Introduction to Computer Vision

Michael J. Black

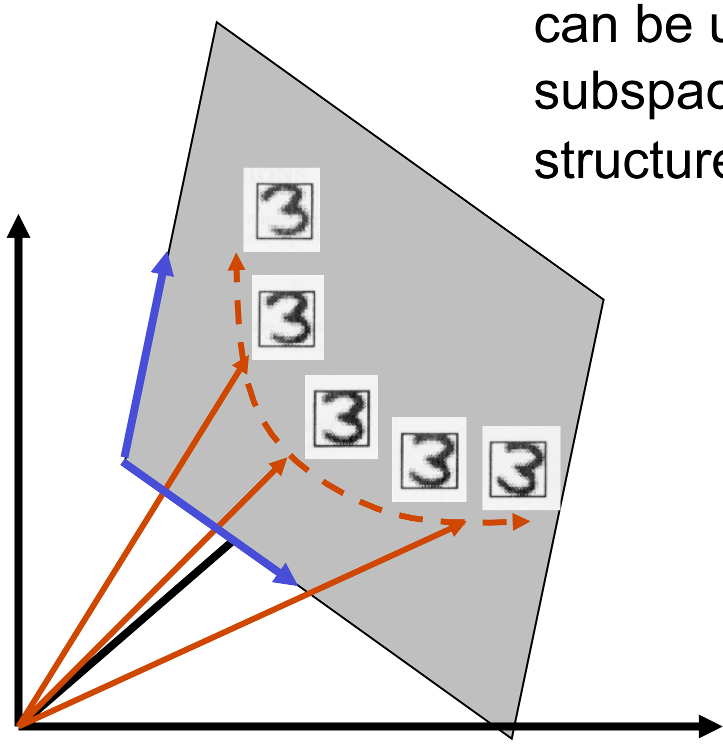
Sept 2009

Covariance and PCA

Goals

- Today: Review covariance and principal component analysis.
  - Prep for homework 2
- Monday, holiday, no class
- Wed start probability and classification

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

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

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

Approximation $\tilde{x}^n$

Error

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

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

Review: 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

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

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

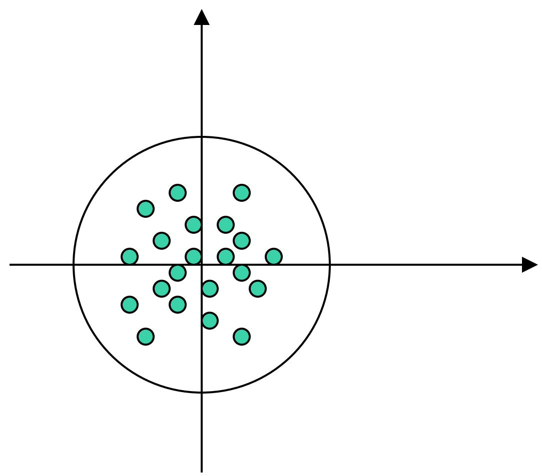
Review: Covariance Matrix

For two random variables x and y we have

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

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

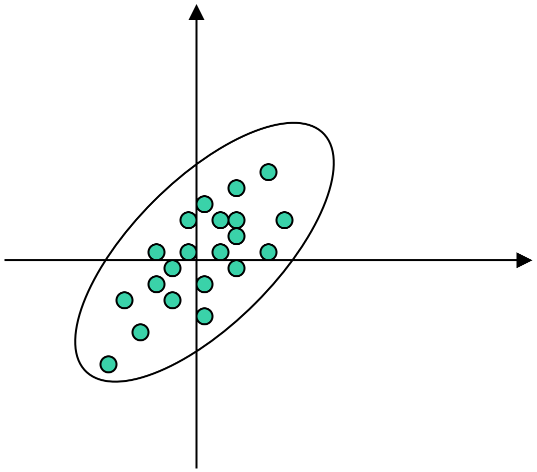
Correlated?



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

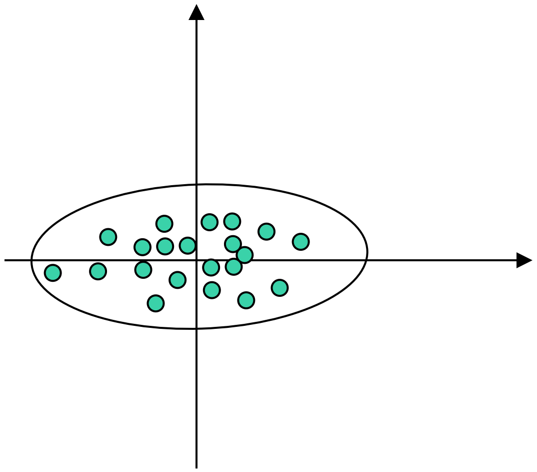
correlation: strength and direction of a linear relationship between two random variables

Correlated?



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

Correlated?



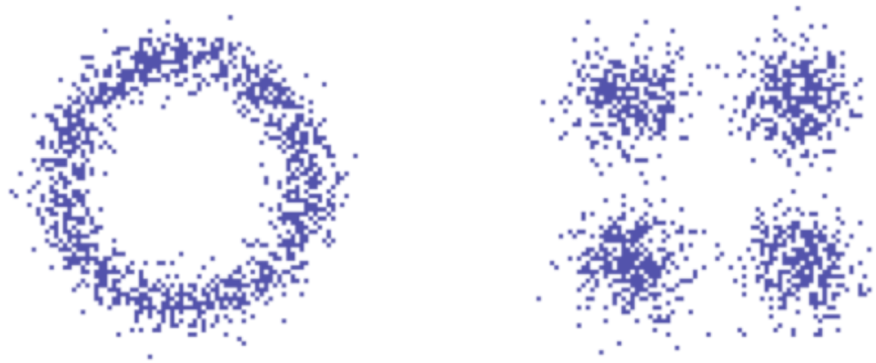
$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

Correlation

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y},$$

Wikipedia

Correlated?



Wikipedia

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 semi-definite*.
We can write it

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

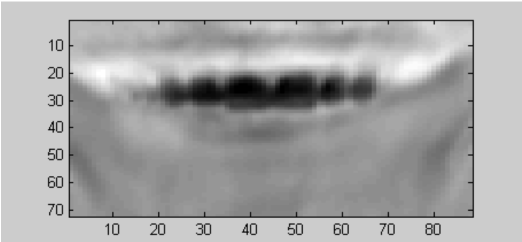
Orthonormal columns
Eigenvectors

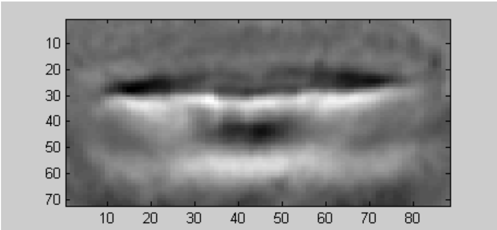
eigenvalues

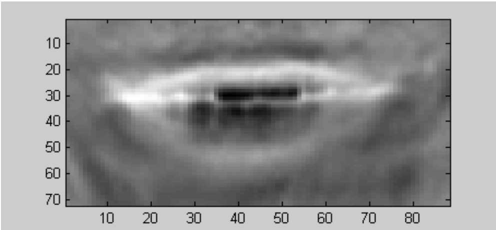
Principal Component Analysis

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

First three eigenvectors:

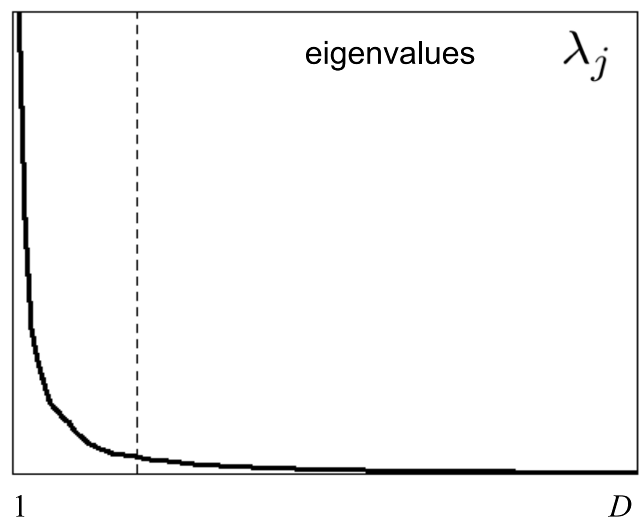






Principal Component Analysis

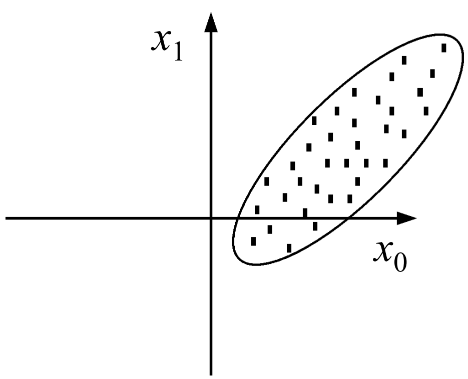
$$C = U\Lambda U^T = \begin{bmatrix} \vec{e}_1 & \cdots & \vec{e}_D \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_D \end{bmatrix} \begin{bmatrix} \vec{e}_1^T \\ \vdots \\ \vec{e}_D^T \end{bmatrix}$$

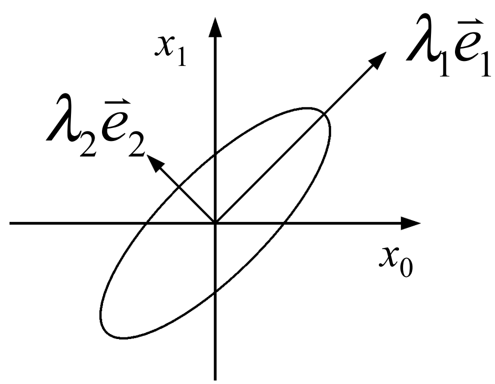


Principal Component Analysis

• Eigenvectors are the *principal directions*, and the eigenvalues represent the variance of the data along each principal direction

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

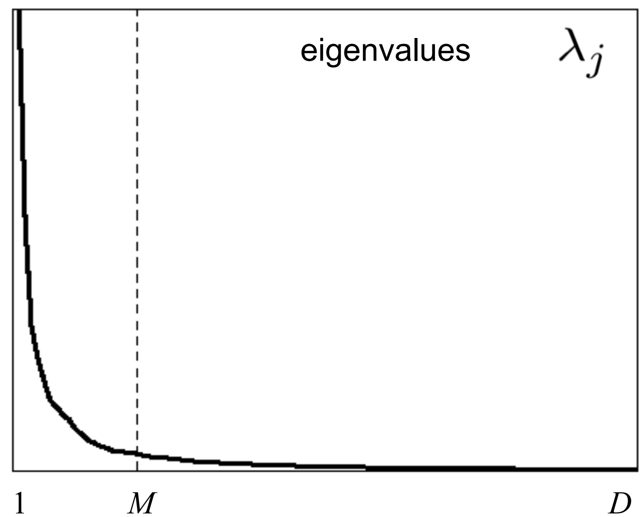
Fleet & Szeliksi

PCA

– Over all rank M bases, $\mathbf{B}$ minimizes the MSE of approximation

$$\sum_{j=M+1}^{D} \lambda_j$$

- Choosing subspace dimension M:
  – look at decay of the eigenvalues as a function of M
  – Larger M means lower expected error in the subspace data approximation

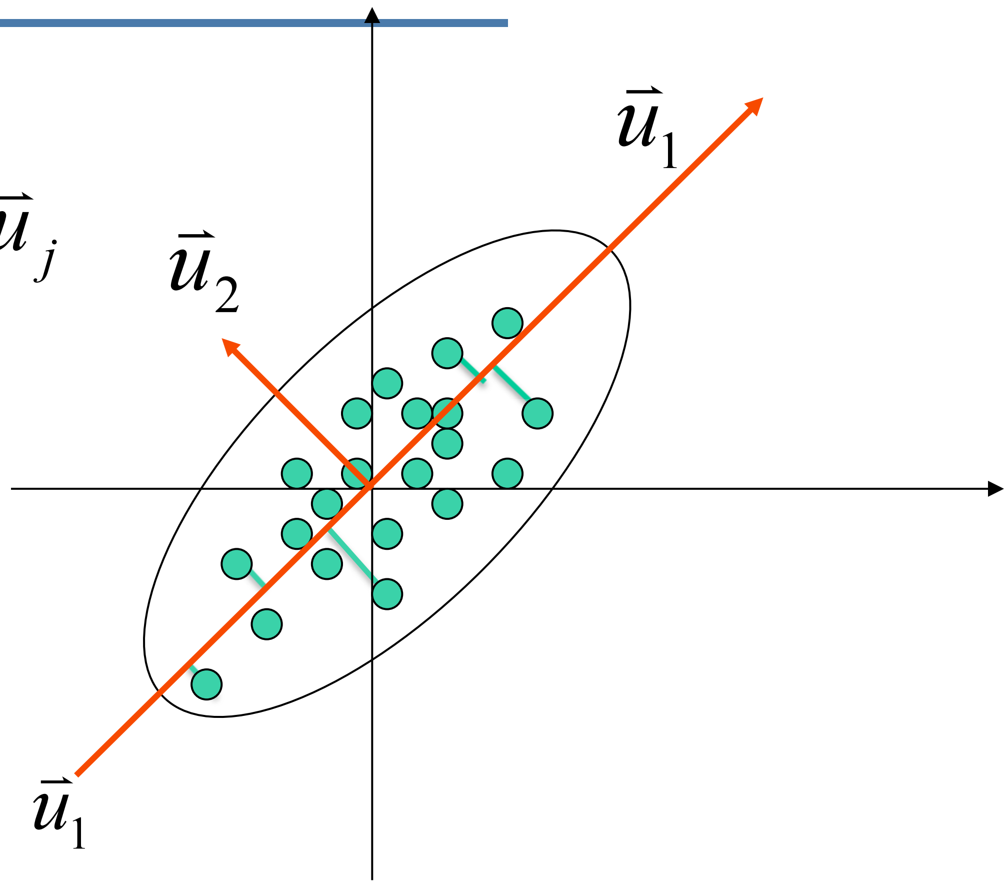


Fleet & Szeliski

Intuition

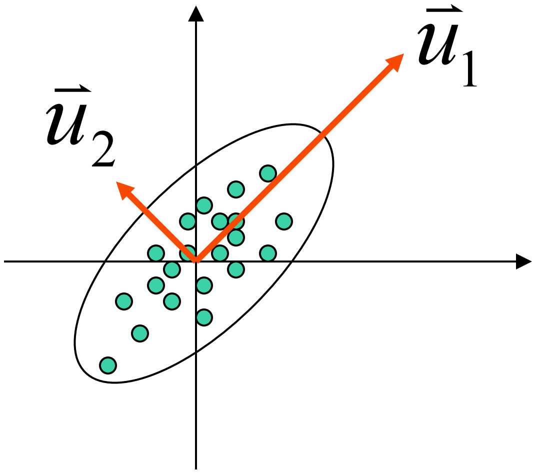
$$\vec{x}^{\,n} - \bar{x} = \sum_{i=1}^{M} a_i \vec{u}_i + \sum_{j=M+1}^{D} b_j \vec{u}_j$$

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



Intuition

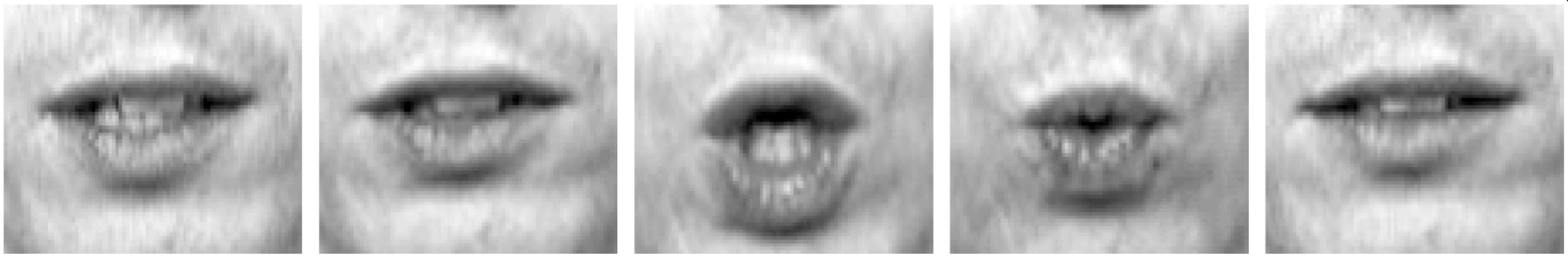
$$\vec{x}^n - \bar{x} = \sum_{i=1}^{M} a_i \vec{u}_i + \sum_{j=M+1}^{D} b_j \vec{u}_j$$



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

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

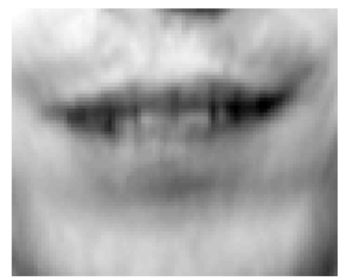
Mouth images

$X =$ 

Images 72x88 pixels.

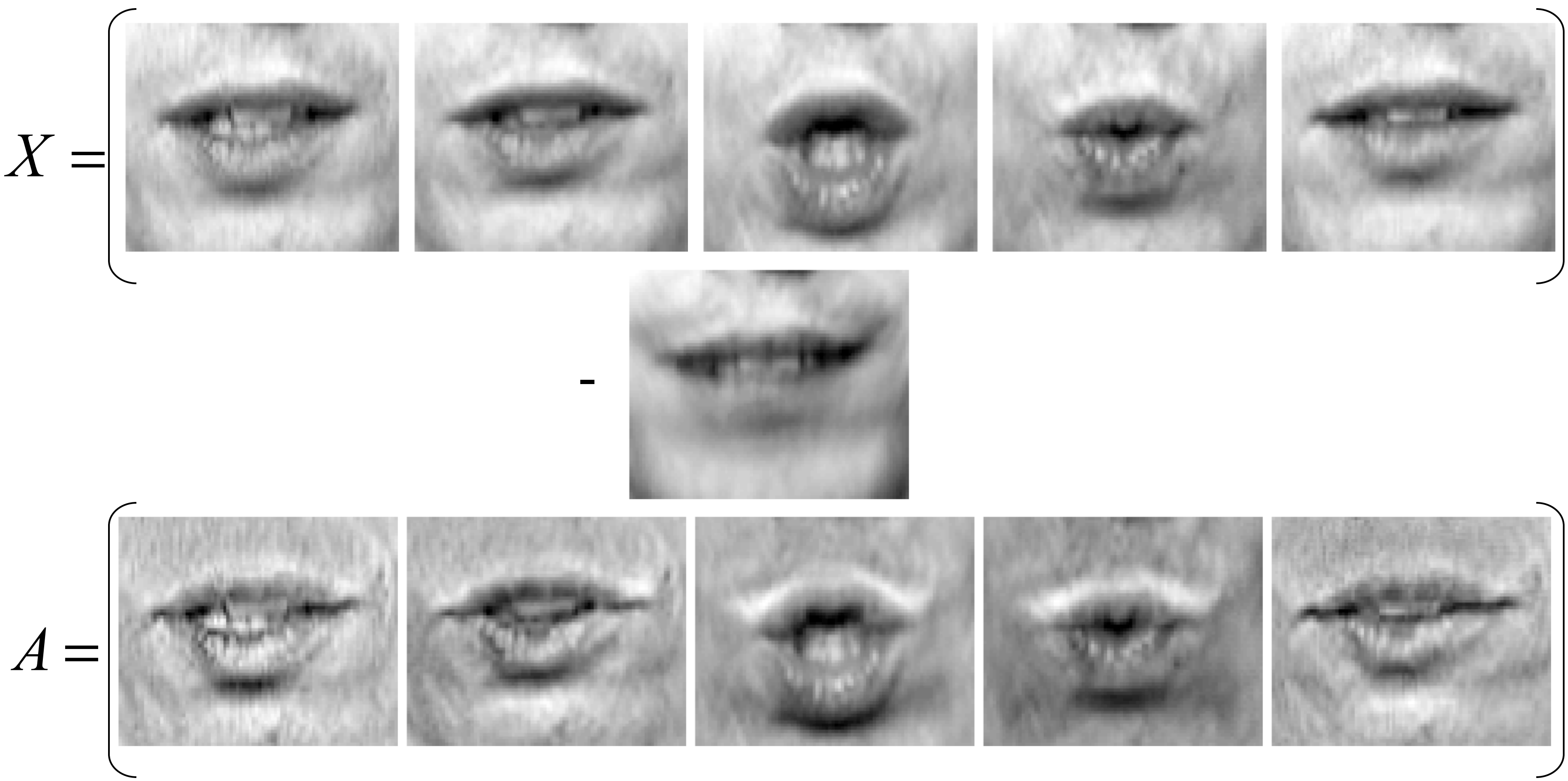
35 example mouths

A is N columns by 6336 pixels.



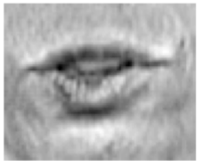
mean

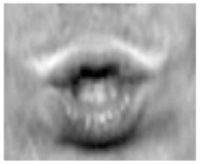
Mouth matrix



Covariance Matrix

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

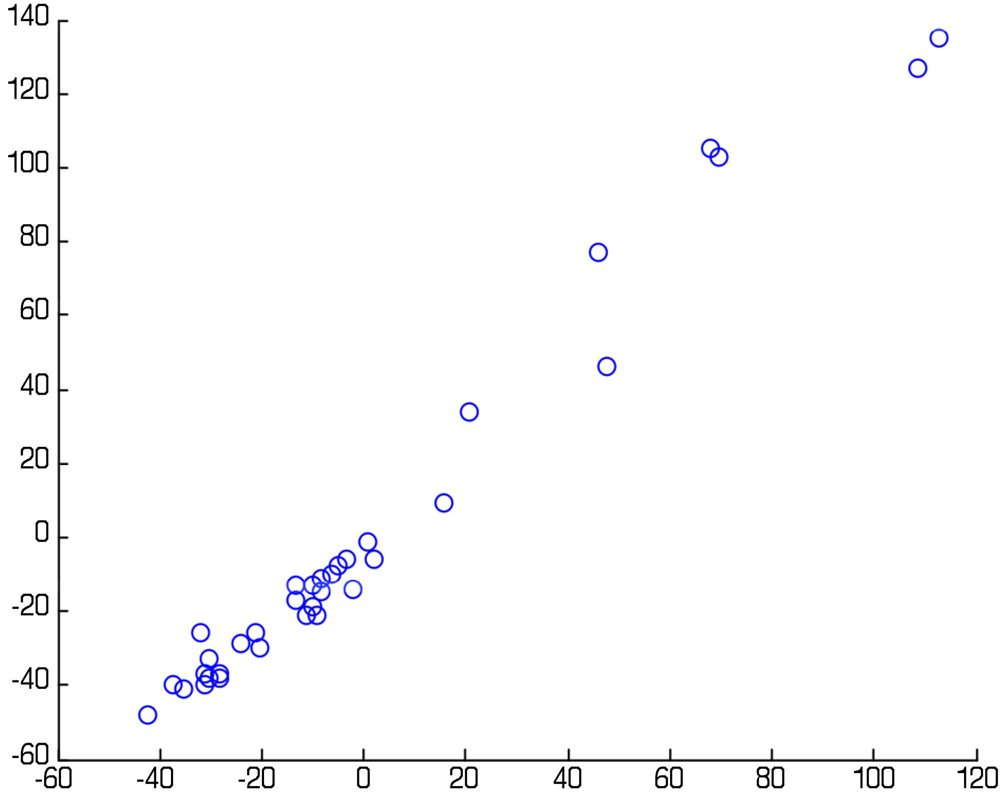


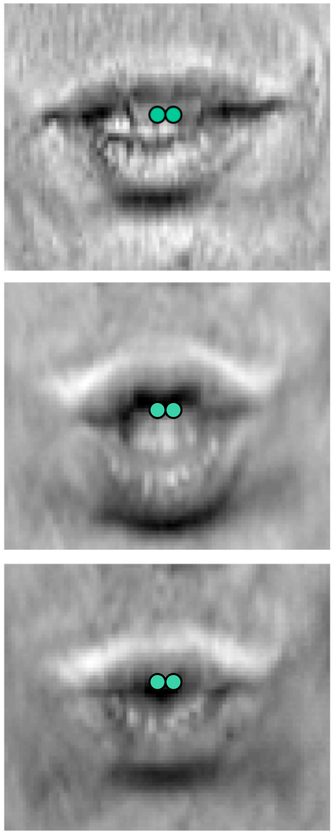


What is

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

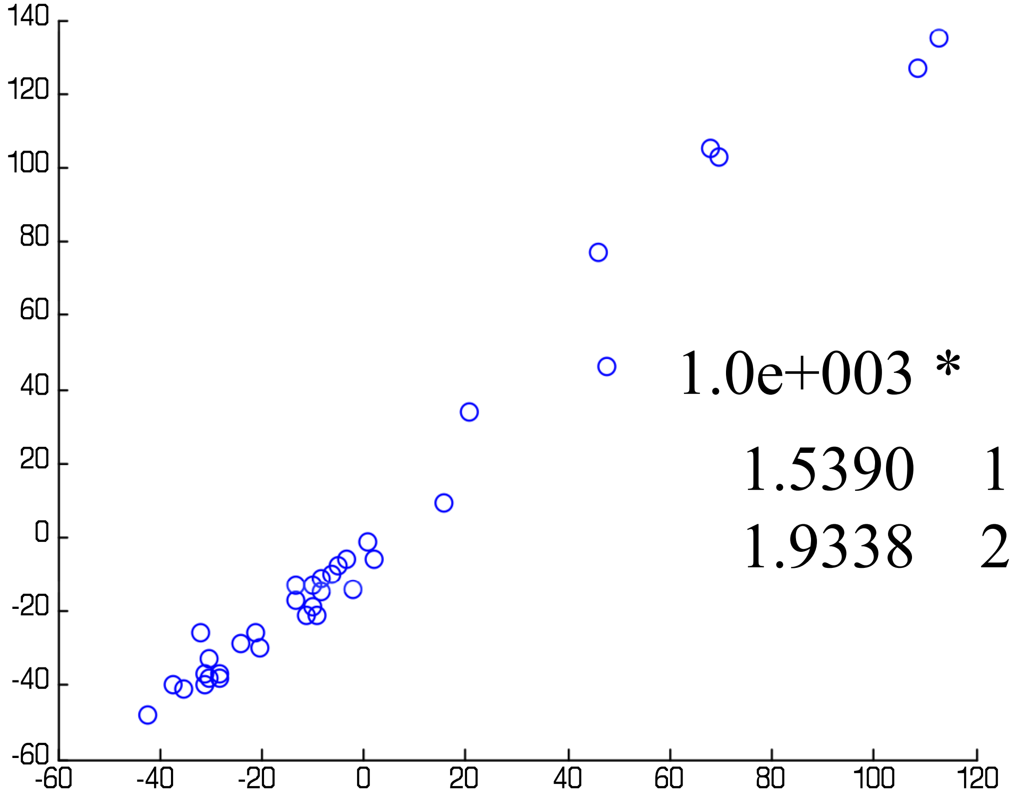
Correlation



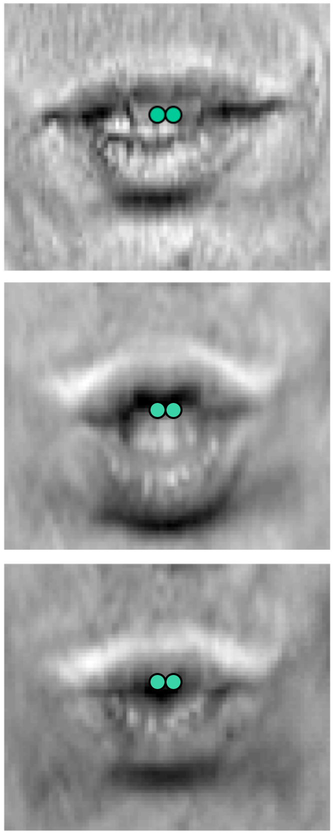


corr(A(:,30*88+46), A(:,30*88+47)) = 0.9864

Covariance

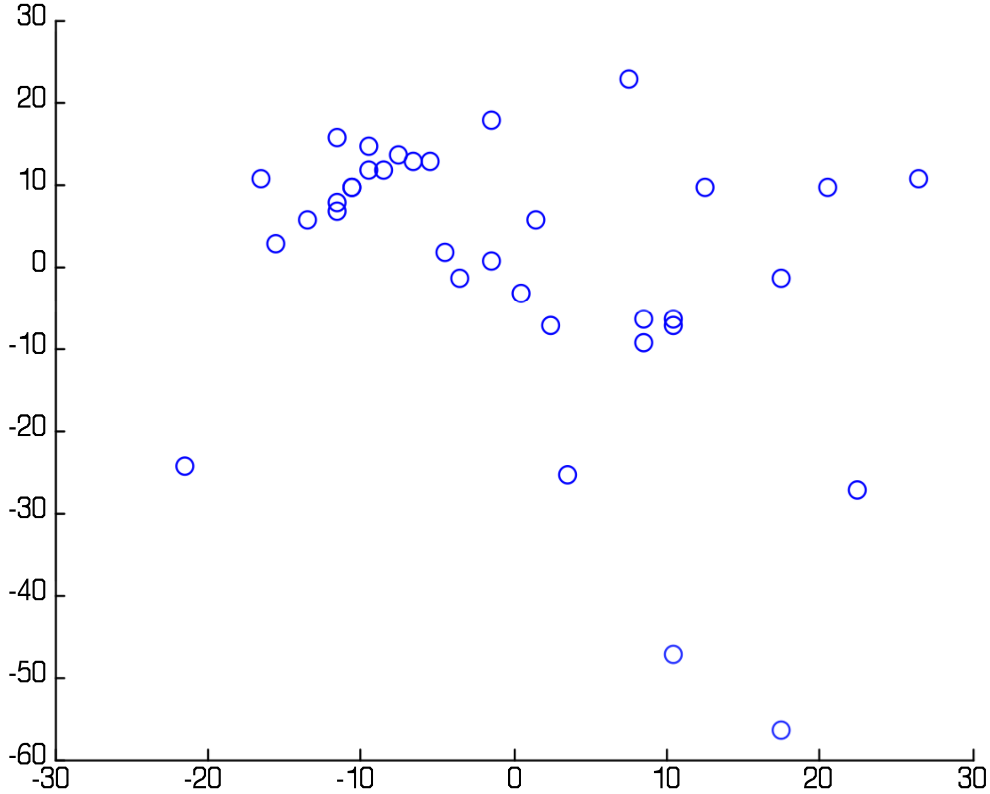


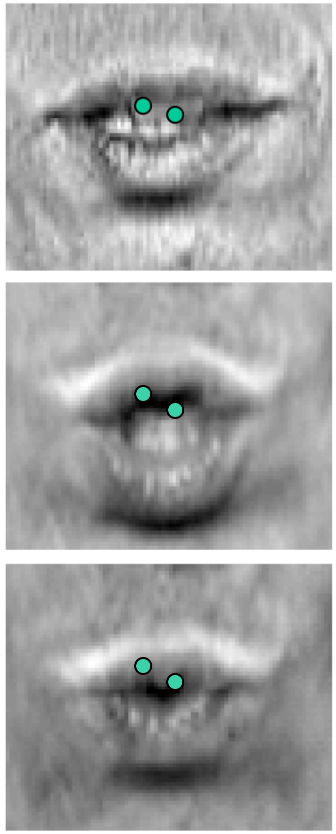
$$1.0e+003 * \begin{matrix} 1.5390 & 1.9338 \\ 1.9338 & 2.4974 \end{matrix}$$



cov(A(:,30*88+46), A(:,30*88+47))

Correlation





corr(A(:, 29*88+40), A(:,30*88+43)) = -0.3641

Covariance Matrix

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

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

Mouth matrix

$$AA^T = \begin{bmatrix} \text{AA' is 6336x6336 pixels.} \end{bmatrix}$$

Mouth matrix

$$AA^T =$$

What does the diagonal look like?

$C = AA^T$

imagesc(reshape(diag(C),72,88));

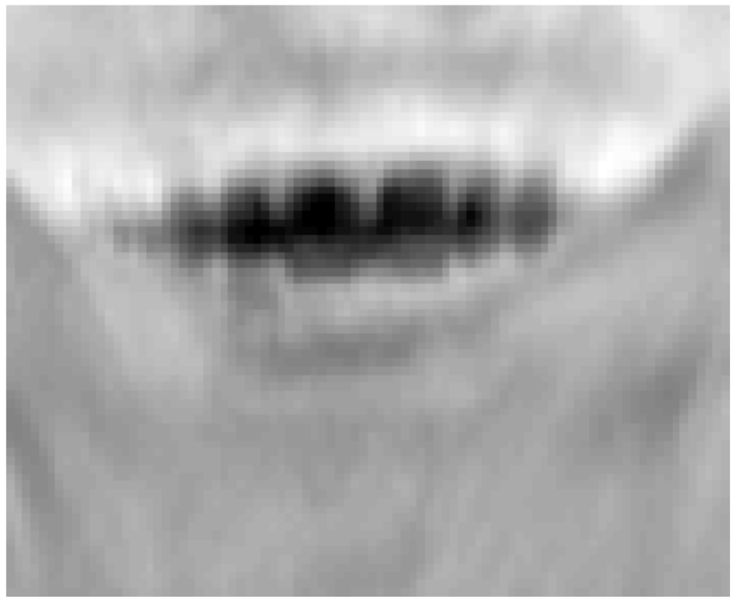
Mouth matrix

$$AA^T =$$

Why?

Mouth matrix

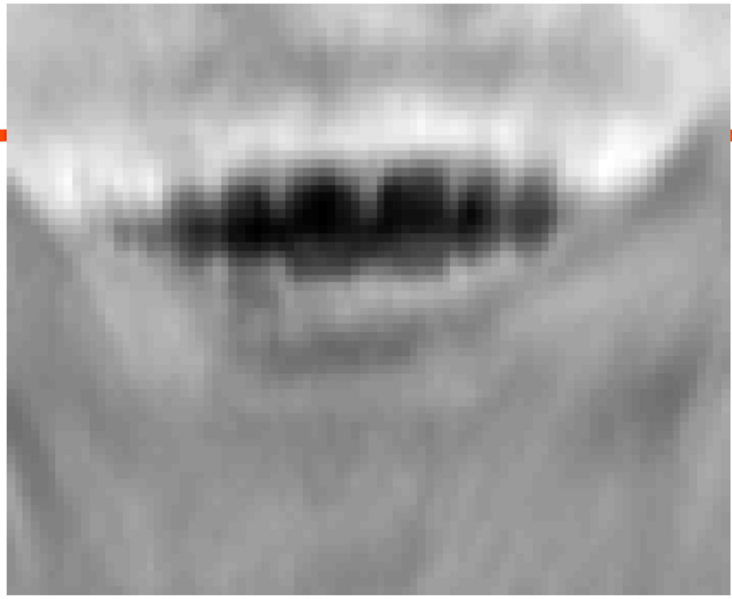
$AA^T =$



```
imagesc(reshape(C(:,1),72,88));
```

Mouth matrix

$$AA^T = \left[\quad \right]$$



imagesc(reshape(C(1,:),72,88)); ?

Mouth matrix

$$AA^T = \left[\quad\right]$$

imagesc(reshape(C(:,36*88+44),72,88));

Computing using SVD

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.

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

Singular Value Decomposition allows us to write A as:

$$A = U\Sigma V^T$$

SVD and PCA

$$A = U \Sigma V^T$$

Orthonormal columns

$$\begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_D \end{bmatrix}$$

Diagonal matrix of *singular values*

SVD and PCA

How are they related?

SVD and PCA

Note:

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

$$= \frac{1}{N-1} U\Sigma V^T (U\Sigma V^T)^T$$

$$= \frac{1}{N-1} U\Sigma V^T V\Sigma U^T$$

$$= \frac{1}{N-1} U\Sigma^2 U^T$$

In other words

$$C\vec{u}_i = \frac{\sigma^2}{N-1}\vec{u}_i$$

i.e. the singular vectors of A are the eigenvectors of the covariance matrix C.

SVD and PCA

- So the columns of *U* are the eigenvectors
- And the eigenvalues are just

$$\lambda_k = \frac{\sigma_k^2}{N-1}$$

Computing using SVD

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

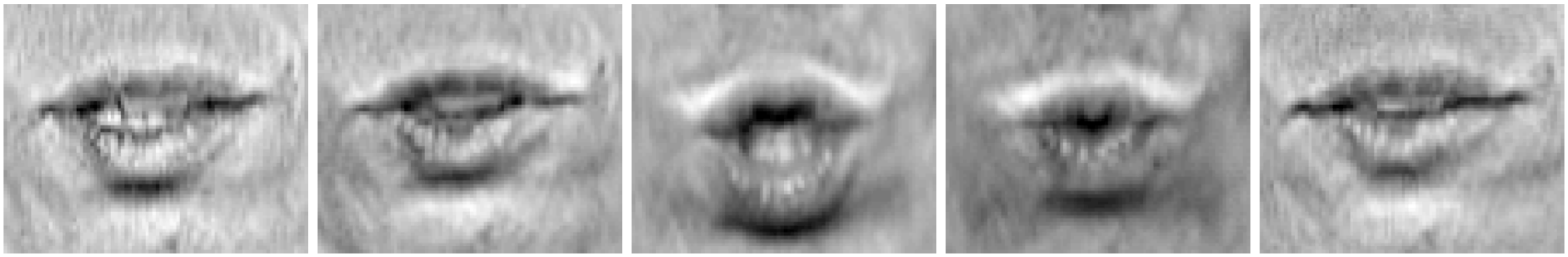
Singular Value Decomposition allows us to write *A* as:

$$A = U\Sigma V^T$$

$$\lambda_k = \frac{\sigma_k^2}{N-1}$$

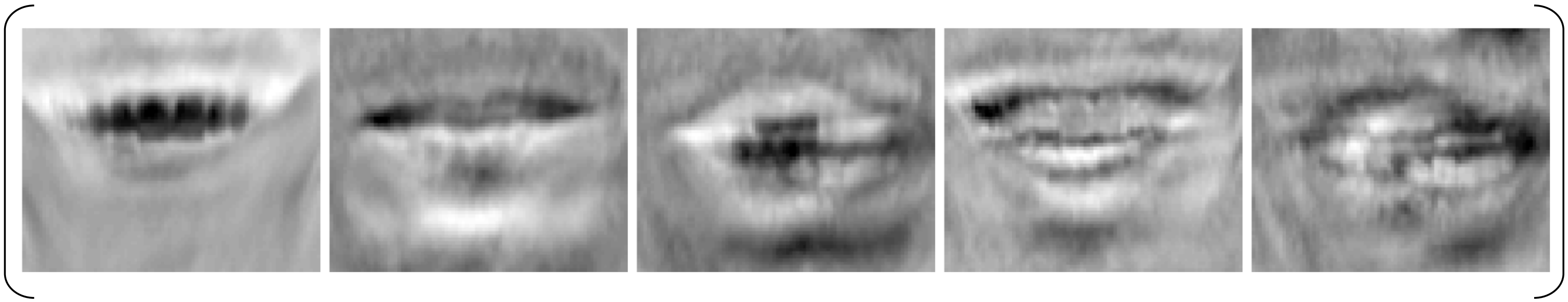
Mouth matrix

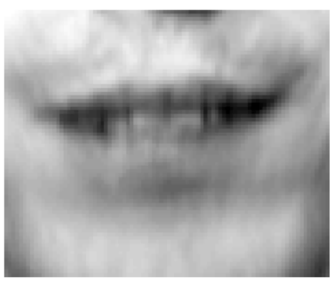
SVD



$$= U\Sigma V^T$$

SVD

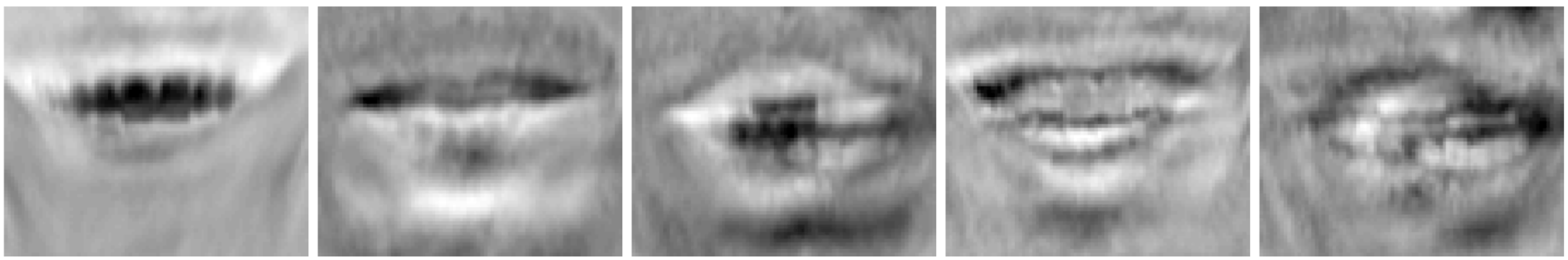
$U =$ 

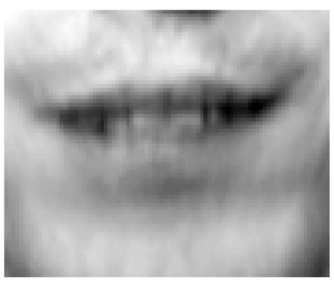


mean

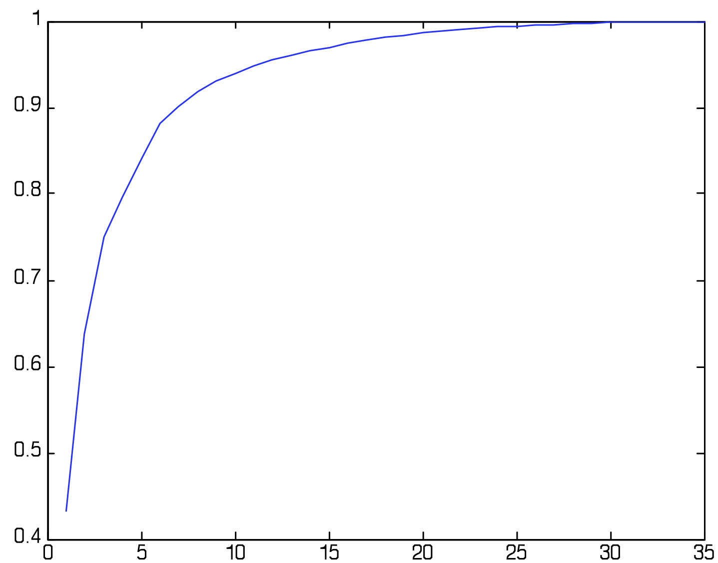
SVD

$U =$





mean



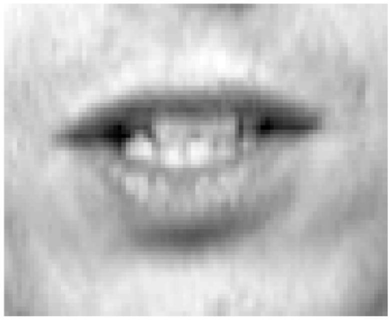
cvar=cumsum(…);

First 5 = 85%

First 6 = 90%

First 11 = 95%

Approximating a mouth



-

=

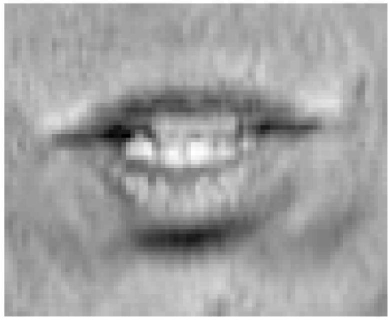
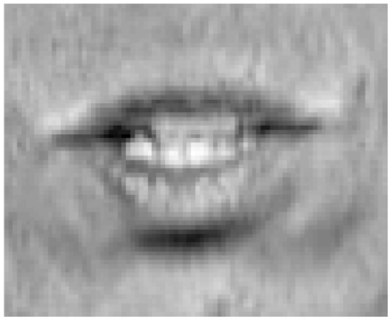
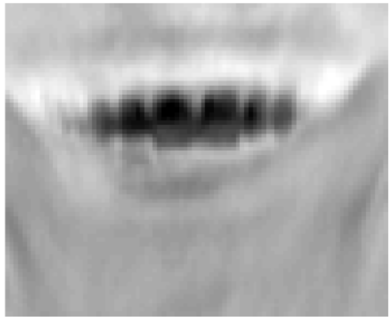


Image to approximate

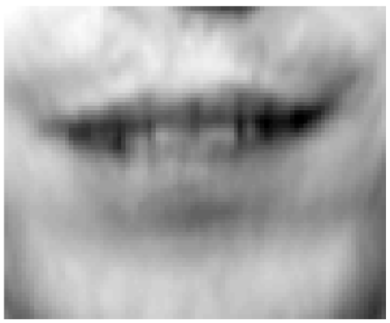
Mean

Approximating a mouth

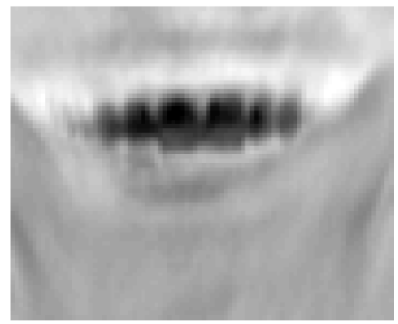
 $\bullet$  $= 587.1616 = a_1$

Project input image onto the first eigen basis (dot product).

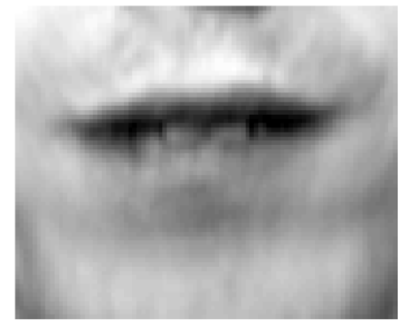
Approximating a mouth



+ 587.1616 *



=



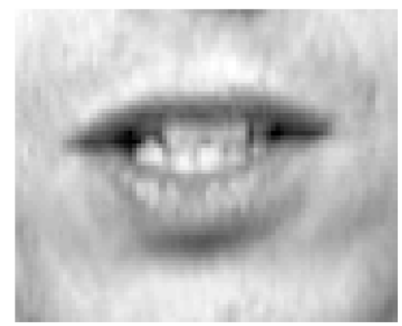
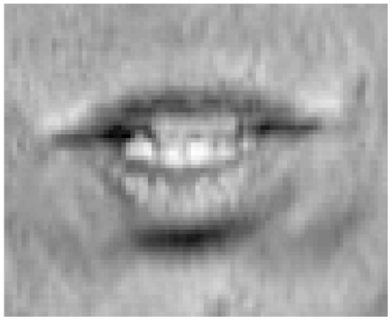
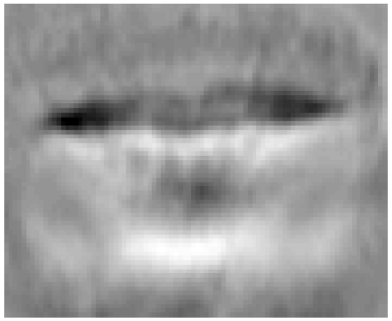


Image to approximate

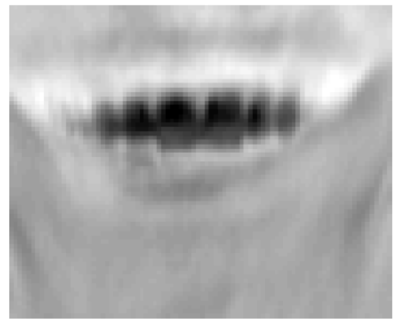
Approximating a mouth

 •  $= -363.8750 = a_2$

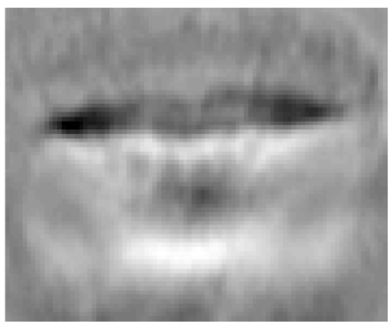
Project input image onto the second basis (dot product).

Approximating a mouth

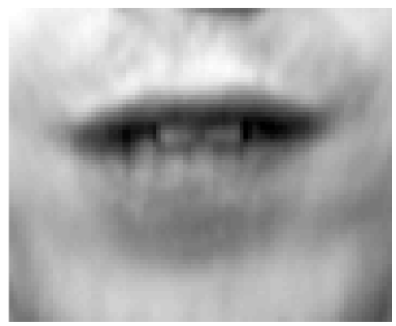
+ 587.1616 *



-363.8750 *



=



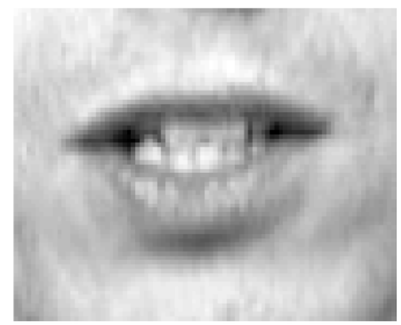
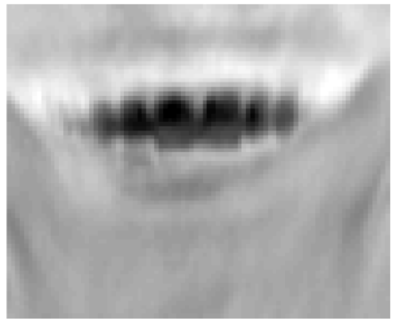


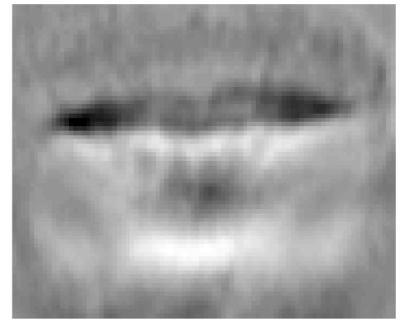
Image to approximate

Approximating a mouth

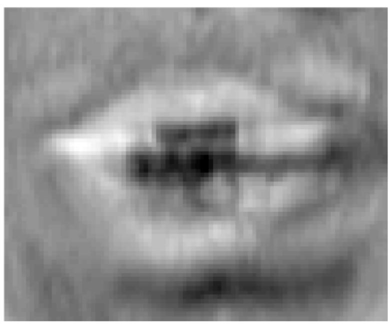
+ 587 *



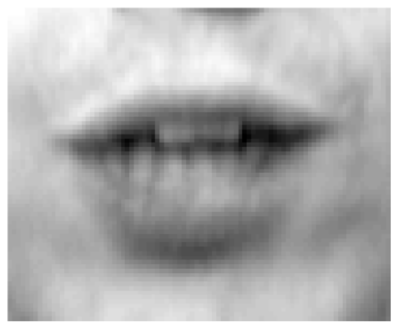
-363 *



-763 *



=



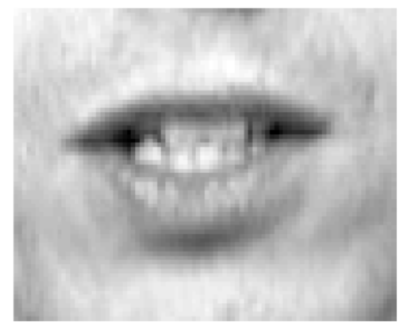
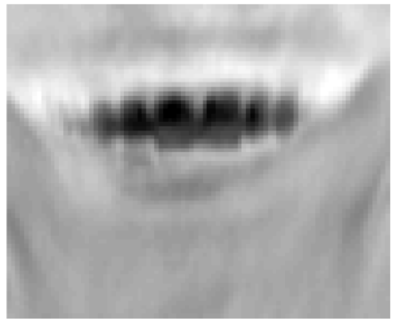


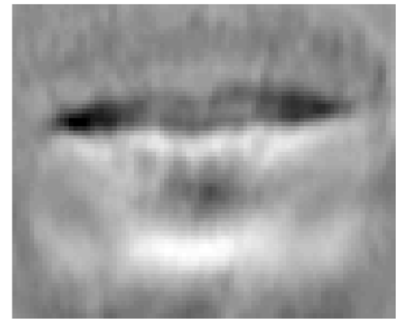
Image to approximate

Approximating a mouth

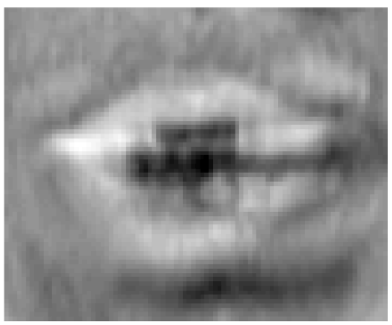
+ 587 *



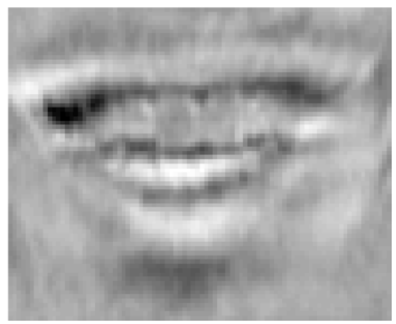
-363 *



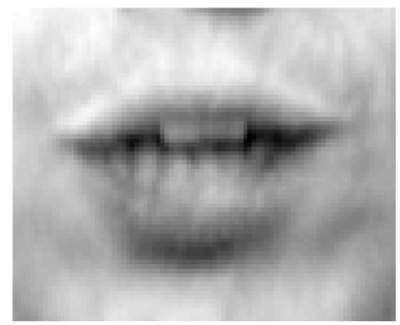
-763 *



-286 *

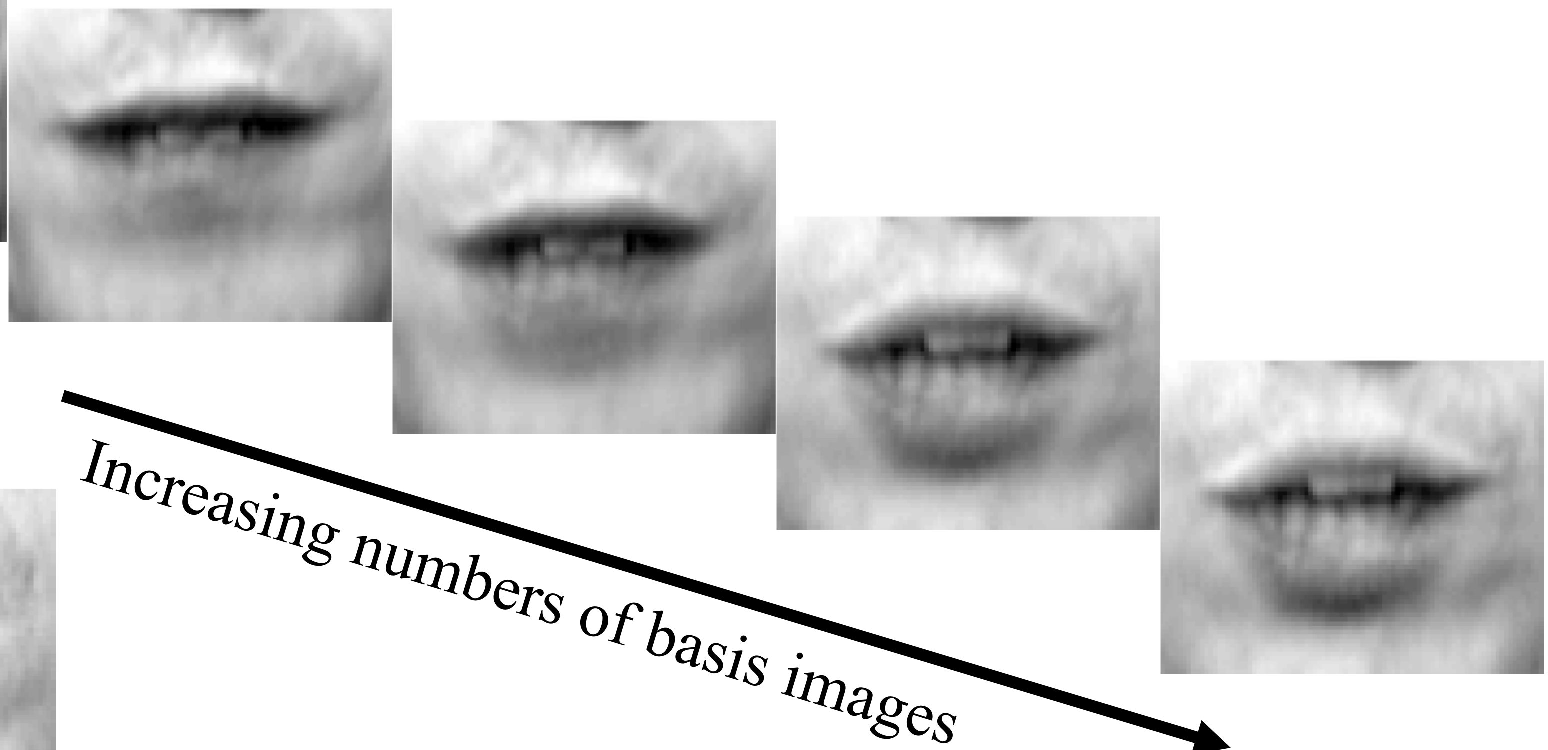


=



Approximating a mouth

mean



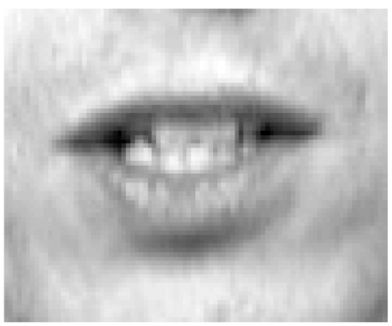


Image to approximate

Bases Revisited

Basis vector

Linear coefficients

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_M \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_{n \times m} \end{bmatrix}$$

Image as a vector

Projection of the image onto a set of basis vectors.

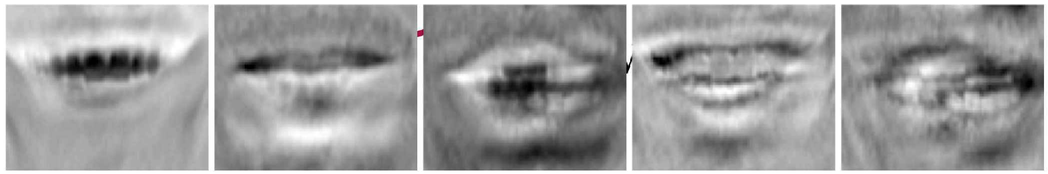
Bases Revisited

$$\vec{c} = B^T \vec{p}$$

$$B\vec{c} = BB^T \vec{p} = \vec{p}$$

$$\vec{p} = B(B^T \vec{p}) = B\vec{c}$$

Bases Revisited

$U =$ 

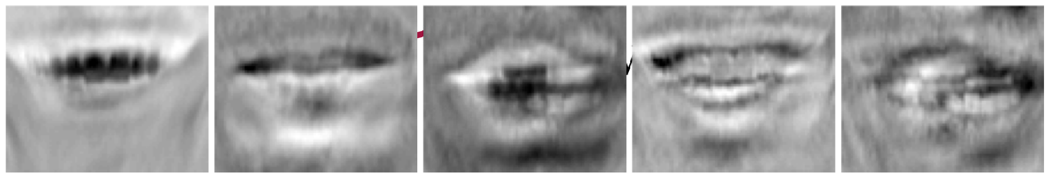
Linear coefficients

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_M \end{bmatrix} = U^T \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_{n \times m} \end{bmatrix}$$

Image as a vector

Projection of the image onto a set of basis vectors.

Bases Revisited

$U =$ 

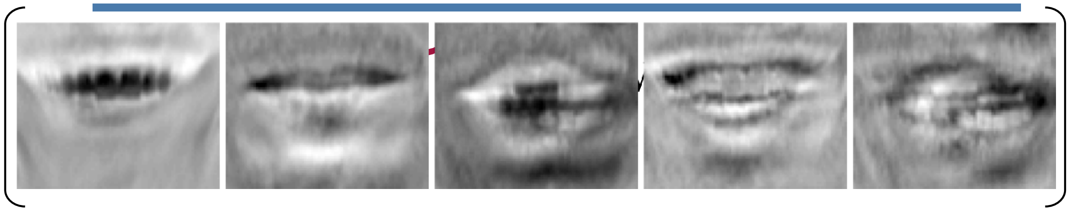
Linear coefficients

$$L = \begin{bmatrix} c_1 & c_1 & & c_1 \\ c_2 & c_2 & & c_2 \\ c_3 & c_3 & \cdots & c_3 \\ \vdots & \vdots & & \vdots \\ c_M & c_M & & c_M \end{bmatrix} = U^T \begin{bmatrix} p_1 & p_1 & & p_1 \\ p_2 & p_2 & & p_2 \\ p_3 & p_3 & \cdots & p_3 \\ \vdots & \vdots & & \vdots \\ p_{n\times m} & p_{n\times m} & & p_{n\times m} \end{bmatrix}$$

Images as a vectors

Projection of the image onto a set of basis vectors.

Bases Revisited

$U =$ 

Linear coefficients

$$L = \begin{bmatrix} c_1 & c_1 & & c_1 \\ c_2 & c_2 & & c_2 \\ c_3 & c_3 & \cdots & c_3 \\ \vdots & \vdots & & \vdots \\ c_M & c_M & & c_M \end{bmatrix}$$

What about LL^T