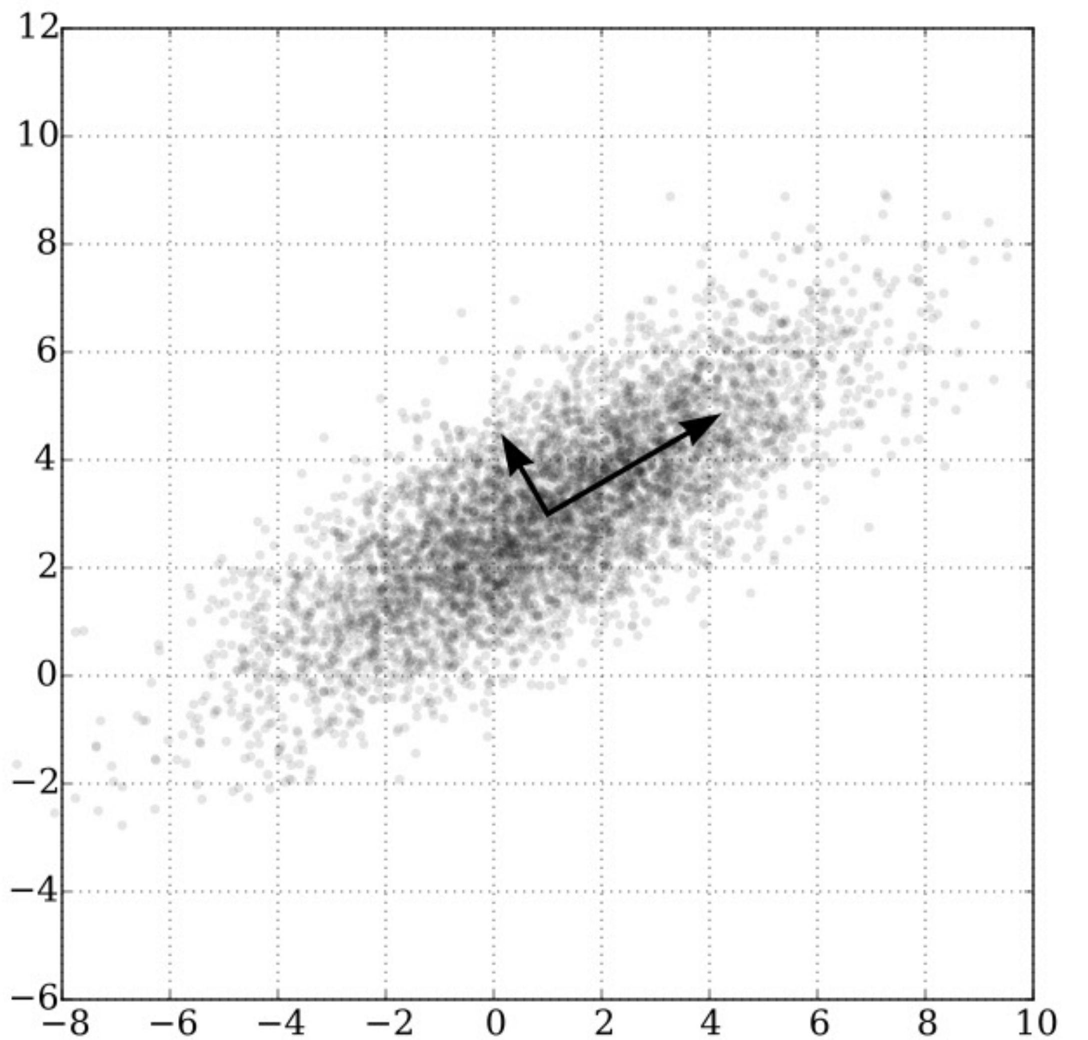
Tutorial 4: PCA

Principal Component Analysis



TA: Zador Pataki patakiz@ethz.ch

https://github.com/Zador-Pataki/viscomp2024

Lossy vs Lossless compression

- Data $x \in A$ e.g. an image
- Compressor $f: A \rightarrow B$
- $f(x)$ = compressed version such that $\mathrm{size}\big(f(x)\big) < \mathrm{size}(x)$
- $f^{-1}(f(x))$ = decoding of the compressed x
- Reconstruction error $\mathrm{E} = \|x - f^{-1}(f(x))\|$

Lossless	Lossy **PCA**
if $E = 0$	if $E \neq 0$
	But good if E small enough

Principal Component Analysis - PCA

- **Non-parametric** method of extracting relevant information from data
- **Orthogonal linear projection** of high dimensional data onto low dimensional subspace

$$f(x) = Ux$$

Properties:

1. **Reconstruct the data well**: minimize error E
2. **Maximize information**:
   maximize the total variance of the encoding $f(x)$

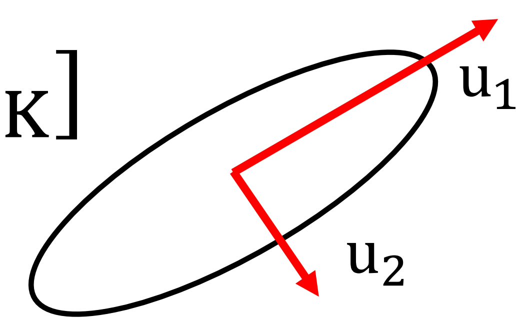
How to calculate U?

- We have a collection of N data samples $x_i \in \mathbb{R}^D$
- We can fit a normal distribution $\mathcal{N}(\mu, \Sigma)$ to the data:

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

covariance
$$\Sigma = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T$$

- U = **eigenmatrix** of Σ, such that $\Sigma = \mathrm{U}\Lambda U^T$ (orthogonal!)
- For PCA: U_K = first K eigenvectors of $\Sigma = [\mathrm{u}_1 \ldots \mathrm{u}_\mathrm{K}]$
- Then $\mathrm{PCA}(\mathrm{x_i}) = \mathrm{f}(\mathrm{x_i}) = {U_K}^T(\mathrm{x_i} - \mu)$



- **K principal components** = directions with largest variance
- Large compression if $\mathrm{K} \ll D$

Using SVD vs Eigendecomposition

- Let X be the D x N data matrix: $X = [\mathrm{x}_1 \ldots \mathrm{x}_\mathrm{N}]$
- Let $\bar{X}$ be the centered data matrix $\bar{X} = \mathrm{X} - \mu$
- Apply SVD: $\bar{X} = USV^T$ where $U^T U = I_D$ and $V^T V = I_N$
- Thus $\Sigma = \bar{X}\bar{X}^T = USV^T VSU^T = US^2U^T$
- Thus we can compute PCA
  with either SVD or Eigendecomposition

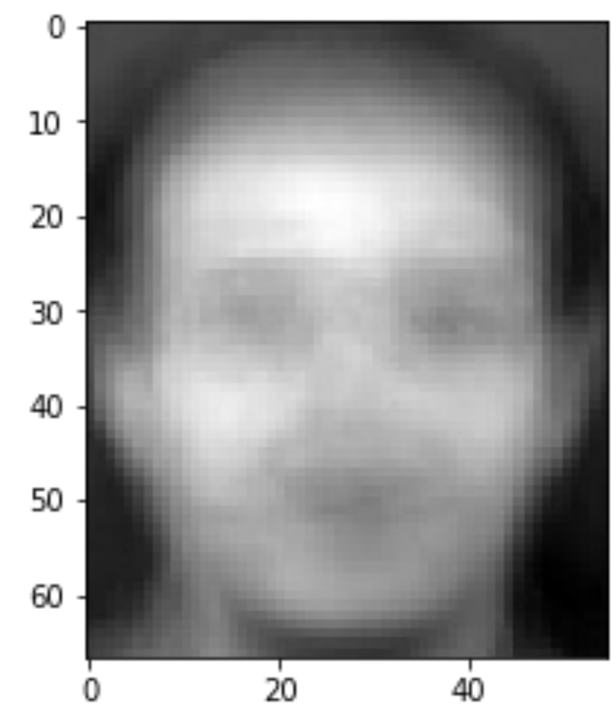
PCA on faces

- Now x_i = images of human faces
- $x_i \in \mathbb{R}^D$ with D = number of pixels = width x height

- AT&T face database: 40 people, 10 expressions each

PCA on faces

Compute:



- Mean $\mu \in \mathbb{R}^D$

- Covariance $\Sigma \in \mathbb{R}^{D \times D}$

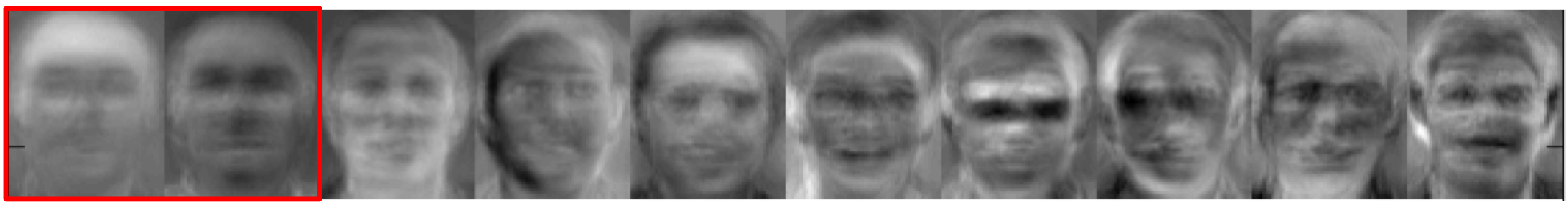
Cannot visualize

PCA on faces

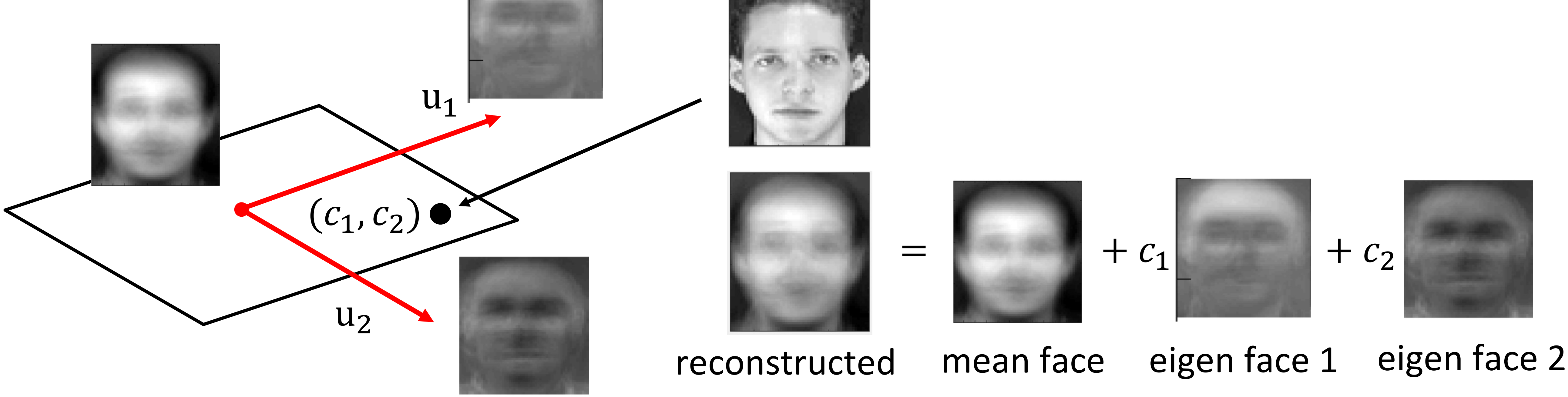
- First 10 eigenvectors $u_{1:K}$ ordered by decreasing eigenvalues

PCA on faces

- First 10 eigenvectors $u_{1:K}$ ordered by decreasing eigenvalues



- Take $K = 2$



PCA compression

- We have $D = 68 \times 56 = 3808$
- If $K = 100$ then each face is represented by only 100 values
- That's a 38x compression!

original reconstructed

PCA compression

$$K = 50$$

$$K = 200$$

Application to Face Detection

George (s38)

- Can you compute the x coordinate of George’s head?

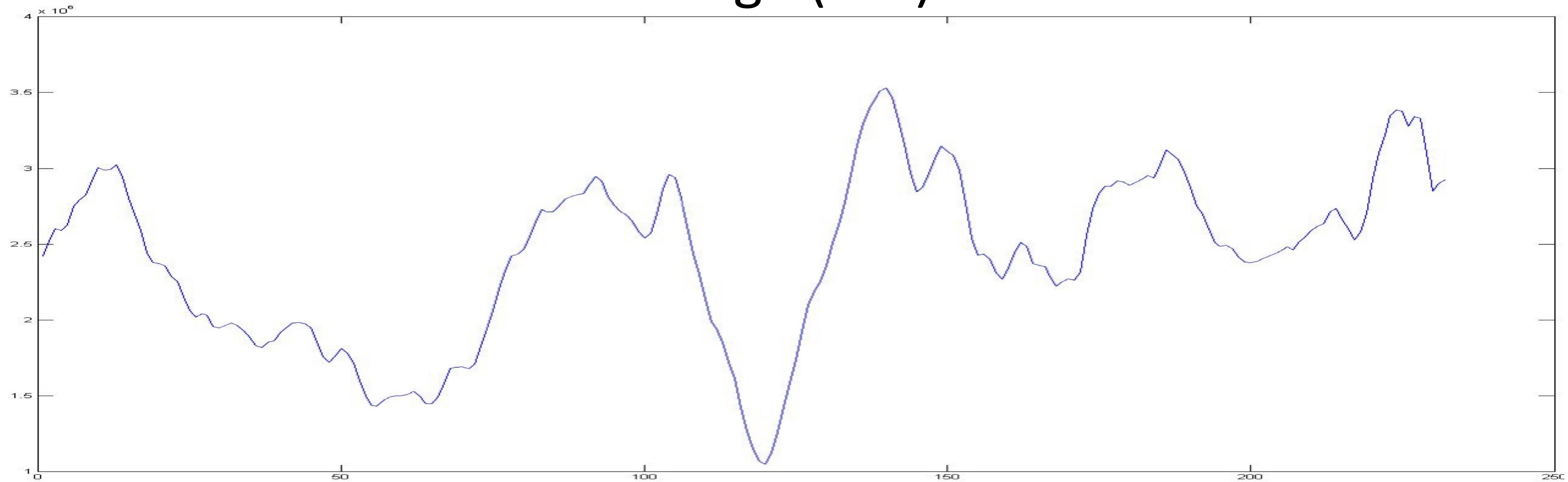
Application to Face Detection

George (s38)

- Steps:
  - Compute eigenfaces (use only the first 20 people)
  - Compress each patch of the image using a sliding window
  - Evaluate the compression error using SSD
  - The patch with the lowest error is George!

Application to Face Detection

George (s38)



Exercise session

- https://github.com/Zador-Pataki/viscomp2024
- Execise: https://colab.research.google.com/drive/1c29AJ6_ZHbFCsjqLeJaeKzMTAi0ADgUc?usp=sharing