

Dimensionality Reduction

Fereshteh Sadeghi

CSEP 546

Motivation

- Clustering
 - One way to summarize a complex real-valued data point with a single categorical variable
- Dimensionality reduction
 - Another way to simplify complex high-dimensional data
 - Summarize data with a lower dimensional real valued vector

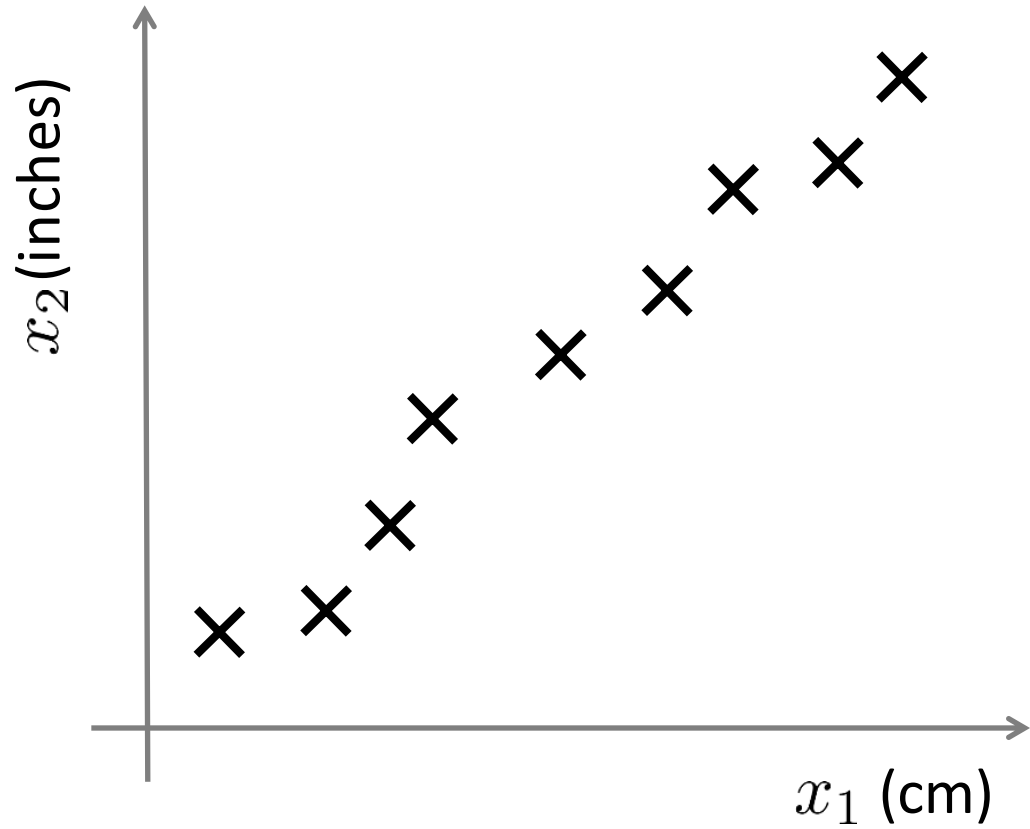
Motivation

- Clustering
 - One way to summarize a complex real-valued data point with a single categorical variable
- Dimensionality reduction
 - Another way to simplify complex high-dimensional data
 - Summarize data with a lower dimensional real valued vector



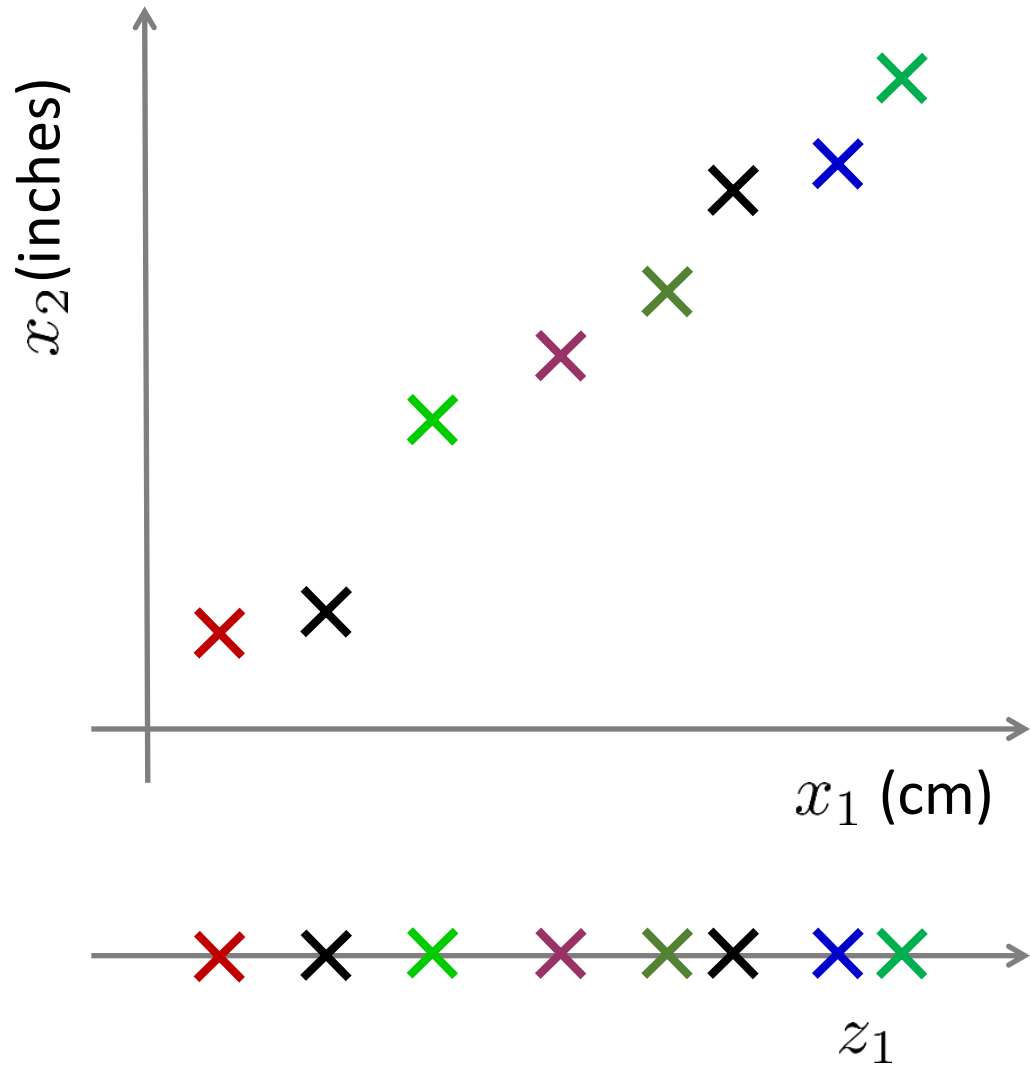
- Given data points in d dimensions
- Convert them to data points in $r < d$ dimensions
- With minimal loss of information

Data Compression



Reduce data from
2D to 1D

Data Compression



Reduce data from
2D to 1D

$$x^{(1)} \rightarrow z^{(1)}$$

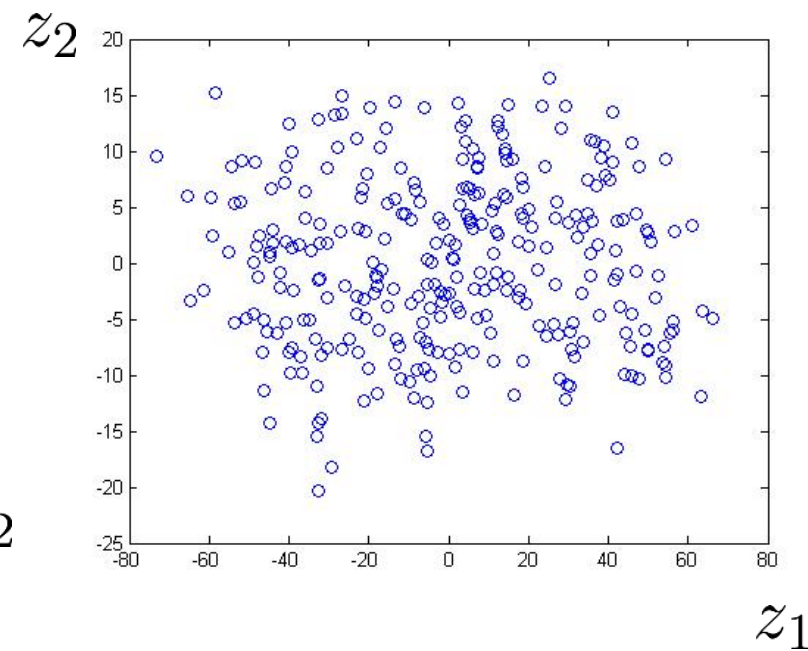
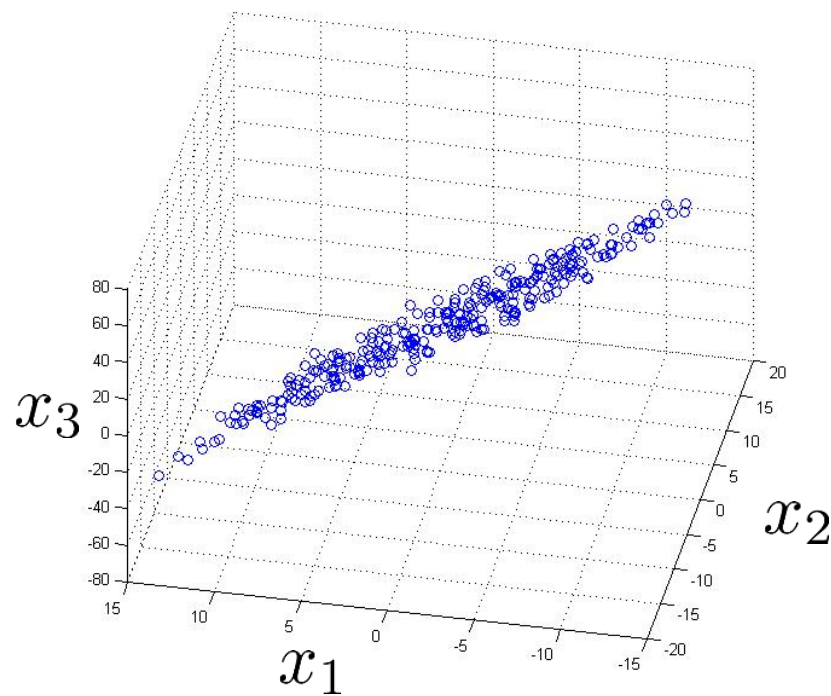
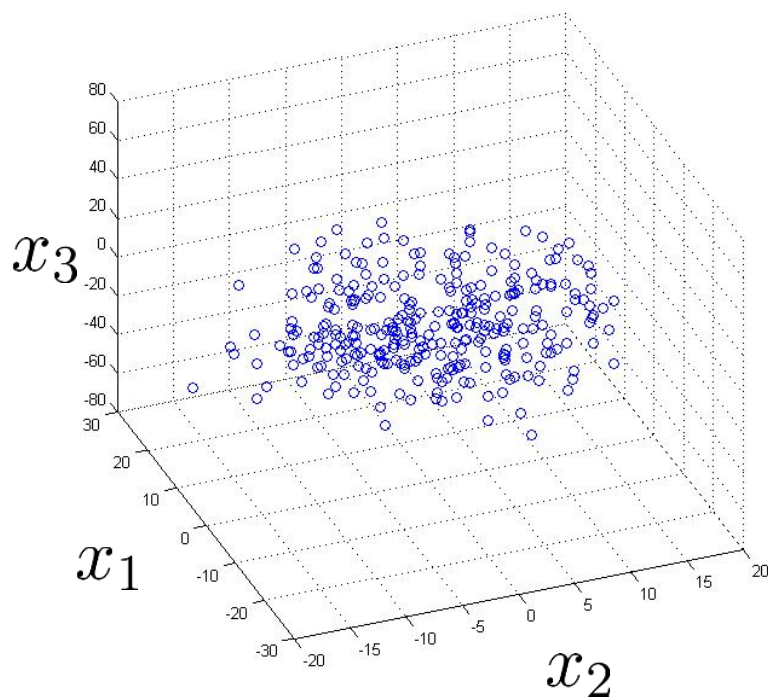
$$x^{(2)} \rightarrow z^{(2)}$$

⋮

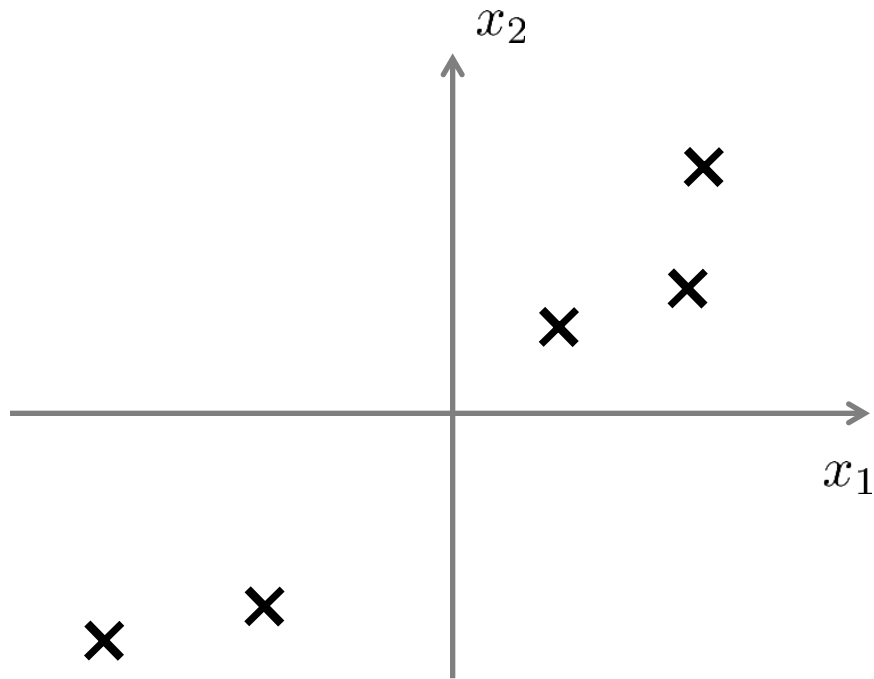
$$x^{(m)} \rightarrow z^{(m)}$$

Data Compression

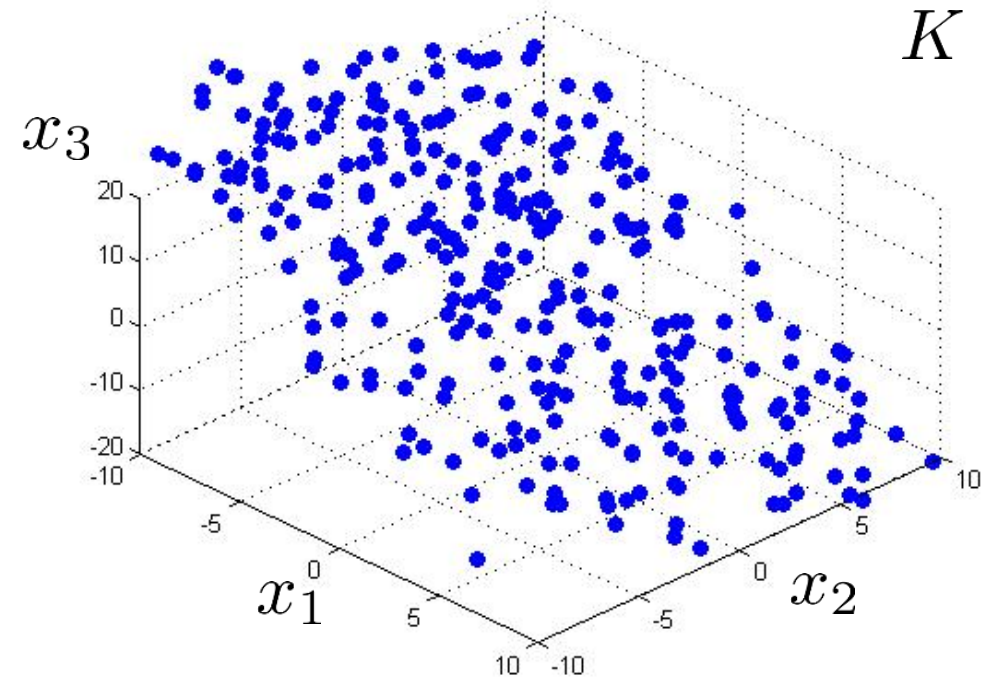
Reduce data from 3D to 2D



Principal Component Analysis (PCA) problem formulation



$$3D \rightarrow 2D$$
$$K = 2$$



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n -dimension to k -dimension: Find k vectors $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Principal Component Analysis

Goal: Find r -dim projection that best preserves variance

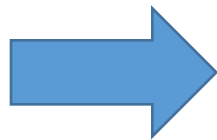
1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

where y is the new point, x is the old one,
and the rows of A are the eigenvectors

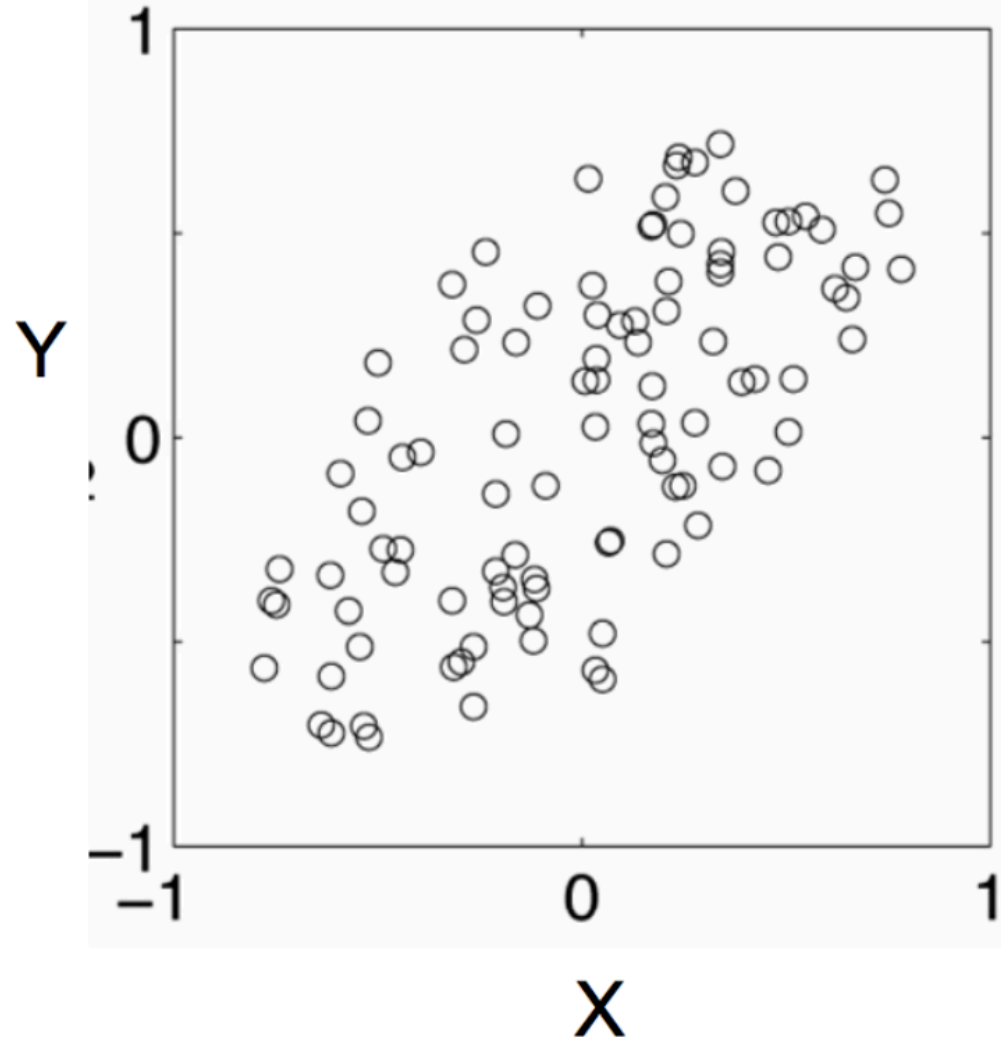
Covariance

- Variance and Covariance:
 - Measure of the “spread” of a set of points around their center of mass(mean)
- Variance:
 - Measure of the deviation from the mean for points in one dimension
- Covariance:
 - Measure of how much each of the dimensions vary from the mean with **respect to each other**

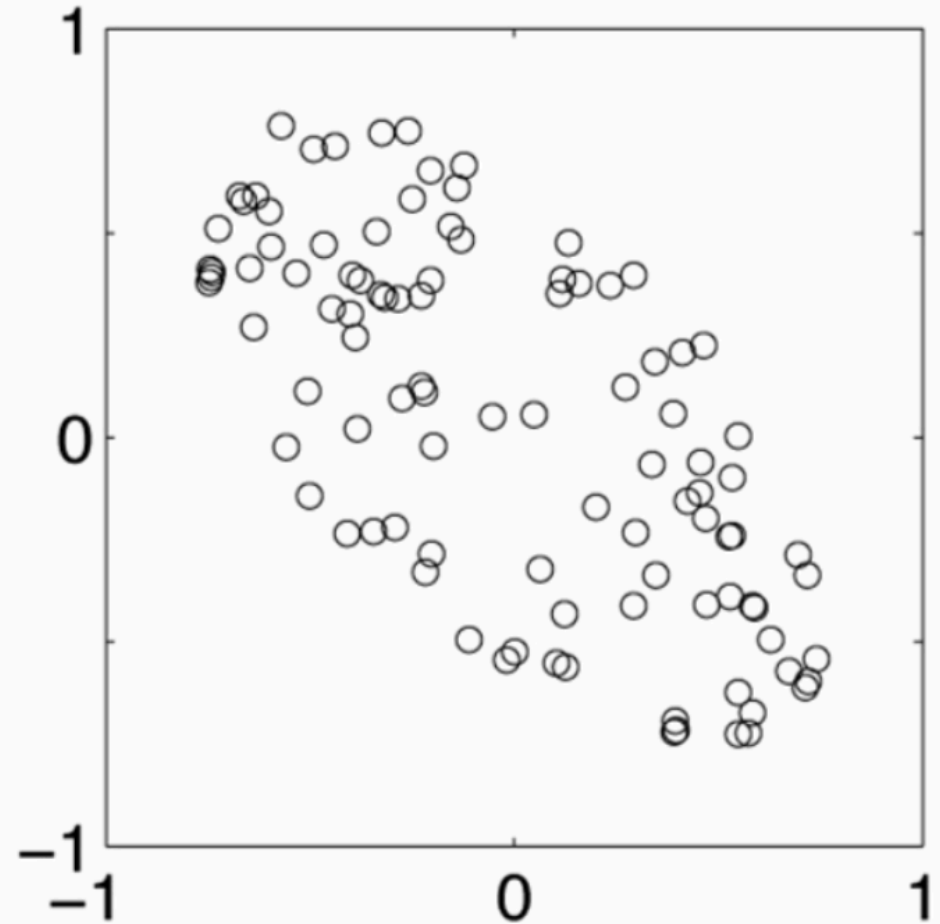


- **Covariance is measured between two dimensions**
- **Covariance sees if there is a relation between two dimensions**
- **Covariance between one dimension is the variance**

positive covariance



negative covariance



Positive: Both dimensions increase or decrease together

Negative: While one increase the other decrease

Covariance

- Used to find relationships between dimensions in high dimensional data sets

$$q_{jk} = \frac{1}{N} \sum_{i=1}^N (X_{ij} - E(X_j)) (X_{ik} - E(X_k))$$



The Sample mean

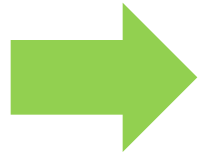
Eigenvector and Eigenvalue

$$Ax = \lambda x$$

A: Square Matirx

λ : Eigenvector or characteristic vector

λ : Eigenvalue or characteristic value



- *The zero vector can not be an eigenvector*
- *The value zero can be eigenvalue*

Eigenvector and Eigenvalue

$$Ax = \lambda x$$

A: Square Matrix

λ : Eigenvector or characteristic vector

λ : Eigenvalue or characteristic value



Example

Show $x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is an eigenvector for $A = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix}$

$$\text{Solution : } Ax = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\text{But for } \lambda = 0, \lambda x = 0 \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Thus, x is an eigenvector of A , and $\lambda = 0$ is an eigenvalue.

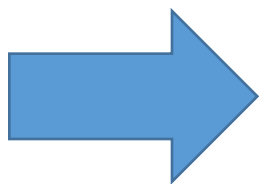
Eigenvector and Eigenvalue

$$Ax = \lambda x \quad \longrightarrow \quad \begin{aligned} Ax - \lambda x &= 0 \\ (A - \lambda I)x &= 0 \end{aligned}$$

If we define a new matrix B: \longrightarrow

$$\begin{aligned} B &= A - \lambda I \\ Bx &= 0 \end{aligned}$$

If B has an inverse: \longrightarrow $x = B^{-1}0 = 0$ **✗ BUT! an eigenvector cannot be zero!!**



x will be an eigenvector of A if and only if B does not have an inverse, or equivalently $\det(B)=0$:

$$\det(A - \lambda I) = 0$$

Eigenvector and Eigenvalue

Example 1: Find the eigenvalues of

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$$

$$\begin{aligned} |\lambda I - A| &= \begin{vmatrix} \lambda - 2 & 12 \\ -1 & \lambda + 5 \end{vmatrix} = (\lambda - 2)(\lambda + 5) + 12 \\ &= \lambda^2 + 3\lambda + 2 = (\lambda + 1)(\lambda + 2) \end{aligned}$$

two eigenvalues: $-1, -2$

Note: The roots of the characteristic equation can be repeated. That is, $\lambda_1 = \lambda_2 = \dots = \lambda_k$.
If that happens, the eigenvalue is said to be of multiplicity k .

Example 2: Find the eigenvalues of

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$|\lambda I - A| = \begin{vmatrix} \lambda - 2 & -1 & 0 \\ 0 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 2 \end{vmatrix} = (\lambda - 2)^3 = 0$$

$\lambda = 2$ is an eigenvalue of multiplicity 3.

Principal Component Analysis

Input: $\mathbf{x} \in \mathbb{R}^D: \mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$

Set of basis vectors: $\mathbf{u}_1, \dots, \mathbf{u}_K$

Summarize a D dimensional vector \mathbf{x} with K dimensional feature vector $h(\mathbf{x})$

$$h(\mathbf{x}) = \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{x} \\ \mathbf{u}_2 \cdot \mathbf{x} \\ \dots \\ \mathbf{u}_K \cdot \mathbf{x} \end{bmatrix}$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$$

Basis vectors are orthonormal

$$\mathbf{u}_i^T \mathbf{u}_j = 0$$

$$\|\mathbf{u}_j\| = 1$$

New data representation $h(\mathbf{x})$

$$z_j = \mathbf{u}_j \cdot \mathbf{x}$$

$$h(\mathbf{x}) = [z_1, \dots, z_K]^T$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$$

New data representation $h(\mathbf{x})$

$$h(\mathbf{x}) = \mathbf{U}^T \mathbf{x}$$

$$h(\mathbf{x}) = \mathbf{U}^T (\mathbf{x} - \mu_0)$$

Empirical mean of the data



$$\mu_0 = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$$

The space of all face images

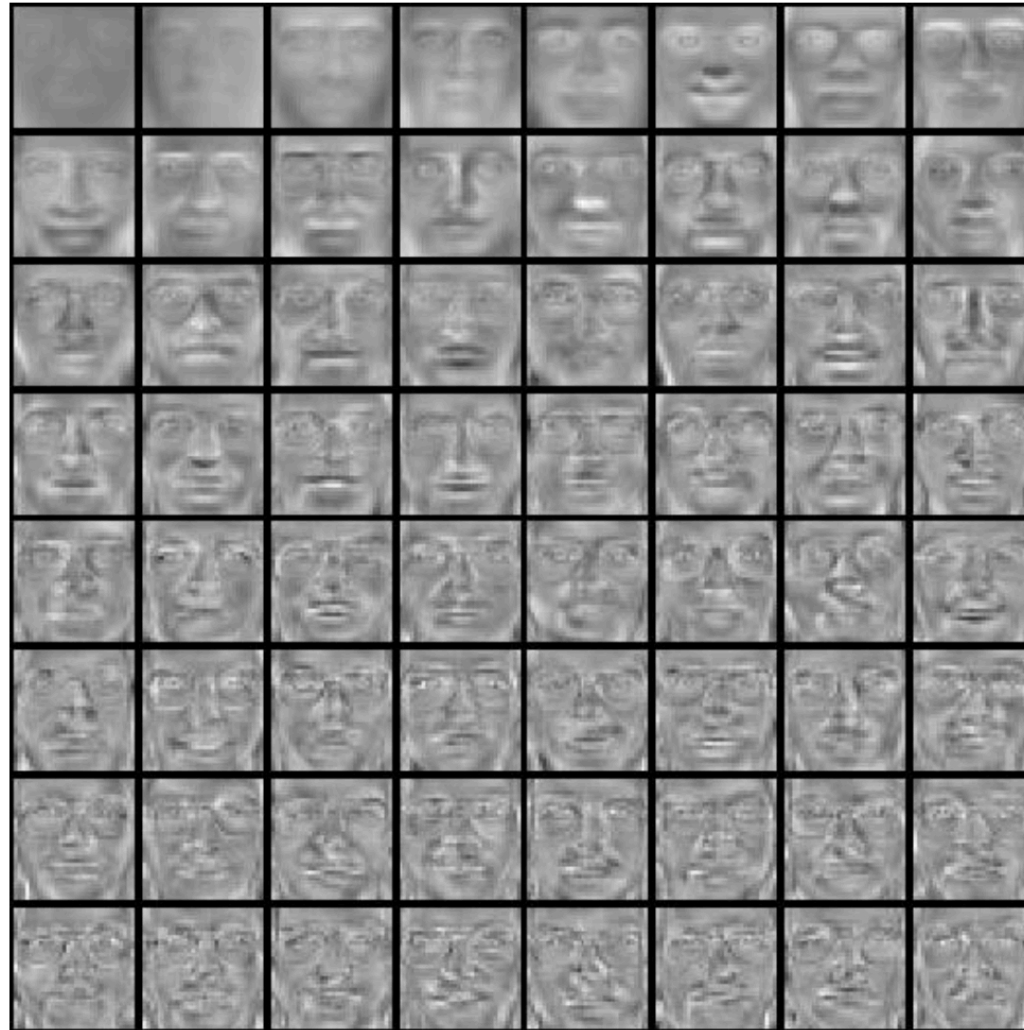
- When viewed as vectors of pixel values, face images are extremely high-dimensional
 - 100x100 image = 10,000 dimensions
 - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images



Eigenfaces example

Top eigenvectors: u_1, \dots, u_k

Mean: μ



Representation and reconstruction

- Face \mathbf{x} in “face space” coordinates:



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

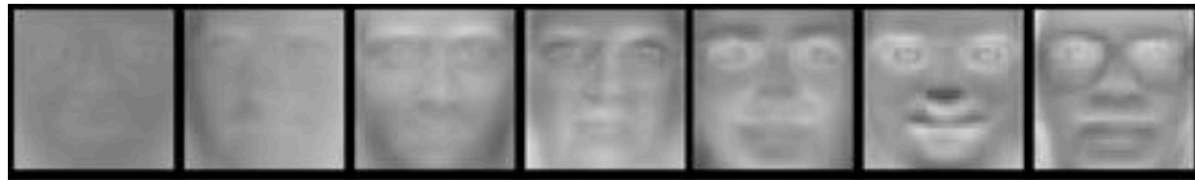
- Reconstruction:



=



+



$\hat{\mathbf{x}}$

=

μ

+

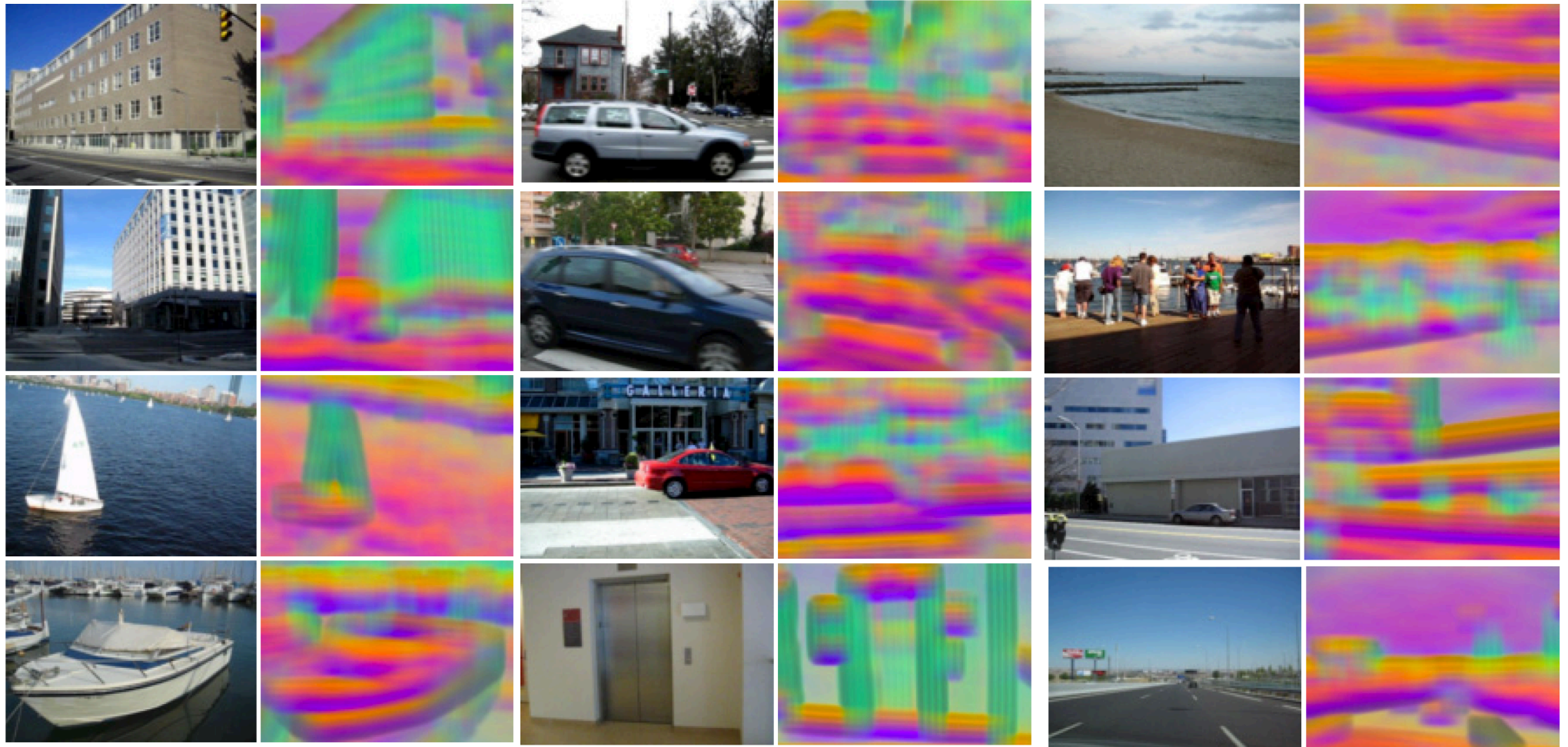
$w_1 u_1 + w_2 u_2 + w_3 u_3 + w_4 u_4 + \dots$

Reconstruction



After computing eigenfaces using 400 face images from ORL face database

SIFT feature visualization



- The top three principal components of SIFT descriptors from a set of images are computed
- Map these principal components to the principal components of the RGB space
- pixels with similar colors share similar structures

Application: Image compression



Original Image

- Divide the original 372x492 image into patches:
 - Each patch is an instance that contains 12x12 pixels on a grid
- View each as a 144-D vector

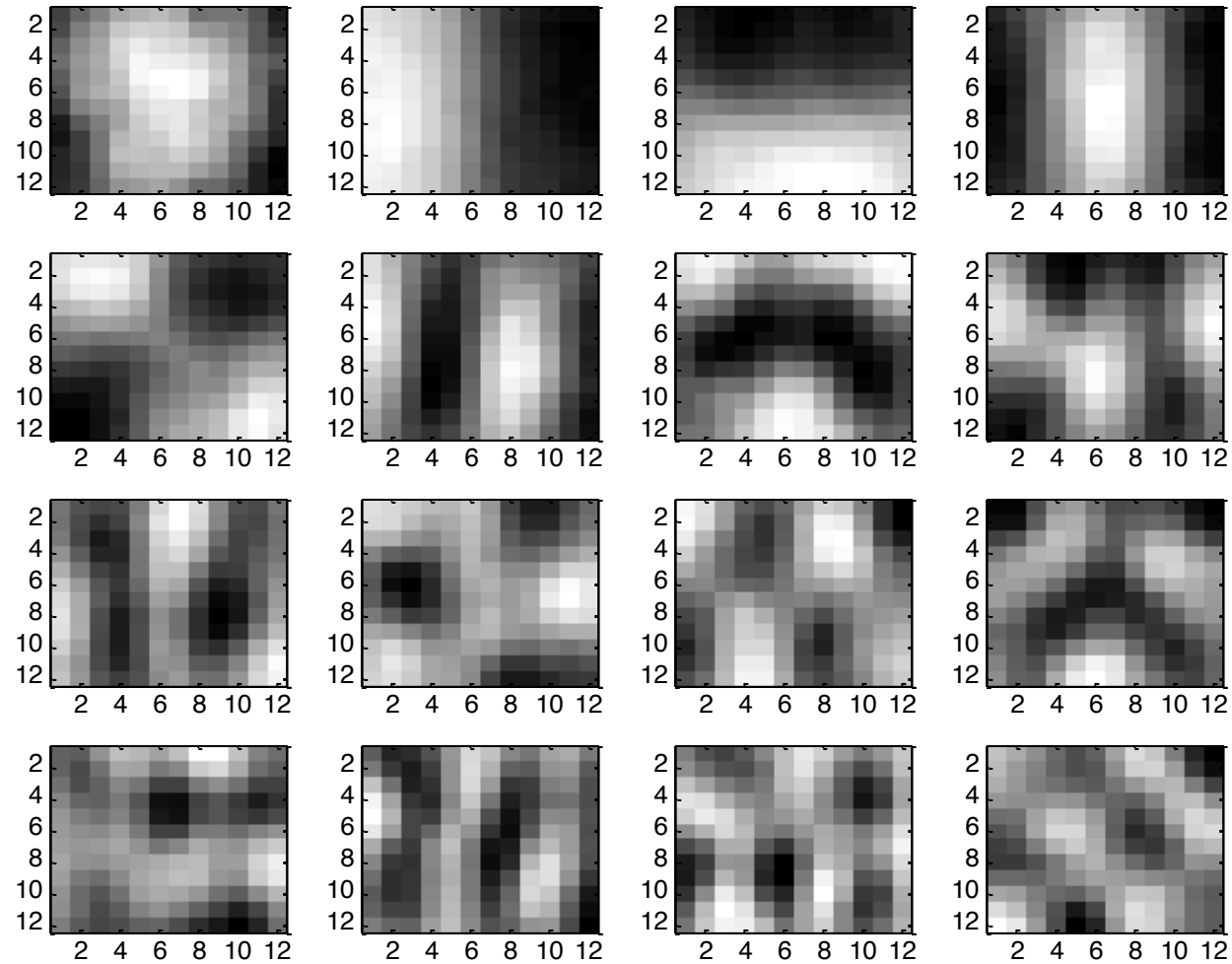
PCA compression: 144D \rightarrow 60D



PCA compression: 144D \rightarrow 16D



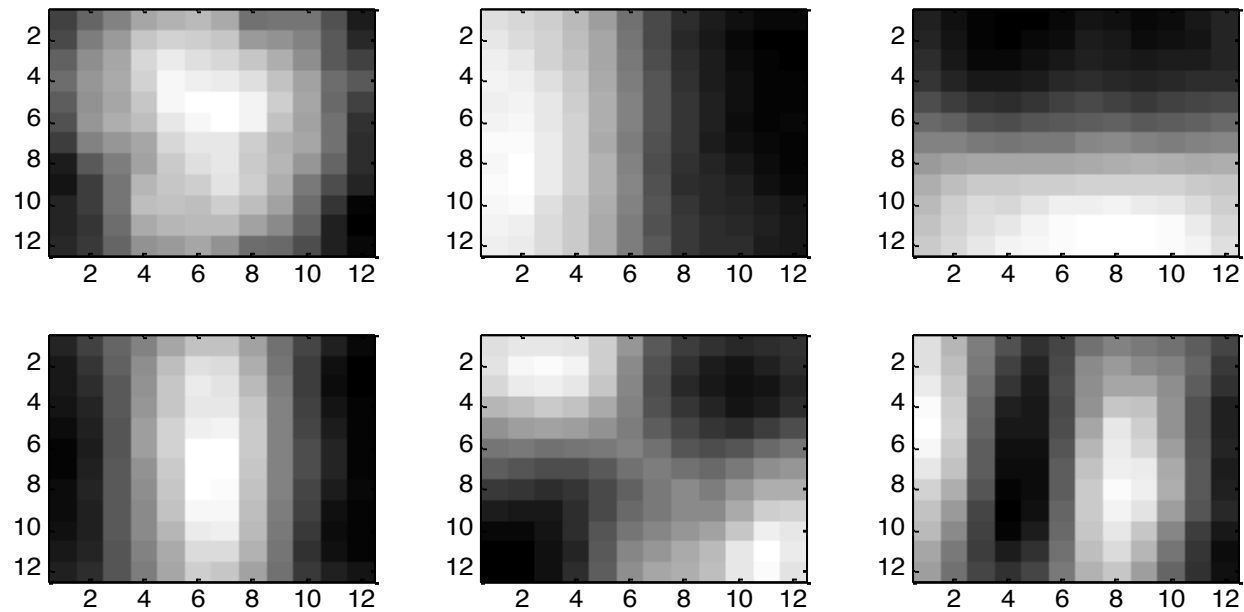
16 most important eigenvectors



PCA compression: 144D) 6D



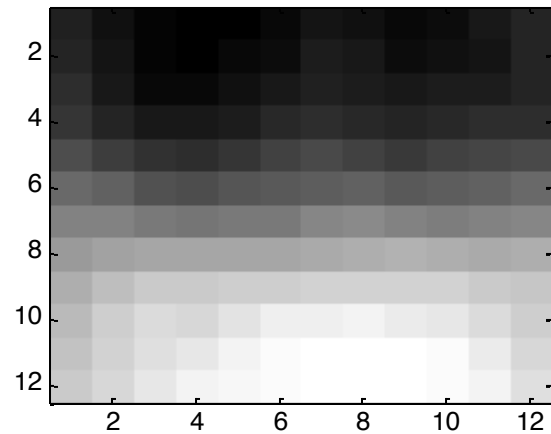
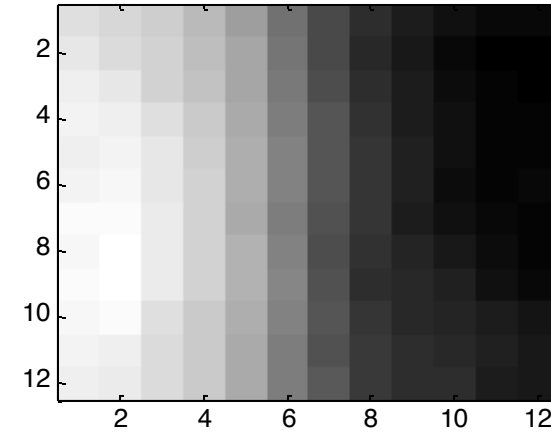
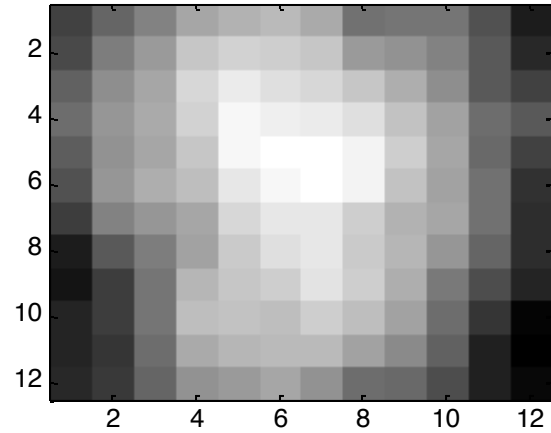
6 most important eigenvectors



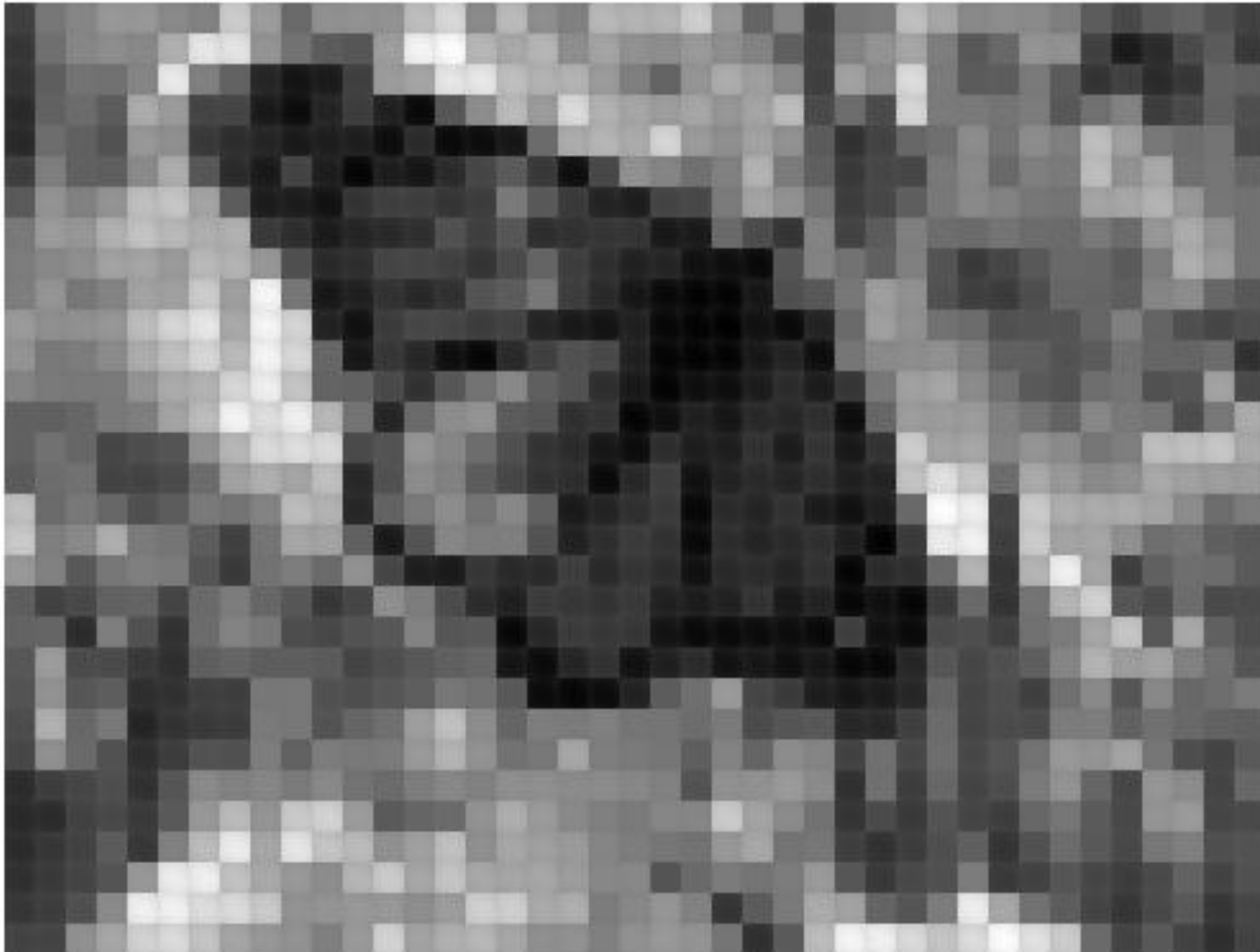
PCA compression: 144D \rightarrow 3D



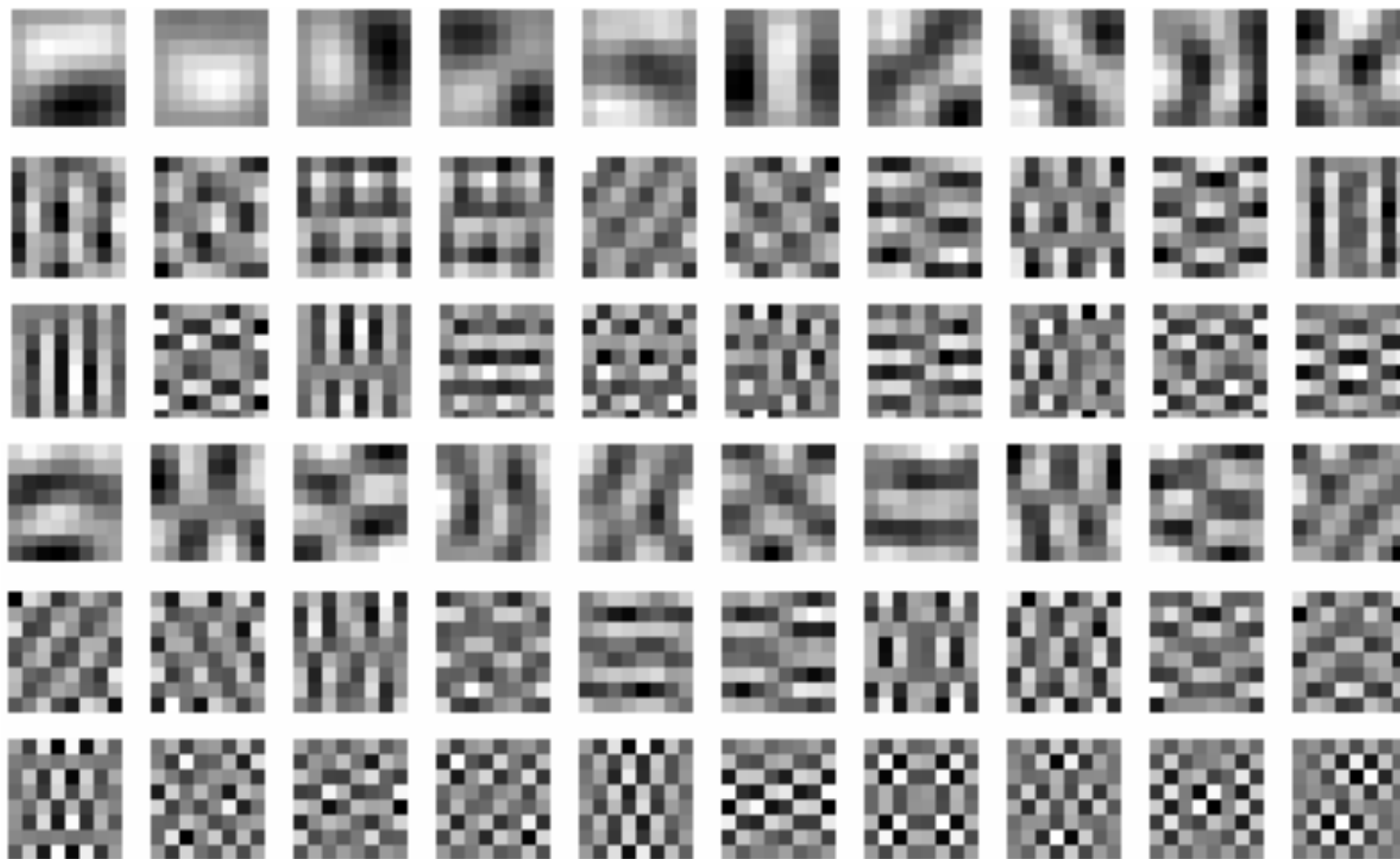
3 most important eigenvectors



PCA compression: 144D \rightarrow 1D

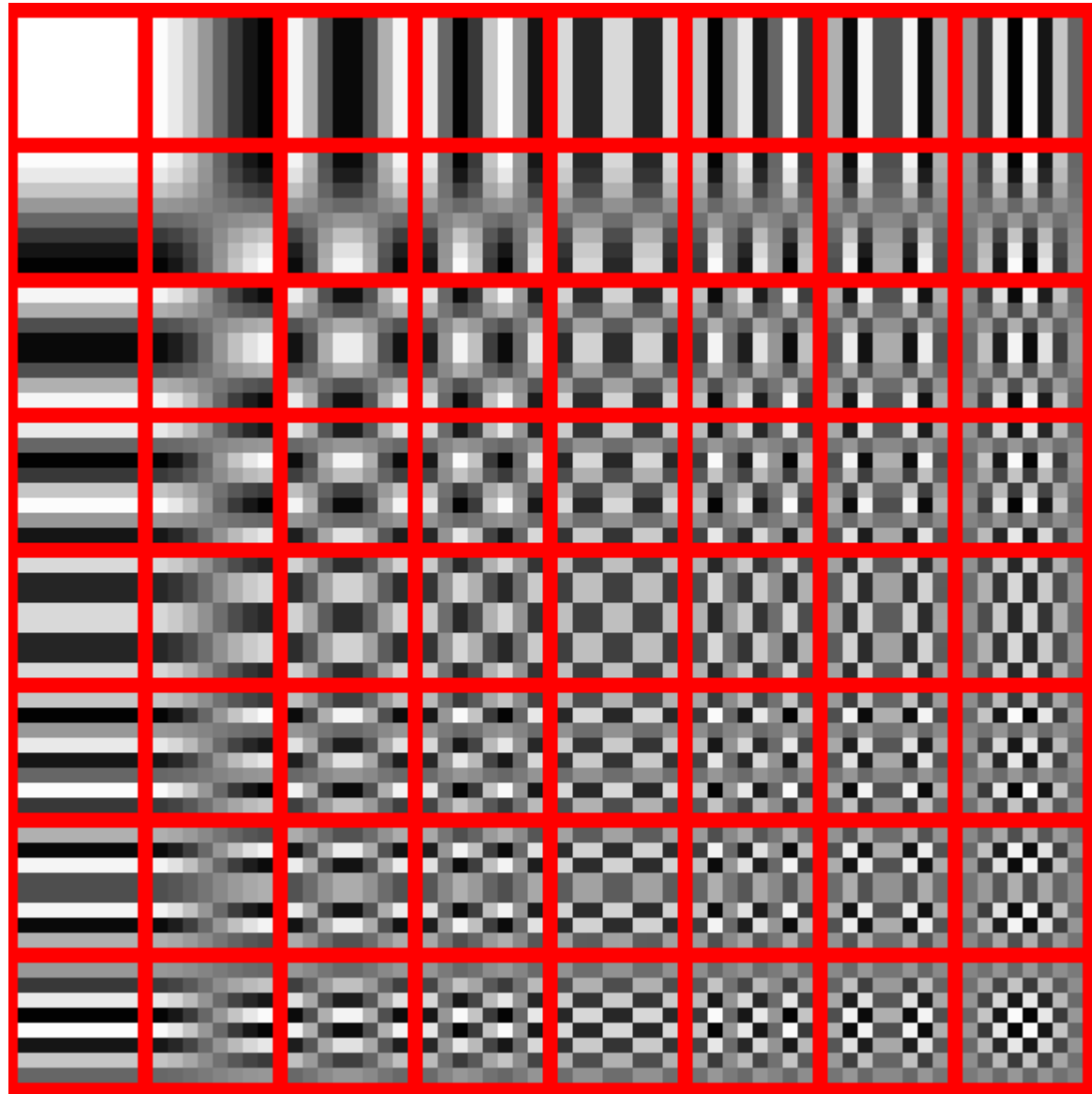


60 most important eigenvectors



Looks like the discrete cosine bases of JPG!...

2D Discrete Cosine Basis



http://en.wikipedia.org/wiki/Discrete_cosine_transform

Dimensionality reduction

- PCA (Principal Component Analysis):
 - Find projection that maximize the variance
- ICA (Independent Component Analysis):
 - Very similar to PCA except that it assumes non-Gaussian features
- Multidimensional Scaling:
 - Find projection that best preserves inter-point distances
- LDA (Linear Discriminant Analysis):
 - Maximizing the component axes for class-separation
- ...
- ...