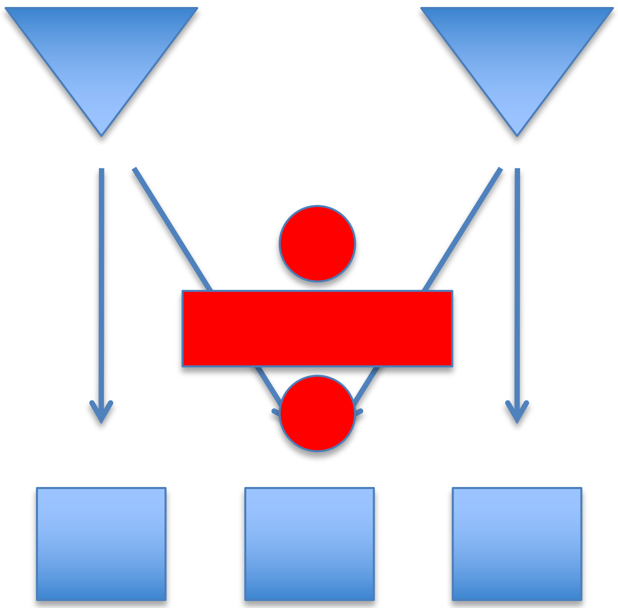
02458 Cognitive Modeling

Natural world statistics

A modest proposal

- The perceptual system is optimized to represent information relevant for survival
  - The dynamic range of neurons match the dynamic range of stimuli
  - Predators have binocular vision
  - Information is represented in an efficient way in the human brain

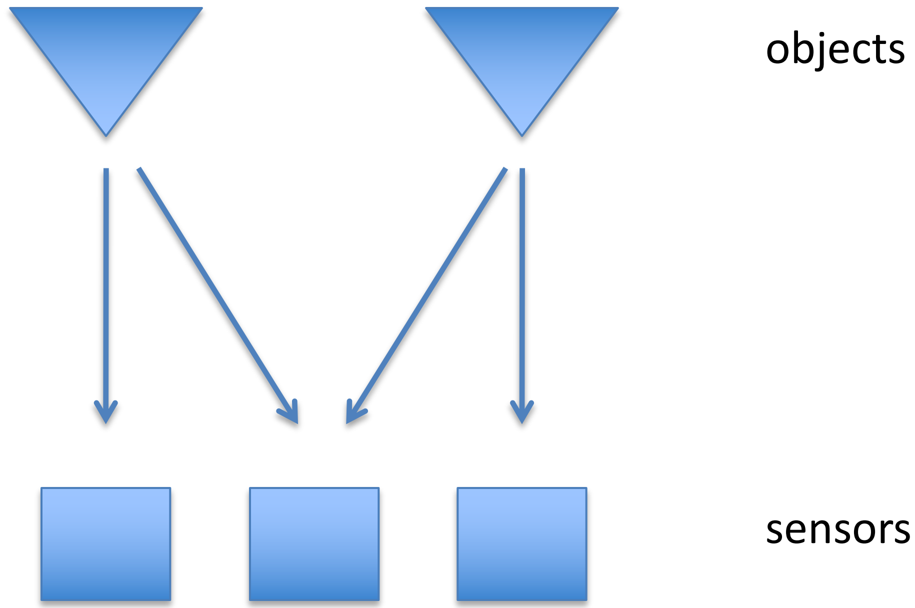
Efficient representations



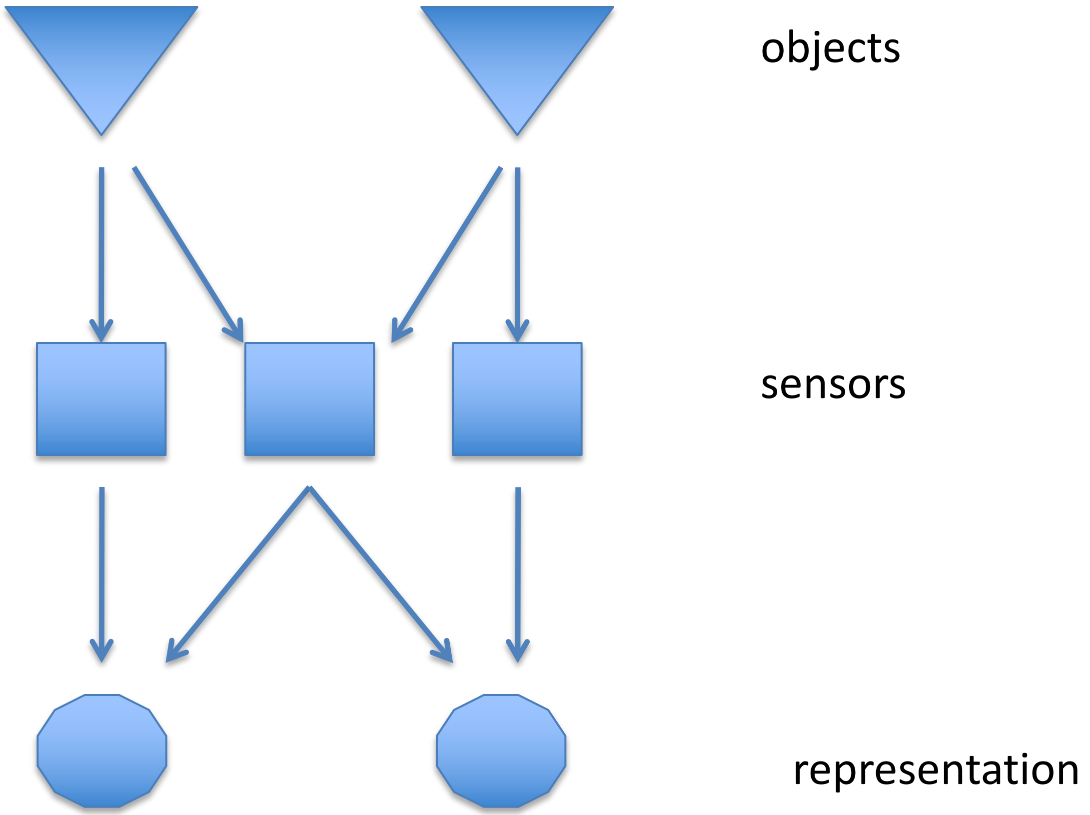
sensors

Efficient representation?

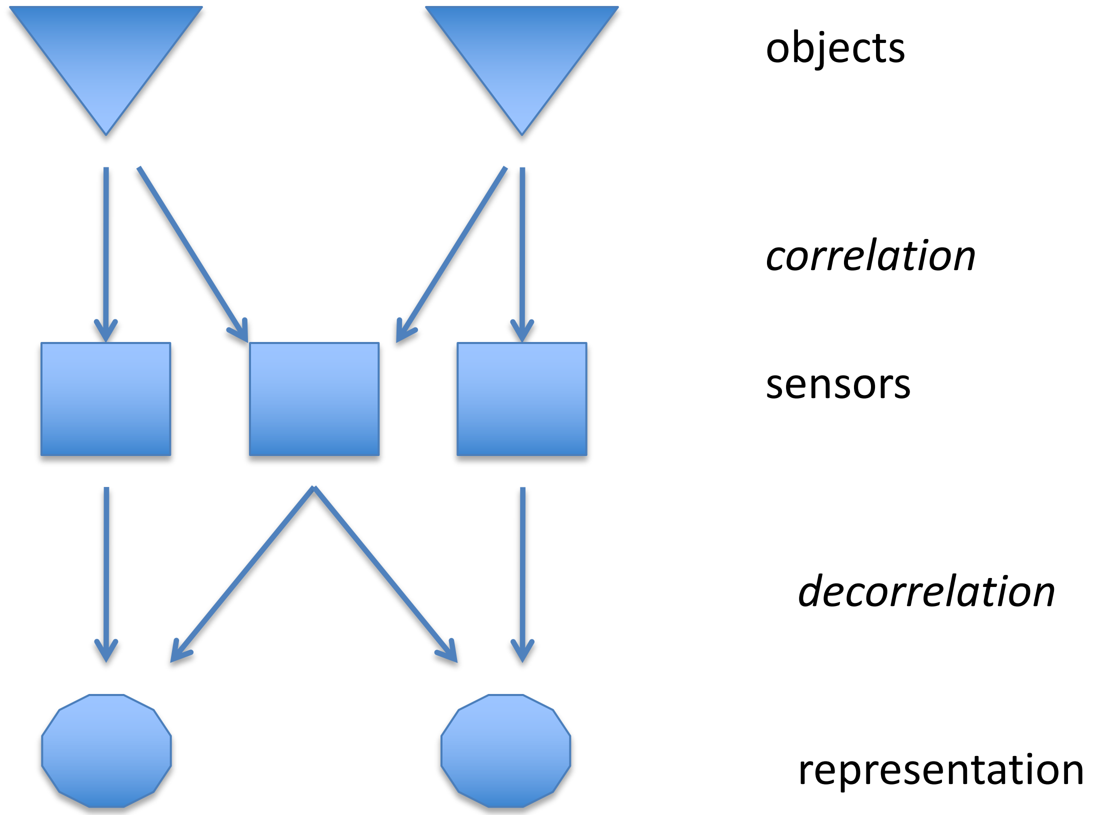
Efficient representations



Efficient representations



Efficient representations



Joint (pixel) probabilities

- Images are parameterized as vectors of pixels
  - For each color channel
  - For now, we just look at luminance (b/w)
- How do we parameterize *P(I)*?
  - As a joint probability $P(I_1,I_2,\ldots,I_N)$
  - Pixels are not independent $P(I_1,I_2)\sim=P(I_1)(I_2)$
  - How do we describe dependence?

Covariance

- Variance: $E\left(\left(x-\mu_x\right)^2\right) \cong E\left(\left(x-\bar{x}\right)^2\right)$
- Covariance: $E\left(\left(x-\mu_y\right)\left(y-\mu_y\right)\right) \cong E\left(\left(x-\bar{x}\right)\left(y-\bar{y}\right)\right)$
- For independent variables:

$$E\left(\left(x-\bar{x}\right)\left(y-\bar{y}\right)\right) = \sum_{i,j} P\left(x_i, y_j\right)\left(x_i-\bar{x}\right)\left(y_j-\bar{y}\right) =$$

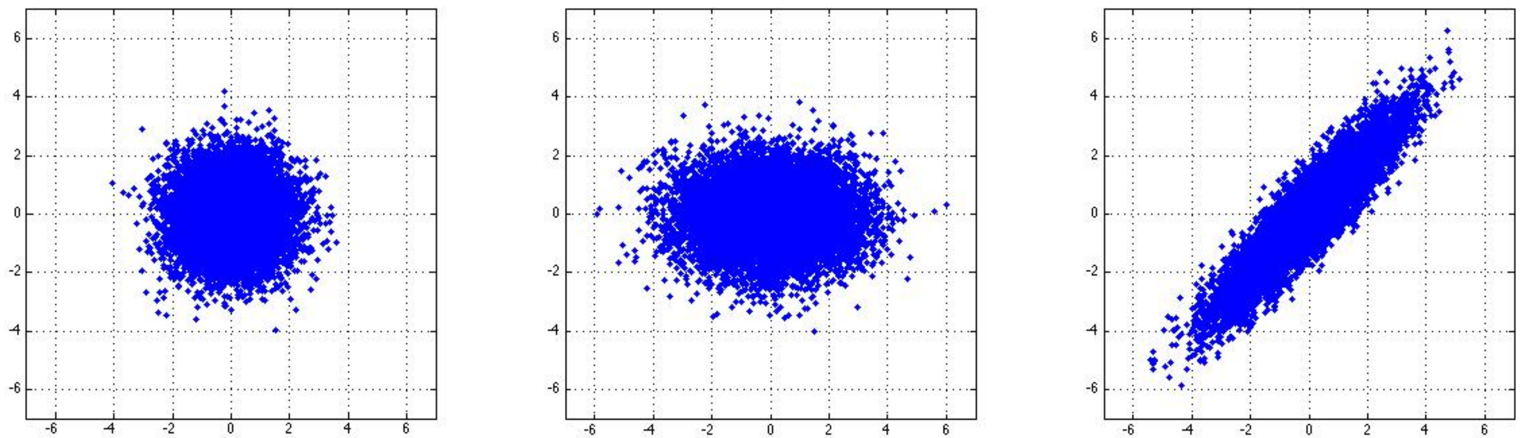
$$\sum_{i,j} P\left(x_i\right)P\left(y_j\right)\left(x_i-\bar{x}\right)\left(y_j-\bar{y}\right) = \sum_i P\left(x_i\right)\left(x_i-\bar{x}\right)\sum_j P\left(y_j\right)\left(y_j-\bar{y}\right) =$$

$$\left(\sum_i P\left(x_i\right)x_i - \bar{x}\sum_i P\left(x_i\right)\right)\left(\sum_i P\left(y_i\right)y_i - \bar{y}\sum_i P\left(y_i\right)\right) =$$

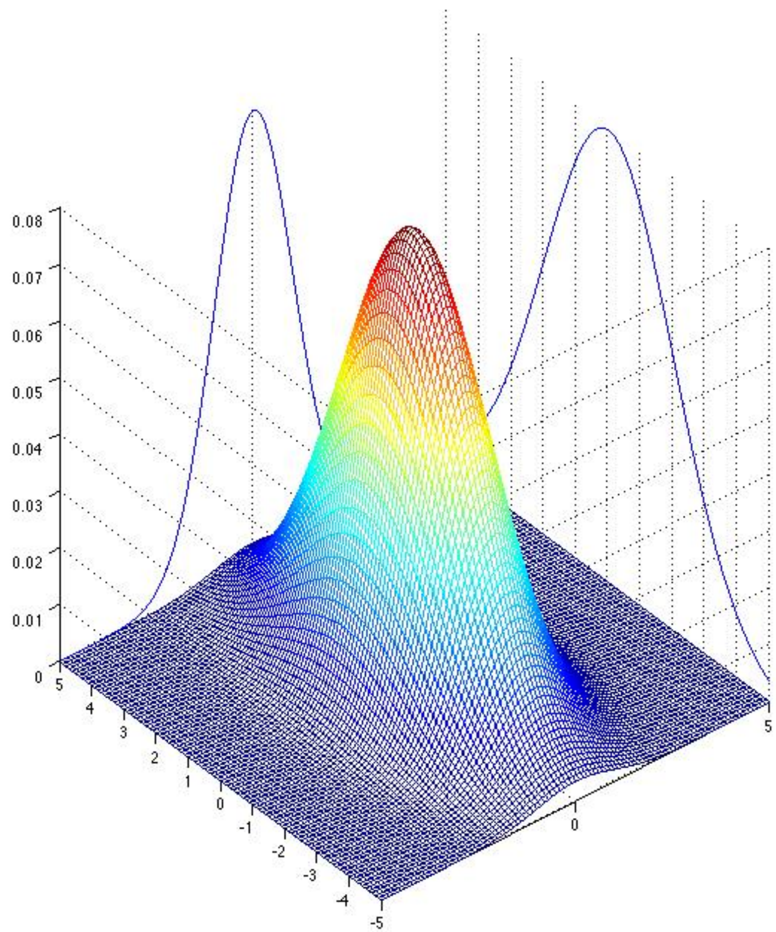
$$\left(\bar{x}-\bar{x}\right)\left(\bar{y}-\bar{y}\right) = 0$$

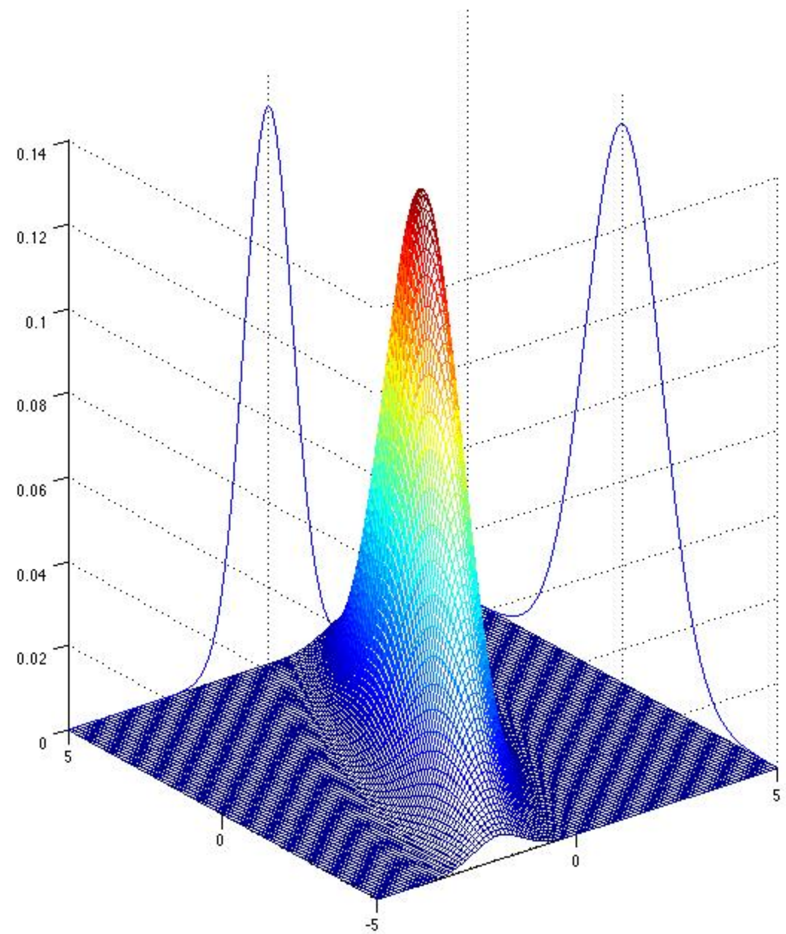
Covariance

- Which of the below data clouds have non-zero covariance?



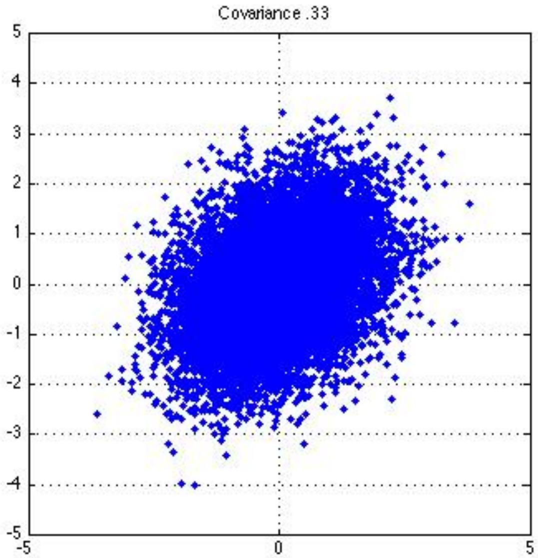
Covariance

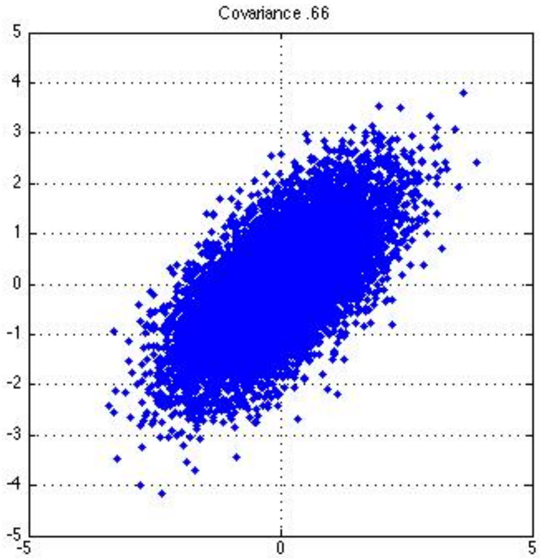

0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
5

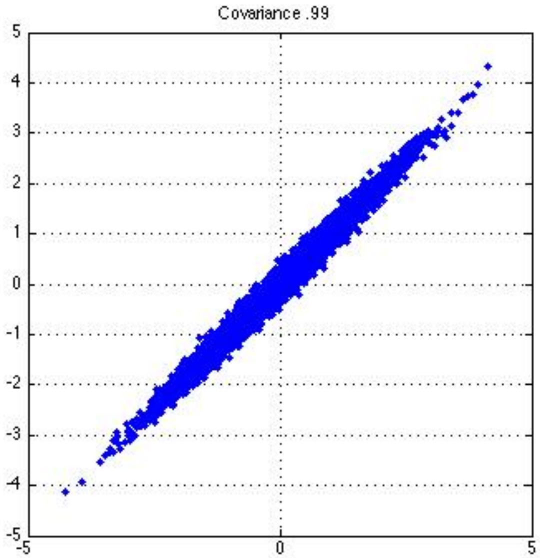

0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
5
0
-5
0
5

Covariance

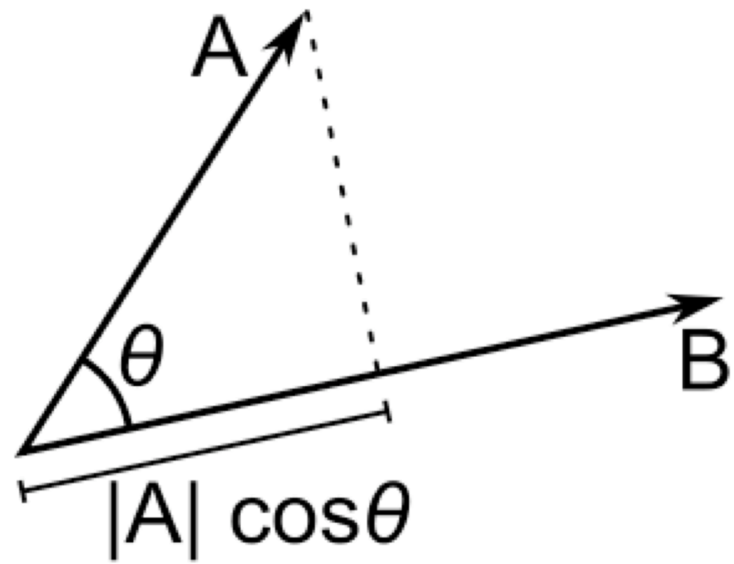
- When covariance -> 1
  - The two variables start becoming identical
  - The 2D representation is redundant
  - A 1D representation would be sufficient





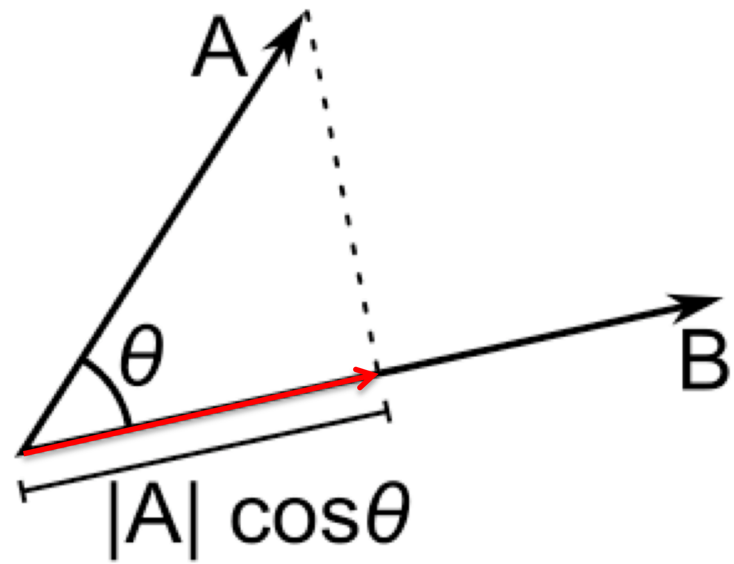


Vector projection



- 2D vector in a 1D basis
  - Projection or least squares approximation
  - If the basis vector B has length 1 then
    $A \cdot B = |A||B|\cos(\theta) \Rightarrow A \cdot B = A^T B = |A|\cos(\theta)$
  - And the approximation of A in B’s direction is $A \approx \left(A^T B\right) B$

Vector projection

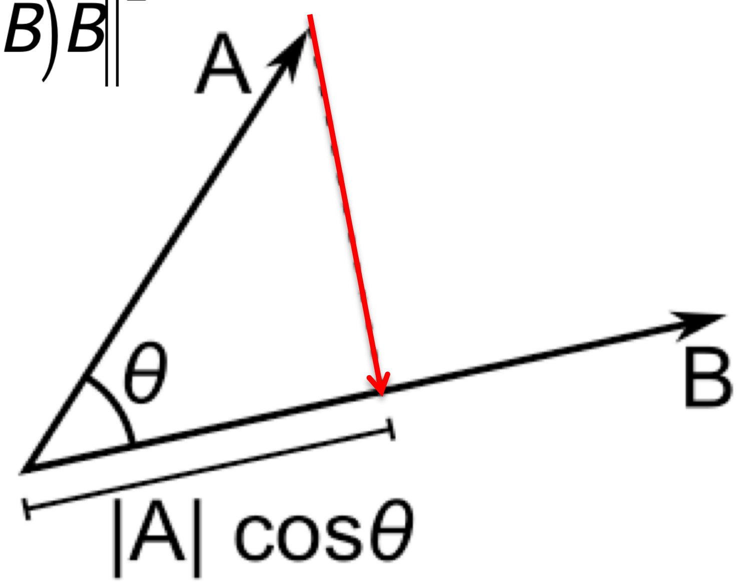


- 2D vector in a 1D basis
  - Projection or least squares approximation
  - If the basis vector B has length 1 then

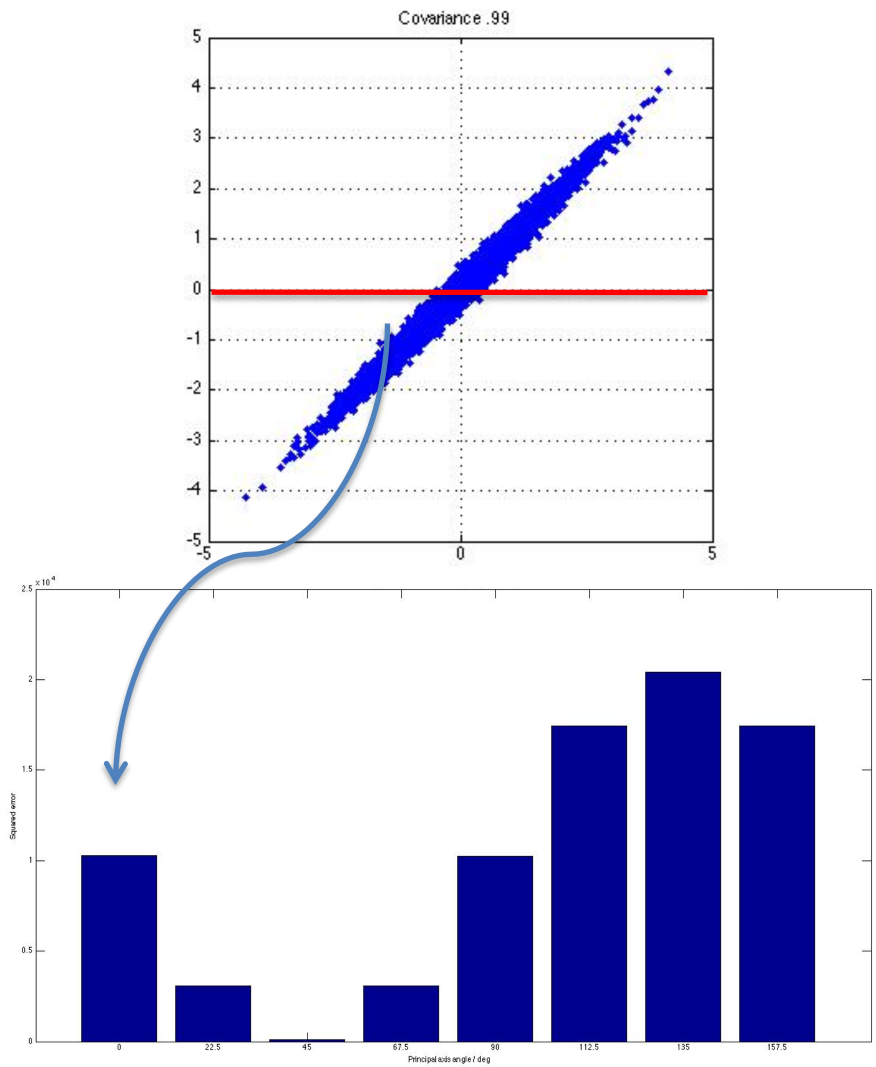
    $$A \cdot B = |A||B|\cos(\theta) \Rightarrow A \cdot B = A^T B = |A|\cos(\theta)$$
  - And the approximation of A in B's direction is $A \approx (A^T B)B$

Vector projection

- How good is the approximation?
  - Error is $A-\left(A^T B\right)B$
  - Squared error is $\left\|A-\left(A^T B\right)B\right\|^2$



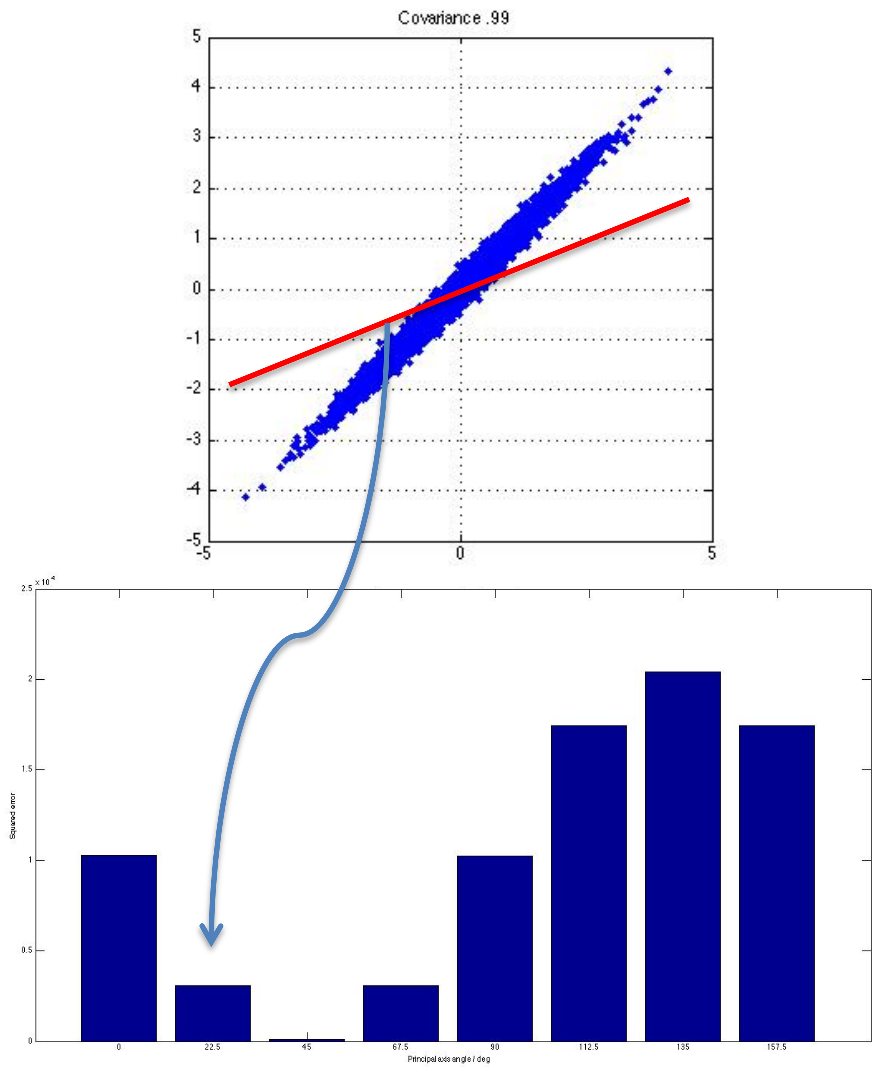
Minimizing the error



Squared Error

Degrees 0 22 45 67 90 112 135 157

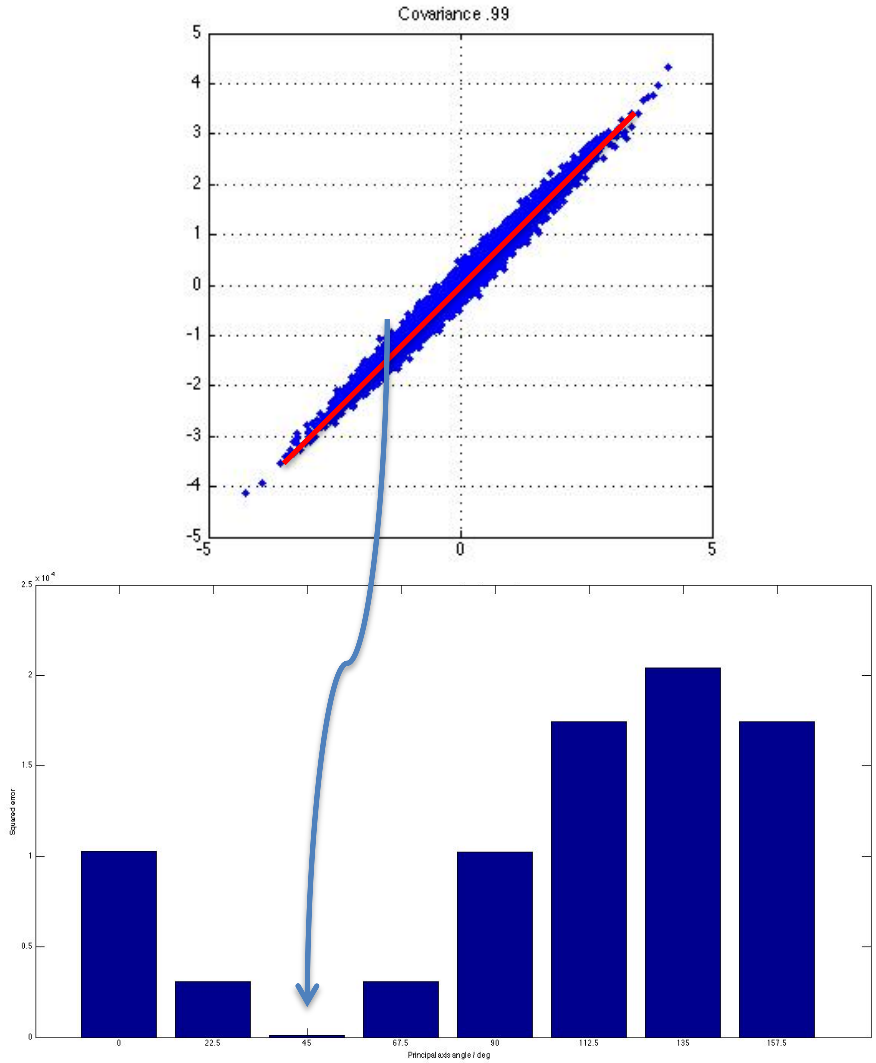
Minimizing the error



Squared Error

Degrees 0 22 45 67 90 112 135 157

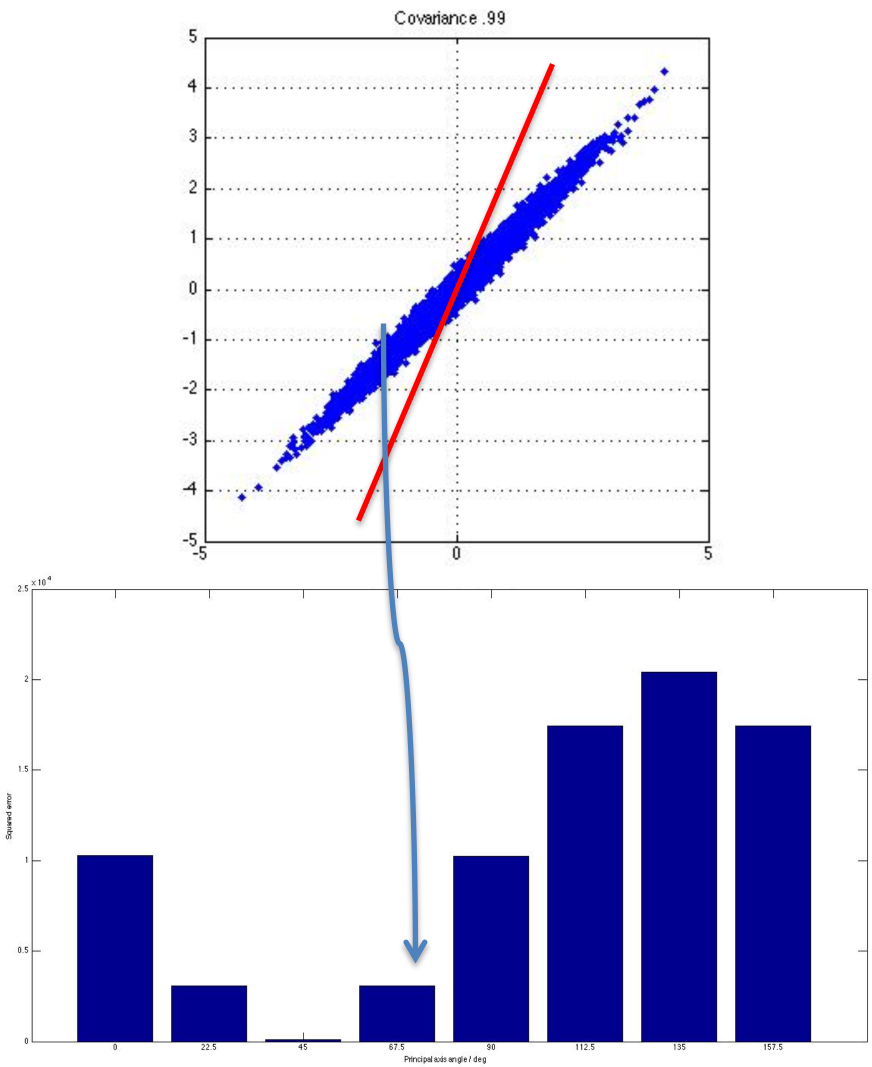
Minimizing the error



Squared Error

Degrees 0 22 45 67 90 112 135 157

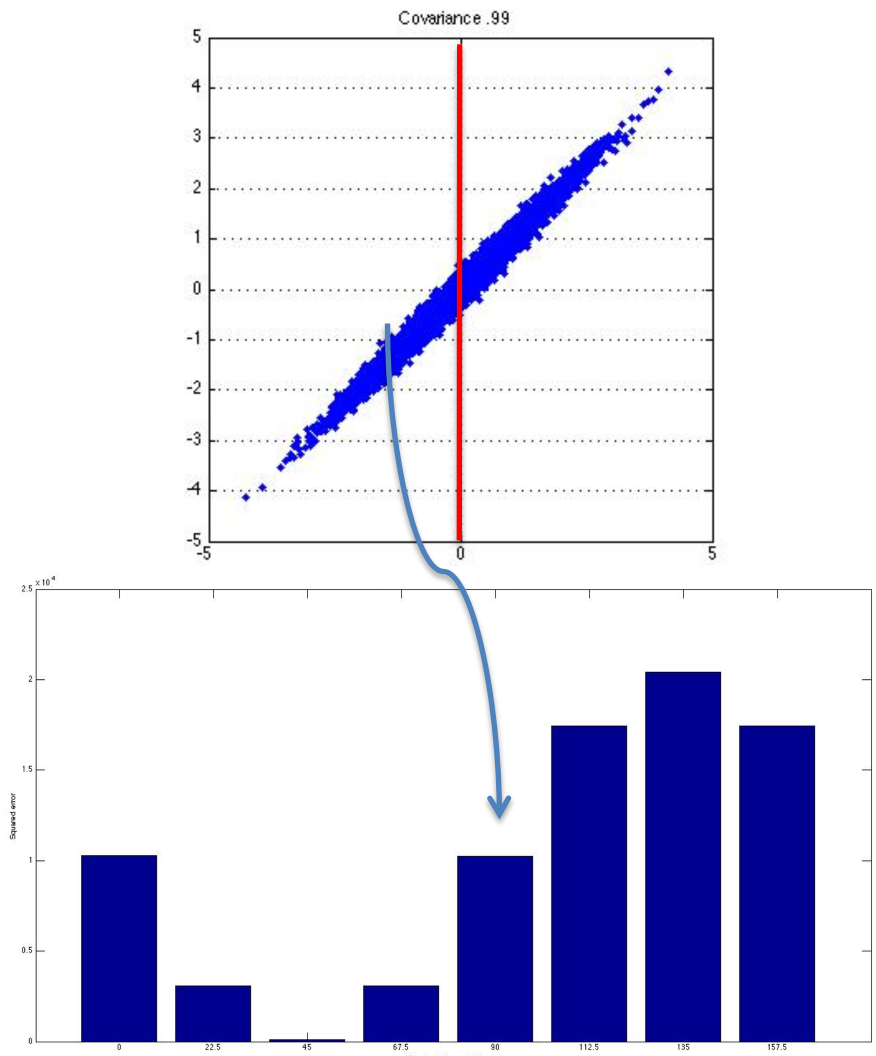
Minimizing the error



Squared Error

Degrees 0 22 45 67 90 112 135 157

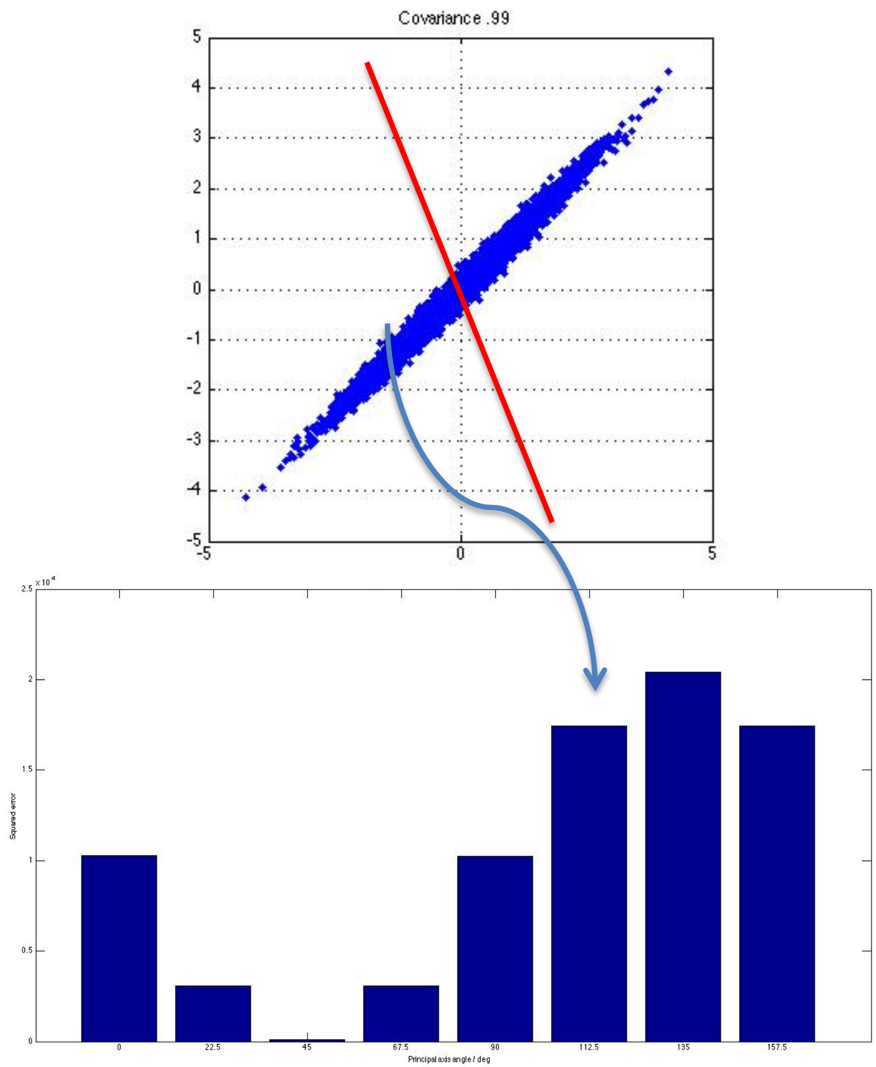
Minimizing the error



Squared Error

Degrees 0 22 45 67 90 112 135 157

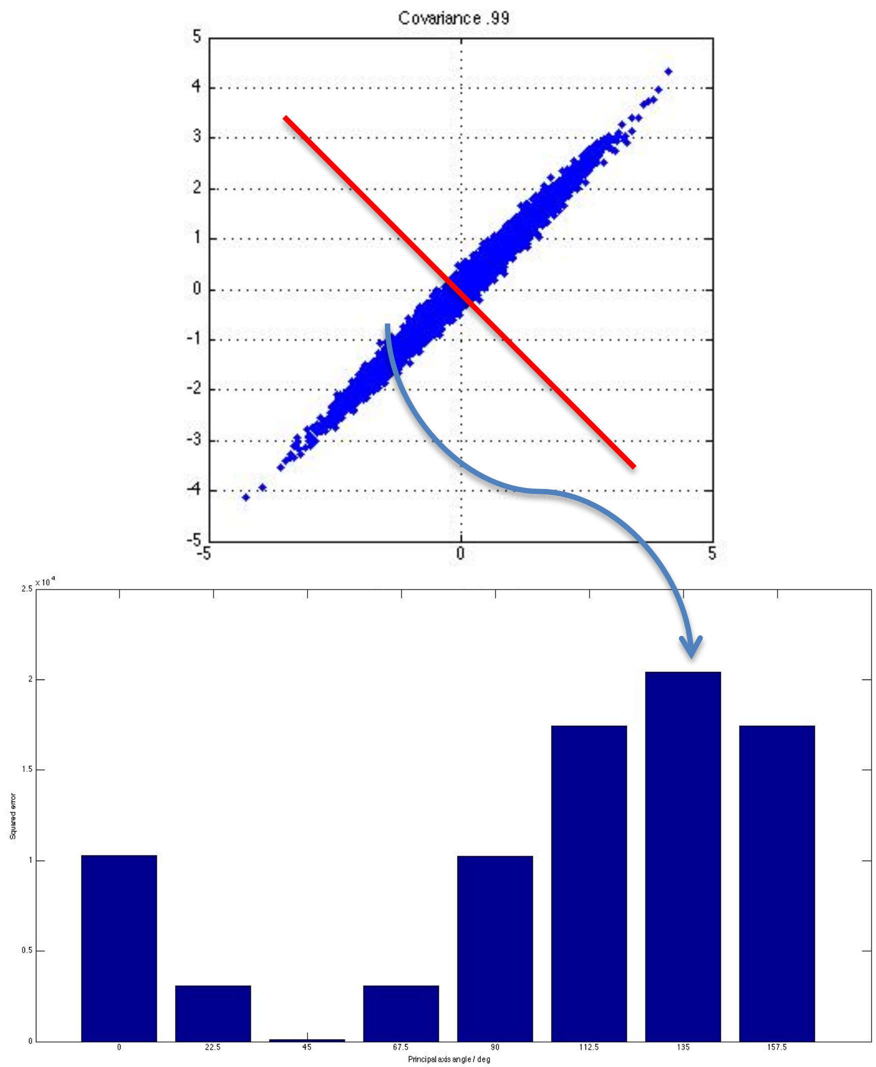
Minimizing the error



Squared Error

Degrees 0 22 45 67 90 112 135 157

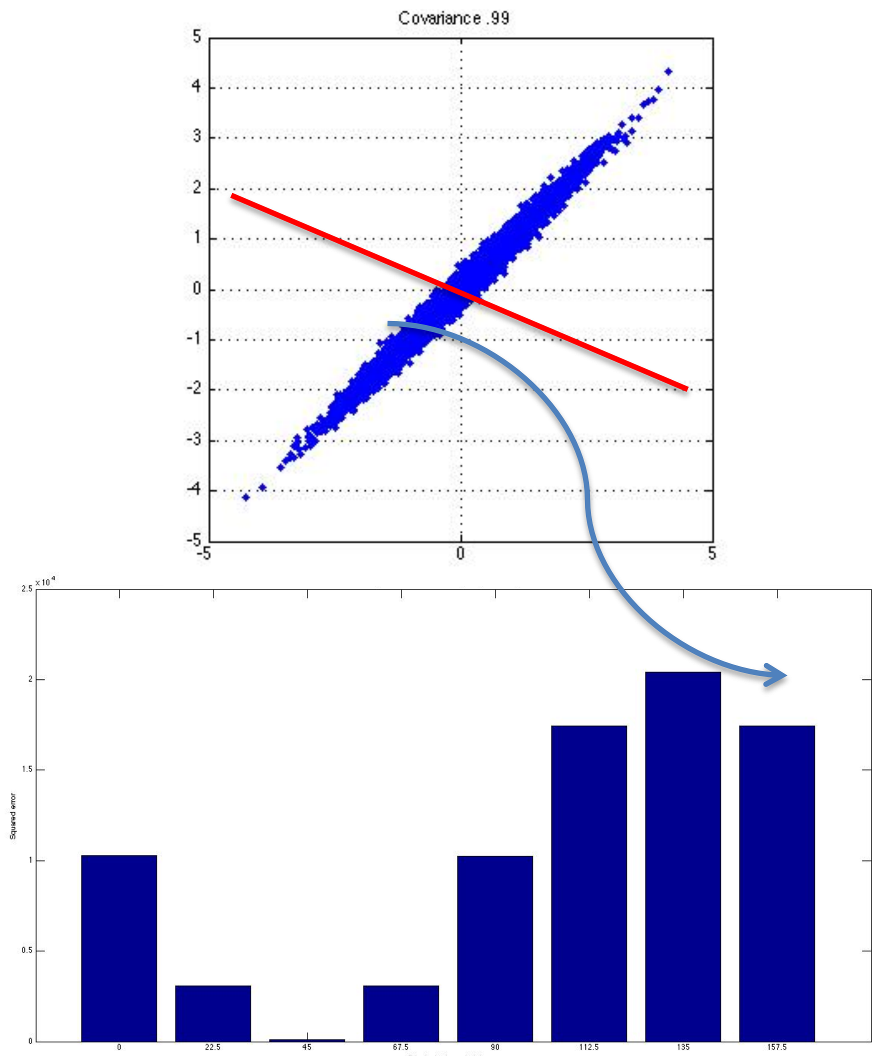
Minimizing the error



Squared Error

Degrees 0 22 45 67 90 112 135 157

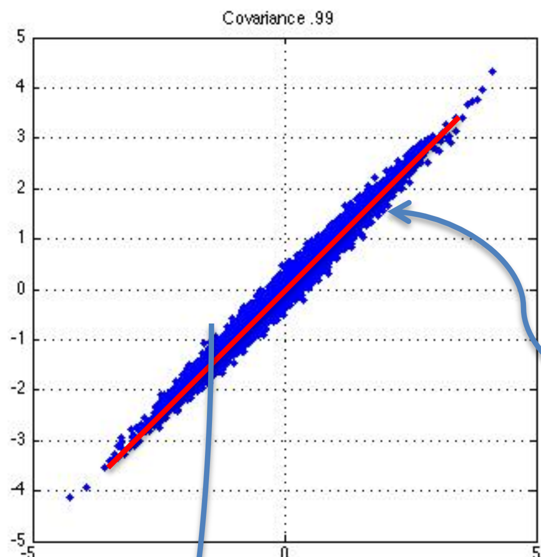
Minimizing the error



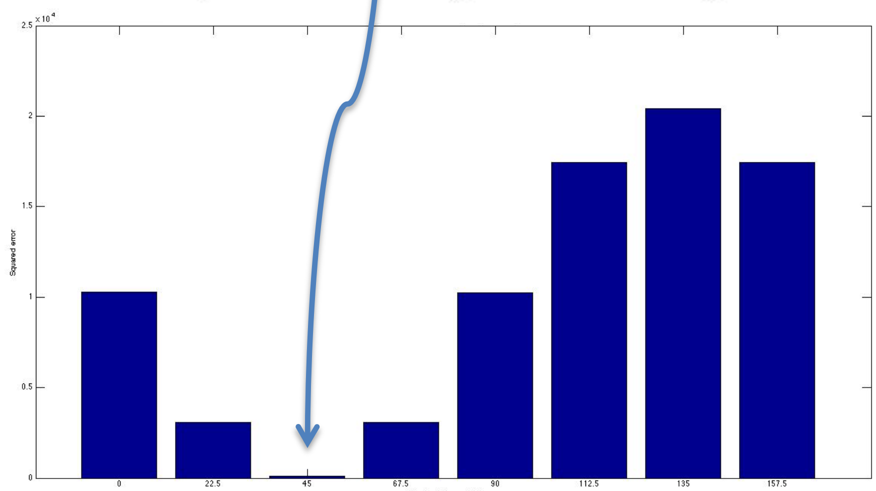
Squared Error

Degrees 0 22 45 67 90 112 135 157

Minimizing the error



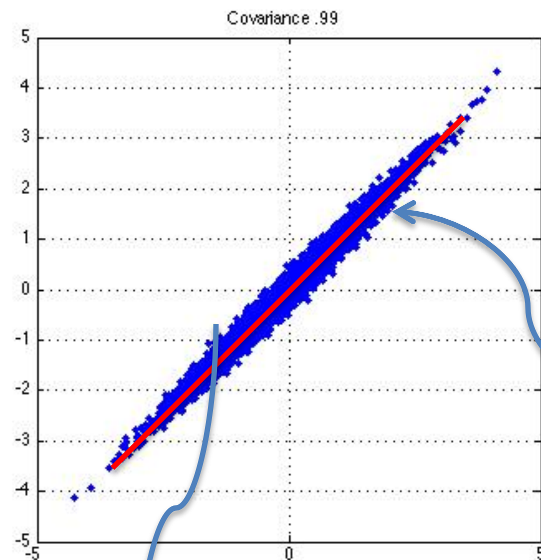
First principal component



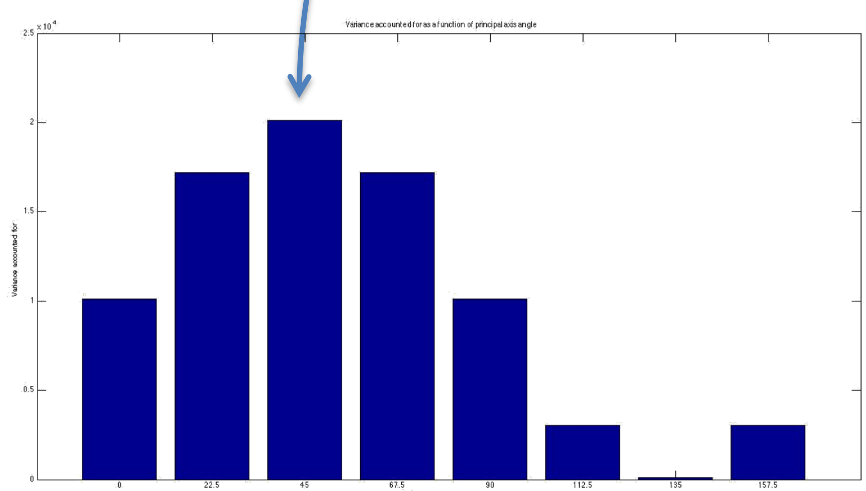
Squared Error

Degrees 0 22 45 67 90 112 135 157

Maximizing the variance *along* the principal component



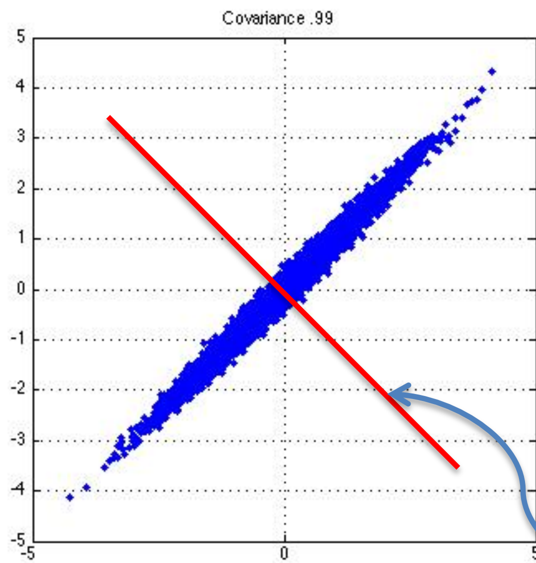
First principal component



Variance along PC1

Degrees 0 22 45 67 90 112 135 157

A principal component for the “error”



Second principal component

- Finding principal components is iterative
  - First find one, then the next
  - Trivial for 2D but not for many dimensions

Principal component analysis (PCA)

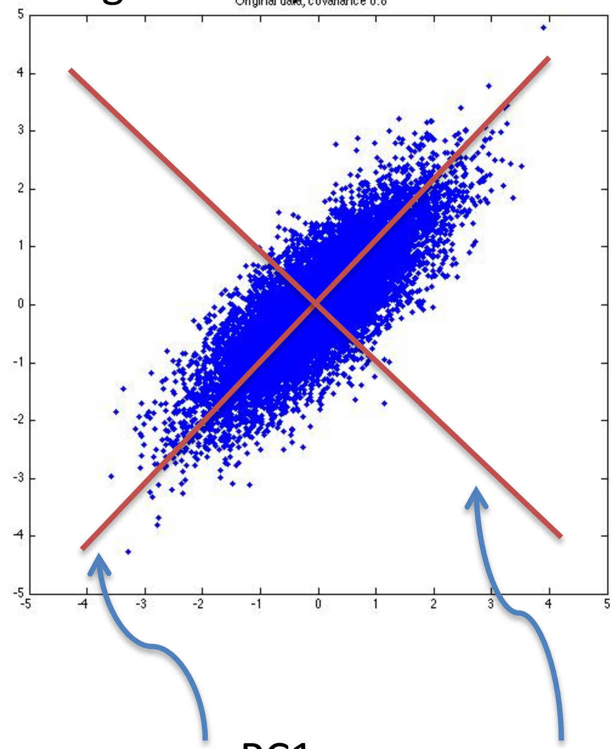
- Find the principal components
  - N N-dimensional vectors (N-by-N matrix)
    - N is the number of measurement dimensions
  - The PC’s are called the coefficients in Matlab
  - PC’s are the eigenvectors of the covariance matrix
  - PC’s are orthogonal – forms a new rotated basis
- Project the original data-set onto the PCs
  - This gives you the “score”

Principal component analysis (PCA)

- The PC's are ordered according to their variance
  - Gives an indication of their "importance"
  - The eigenvalues of the covariance matrix

Principal component analysis (PCA)

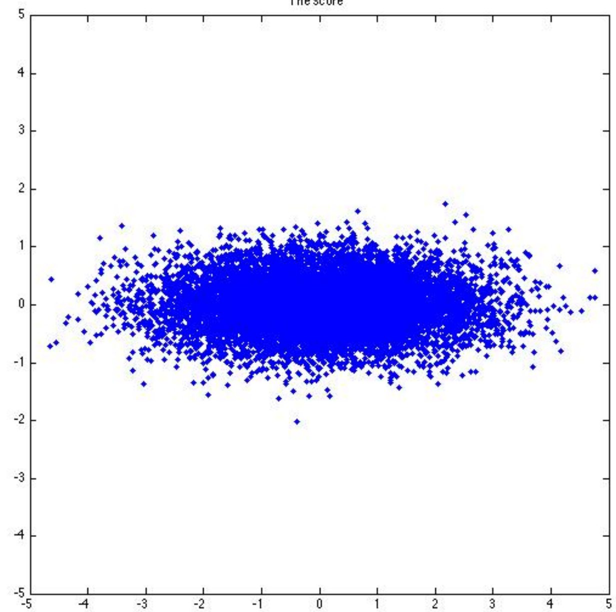
Original data, covariance 0.8



PC1

PC2

The score



Notice greatest variance along 1st dimension

Image PCA

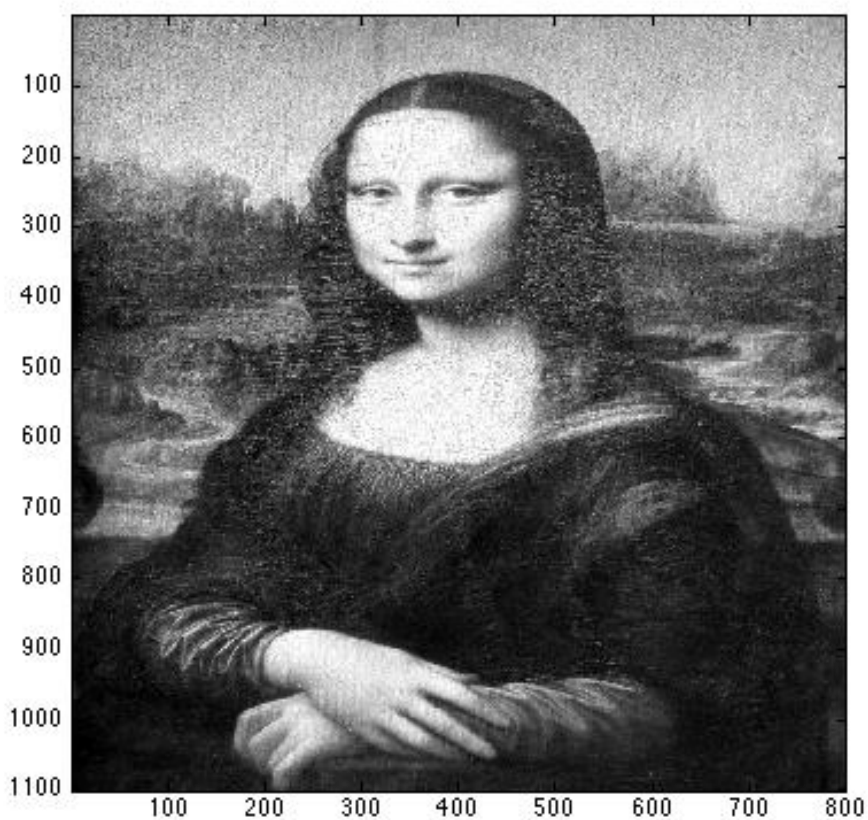
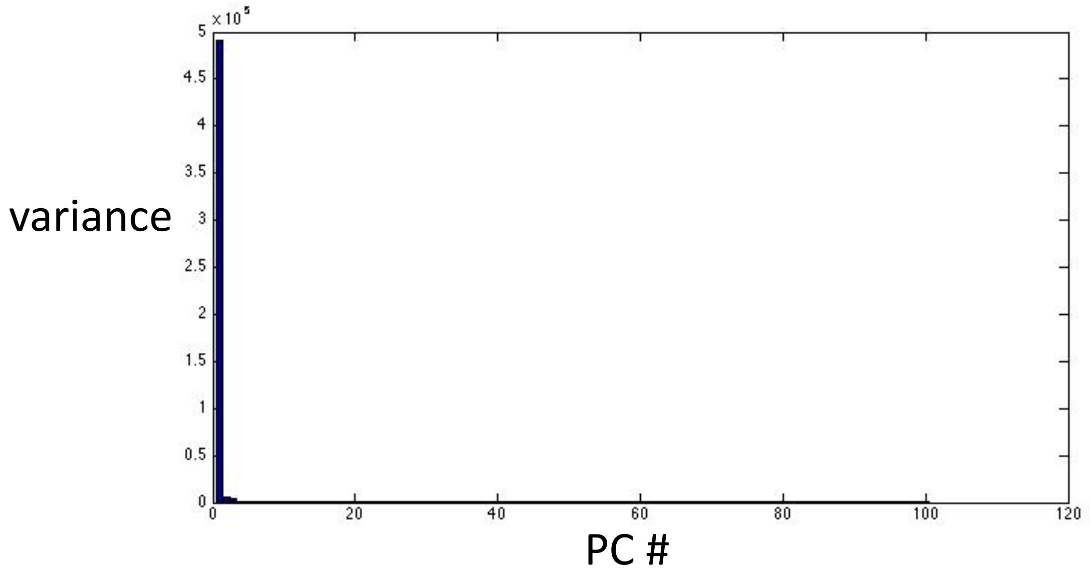

100
200
300
400
500
600
700
800
900
1000
1100
100
200
300
400
500
600
700
800

Image PCA

- Mona Lisa is just a bunch of samples from image patch space
- Given an 1100x800 pixel image
  - 110X80 = 8800 10-by-10 patches
  - A 10-by-10 patch = 100-dimensional vector
- Enough for PCA
- How dimensions are relevant / informative?
- How many dimensions are redundant?

Image PCA

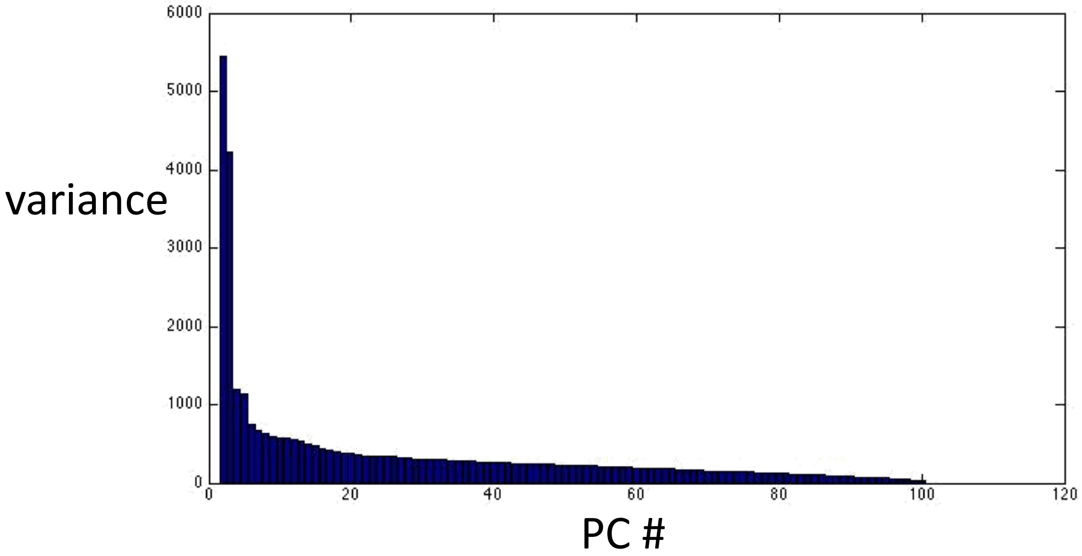
- Sorted eigenvalues



- Shows that PC1 dominates!
  - Let’s take it out to have a closer look at the others…

Image PCA

- Sorted eigenvalues 2-100



- PC2 and PC3 also carries a lot of variance
- PC4-6 carries some variance
- But then it’s just downhill

Image PCA

- What are these PCs?
- Let’s have a look!

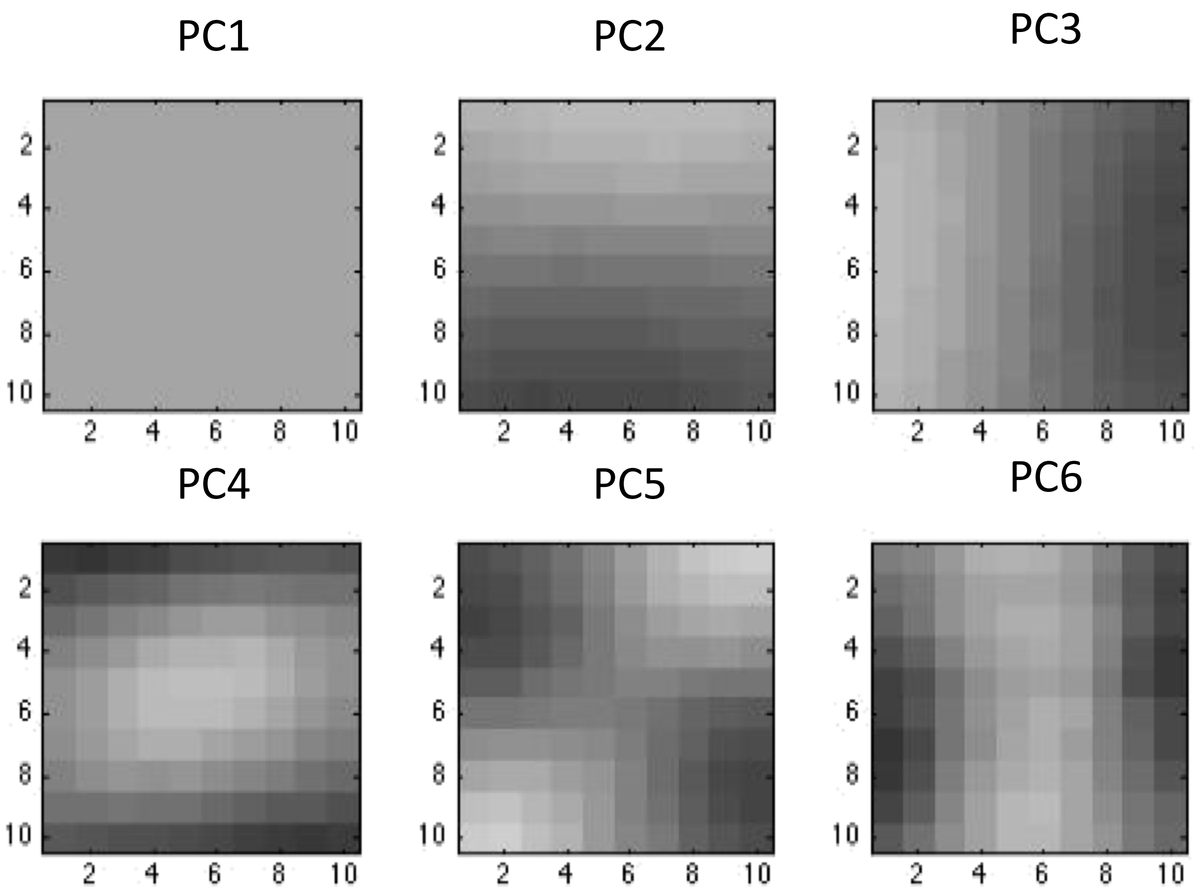


Image PCA

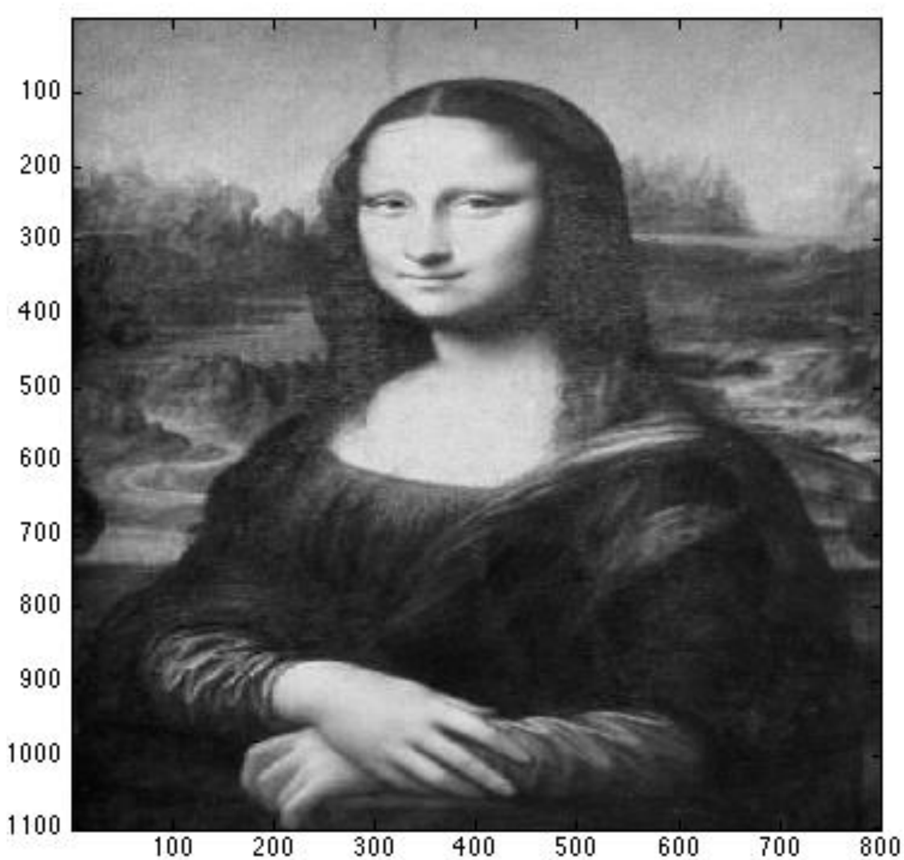
- The principal components look like edge filters
  - Similar to simple cell receptive fields
  - Similar to a local Fourier analysis
    - But data-driven!

Image PCA

- What about the rest (94) PC’s
- Do we need them?
- Let’s reconstruct the image from the first 6 PCs

Image PCA

Who's who?



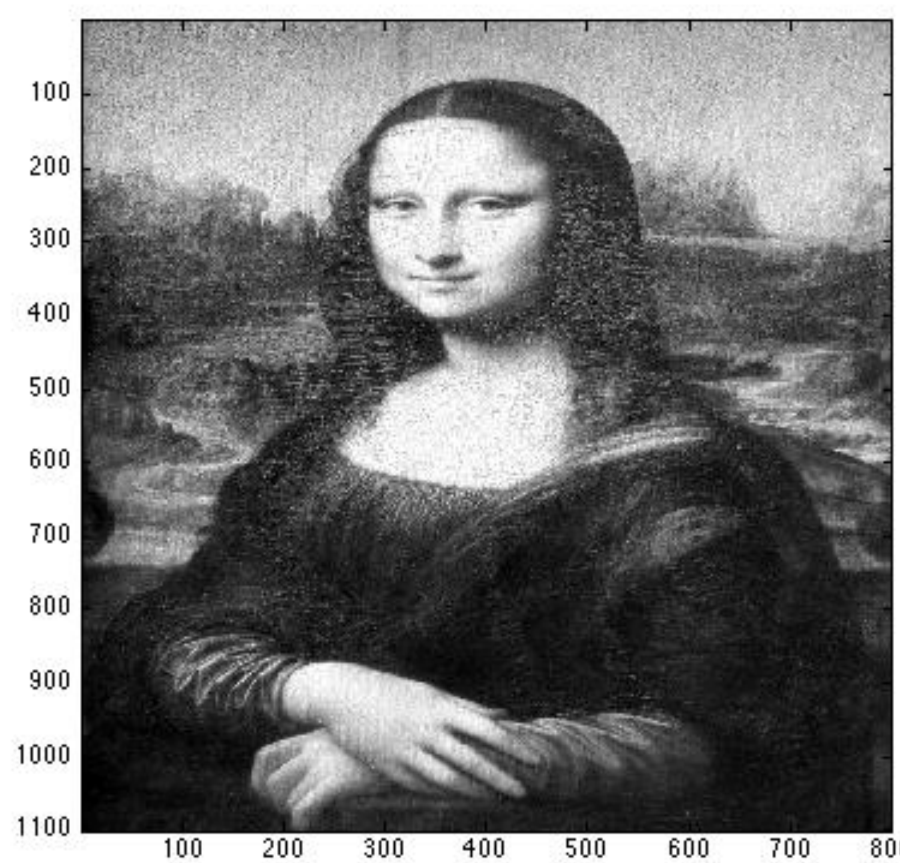
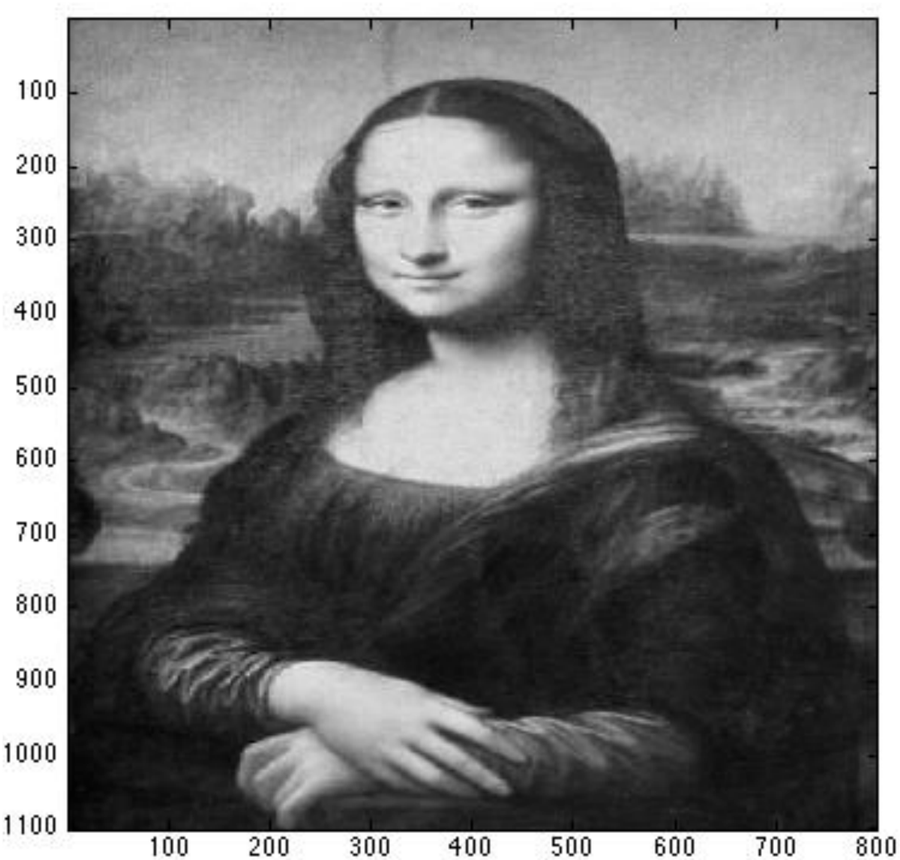
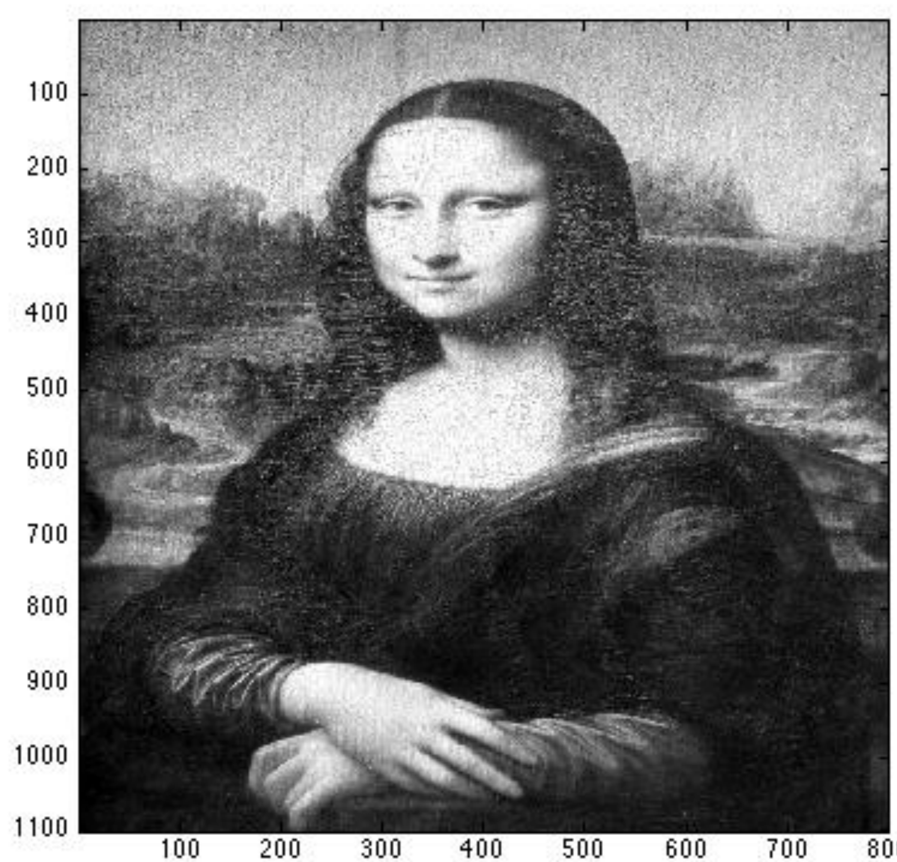


Image PCA

Who's who?



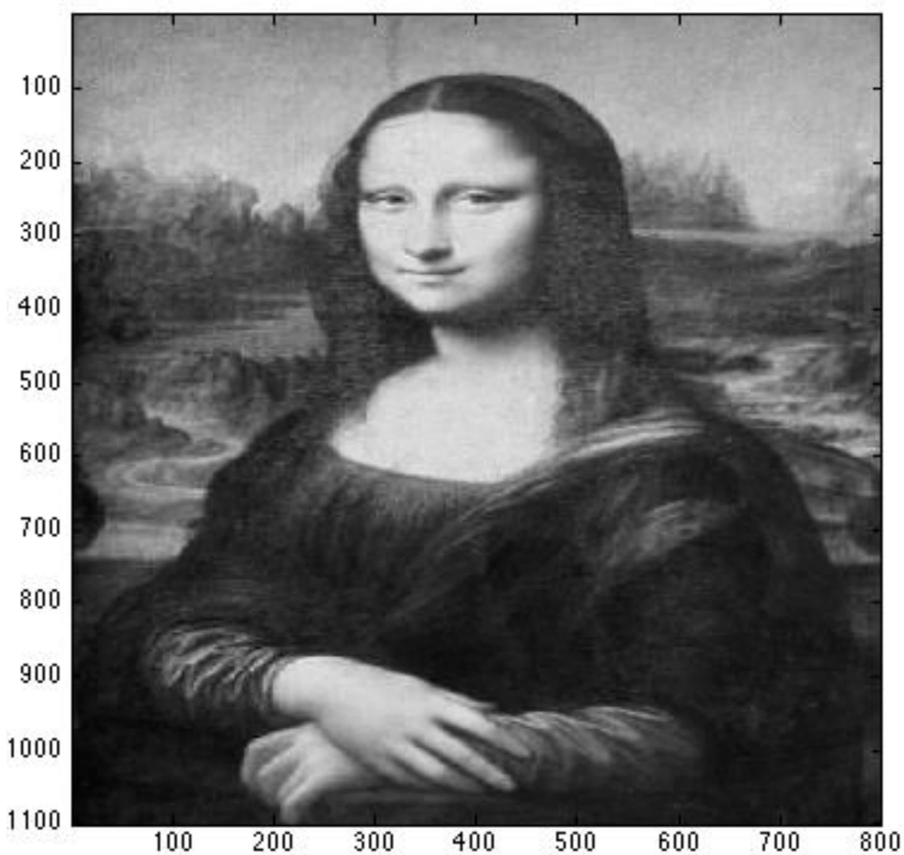
94 of 100 dimensions missing



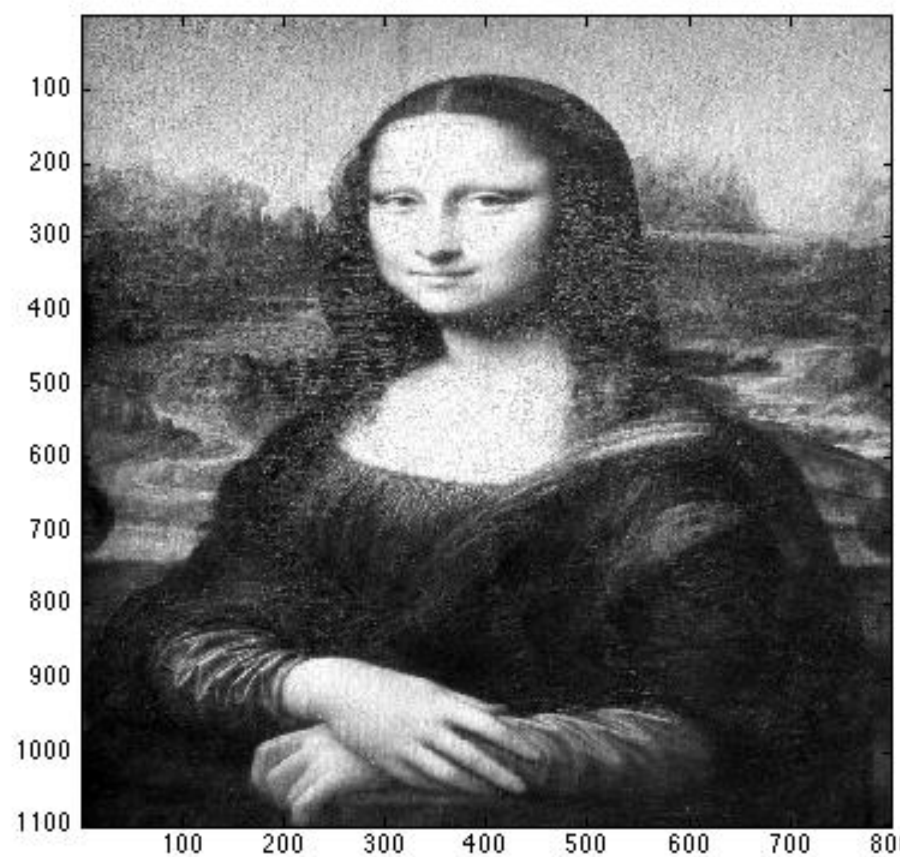
The real McCoy!

Image PCA

Who's prettier?



94 of 100 dimensions missing



The real McCoy!

Image PCA

- PCA
  - Provides a prioritized list of “sensors”
  - Denoises
  - Compress images
  - Provides a superior representation of images
    - Compared to a pixel-by-pixel representation
  - A candidate for image components

Now to the exercises..