Introduction to Computer Vision

Michael J. Black

Oct. 2009

Lecture 11:

Images as vectors.

Sub-space methods.

Goals

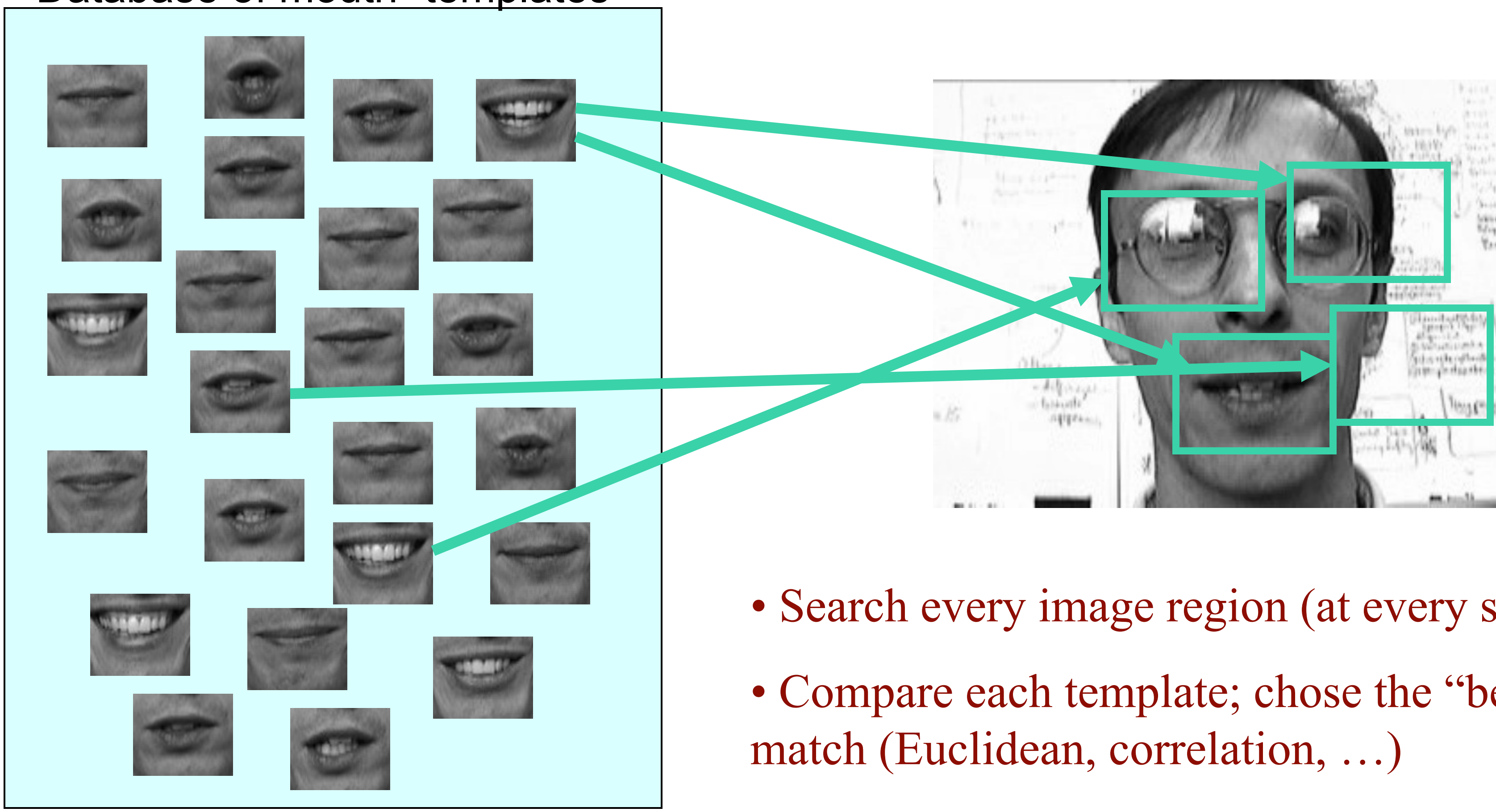
- Images as vectors in a high dimensional space
- Subspace methods (eigen analysis)
- Covariance and principal component analysis

Search and Recognition

1. How can we find the mouth?
2. How can we recognize the "expression"?

Naïve Appearance-Based Approach

Database of mouth "templates"



• Search every image region (at every scale).

• Compare each template; chose the "best" match (Euclidean, correlation, …)

Appearance-Based Methods

Represent objects by their appearance in an ensemble of images, including different poses, illuminants, configurations of shape, …

Approaches covered here:

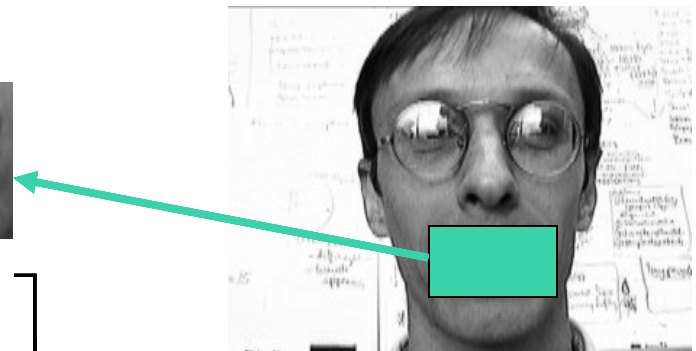
- Subspace (eigen) Methods
- Local Invariant Image Features

Fleet & Szeliski

Images as Vectors

$$= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n \times m} \end{bmatrix}$$

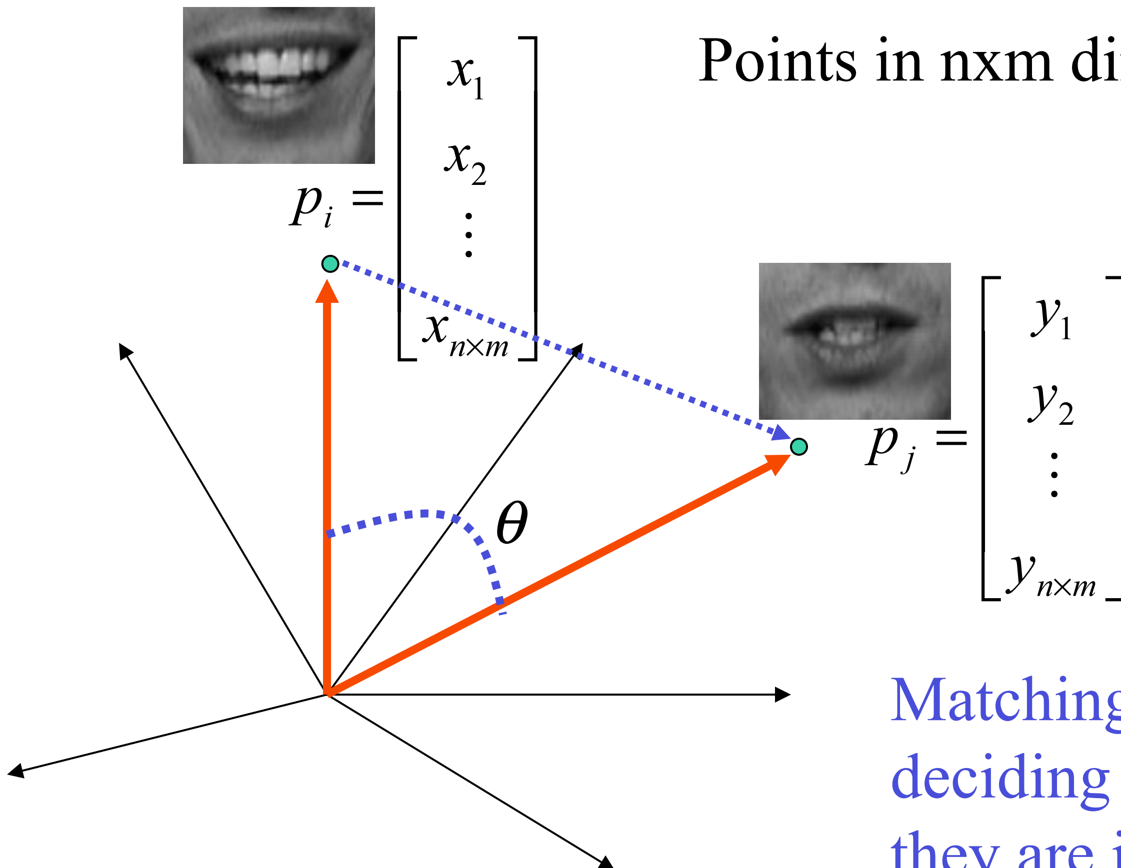
$$= \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n \times m} \end{bmatrix}$$



e.g. standard lexicographic ordering

Images as Points

Points in nxm dimensional space



Matching involves deciding how far apart they are in this space.

SSD Matching

- An alternative to correlation is to minimize the Sum of Squared Differences (SSD)

$$E(p_1, p_2) = \sum_{i=1:n} (p_1(i) - p_2(i))^2$$

- Distance metric.
- Euclidean distance = sqrt(E)

Template Methods

Image templates (simplest view-based method – straw man)

- keep an image of every object from different viewing directions, lighting conditions, etc.
- nearest neighbor cross-correlation matching with images in model database (or robust matching for clutter & occlusion)

Obvious problems:

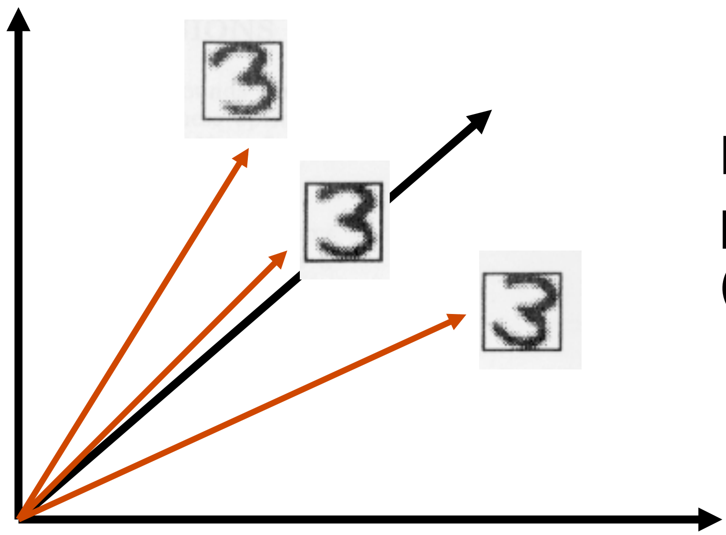
- storage and computation costs become unreasonable as the number of objects increases
- may require very large ensemble of 'training' images

Fleet & Szeliski

Subspace Methods

How can we find more efficient representations for the ensemble of views, and more efficient methods for matching?

- Idea: images are not random… especially images of the same object that have similar appearance

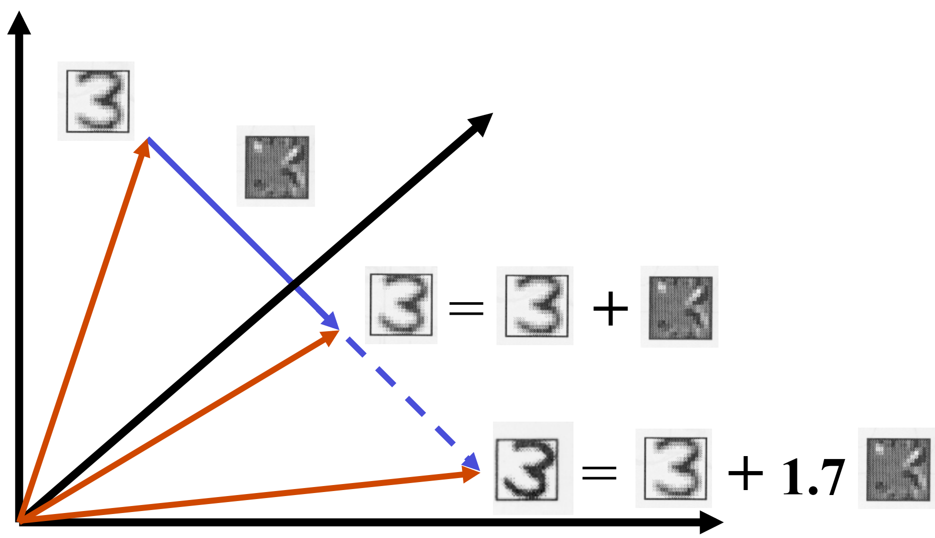


E.g., let images be represented as points in a high-dimensional space (e.g., one dimension per pixel)

Fleet & Szeliski

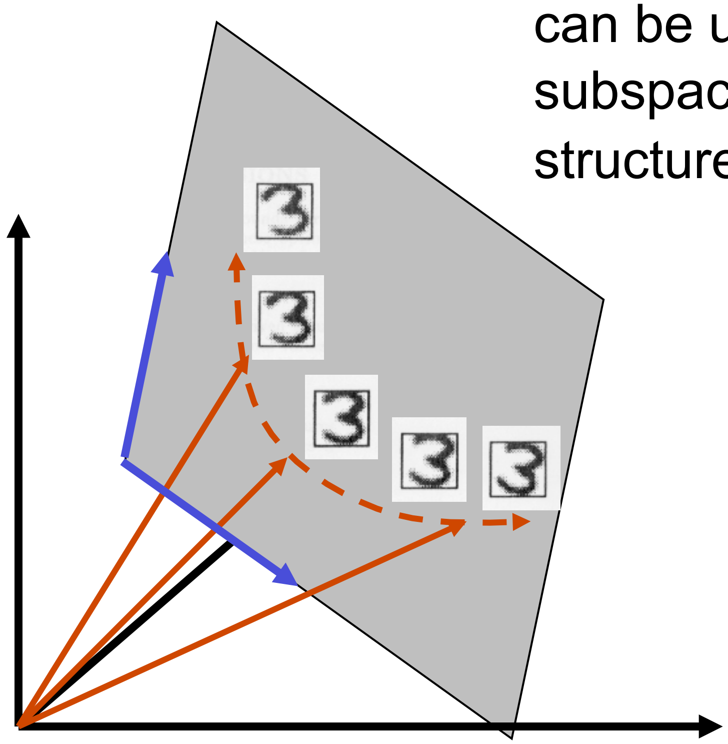
Linear Dimension Reduction

Given that differences are structured, we can use *'basis images'* to transform images into other images in the same space.



Fleet & Szeliski

Linear Dimension Reduction



What linear transformations of the images can be used to define a lower-dimensional subspace that captures most of the structure in the image ensemble?

Fleet & Szeliski

Approach

- Find a lower dimensional representation that captures the variability in the data.
- Search using this low dimensional model.

Goal

Data point *n*

$$\vec{x}^n \in \Re^D$$

Low dim representation:

$$\vec{z}^n \in \Re^M \quad M << D$$

Map $\vec{x}^n \rightarrow \vec{z}^n$

Observation

I can always write a vector as:

Kronecker delta =1 if i=j, 0 otherwise.

$$\vec{x} = \sum_{i=1}^{D} a_i \vec{u}_i \quad \text{where} \quad \vec{u}_i^T \vec{u}_j = \delta_{ij}$$

orthonormal

Example:

$$\begin{bmatrix} 3 \\ 7 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 7 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Observation

$$\vec{x} = a_1\vec{u}_1 + a_2\vec{u}_2$$

$$\begin{aligned} a_1 &= \vec{u}_1^T(\vec{x} - a_2\vec{u}_2) \\ &= \vec{u}_1^T\vec{x} - a_2\vec{u}_1^T\vec{u}_2 \\ &= \vec{u}_1^T\vec{x} \end{aligned}$$

Projection

More generally

$$\underbrace{a_i}_{\text{Scalar coefficient}} = \underbrace{\vec{u}_i^{\,T} \vec{x}}_{\text{projection}}$$

Observation

$$\vec{x}^{\,n} = \sum_{i=1}^{M} a_i \vec{u}_i + \sum_{j=M+1}^{D} b_j \vec{u}_j$$

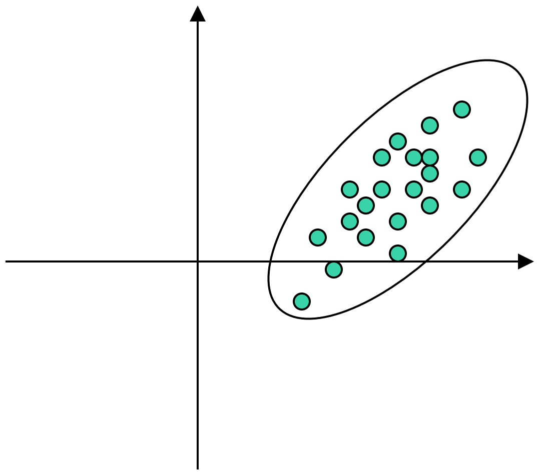
Approximation $\tilde{x}^n$

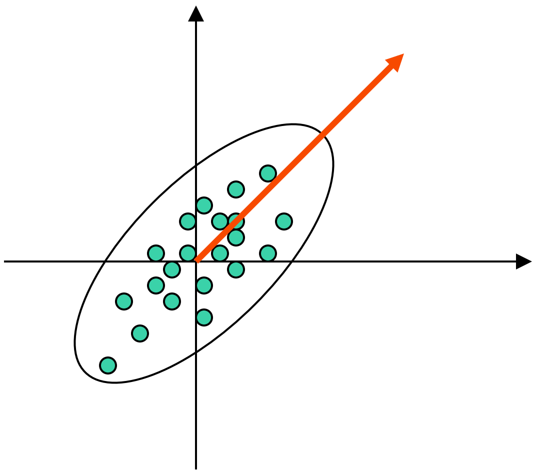
Error

Want the M bases that minimize the mean squared error over the training data

$$\min E_M = \sum_{n=1}^{N} \left\| \vec{x}^{\,n} - \tilde{x}^n \right\|^2$$

Simple 2D example

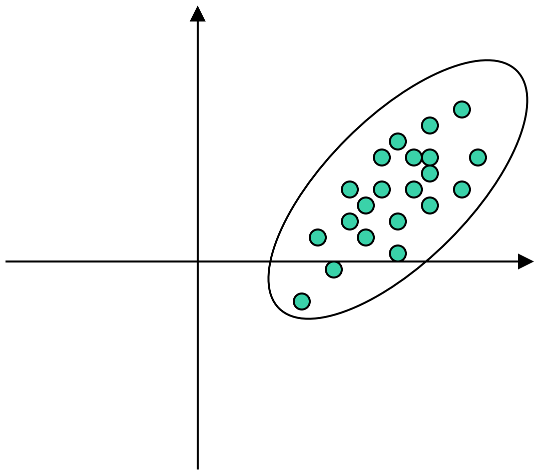


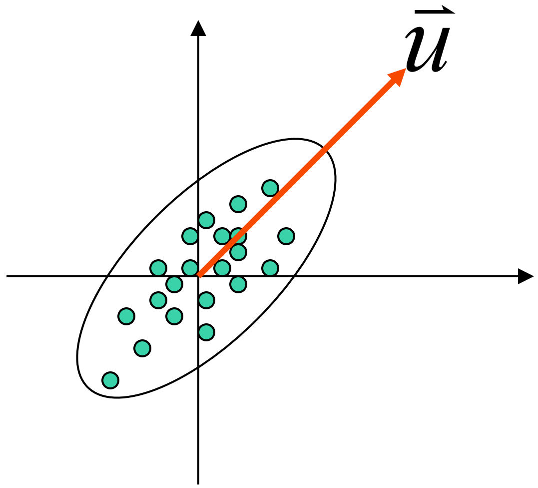


If I give you the mean and one vector to represent the data, what vector would you choose?

Why?

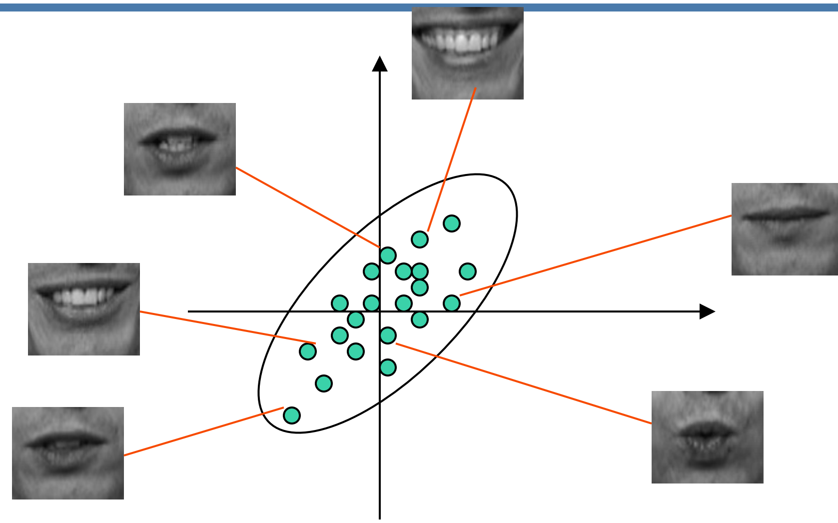
Simple 2D example





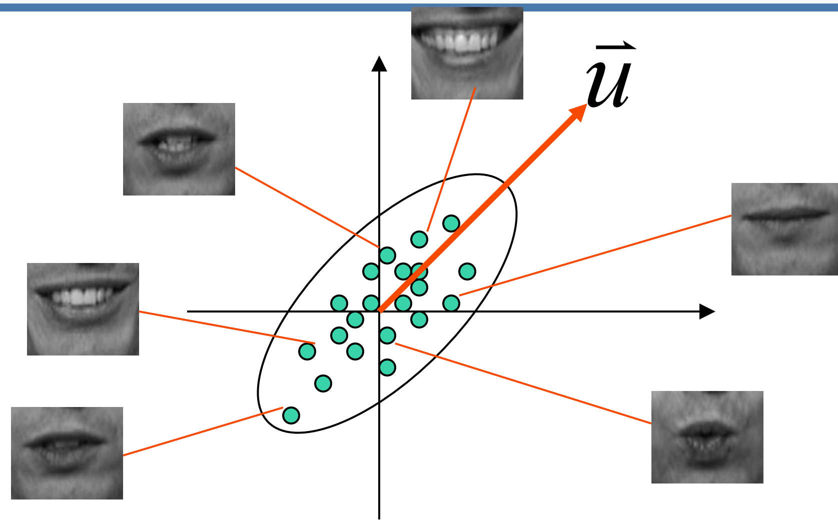
$$\vec{x}^{\,n} \approx \bar{x} + a\vec{u}$$

Mouths



$$X = \left[\vec{x}^1 \quad \vec{x}^2 \quad \ldots \quad \vec{x}^N \right] =$$

Mouths



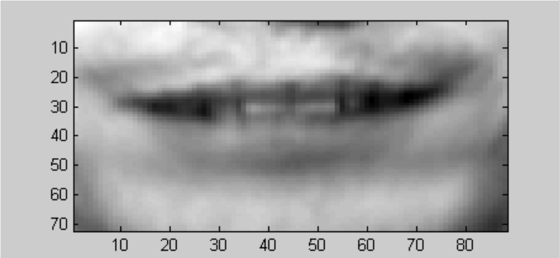
$$X = \begin{bmatrix} \vec{x}^1 & \vec{x}^2 & \ldots & \vec{x}^N \end{bmatrix} =$$

Recall our goal: $\vec{x}^n \approx \bar{x} + a\vec{u}$

Statistics Review

Sample Mean

$$\bar{x} = \langle x \rangle = \frac{1}{N} \sum_{i=1}^{N} \vec{x}^{\,i}$$



Sample Variance

$$\sigma^2 = \left\langle (x - \bar{x})^2 \right\rangle = \mathrm{var}(\vec{x}) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

Statistics Review

Multiple variables: covariance.

$$\operatorname{cov}(x,y) = \sigma_{xy} = \left\langle (x-\bar{x})(y-\bar{y}) \right\rangle$$

$$= \left\langle xy \right\rangle - \left\langle x \right\rangle \left\langle \bar{y} \right\rangle - \left\langle y \right\rangle \left\langle \bar{x} \right\rangle + \left\langle x \right\rangle \left\langle y \right\rangle$$

$$= \left\langle xy \right\rangle - \left\langle x \right\rangle \left\langle y \right\rangle - \left\langle y \right\rangle \left\langle x \right\rangle + \left\langle x \right\rangle \left\langle y \right\rangle$$

$$= \left\langle xy \right\rangle - \left\langle x \right\rangle \left\langle y \right\rangle$$

Special case: variance.

$$\operatorname{cov}(x,x) = \left\langle x^2 \right\rangle - \left\langle x \right\rangle^2 = \sigma_x^2$$

Statistical Correlation

The covariance of two random variables X and Y provides a measure of how strongly correlated these variables are, and the derived quantity

$$\mathrm{cor}(x, y) = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y}$$

(Same as correlation coefficient, *r,* defined earlier.)

Statistical Correlation

$$\operatorname{cor}(x, y) = \frac{\operatorname{cov}(x, y)}{\sigma_x \sigma_y}$$

$$r = \frac{\sum_{k,l} (f(k,l) - \bar{f})(g(k,l) - \bar{g})}{\sqrt{\left(\sum_{k,l} (f(k,l) - \bar{f})^2\right)\left(\sum_{k,l} (g(k,l) - \bar{g})^2\right)}}$$

Covariance Matrix

For two random variables x and y we have

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

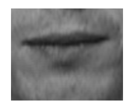
$$C = \frac{1}{N-1}\sum_{n=1}^{N}(\vec{x}^n - \bar{x})(\vec{x}^n - \bar{x})^T$$

Outer product

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} x_1 x_1 & x_1 x_2 \\ x_2 x_1 & x_2 x_2 \end{bmatrix}$$

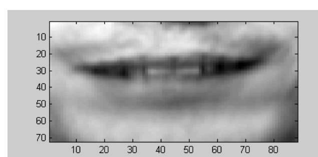
Covariance Matrix

$$X = \begin{bmatrix} \vec{x}^{1} & \vec{x}^{2} & \ldots & \vec{x}^{N} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^N \\ x_2^1 & x_2^2 & \cdots & x_2^N \\ \vdots & \vdots & & \vdots \\ x_D^1 & x_D^2 & \cdots & x_D^N \end{bmatrix}$$



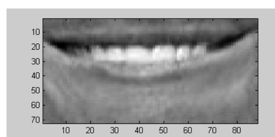
Mean

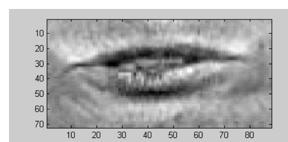
$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x^{i}$$

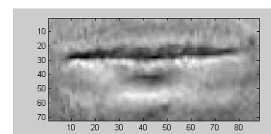


Covariance Matrix

$$A = X - \bar{x} = \begin{bmatrix} x_1^1 - \bar{x}_1 & x_1^2 - \bar{x}_1 & \cdots & x_1^N - \bar{x}_1 \\ x_2^1 - \bar{x}_2 & x_2^2 - \bar{x}_2 & \cdots & x_2^N - \bar{x}_2 \\ \vdots & \vdots & & \vdots \\ x_D^1 - \bar{x}_D & x_D^2 - \bar{x}_D & \cdots & x_D^N - \bar{x}_D \end{bmatrix}$$







What is

$$\frac{1}{N-1} AA^T$$

Covariance Matrix

$$AA^T = \begin{bmatrix} x_1^1 - \bar{x}_1 & x_1^2 - \bar{x}_1 & \cdots & x_1^N - \bar{x}_1 \\ x_2^1 - \bar{x}_2 & x_2^2 - \bar{x}_2 & \cdots & x_2^N - \bar{x}_2 \\ \vdots & \vdots & & \vdots \\ x_D^1 - \bar{x}_D & x_D^2 - \bar{x}_D & \cdots & x_D^N - \bar{x}_D \end{bmatrix} \begin{bmatrix} x_1^1 - \bar{x}_1 & x_2^1 - \bar{x}_2 & \cdots & x_D^1 - \bar{x}_D \\ x_1^2 - \bar{x}_1 & x_2^2 - \bar{x}_2 & \cdots & x_D^2 - \bar{x}_D \\ \vdots & \vdots & & \vdots \\ x_1^N - \bar{x}_1 & x_2^N - \bar{x}_2 & \cdots & x_D^N - \bar{x}_D \end{bmatrix}$$

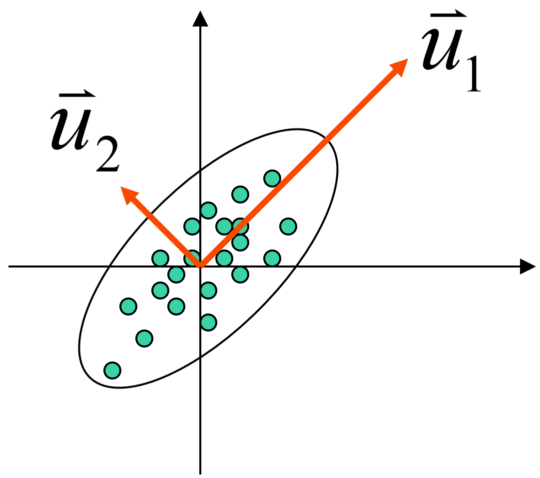
$$AA^T = \begin{bmatrix} \sum_{j=1}^{N} (x_1^j - \bar{x}_1)^2 & \sum_{j=1}^{N} (x_1^j - \bar{x}_1)(x_2^j - \bar{x}_2) & \cdots \\ \sum_{j=1}^{N} (x_2^j - \bar{x}_2)(x_1^j - \bar{x}_1) & \sum_{j=1}^{N} (x_2^j - \bar{x}_2)^2 & \cdots \\ \vdots & \vdots & \vdots \\ & & \cdots \end{bmatrix}$$

Size?

Intuition

$$\vec{x}^n - \bar{x} = \sum_{i=1}^{M} a_i \vec{u}_i + \sum_{j=M+1}^{D} b_j \vec{u}_j$$

$$\hat{x}^n = \sum_{i=1}^{M} a_i \vec{u}_i + \bar{x}$$

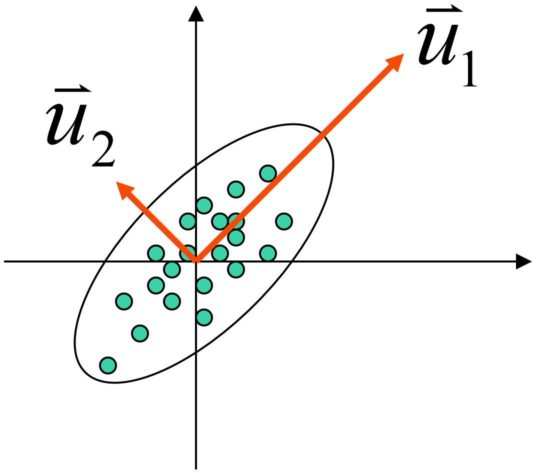


Projecting onto $\vec{u}_1$ captures the majority of the variance and hence projecting onto it minimizes the error

Intuition

$$\vec{x}^n - \bar{x} = \sum_{i=1}^{M} a_i \vec{u}_i + \sum_{j=M+1}^{D} b_j \vec{u}_j$$

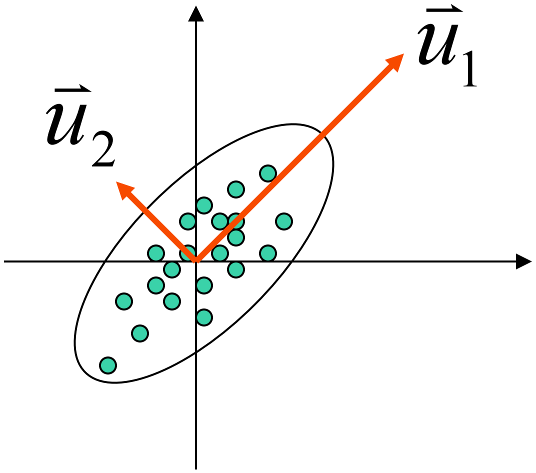
$$\min E_M = \sum_{n=1}^{N} \left\| \vec{x}^n - \hat{x}^n \right\|^2$$



Note that these axes are orthogonal and decorrelate the data; ie in the coordinate frame of these axes, the data is uncorrelated.

Intuition

$$\vec{x}^n - \bar{x} = \sum_{i=1}^{M} a_i \vec{u}_i + \sum_{j=M+1}^{D} b_j \vec{u}_j$$



$$\min E_M = \sum_{n=1}^{N} \left\| \vec{x}^n - \hat{x}^n \right\|^2$$

So how do we find these directions of maximum variance? This is key.

Principal Component Analysis

Let $X = [\vec{x}^1 \cdots \vec{x}^N]$

Compute the mean column vector: $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x^i$

Subtract the mean from each column.

$$A = X - \bar{x} = [(\vec{x}^1 - \bar{x}) \cdots (\vec{x}^N - \bar{x})]$$

Covariance matrix can be written

$$C = \frac{1}{N-1} A A^T$$

Principal Component Analysis

C is real, symmetric, positive definite. We can write it

$$C = U \Lambda U^T = \begin{bmatrix} & & \\ \vec{e}_1 & \cdots & \vec{e}_D \\ & & \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_D \end{bmatrix} \begin{bmatrix} \vec{e}_1^{\,T} \\ \vdots \\ \vec{e}_D^{\,T} \end{bmatrix}$$

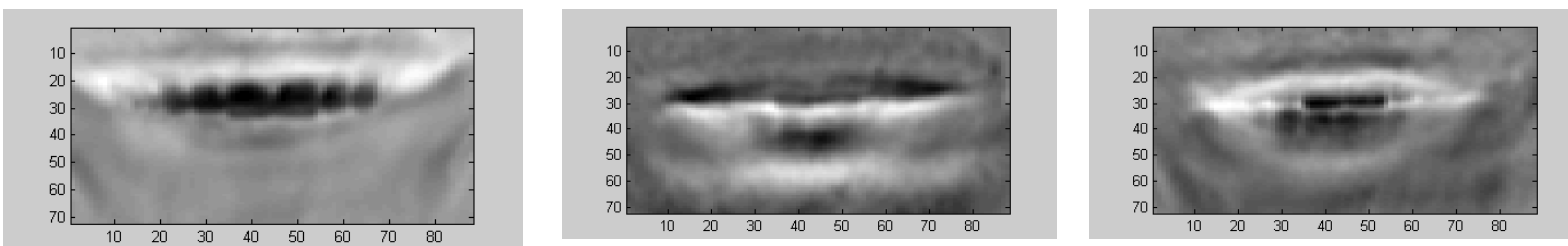
Orthonormal columns
Eigenvectors

eigenvalues

Principal Component Analysis

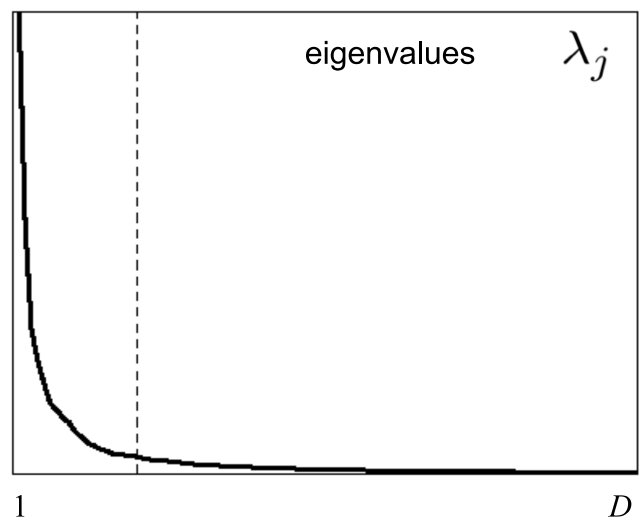
$$C = U \Lambda U^T = \begin{bmatrix} \vec{e}_1 & \cdots & \vec{e}_D \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_D \end{bmatrix} \begin{bmatrix} \vec{e}_1^{\,T} \\ \vdots \\ \vec{e}_D^{\,T} \end{bmatrix}$$

First three eigenvectors:



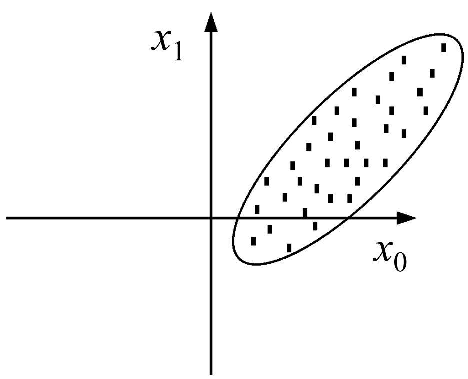
Principal Component Analysis

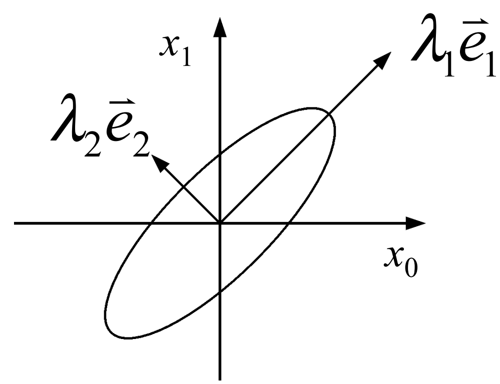
$$C = U\Lambda U^T = \begin{bmatrix} \vec{e}_1 & \cdots & \vec{e}_D \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_D \end{bmatrix} \begin{bmatrix} \vec{e}_1^T \\ \vdots \\ \vec{e}_D^T \end{bmatrix}$$



Principal Component Analysis

- Eigenvectors are the *principal directions*, and the eigenvalues represent the variance of the data along each principal direction
  - ∗ l_k is the marginal variance along the principal direction $\vec{e}_k$





Fleet & Szeliski

Principal Component Analysis

- The first principal direction $\vec{e}_1$ is the direction along which the variance of the data is maximal, i.e. it maximizes

$$\vec{e}^T C \vec{e} \quad \text{where} \quad \vec{e}^T \vec{e} = 1$$

- The second principal direction maximizes the variance of the data in the orthogonal complement of the first eigenvector.
- etc.

Fleet & Szeliski

Principal Component Analysis

- PCA Approximate Basis: If $\lambda_k \approx 0$ for $k > M$ for some $M << D$, then we can approximate the data using only M of the principal directions (basis vectors):
  - If $\mathbf{B} = [\vec{e}_1, \ldots, \vec{e}_M]$, then for all points

$$\vec{x}^n \approx \mathbf{B}\vec{a}^n + \bar{x}$$

where $$a_k^n = (\vec{x}^n - \bar{x})^T \vec{e}_k$$

Fleet & Szeliksi