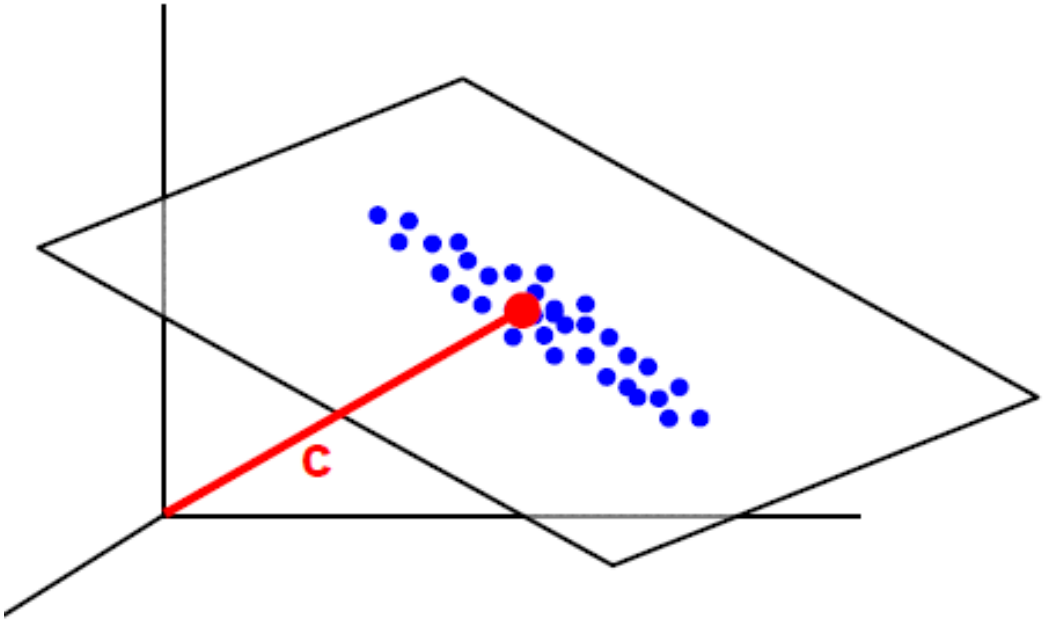
- Distribution of x (see end of slides)

$$P(x, h|\theta) = \frac{1}{\sqrt{(2\pi)^{n/2}|\Sigma|}} \exp\left(\begin{pmatrix} h \\ x-\mu \end{pmatrix}^T \begin{bmatrix} I & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \Psi \end{bmatrix} \begin{pmatrix} h \\ x-\mu \end{pmatrix}\right)$$

- Density estimation: recover optimal μ, Λ, Ψ, for a set of data X

Continuous Hidden Variables: Factor Analysis

- Find low-dimensional subspace (`skills') explaining data
- Hidden variables: coordinates on subspace
  - E-step: posterior on coordinates
  - M-step: subspace



EM for Factor Analysis

- E-step: distribution on h(skills), conditioned on x (grades)

$$P(x,h|\theta) = \frac{1}{\sqrt{(2\pi)^{n/2}|\Sigma|}} \exp\left(\begin{pmatrix} h \\ x-\mu \end{pmatrix}^T \begin{bmatrix} I & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \Psi \end{bmatrix} \begin{pmatrix} h \\ x-\mu \end{pmatrix}\right)$$

$$P(x,h|\theta) = P(h|x,\theta)P(x|\theta)$$

$$P(h|x) = \frac{1}{(2\pi)^{k/2}|\Sigma_{1|2}|} \exp\left((h-\mu_{1|2})^T \Sigma_{1|2}^{-1}(h-\mu_{1|2})\right)$$

$$\Sigma_{1|2} = I - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda$$

$$\mu_{1|2} = \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}(x-\mu)$$

- M-step: plug in distribution on h, and maximize w.r.t. parameters

$$\Lambda = \left(\sum_{i=1}^N x^i E(h^{iT})\right)\left(\sum_{i=1}^N E(h^i h^{iT})\right)^{-1} \qquad \Psi = \frac{1}{N}\mathrm{diag}\left\{\left(\sum_{i=1}^N x^i x^{iT} - \Lambda E(h^i)x^{iT}\right)\right\}$$

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

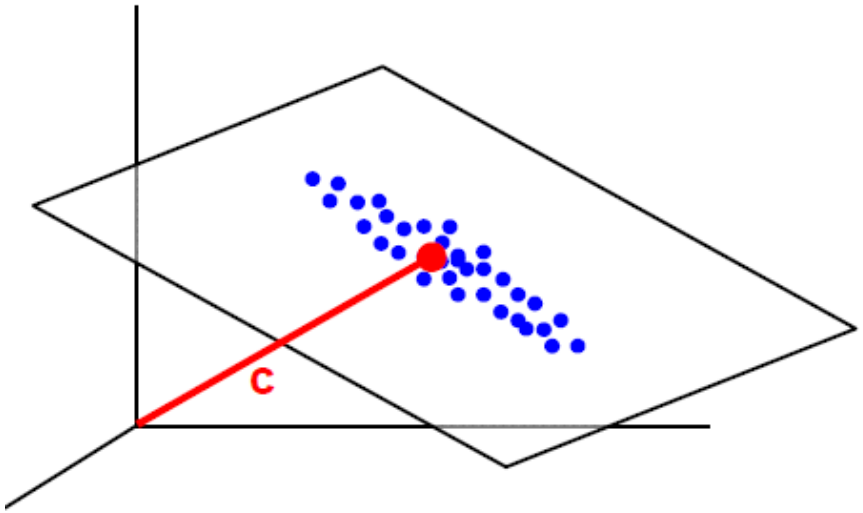
Principal Component Analysis (PCA)

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

$$x^i = \sum_{j=1}^{M} h_j^i b_j$$

$$= \sum_{j=1}^{M} (h^{iT} b_j) b_j$$

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



Relation with Factor Analysis?

- PCA criterion:

$$\min_{\Lambda,h} \frac{1}{N} \sum_{i=1}^{N} (x^i - \Lambda h^i)^2$$

- Regularize solution

$$\min_{B,h} \frac{1}{N} \sum_{i=1}^{N} (x^i - \Lambda h^i)^2 + c|h^i|^2$$

- Equivalently:

$$\min_{B,h} \underbrace{\frac{1}{2\sigma^2} \frac{1}{N} \sum_{i=1}^{N} (x^i - \Lambda h^i)^2}_{-\log P(x|h,\Lambda)+a_1} + \underbrace{|h^i|^2}_{-\log P(h)+a_2}$$

- Difference from FA:

$$\Psi \rightarrow \sigma^2 I.$$

- What we gain: no EM, factorization-based estimate of Λ, h
- What we lose: proper probabilistic framework.

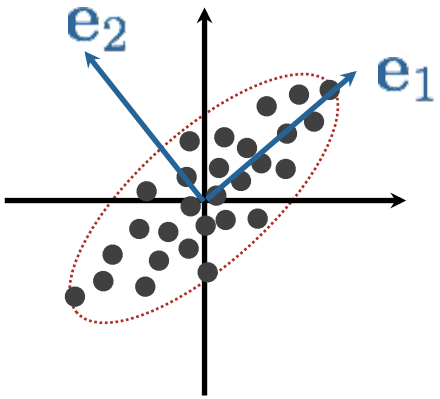
Principal component analysis

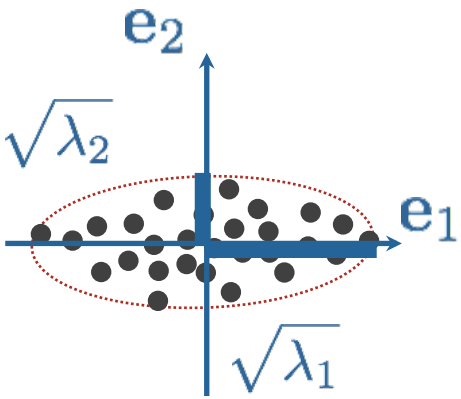
- The *k* orthogonal directions that capture most of the data variance are the *k* leading (largest-eigenvalue) covariance eigenvectors

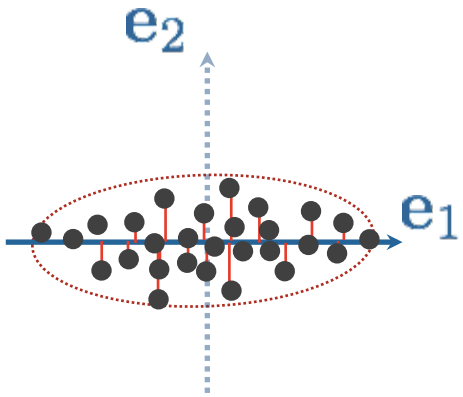
Factor Analysis	PCA
Λ matrix	Leading K eigenvectors of covariance
Hidden variables	Inner product of data with eigenvectors

PCA: decorrelation/dimensionality reduction

- `Hidden variables': projection onto eigenvectors of covariance matrix







$$\Sigma = \begin{pmatrix} 2.45 & 1.2 \\ 1.2 & 2.1 \end{pmatrix} \qquad \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Dimensionality reduction

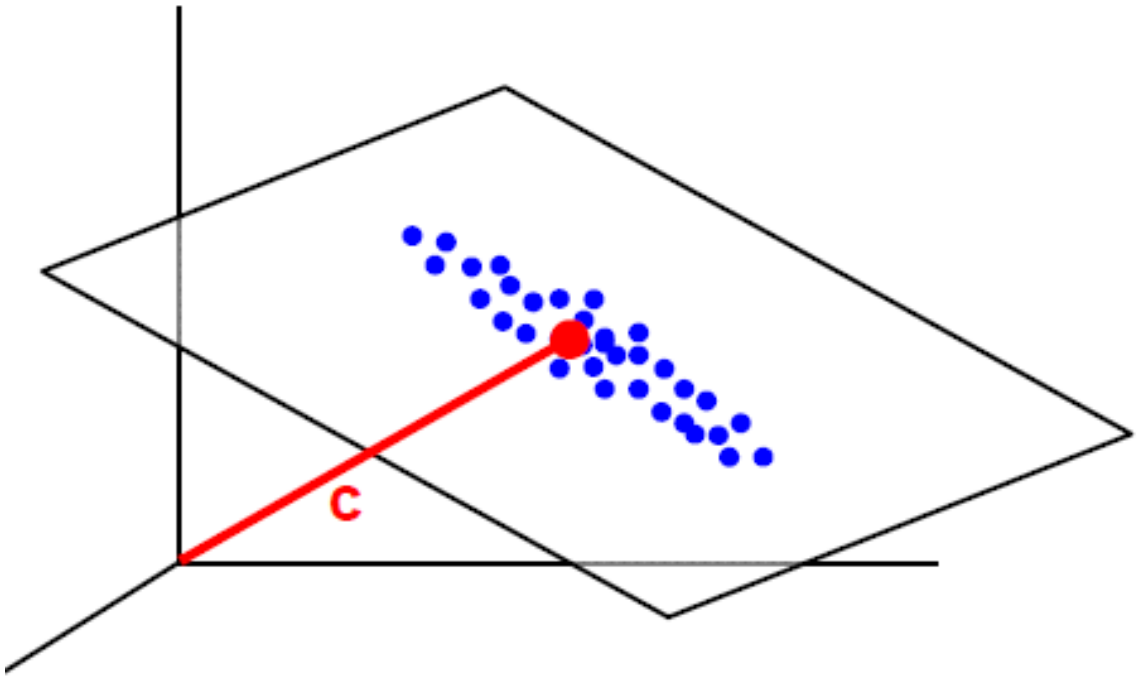
Grades in 60 courses -> Good in math, computer science

Lecture outline

- Bayes' rule and generative models
- Density estimation
  - Gaussian distributions
  - Mixture-of-Gaussian models
    - Expectation-Maximization algorithm and hidden variables
  - Factor Analysis & PCA
  - Mixed Discrete/Continuous hidden variable models
- Parametric deformable models

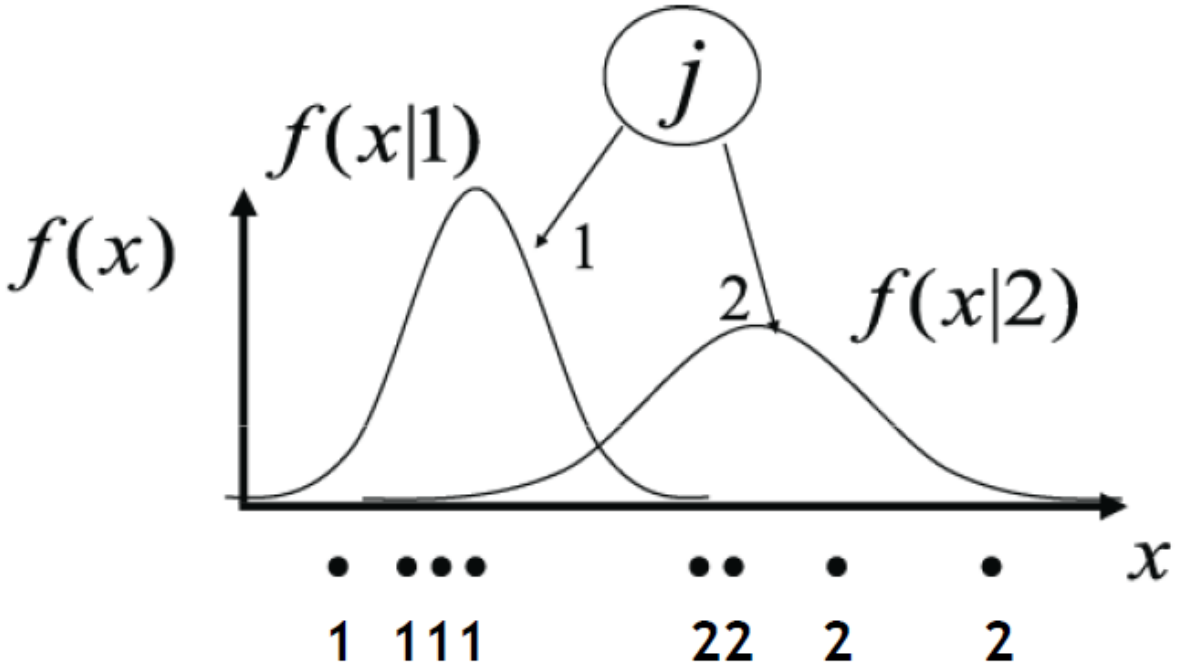
Continuous Hidden Variables: Factor Analysis

- Also known as Dimensionality Reduction



Discrete hidden variables: Mixture of Gaussians

- Also known as Clustering

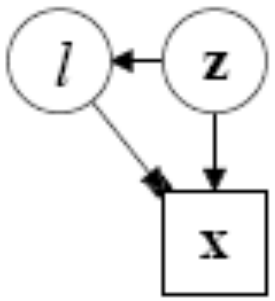


Transformation-invariant image averaging

Consider shift as a hidden variable, l

Estimate model with EM

Shift $p(\mathbf{z})$



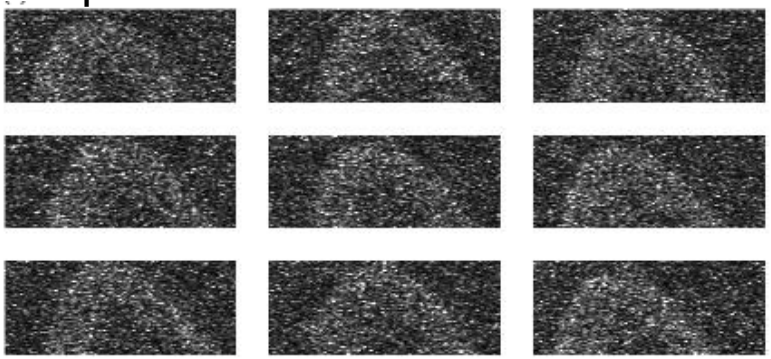
Deformation-free image

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

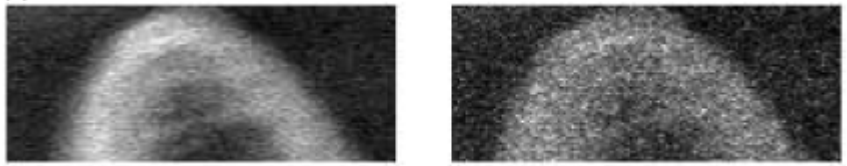
Observed Image

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

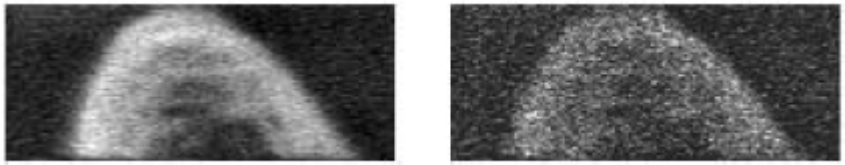
Input



Plain mean & std



With transformation & EM



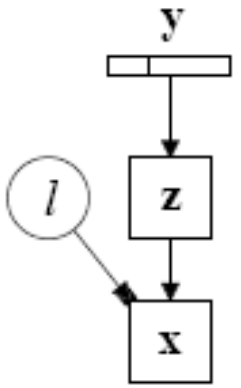
Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, 2003

Transformed Components Analysis

Latent variables for synthesis (continuous)

Latent variables for shift (discrete)

Estimate mean basis using EM

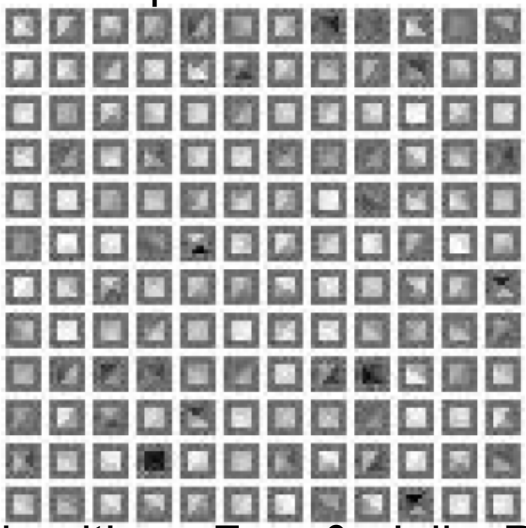


Plain mean & PCA

With offset

Input

Samples of model

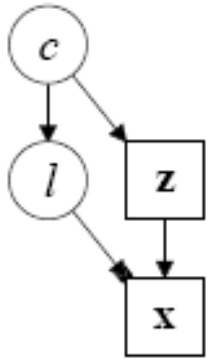


Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, PAMI 2003

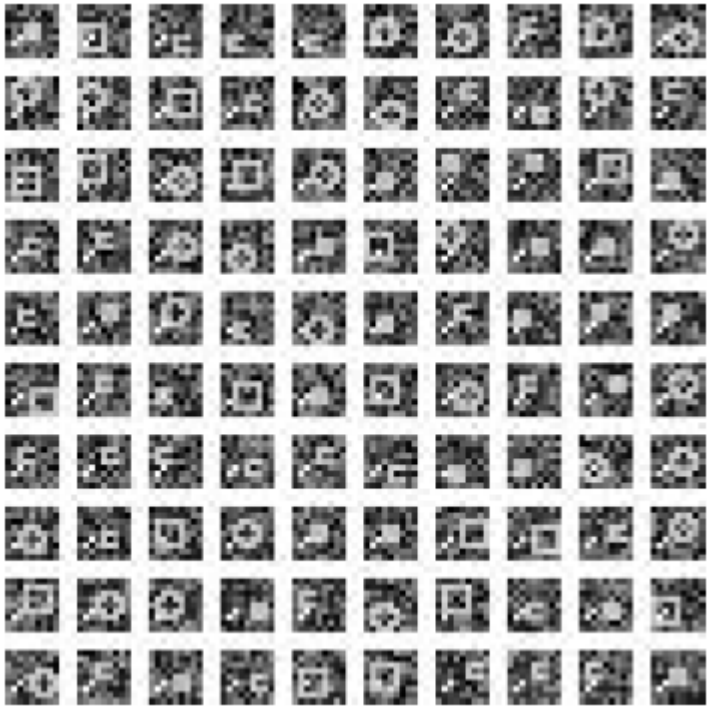
Transformed Mixture of Gaussians

Latent variables for cluster (discrete)

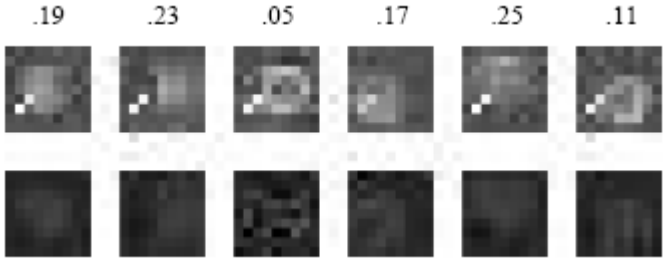
Latent variables for shift (discrete)



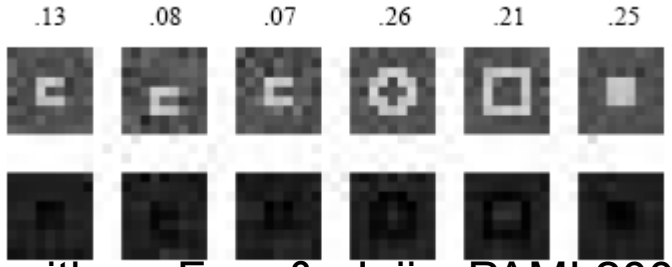
Input



Plain Mixture-of-Gaussians

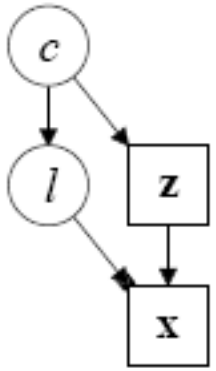


With offset



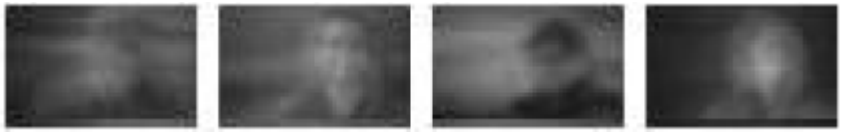
Transformation-Invariant Clustering Using the EM Algorithm, Frey &. Jojic, PAMI 2003

Transformed Mixture of Gaussians



Input

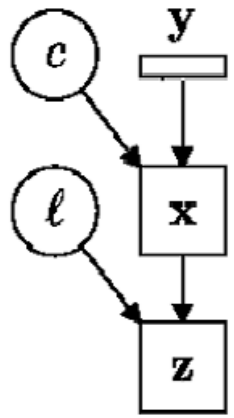
Plain Mixture-of-Gaussians



With offset

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

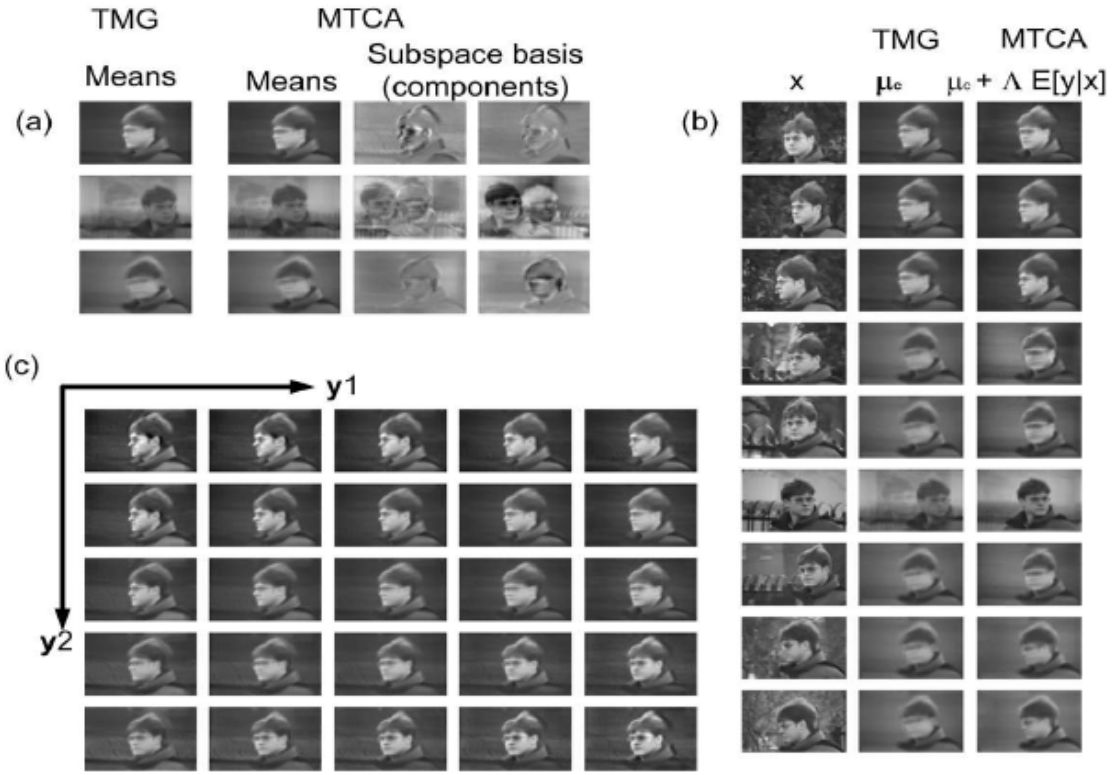
Mixture of Transformed Components



Latent variables for cluster

Latent variables for components

Latent variables for shift



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

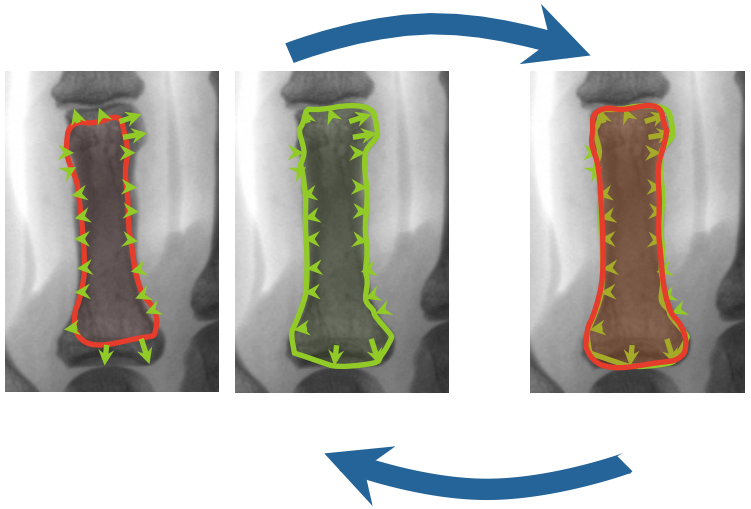
Lecture outline

Bayes' rule and generative models

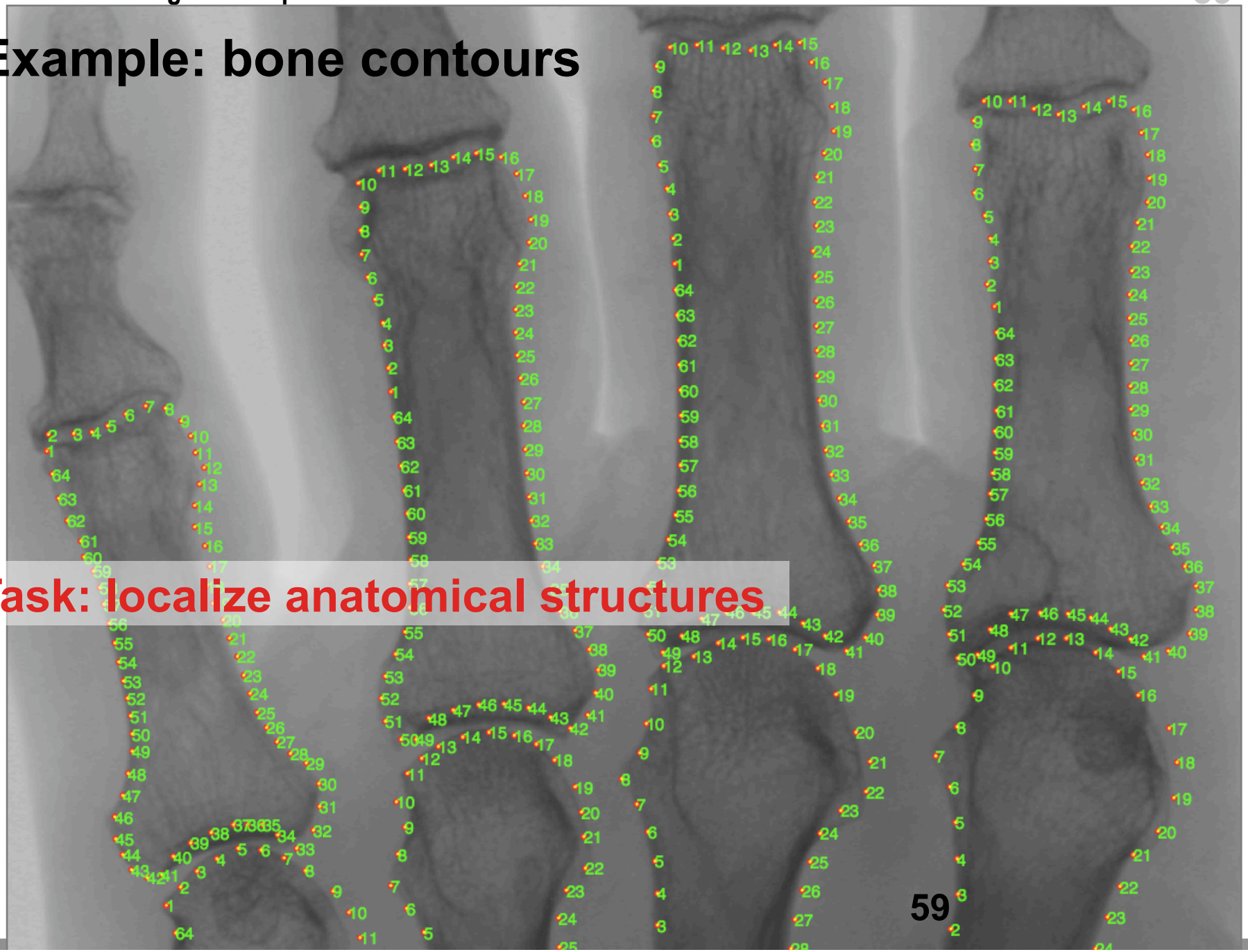
Density estimation

Parametric deformable models

- Statistical active shape models
- Eigenfaces
- Active appearance models
- 3D Morphable models



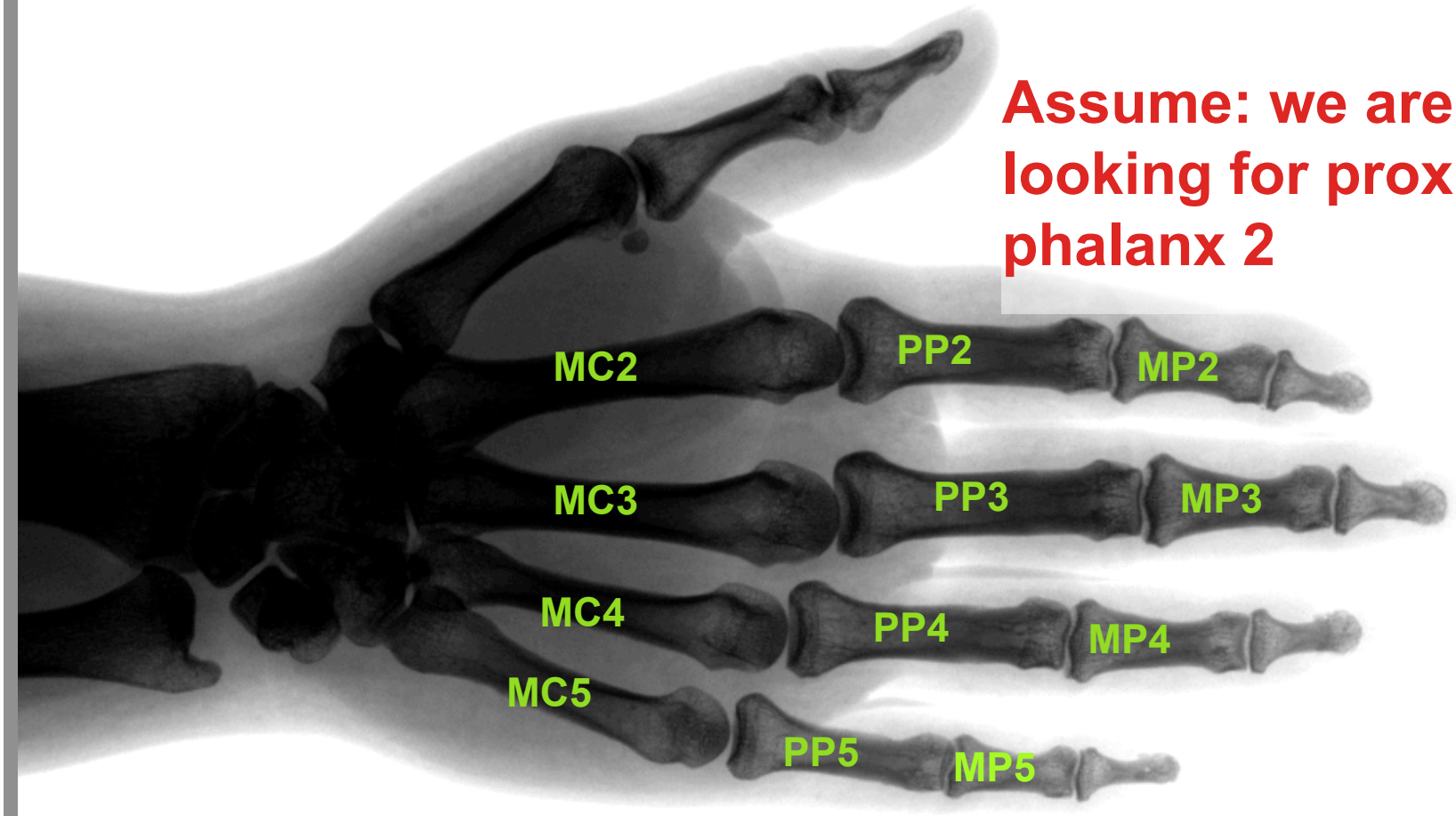
Example: bone contours



Task: localize anatomical structures

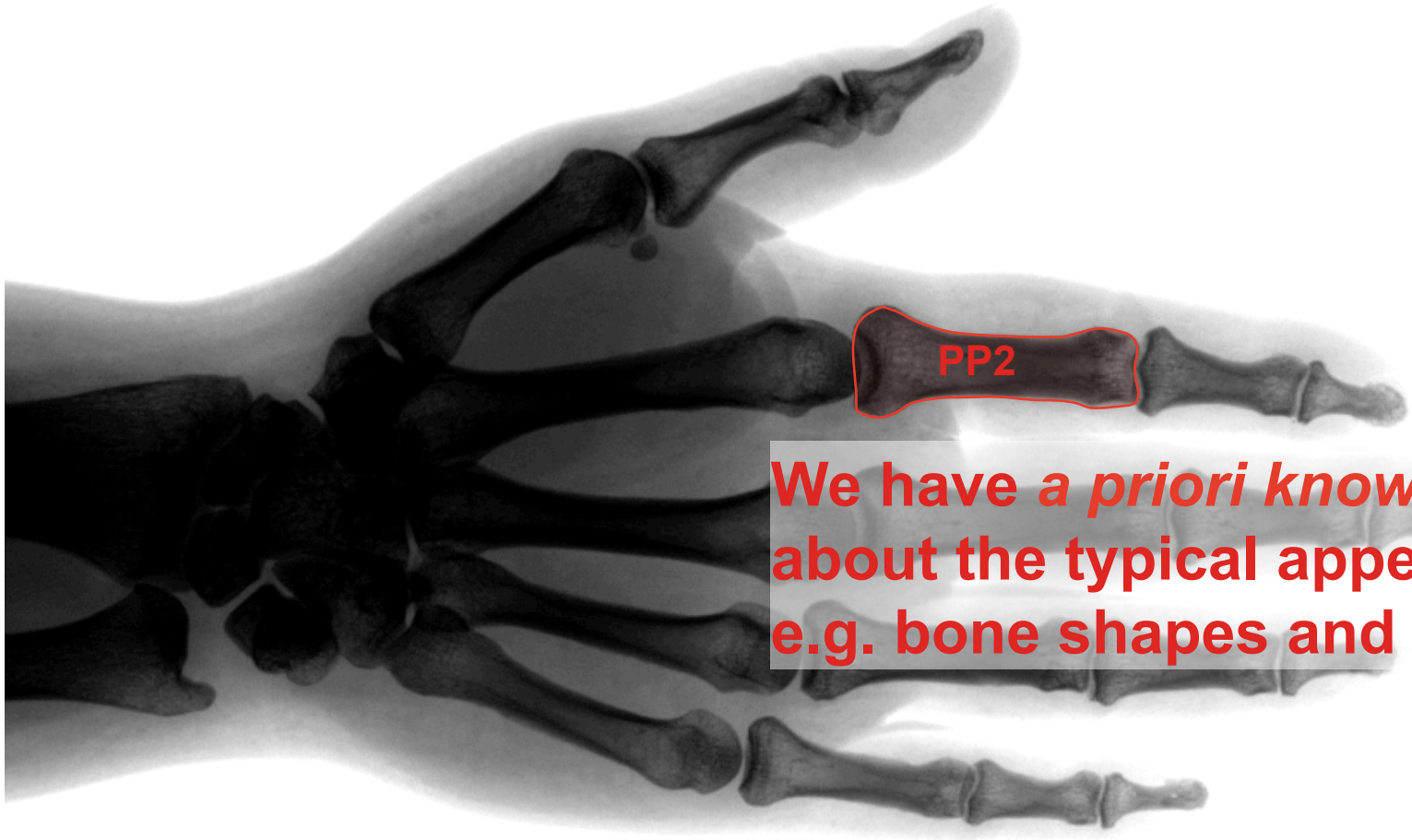
Task: Analyze a hand radiograph

Task: Analyze a hand radiograph



Assume: we are looking for proximal phalanx 2

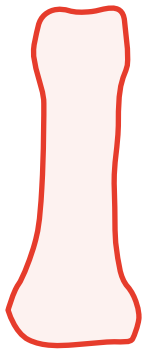
Analyzing a hand radiograph



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?

Statistical Shape Models



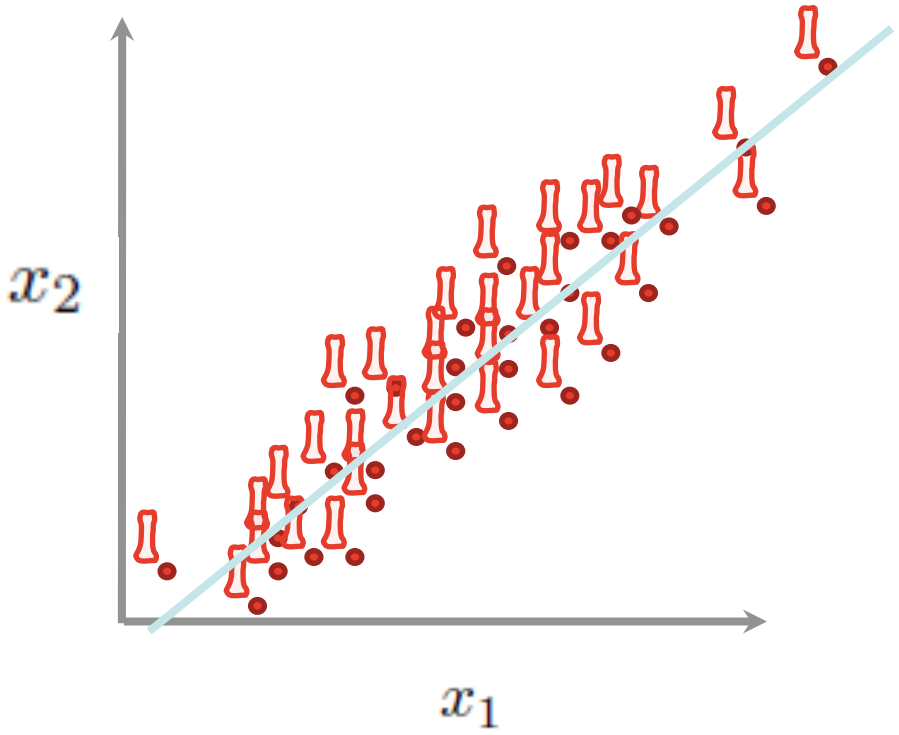
$$\mathbf{x}_i = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_m \\ y_m \end{pmatrix}$$

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

Learning: Model Acquisition
Inference: Model Fitting

The space of all bone shapes

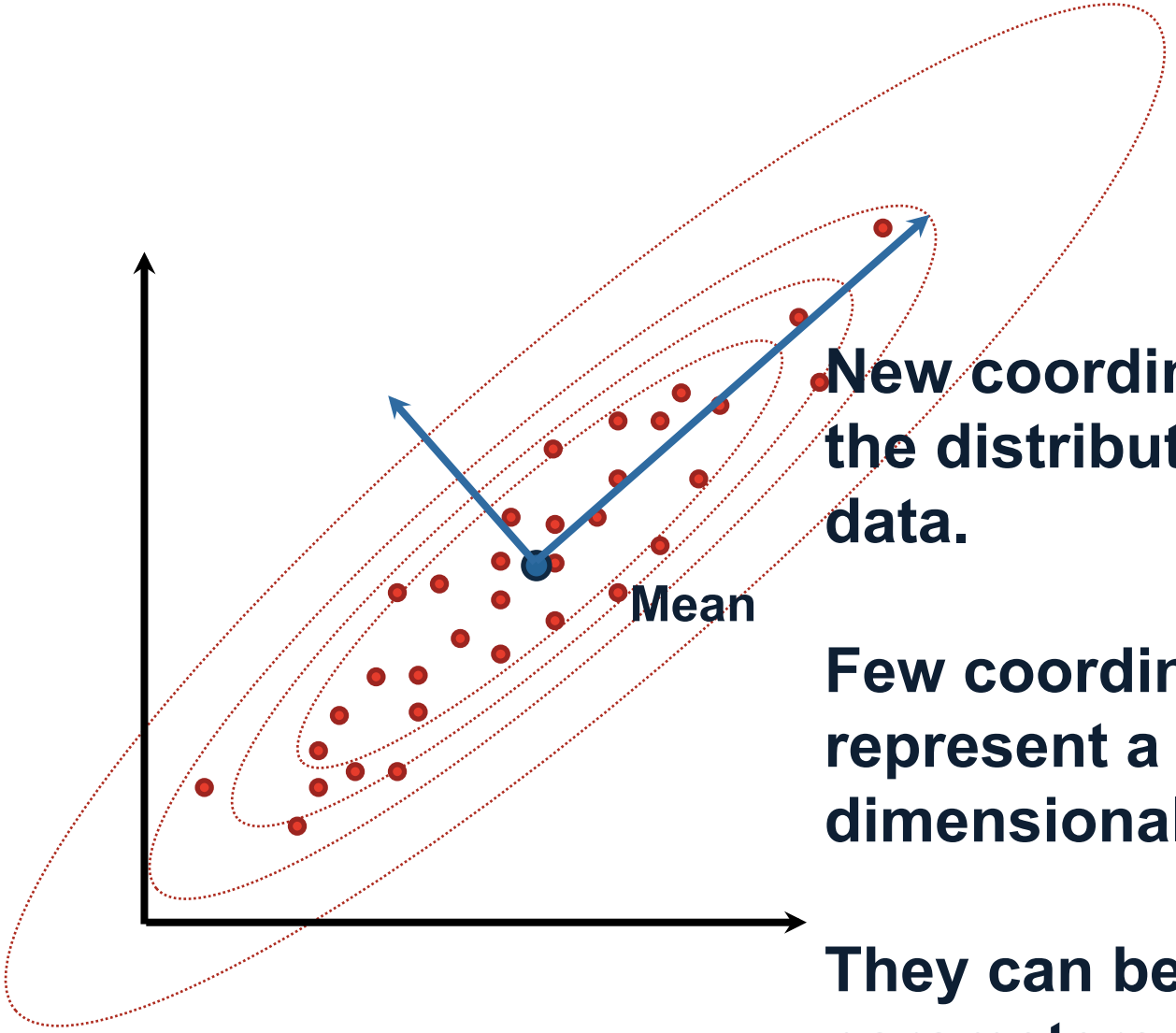
- Bone shapes: vectors in R^{2m}



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

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

New subspace: `better' coordinate system

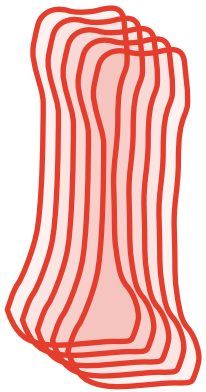


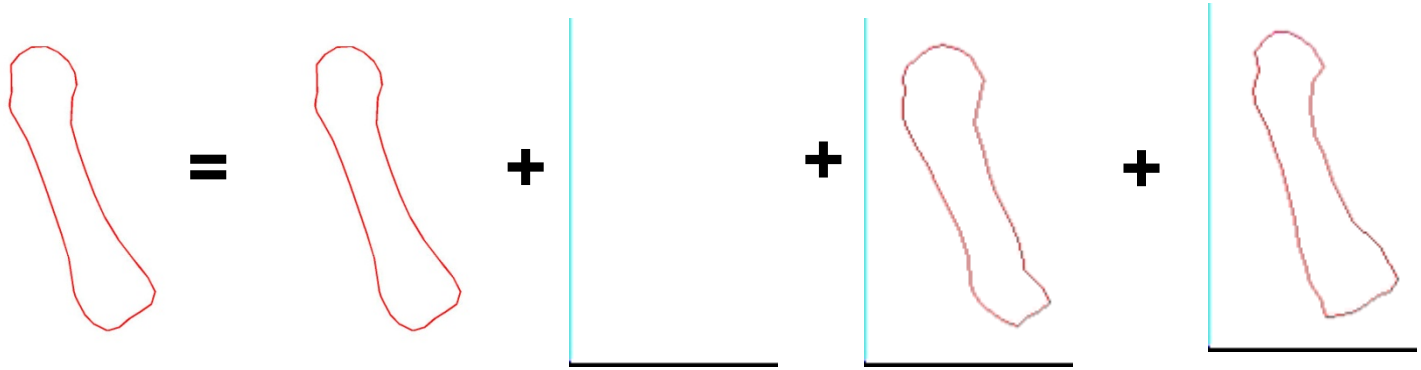
New coordinates reflect the distribution of the data.

Few coordinates suffice to represent a high dimensional vector

They can be viewed as parameters of a model

Using PCA to model shape

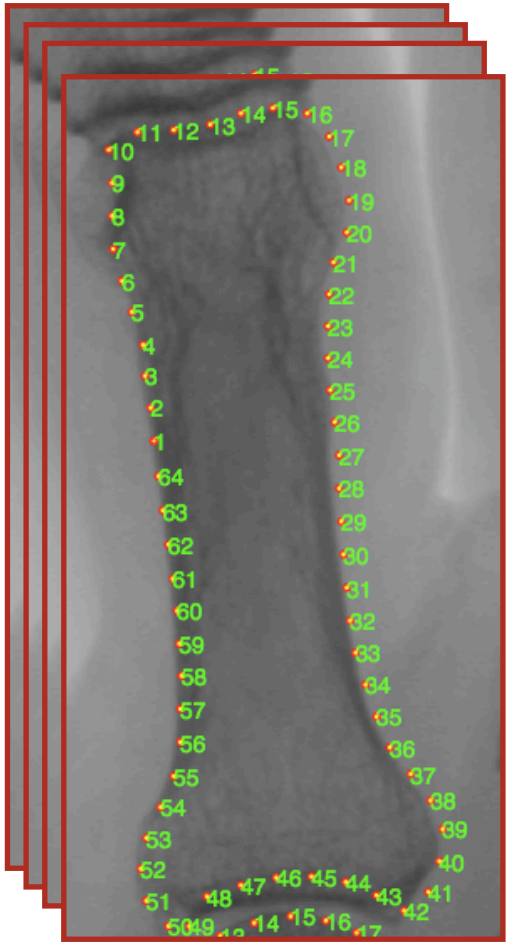




$$\mathbf{x}_{new} = \hat{\mathbf{m}} + b_1\mathbf{e}_1 + b_2\mathbf{e}_2 + b_3\mathbf{e}_3$$

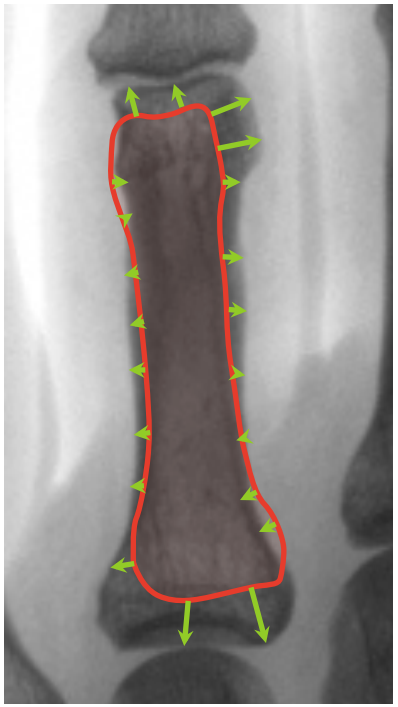
Active shape models (ASM)

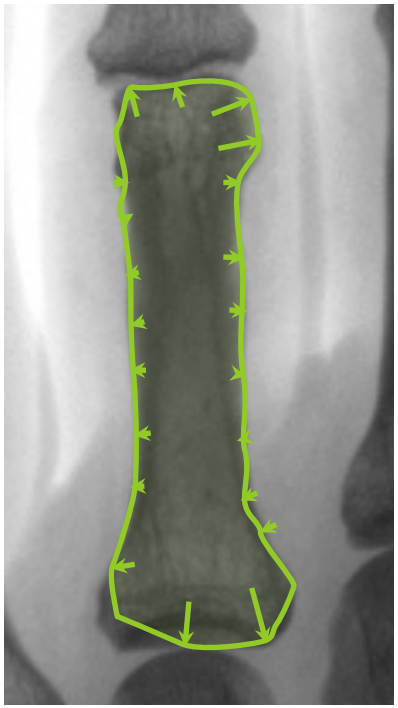
- A set of training examples (images)
- A set of landmarks, that are present on all images
- Build a statistical model of shape variation (PCA)
- Build a statistical model of the local texture (PCA)
- Use the model for the search in a new image



ASM search

Initialize

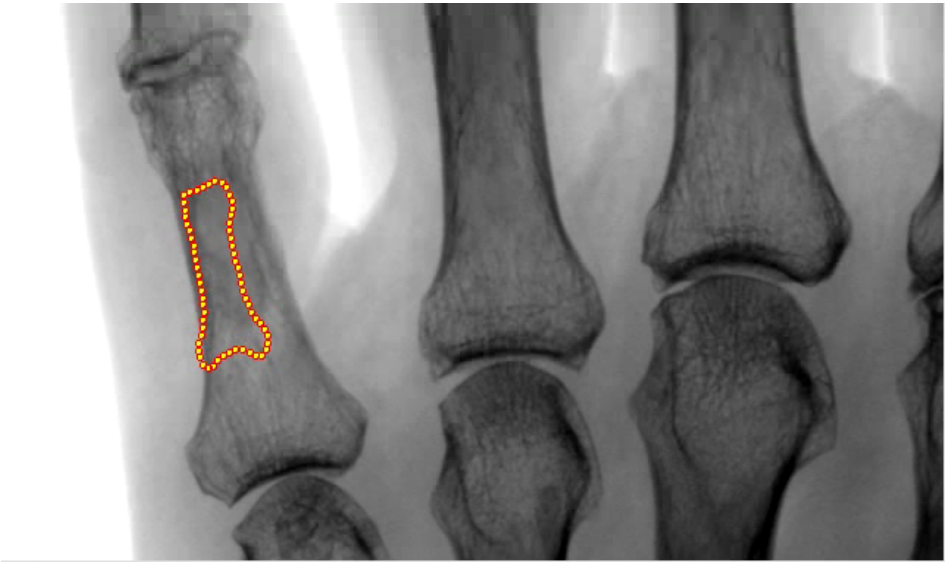


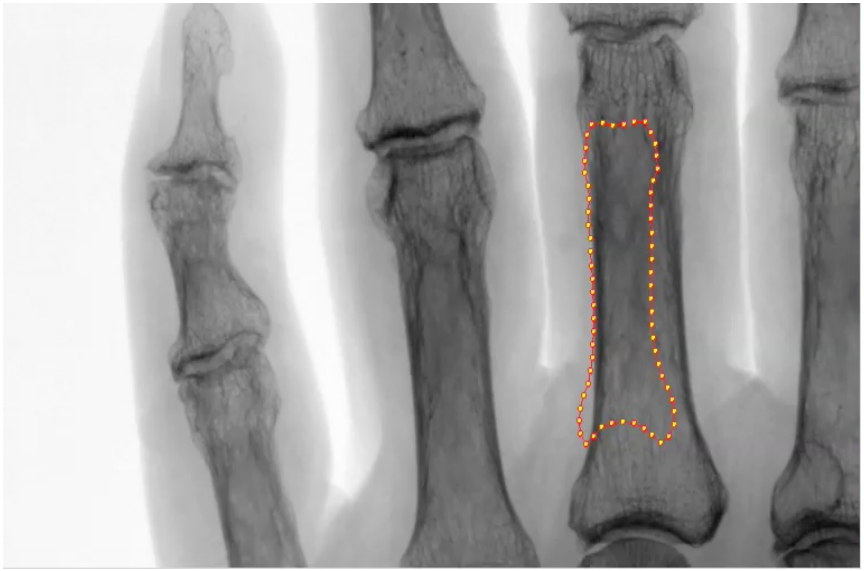


Adjust to texture

Fit to shape model

ASM search





Lecture outline

Bayes' rule and generative models

Density estimation

Parametric deformable models

Statistical active shape models

Eigenfaces

Active appearance models

3D Morphable models

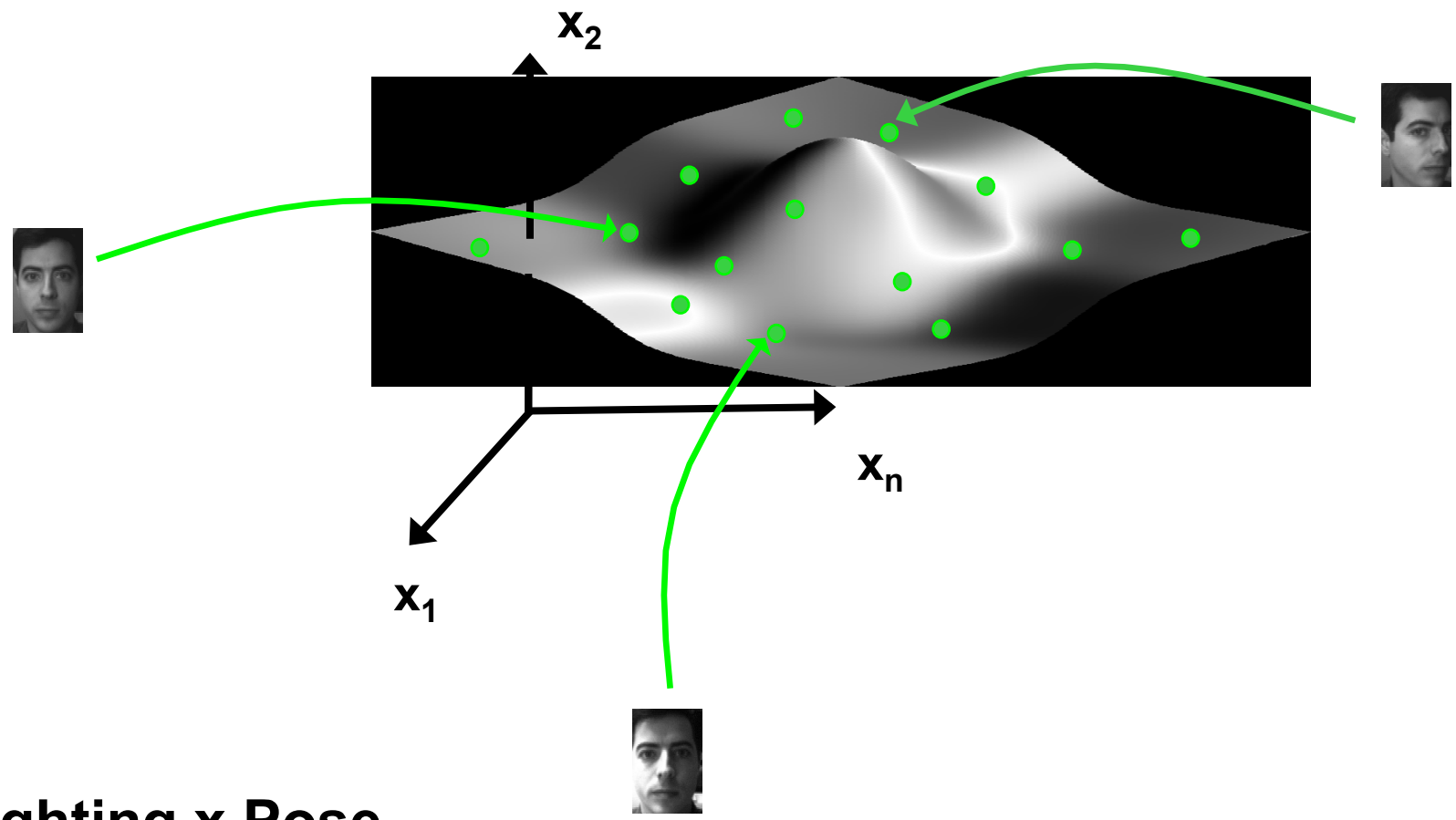
$$\hat{\mathbf{x}} = \boldsymbol{\mu} + w_1\mathbf{u}_1 + w_2\mathbf{u}_2 + w_3\mathbf{u}_3 + w_4\mathbf{u}_4 + \ldots$$

Appearance modelling for faces

- When viewed as vectors of pixel values, face images are extremely high-dimensiona
  - 100x100 image = 10,000 dimensions
- Very few vectors correspond to valid face images

- Original coordinates are not revealing about face properties
- We want to model the subspace (`manifold') of face images

Continuous Hidden Variables: Appearance Manifolds



Lighting x Pose

[Murase and Nayar 1993]

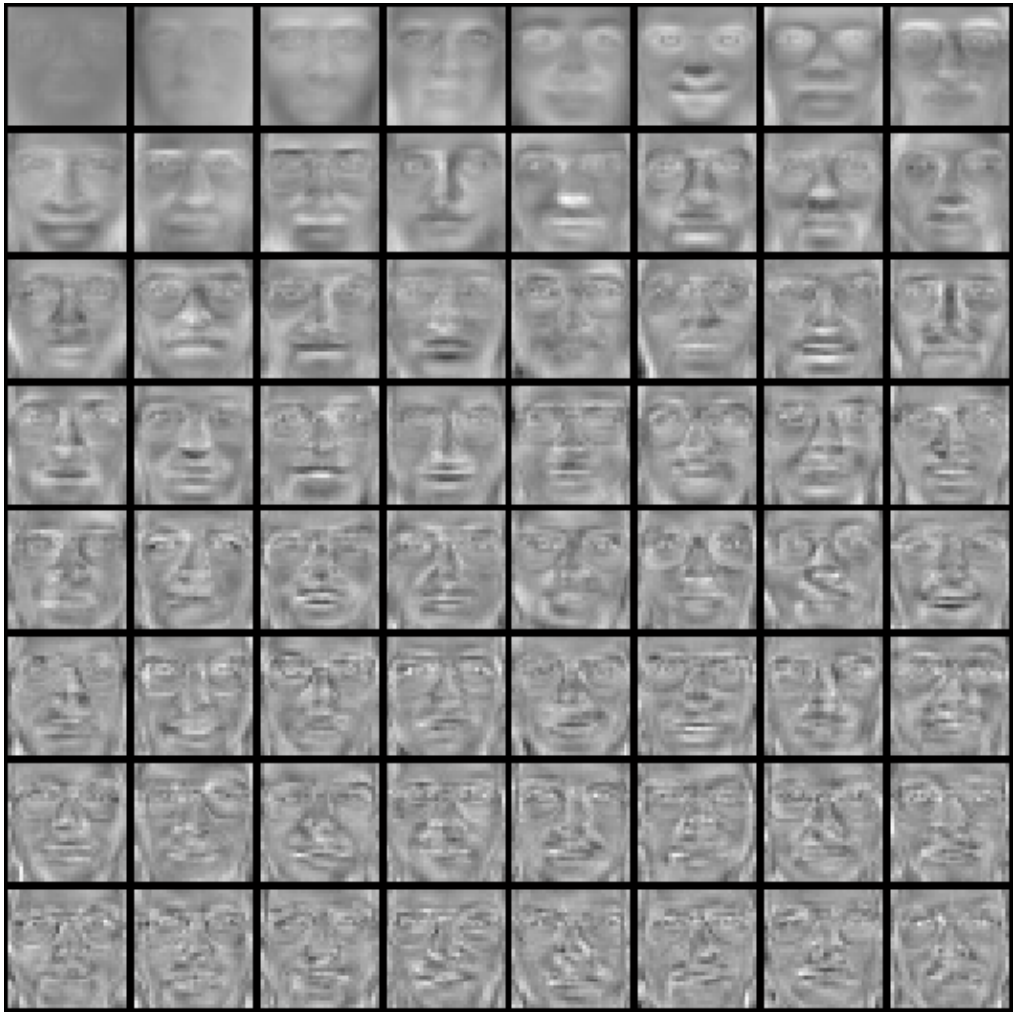
Eigenfaces (Murase & Nayar, 91)

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

Eigenfaces

Mean: $\boldsymbol{\mu}$

Top eigenvectors: $\mathbf{u}_1,\ldots\mathbf{u}_k$



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:

$$\hat{\mathbf{x}} = \boldsymbol{\mu} + w_1\mathbf{u}_1 + w_2\mathbf{u}_2 + w_3\mathbf{u}_3 + w_4\mathbf{u}_4 + \ldots$$

Limitations

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



Lecture outline

Bayes' rule and generative models

Density estimation

Parametric deformable models

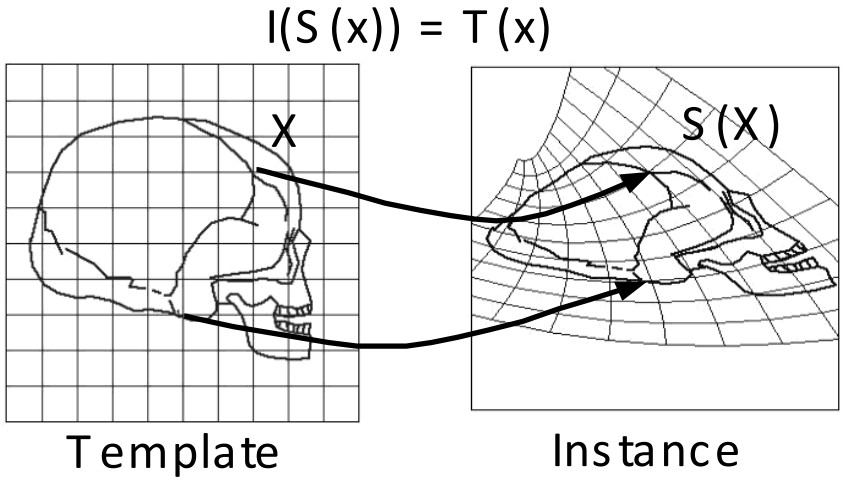
- Statistical active shape models
- Eigenfaces
- Active appearance models
- 3D Morphable models

Active Appearance Models (AAMs)

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

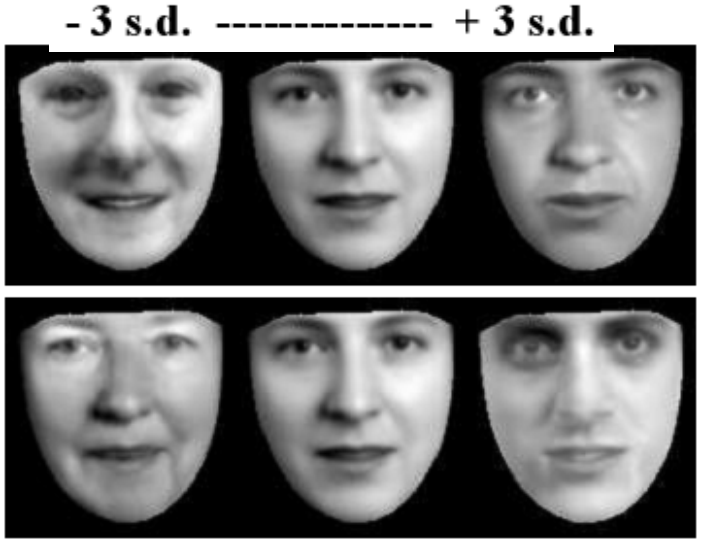
Appearance: $$\mathcal{T}(\mathbf{x};\mathbf{t}) = \sum_i \mathbf{t}_i T_i(\mathbf{x})$$

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

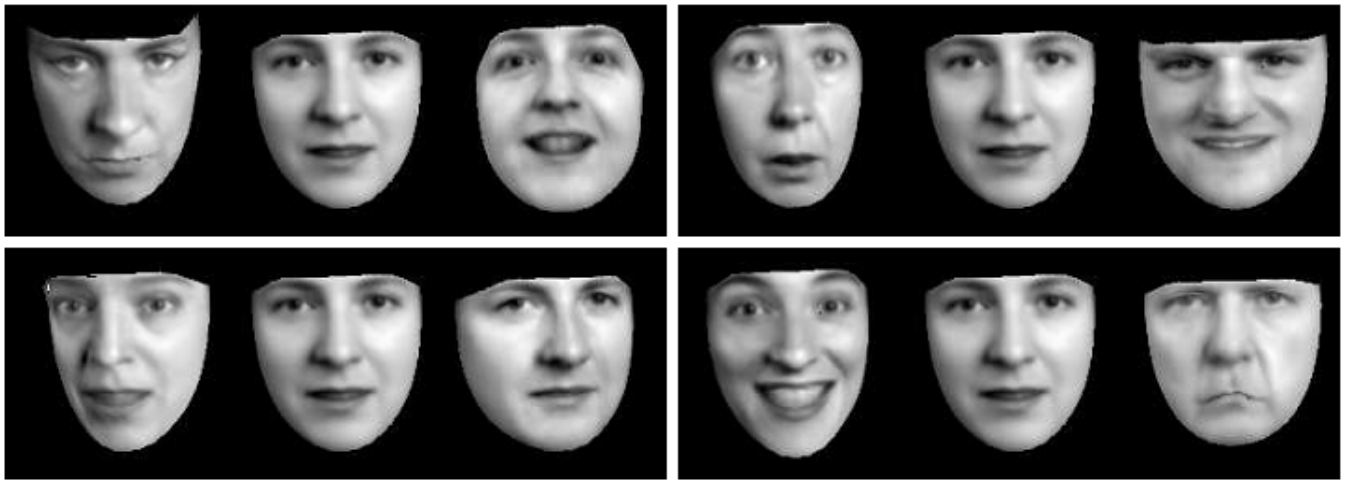


Playing with the AAM parameters

First two modes of shape variation



First two modes of gray-level variation



First four modes of appearance variation

Active Appearance Model Search (Results)

AAM Search

Lecture outline

Bayes' rule and generative models

Density estimation

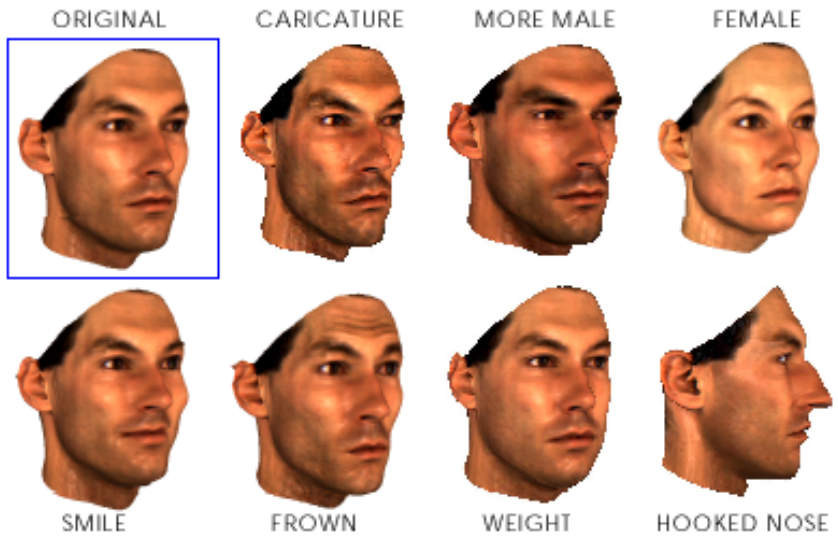
Parametric deformable models

Statistical active shape models

Eigenfaces

Active appearance models

3D Morphable models



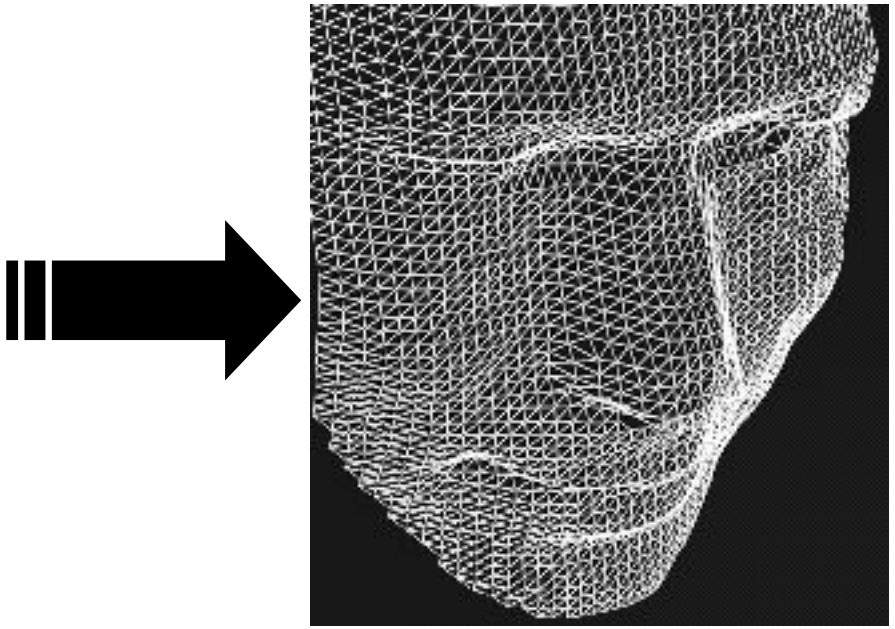
3-D surface acquisition

Laser Range Scanners

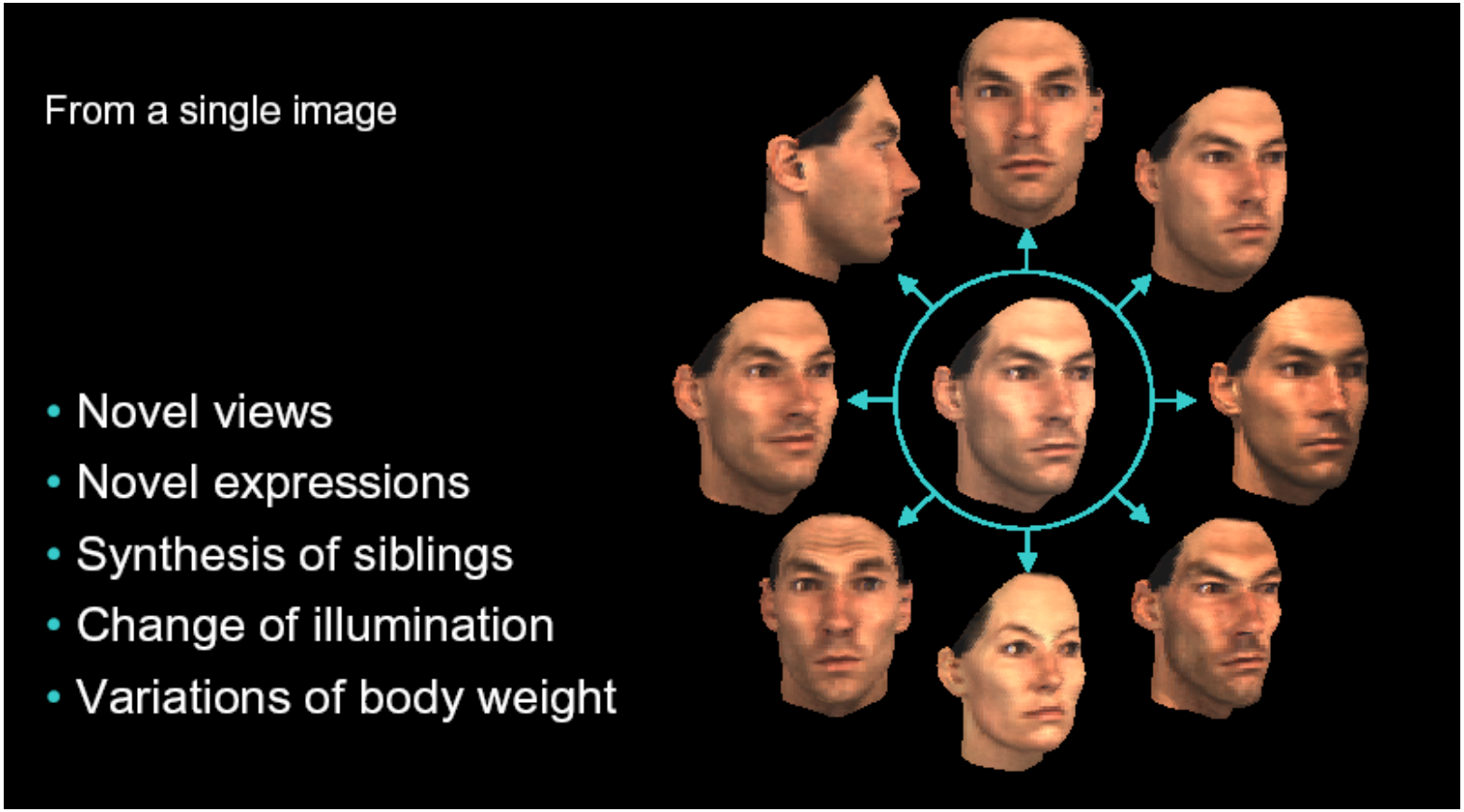
Stereo Cameras

Structured Light (Kinect)

Photometric Stereo

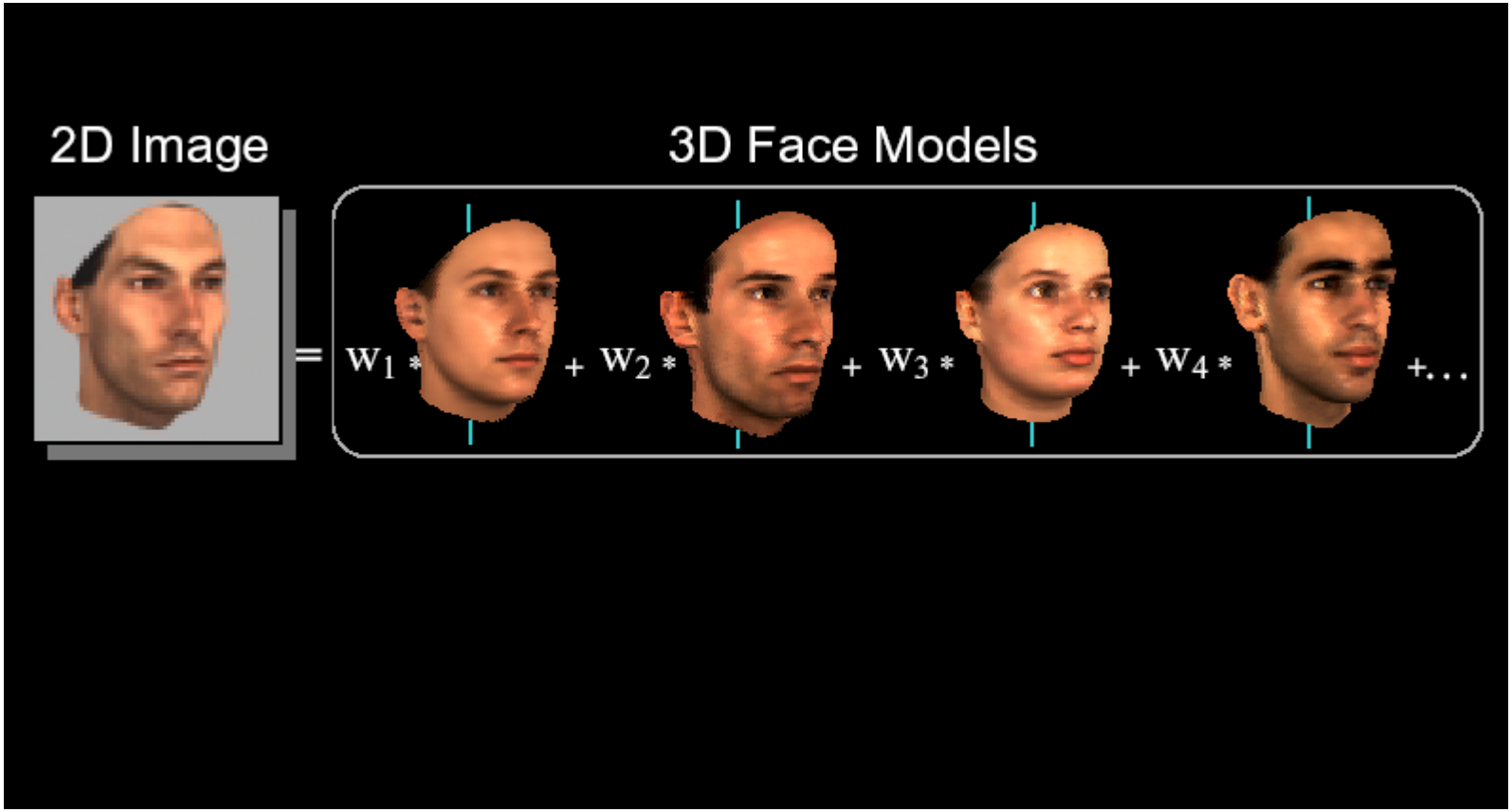


What can we do with 3d shape models?



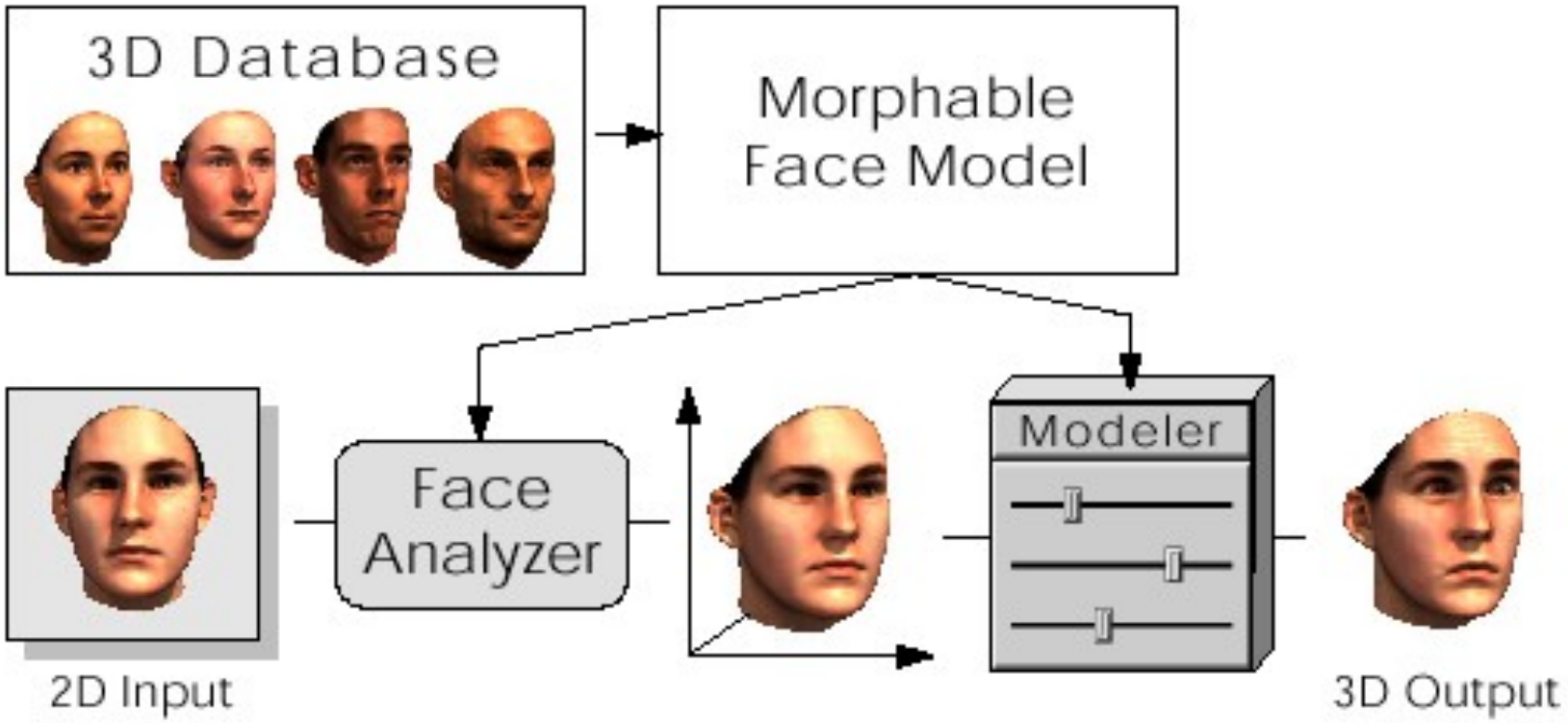
[Blanz and Vetter 1999, 2003]

Building a Morphable Face Model



[Blanz and Vetter 1999, 2003]

3-D Morphable Models



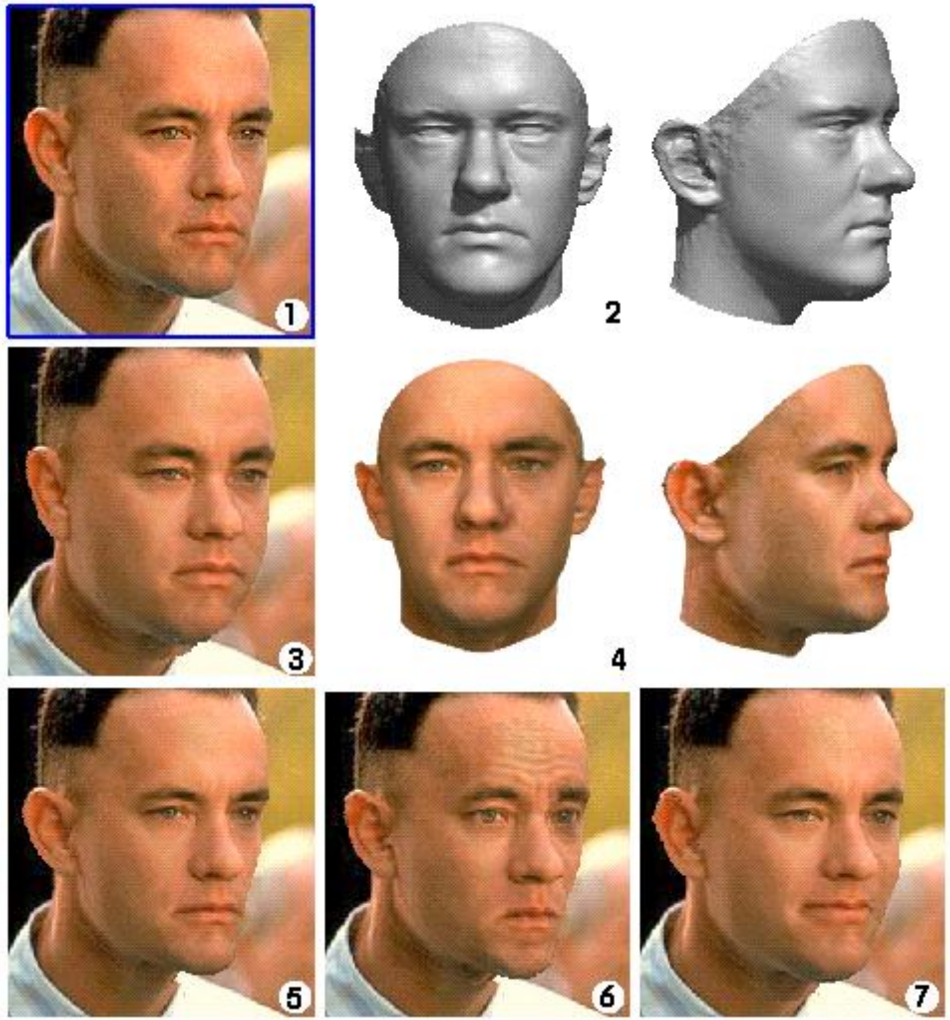
[Blanz and Vetter 1999, 2003]

3D Morphable models

Recover Shape

Synthesize new views

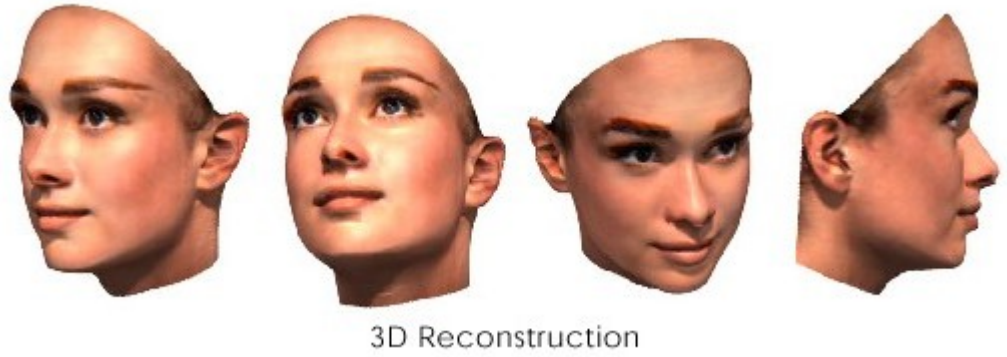
Synthesize new expressions



3-D Morphable Model fitting

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

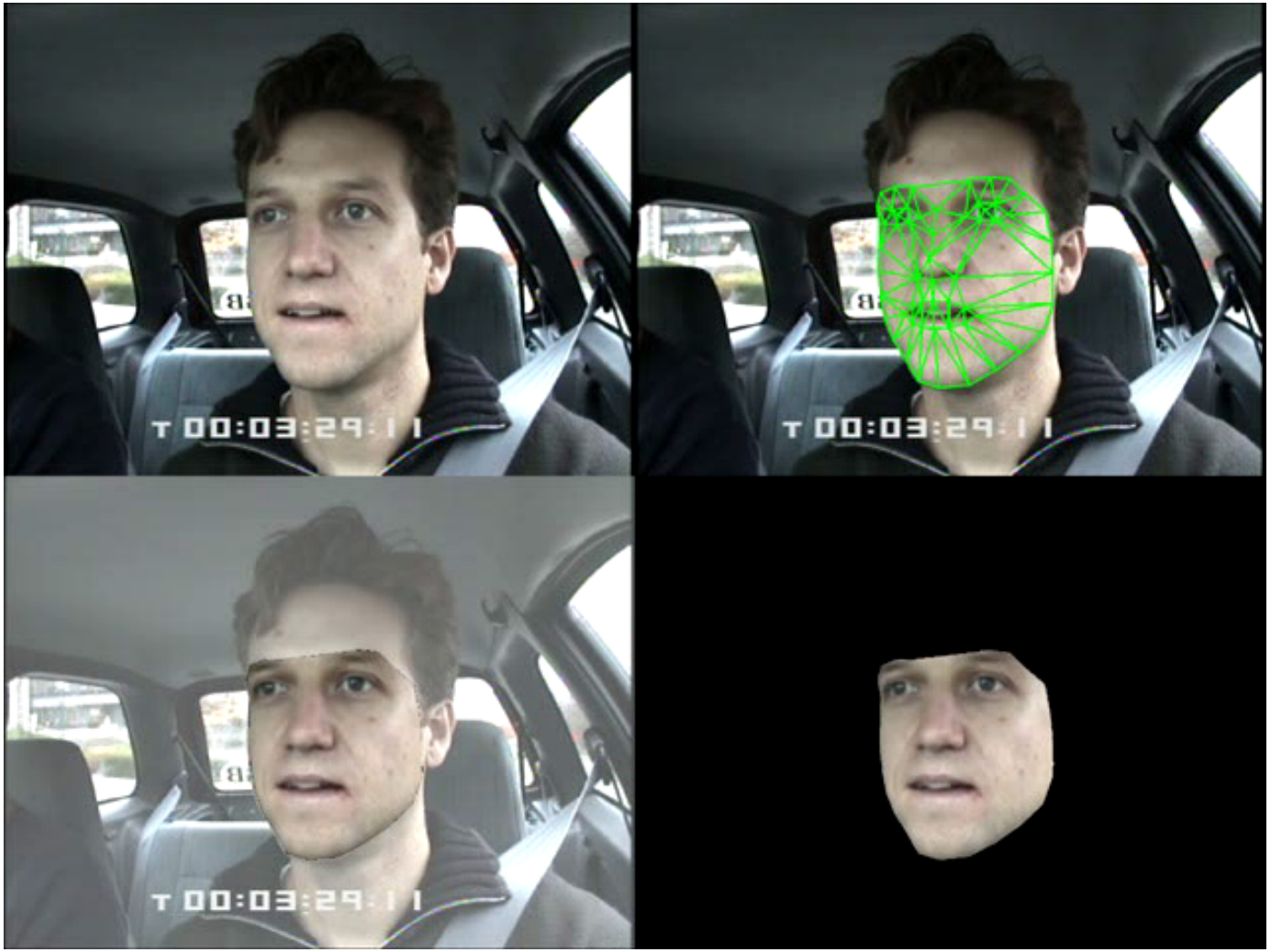
- And then



3D AAM for face tracking

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

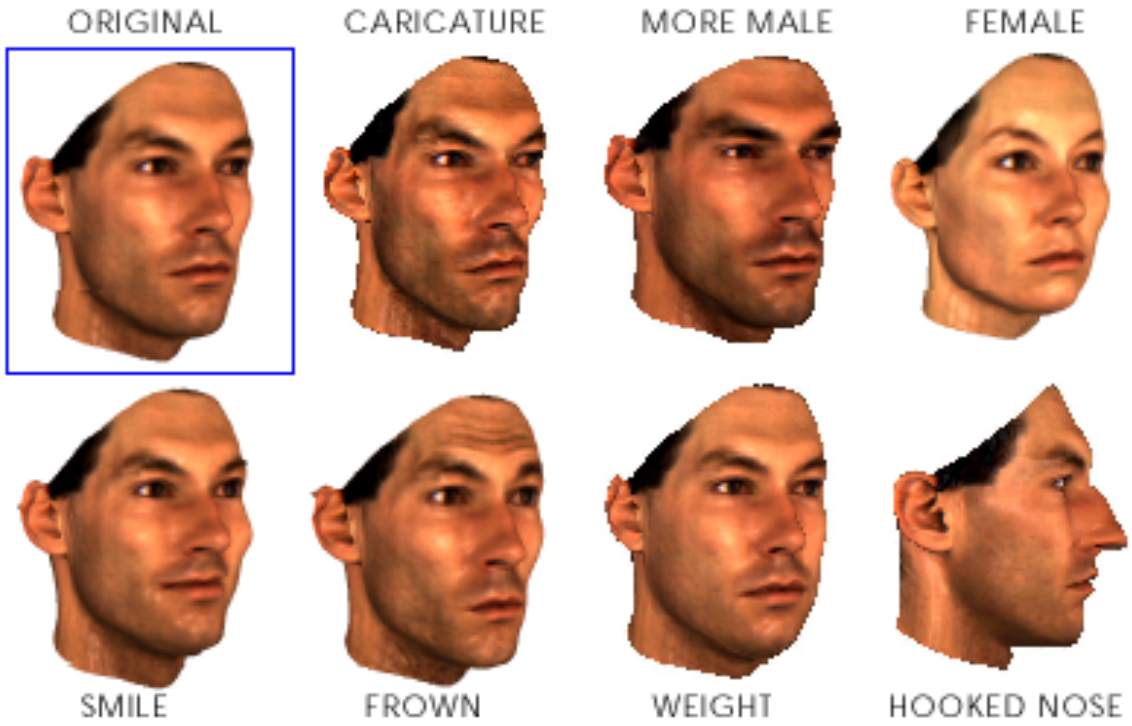
3D AAM for face tracking



Playing with Facial Attributes

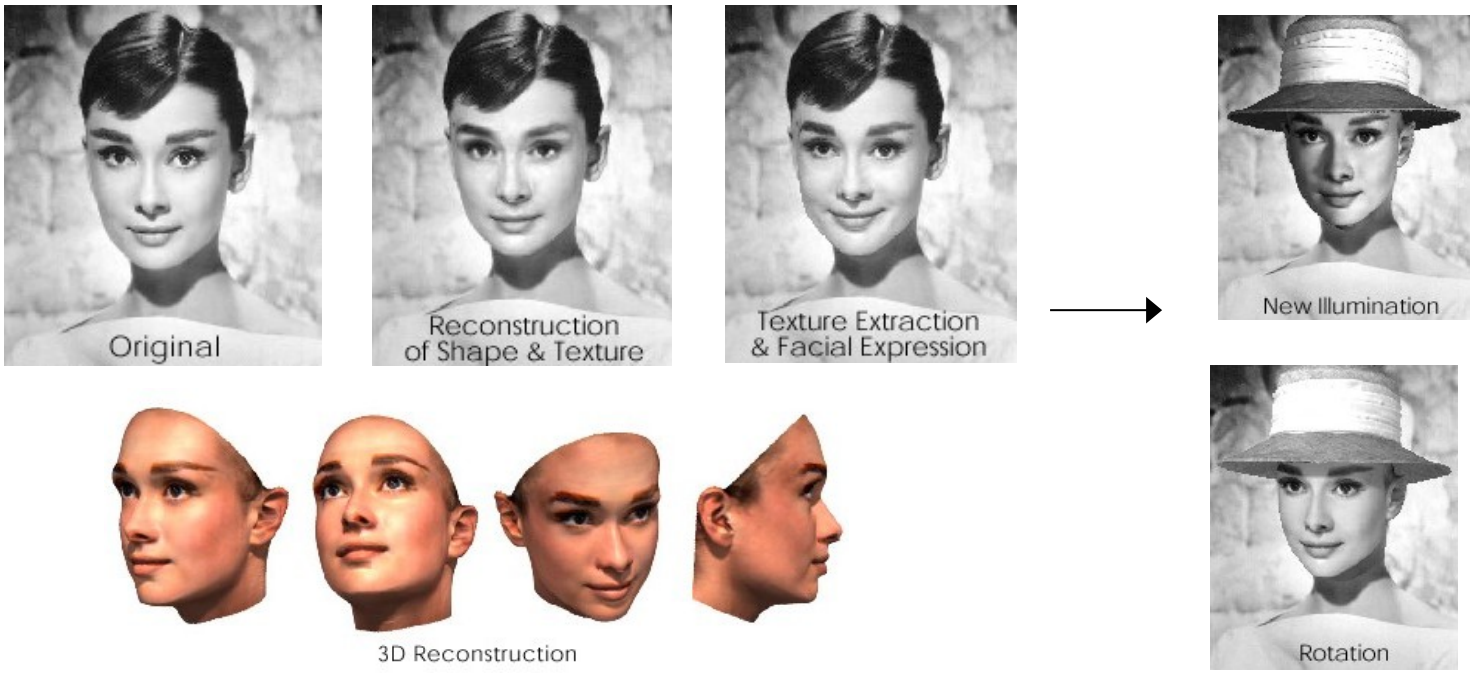
Several classes of attributes are modeled:

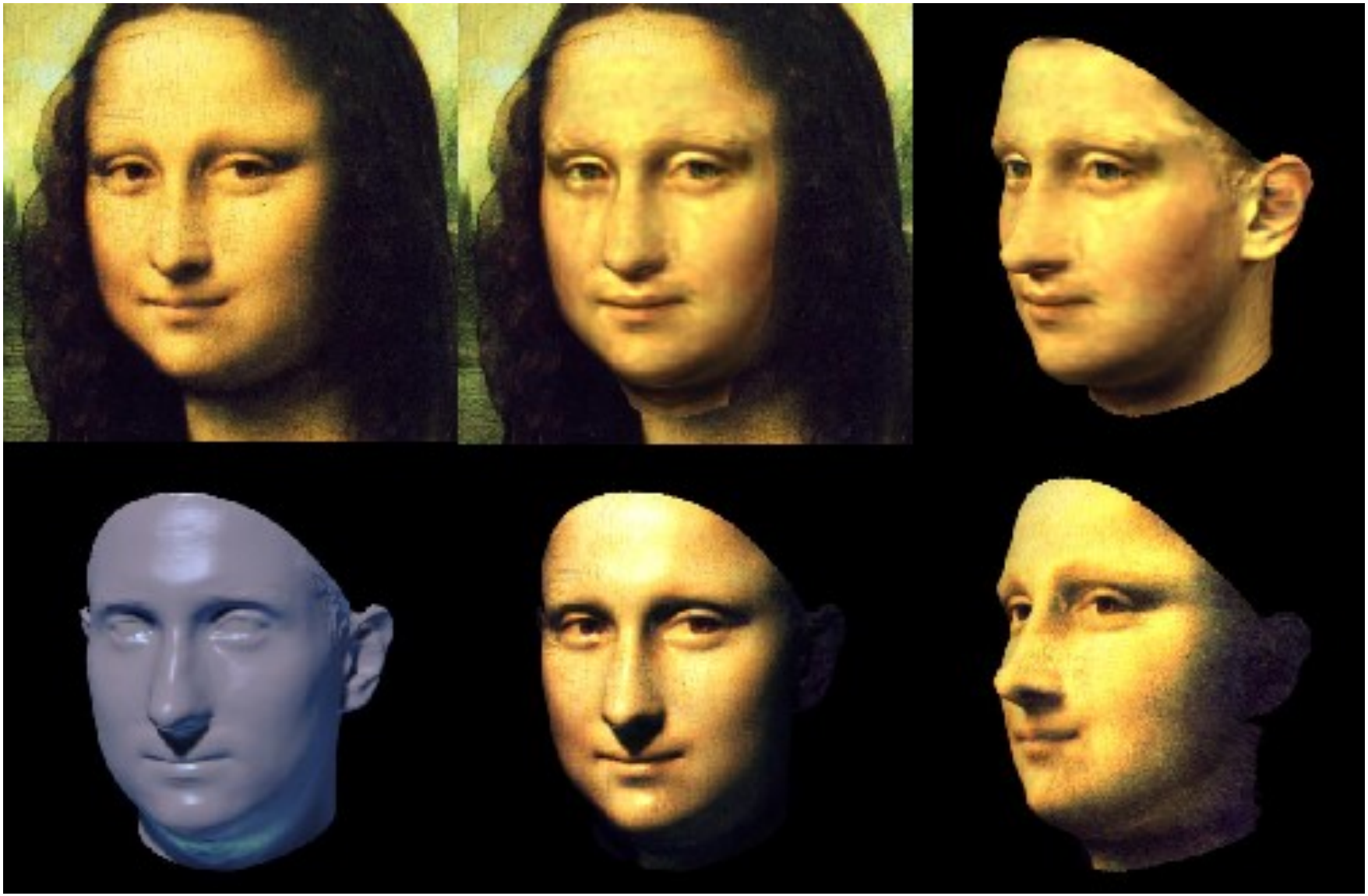
- Facial expressions (smile, frown)
- Individual characteristics (double chin, hooked nose, ‘maleness’)
- Distinctiveness

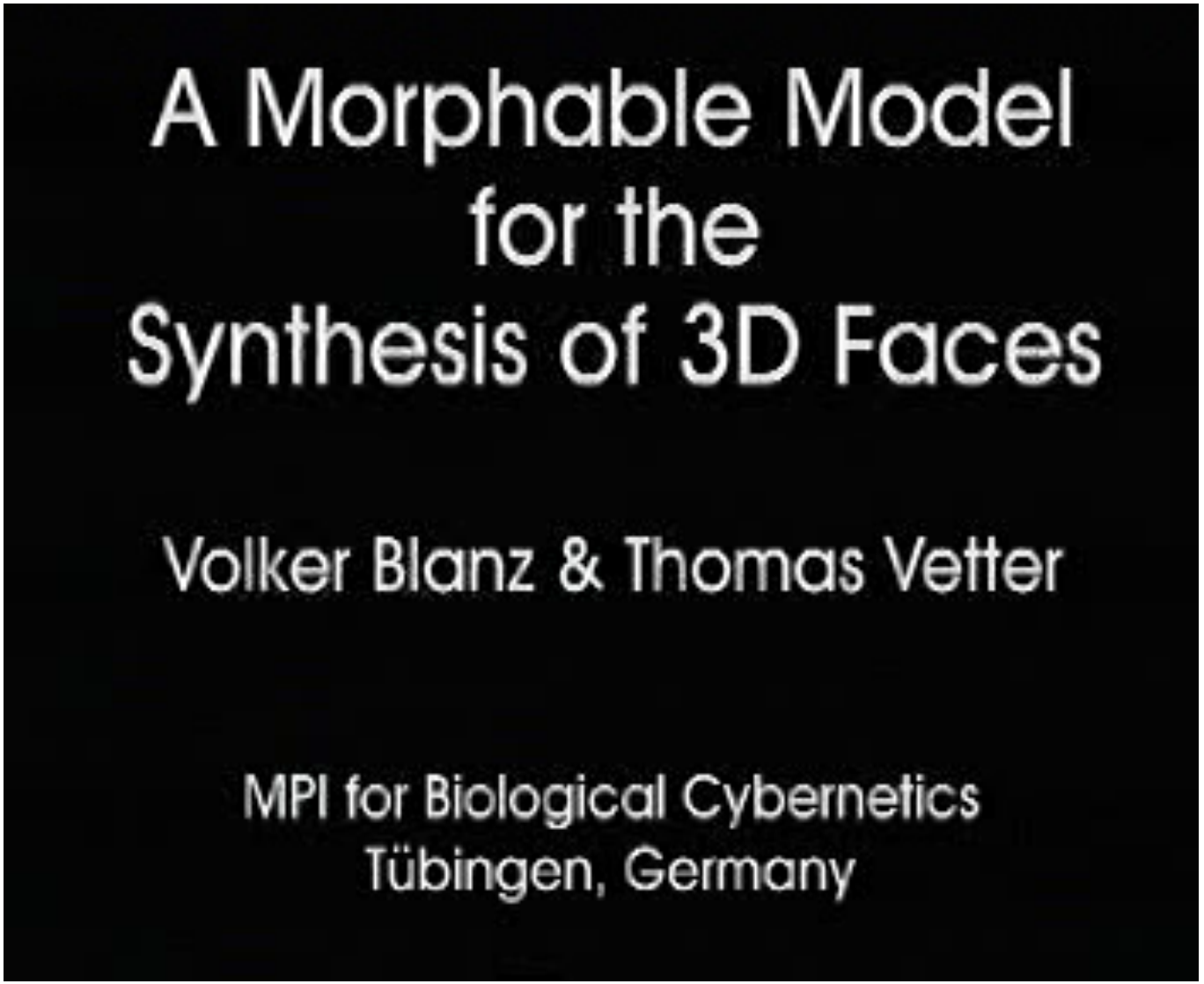


Manipulating Facial Attributes via Deformations

- For each face in the database, two scans are recorded: $S_{neutral}$, and $S_{expression}$.
- The difference vector $\Delta S = S_{expression} - S_{neutral}$ is saved and later on simply added to the 3D reconstruction of the input image.



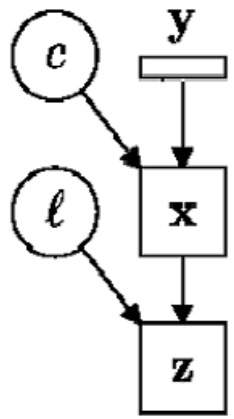



A Morphable Model
for the
Synthesis of 3D Faces
Volker Blanz & Thomas Vetter
MPI for Biological Cybernetics
Tübingen, Germany

Current State-of-the-art

Face Detection, Pose Estimation, and Landmark Localization in the Wild, X. Zhu and D. Ramanan, CVPR 2012

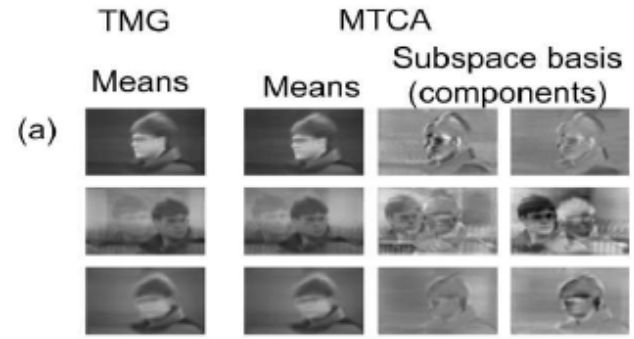
Continuous + Discrete Hidden Variables: Mixture of Transformed Components

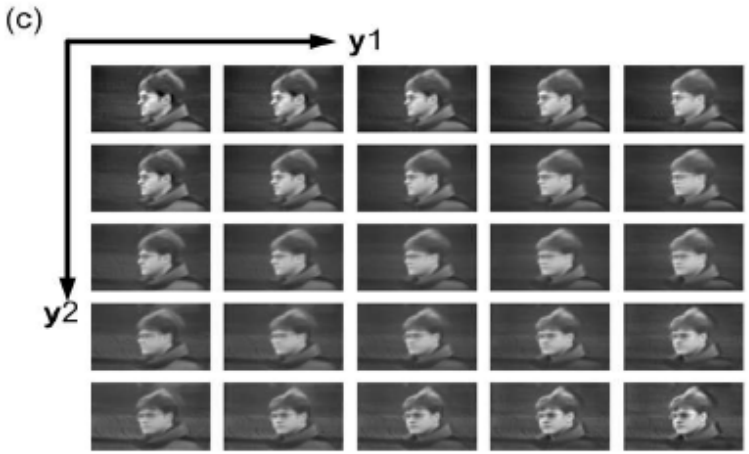


Latent variables for cluster

Latent variables for components

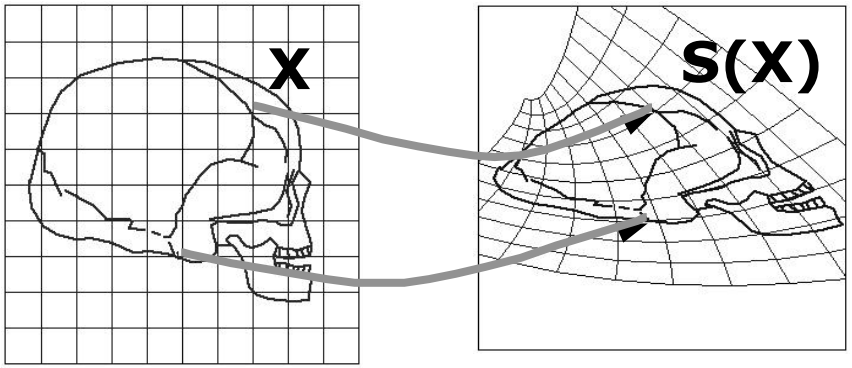
Latent variables for shift





Nonrigid deformations: AAMS

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



- Active Appearance Models

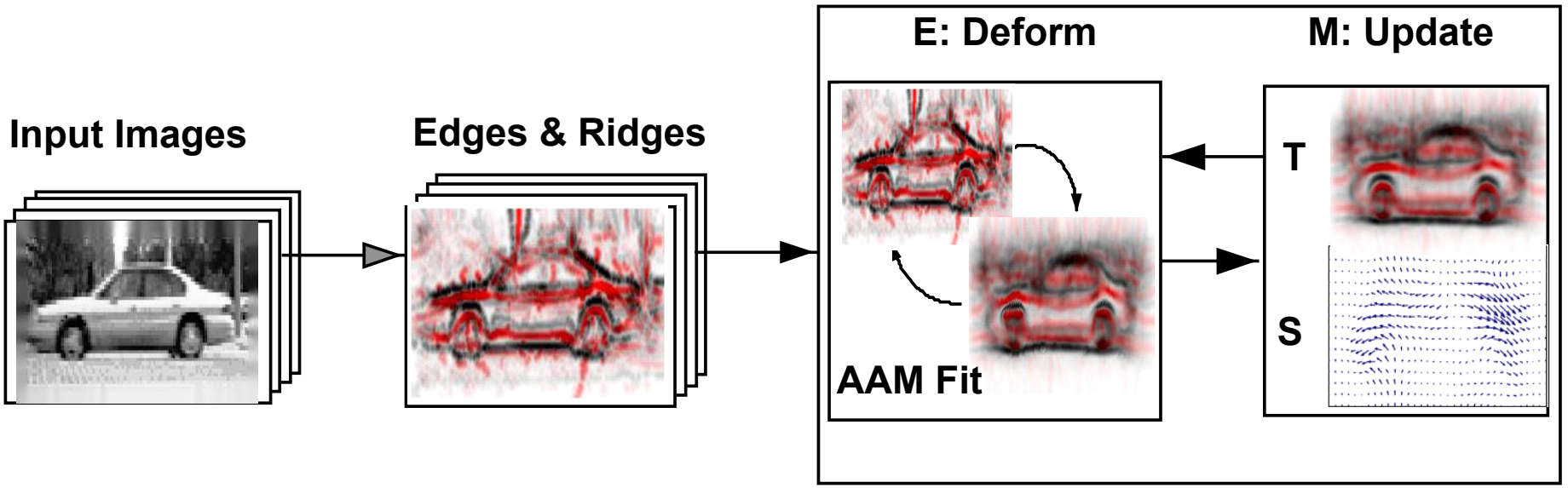
$$\mathcal{S}(\mathbf{x};\mathbf{s}) = \sum_i \mathbf{s}_i S_i(\mathbf{x}) \qquad \mathcal{T}(\mathbf{x};\mathbf{t}) = \sum_i \mathbf{t}_i T_i(\mathbf{x})$$

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 (x)



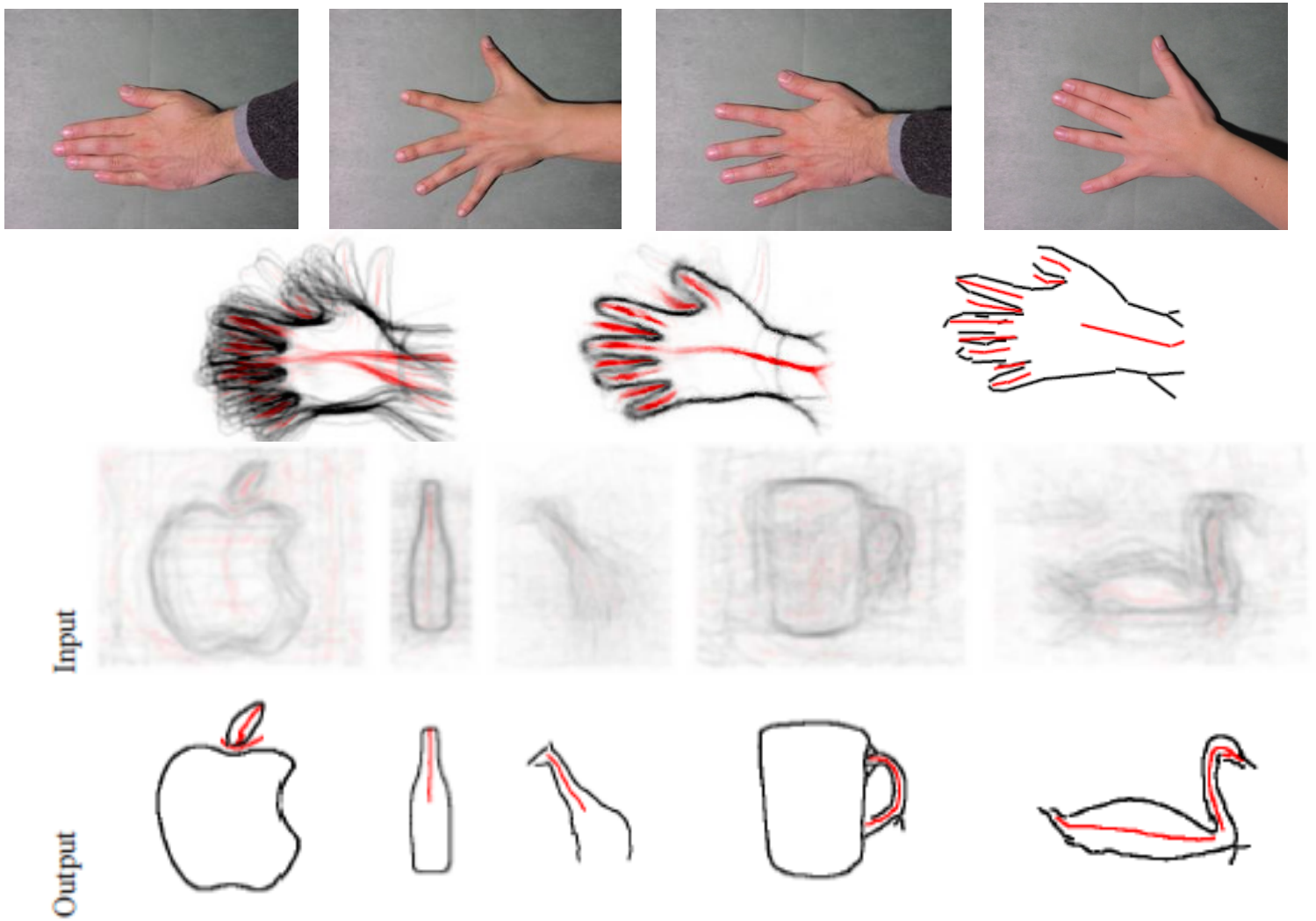
Template

1st basis element

2nd basis element

Recovering Object Contours (2007)

Hand, apple, giraffe, mug, swan models (2008)

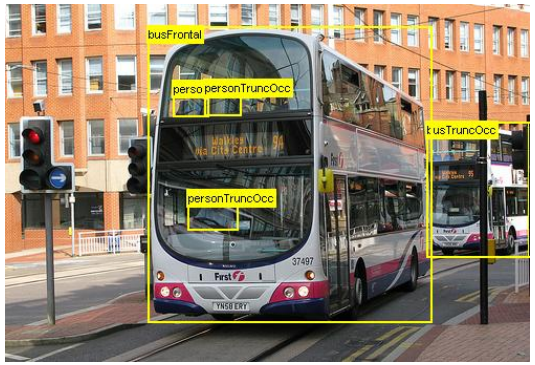


I. Kokkinos and A. Yuille,
Inference and Learning for Hierarchical Shape Models, IJCV 2011

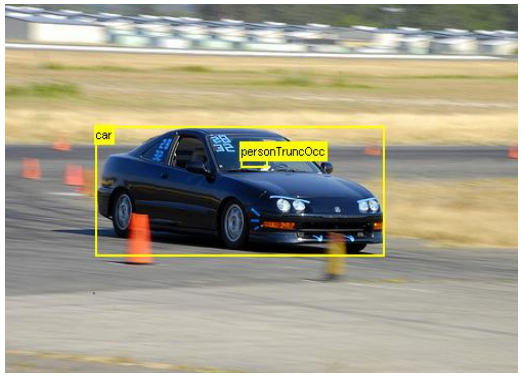
Pascal Dataset

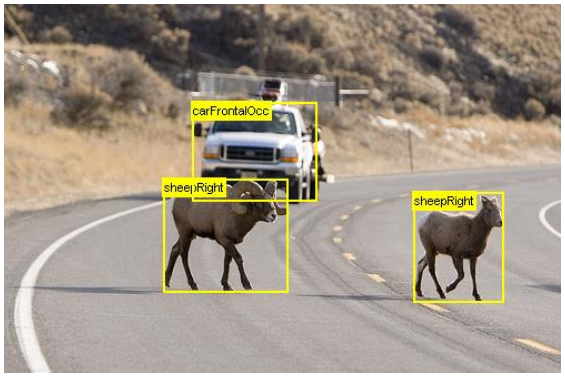
20 Categories, 25000 images

`Bus'

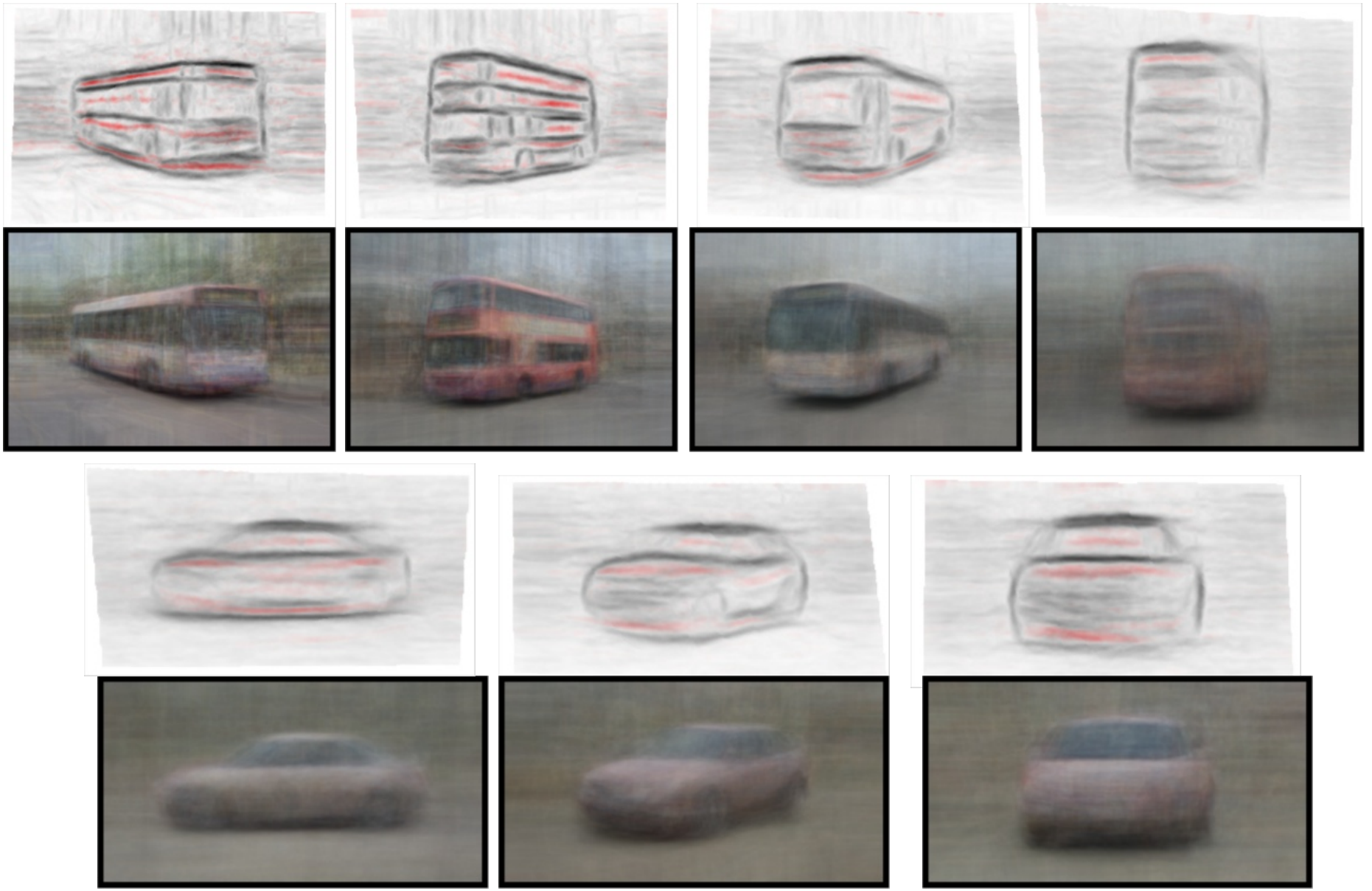


`Car'

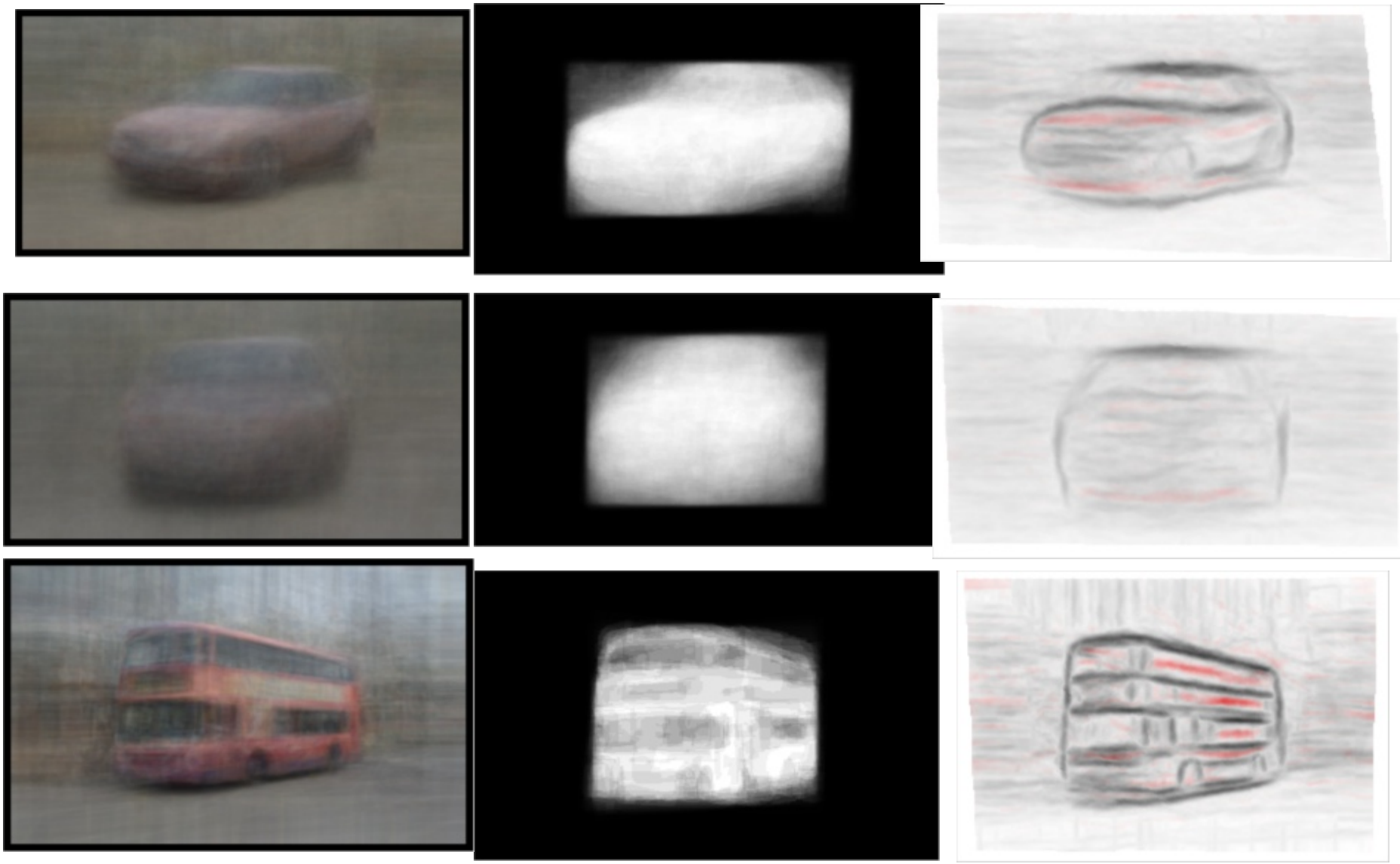




Multi-view models (2011)

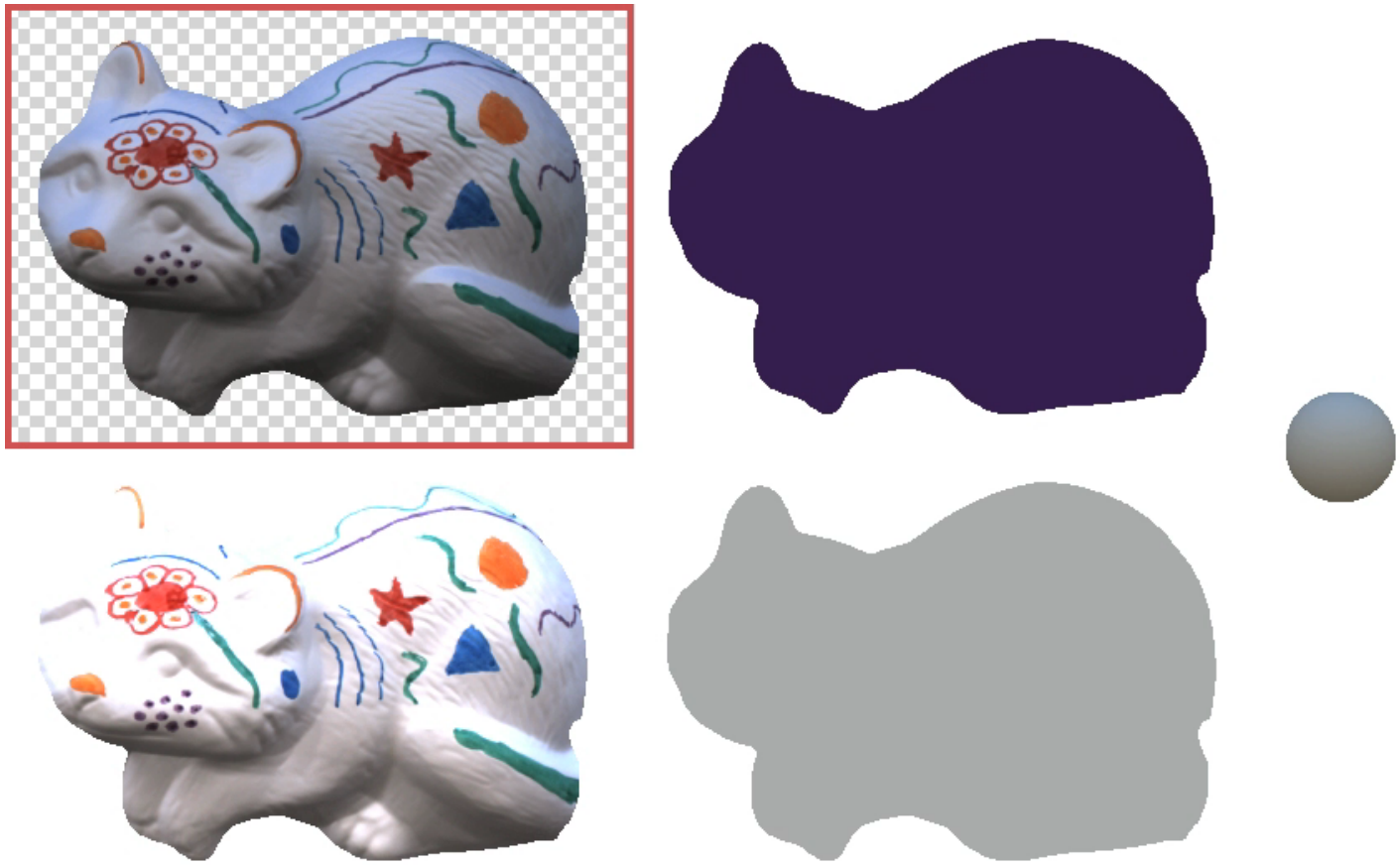


What's next? Segmentation



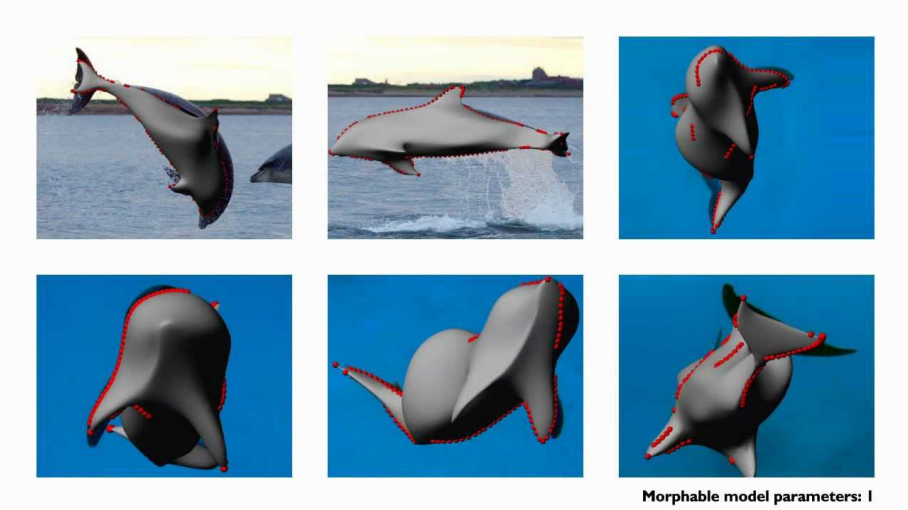
What's next? Segmentation

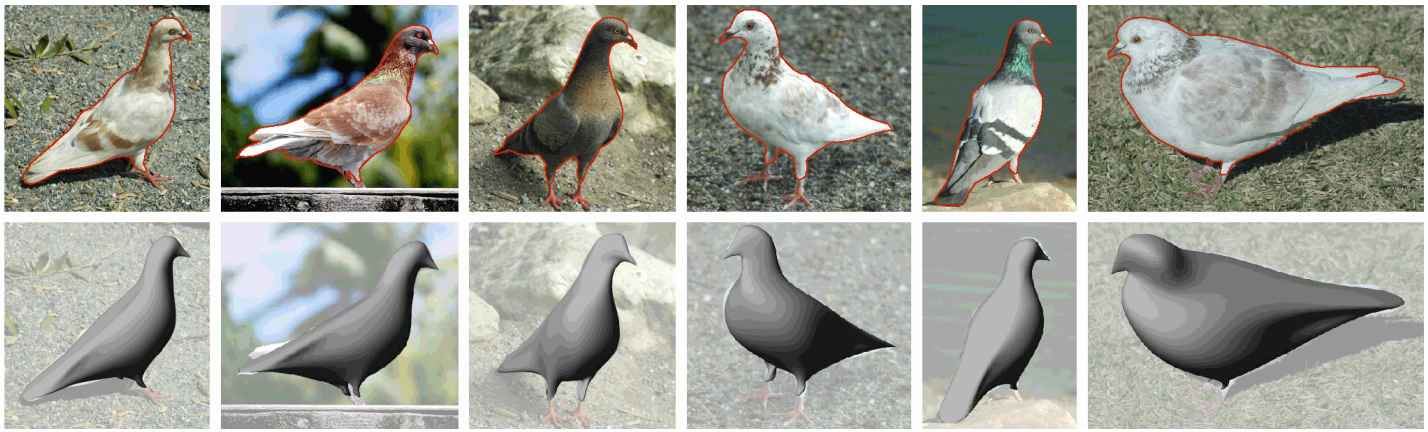
What's next? Shape-from-shading



Color Constancy, Intrinsic Images, and Shape Estimation
Jonathan T. Barron, Jitendra Malik
European Conference on Computer Vision (ECCV), 2012

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/

APPENDIX

$$P(y=1|x) = \frac{P(x|y=1)P(y=1)}{\sum_{y'=0}^{1} P(x|y')P(y')}$$

$$= \frac{\frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma_1|}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)\pi}{\frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma_0|}} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)\right)(1-\pi) + \frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma_1|}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)\pi}$$

$$= \frac{1}{1+\frac{1-\pi}{\pi}\sqrt{\frac{|\Sigma_0|}{|\Sigma_1|}} \exp(\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) - \frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0))}$$

$$= \frac{1}{1+\exp(x^TAx + x^TB + C)}$$

$$A = \frac{1}{2}\Sigma_1^{-1} - \frac{1}{2}\Sigma_0^{-1}$$

$$B = -\Sigma_1^{-1}\mu_1 + \Sigma_0^{-1}\mu_0$$

$$C = \frac{1}{2}\left[\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_0^T\Sigma_0^{-1}\mu_0\right] + \log(\frac{1-\pi}{\pi}\sqrt{\frac{|\Sigma_0|}{|\Sigma_1|}})$$

Factor Analysis: Generative Model

- Hidden variables
- Observations
  - noise covariance matrix
- Linear model
- Distribution of x

$$\begin{aligned} h &\propto N(0, I) \\ x &\propto N(\mu + \Lambda h, \Psi) \end{aligned}$$

$$x = \mu + \Lambda h + w, \quad w \propto N(0, \Psi)$$

$$\begin{aligned} E(x) &= E(\mu + \Lambda h + w) \\ &= \mu + \Lambda E(h) + E(w) \\ &= \mu \end{aligned}$$

$$\begin{aligned} Var(x) &= E((x-\mu)(x-\mu)^T) \\ &= E((\Lambda h + w)(\Lambda h + w)^T) \\ &= E(\Lambda h h^T \Lambda^T + 2\Lambda h w^T + w w^T) \\ &= \Lambda E(h h^T) \Lambda^T + 2\Lambda E(h) E(w^T) + E(w w^T) \\ &= \Lambda \Lambda^T + \Psi \end{aligned}$$

Full observation distribution

- Consider covariance of x, h:

$$\begin{aligned} Cov(x,h) &= E(x((\mu+\Lambda h+w)-\mu)^T) \\ &= E(x(\Lambda h+w)^T) \\ &= E(xh^T\Lambda^T)+E(xh^T) \\ &= \Lambda^T \end{aligned}$$

- Full observations

$$z=\begin{bmatrix} h \\ x \end{bmatrix}$$

- Distribution

$$z \propto N\left(\begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} I & \Lambda^T \\ \Lambda & \Lambda\Lambda^T+\Psi \end{bmatrix}\right)$$

- We will need to write

$$P(x,h)=P(x|h)P(h)$$

- Problem: non-diagonal matrix

Block matrix diagonalization

$$F = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$\begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix} F = \begin{bmatrix} A - BD^{-1}C & 0 \\ C & D \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix}}_{X} F \underbrace{\begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix}}_{h}$$

- Schur Complement

$$XFY = h \rightarrow F^{-1} = Yh^{-1}h$$

$$F/D \equiv A - BD^{-1}C$$

$$\begin{aligned} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} &= \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \begin{bmatrix} F/D & 0 \\ 0 & D \end{bmatrix}^{-1} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \begin{bmatrix} (F/D)^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} (F/D)^{-1} & -(F/D)^{-1}BD^{-1} \\ -D^{-1}C(F/D)^{-1} & -D^{-1}C(F/D)^{-1}BD^{-1} + D^{-1} \end{bmatrix} \end{aligned}$$

Factorizing a Gaussian distribution

$$P(x_1, x_2) \propto \exp\left(-\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T\right)$$

$$= \exp\left(-\frac{1}{2}\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T \begin{bmatrix} I & 0 \\ -\Sigma_{22}^{-1}\Sigma_{21} & I \end{bmatrix} \begin{bmatrix} (\Sigma/\Sigma_{22})^{-1} & 0 \\ 0 & \Sigma_{22}^{-1} \end{bmatrix} \begin{bmatrix} I & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T\right)$$

$$= \exp\left(-\frac{1}{2}(x_1 - \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2))^T (\Sigma/\Sigma_{22})^{-1}(x_1 - \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2))\right)$$

$$\exp\left(-\frac{1}{2}(x_2 - \mu_2)^T \Sigma_{22}^{-1}(x_2 - \mu_2)\right)$$

$$= \underbrace{\exp\left(-\frac{1}{2}(x_1 - \mu_{1|2})^T \Sigma_{1|2}^{-1}(x_1 - \mu_{1|2})\right)}_{\propto P(x_1|x_2)} \underbrace{\exp\left(-\frac{1}{2}(x_2 - \mu_2)^T \Sigma_{22}^{-1}(x_1 - \mu_2)\right)}_{\propto P(x_2)}$$

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \qquad \Sigma_{1|2} = \Sigma/\Sigma_{22} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}^{-1}$$

PCA criterion

- Minimize reconstruction error of training set

$$
\begin{aligned}
\sum_{i=1}^{N} |x^i - \bar{x}^i|^2 &= \sum_{i=1}^{N} \left(\sum_{j=K+1}^{M} (x^{i^T} b_j) b_j \right)^2 \\
&= \sum_{i=1}^{N} \sum_{j=K+1}^{M} \left(x^{i^T} b_j \right)^2 \\
&= \sum_{i=1}^{N} \sum_{j=K+1}^{M} \left(x^{i^T} b_j \right)^2 \\
&= \sum_{i=1}^{N} \sum_{j=K+1}^{M} (b_j^T x^i) x^{i^T} b_j \\
&= \sum_{j=K+1}^{M} u_j^T \sum_{i=1}^{N} x^i x^{i^T} b_j \\
&= N \sum_{j=K+1}^{M} b_j^T S b_j
\end{aligned}
$$

Spectral Decomposition of a matrix

$$
\begin{aligned}
U &= [u_1, u_2, \ldots, u_N] \\
U^T U &= I
\end{aligned}
$$

$$
\begin{aligned}
S &= SUU^T \\
&= S[u_1, \ldots, u_N]U^T \\
&= [Su_1, \ldots, Su_N]U^T \\
&= [\lambda_1 u_1, \ldots, \lambda_N u_N]U^T \\
&= \sum_{i=1}^{N} \lambda_i u_i u_i^T \\
&= UDU^T
\end{aligned}
$$

$$
\begin{aligned}
U^T SU &= D \\
u_i^T S u_i &= \lambda_i
\end{aligned}
$$

$$\min b^T S b = \lambda_N$$

$$\min \sum_{j=K+1}^{M} u_j^T S u_j = \sum_{j=K+1}^{M} \lambda_j$$

Principal Component Analysis

- Given: N data points $x_1, \ldots, x_N$ in R^d
- We want to find a new set of features that are linear combinations of original ones:

$$u(x_i) = u^T(x_i - \mu)$$

(μ: mean of data points)

- What unit vector u in R^d captures the most variance of the data?

Principal Component Analysis

- Variance of projection on u:

$$var(u) = \frac{1}{N}\sum_{i=1}^{N-1} \underbrace{\mathbf{u}^{\mathrm{T}}(\mathbf{x}_i - \mu)}_{\text{Projection of data point}}(\mathbf{u}^{\mathrm{T}}(\mathbf{x}_i - \mu))^{\mathrm{T}}$$

$$= \mathbf{u}^{\mathrm{T}}\underbrace{\left[\sum_{i=1}^{N-1}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^{\mathrm{T}}\right]}_{\text{Covariance matrix of data}}\mathbf{u}$$

$$= \mathbf{u}^{\mathrm{T}}\Sigma\mathbf{u}$$

Direction: Unit norm vector

The direction that maximizes the variance: the eigenvector associated with the largest eigenvalue of Σ