Face Recognition

CSE 576

Face recognition: once you've detected and cropped a face, try to recognize it

Face recognition: overview

- Typical scenario: few examples per face, identify or verify test example
- What's hard: changes in expression, lighting, age, occlusion, **viewpoint**
- Basic approaches (all nearest neighbor)
  1. Project into a new subspace
  2. Measure face features

Typical face recognition scenarios

- Verification: a person is claiming a particular identity; verify whether that is true
  - E.g., security
- Closed-world identification: assign a face to one person from among a known set
- General identification: assign a face to a known person or to “unknown”

What makes face recognition hard?

Expression

What makes face recognition hard?

Lighting

What makes face recognition hard?

Occlusion

What makes face recognition hard?

Viewpoint

Simple idea for face recognition

1. Treat face image as a vector of intensities

$\mathbf{x}$

2. Recognize face by nearest neighbor in database

$\mathbf{y}_1 \dots \mathbf{y}_n$

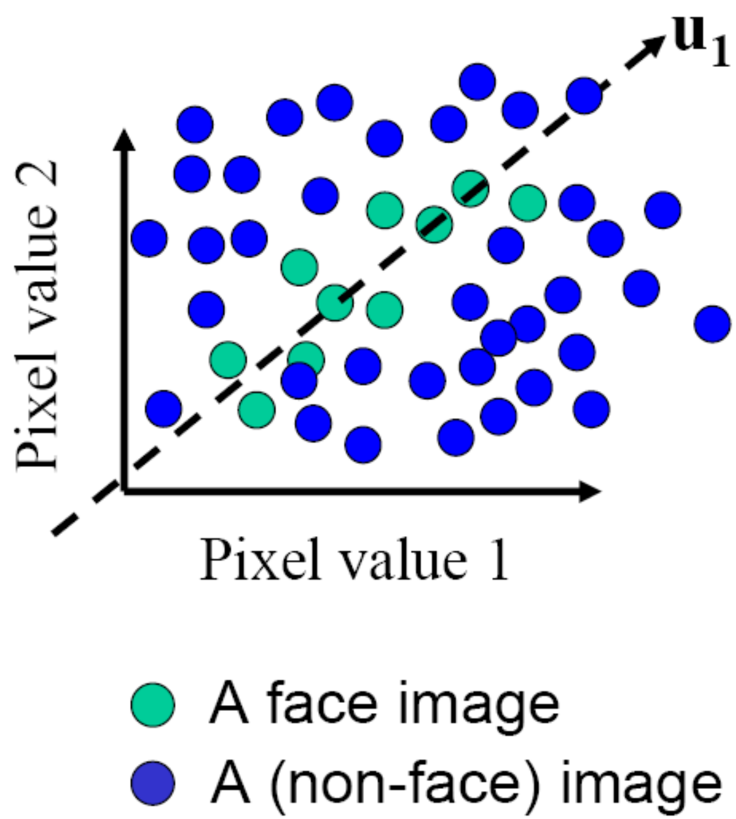
$$k = \underset{k}{\operatorname{argmin}} \|\mathbf{y}_k - \mathbf{x}\|$$

The space of all face images

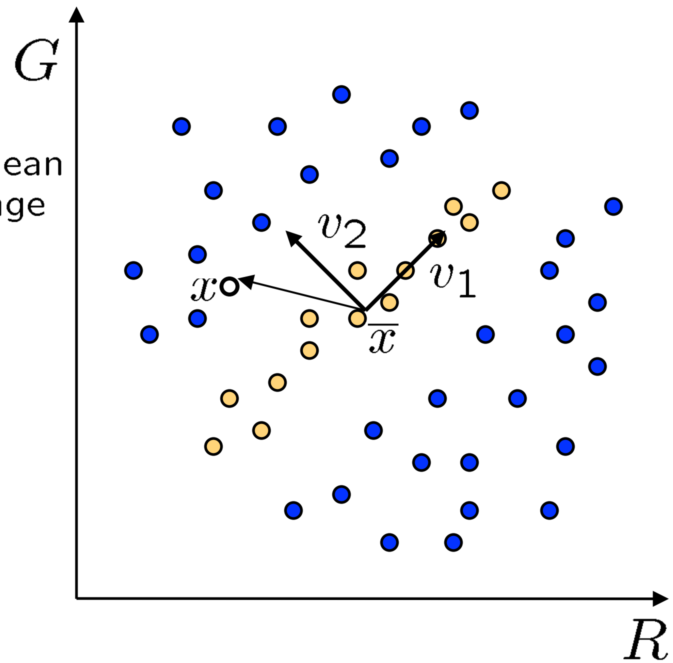
- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100x100 image = 10,000 dimensions
  - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images

The space of all face images

- Idea: construct a low-dimensional linear subspace that best explains the variation in the set of face images



Linear subspaces



$\overline{x}$ is the mean of the orange points

Consider the variation along direction v among all of the orange points:

$$var(\mathbf{v}) = \sum_{\text{orange point } \mathbf{x}} \|(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\|^2$$

What unit vector v minimizes var?

$$\mathbf{v}_2 = min_{\mathbf{v}} \; \{var(\mathbf{v})\}$$

What unit vector v maximizes var?

$$\mathbf{v}_1 = max_{\mathbf{v}} \; \{var(\mathbf{v})\}$$

$$\begin{aligned} var(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\| \\ &= \sum_{\mathbf{x}} \mathbf{v}^{\mathbf{T}}(\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}}\mathbf{v} \\ &= \mathbf{v}^{\mathbf{T}}\left[\sum_{\mathbf{x}}(\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}}\right]\mathbf{v} \\ &= \mathbf{v}^{\mathbf{T}}\mathbf{A}\mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}}(\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \end{aligned}$$

Solution: v_1 is eigenvector of A with largest eigenvalue
v_2 is eigenvector of A with smallest eigenvalue

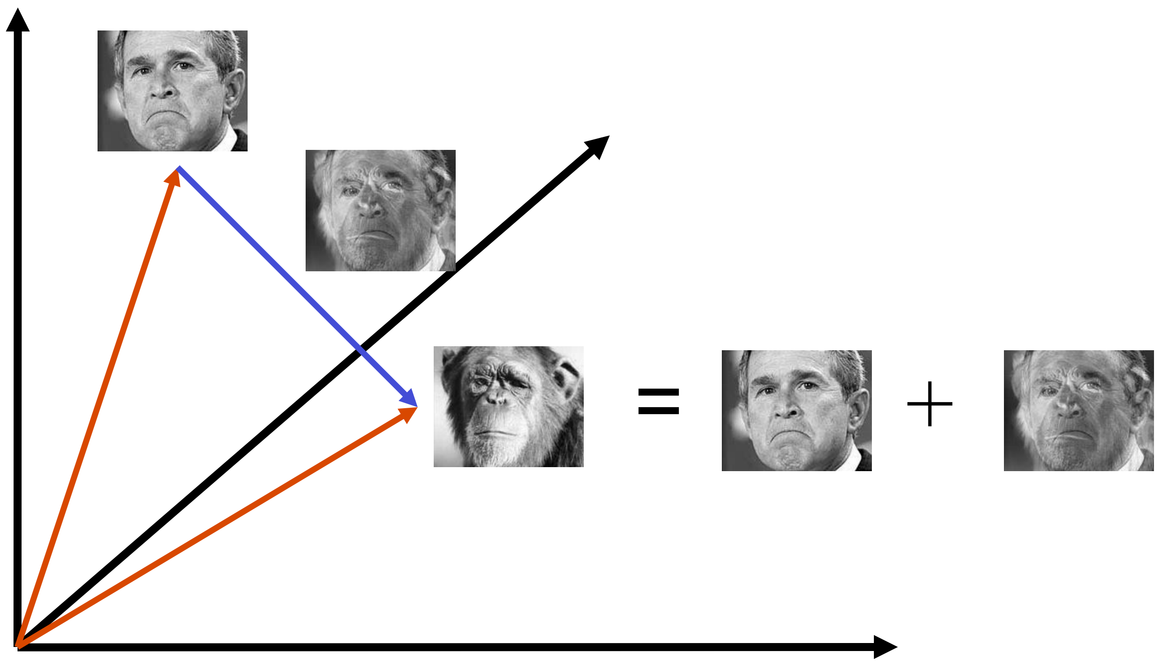
Principal component analysis (PCA)

- Suppose each data point is N-dimensional
  - Same procedure applies:

$$
\begin{aligned}
var(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\| \\
&= \mathbf{v}^{\mathbf{T}} \mathbf{A} \mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}}
\end{aligned}
$$

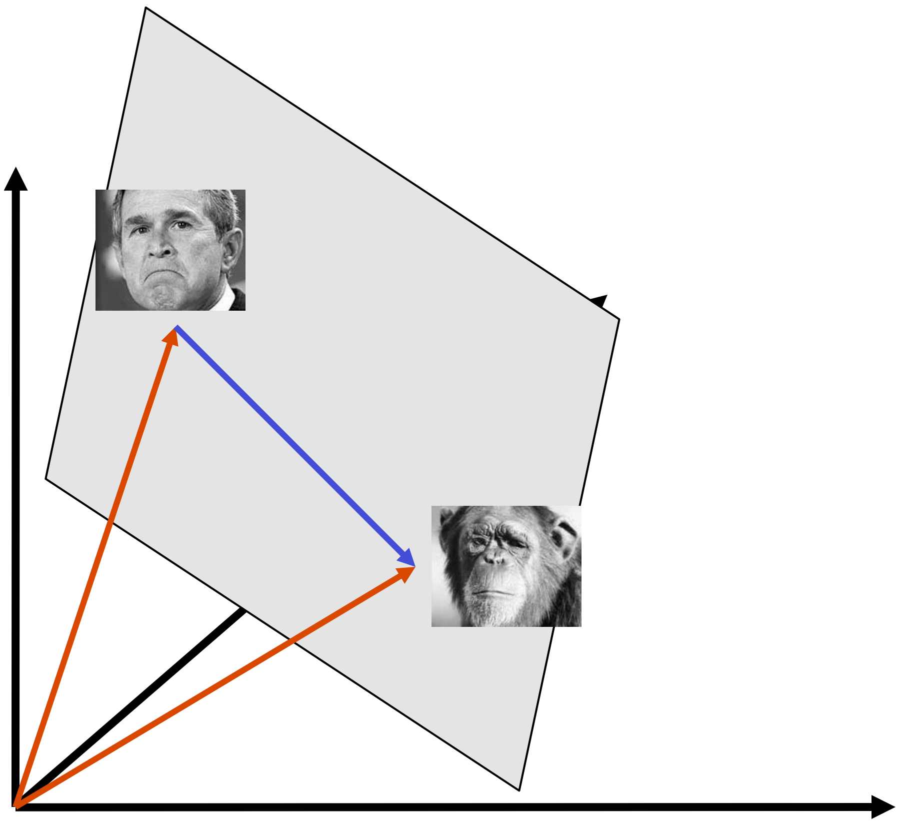
  - The eigenvectors of A define a new coordinate system
    - eigenvector with largest eigenvalue captures the most variation among training vectors x
    - eigenvector with smallest eigenvalue has least variation
  - We can compress the data by only using the top few eigenvectors
    - corresponds to choosing a "linear subspace"
      - represent points on a line, plane, or "hyper-plane"
    - these eigenvectors are known as the principal components

The space of faces



- An image is a point in a high dimensional space
  - An N x M image is a point in R^{NM}
  - We can define vectors in this space as we did in the 2D case

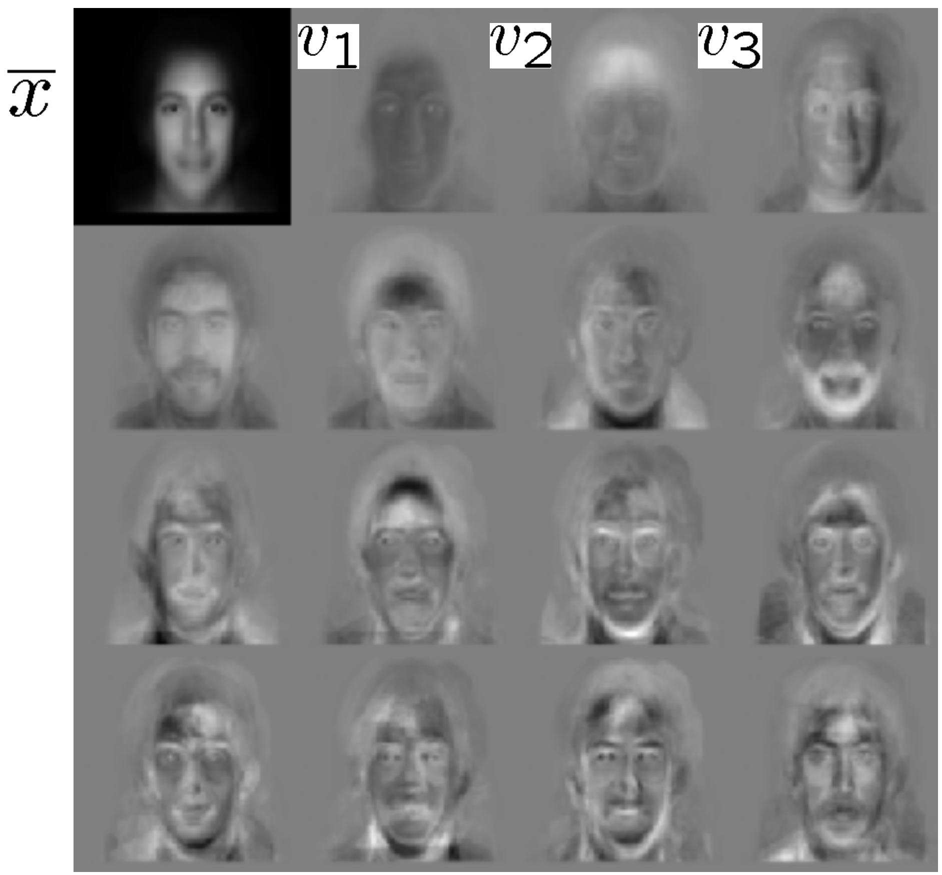
Dimensionality reduction



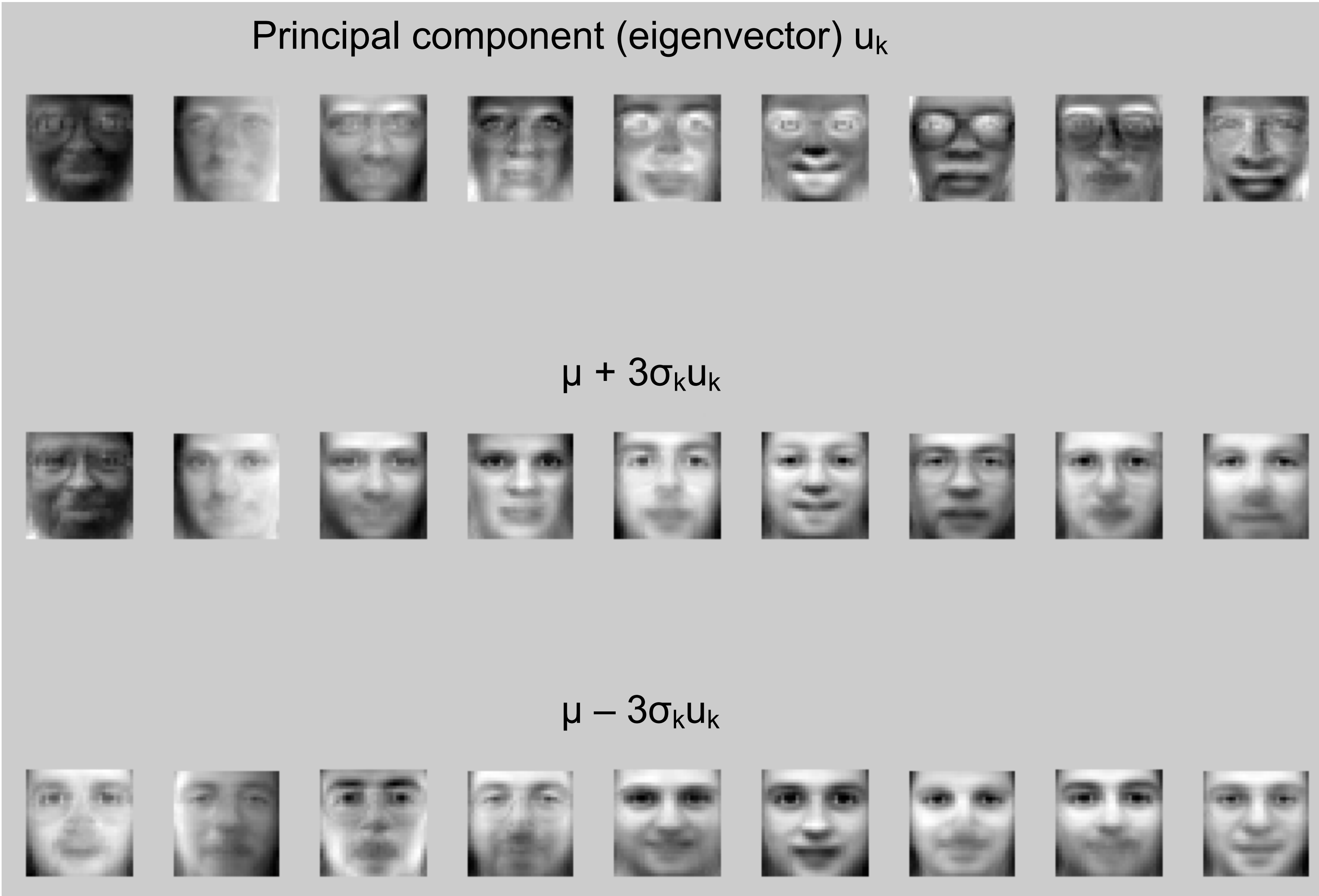
- The set of faces is a "subspace" of the set of images
  - Suppose it is K dimensional
  - We can find the best subspace using PCA
  - This is like fitting a "hyper-plane" to the set of faces
    - spanned by vectors $v_1, v_2, ..., v_K$
    - any face $\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_k\mathbf{v_k}$

Eigenfaces

- PCA extracts the eigenvectors of A
  - Gives a set of vectors v_1, v_2, v_3, ...
  - Each one of these vectors is a direction in face space
    - what do these look like?



Visualization of eigenfaces



Projecting onto the eigenfaces

- The eigenfaces $v_1, \ldots, v_K$ span the space of faces
  - A face is converted to eigenface coordinates by

$$\mathbf{x} \rightarrow (\underbrace{(\mathbf{x}-\overline{\mathbf{x}}) \cdot \mathbf{v_1}}_{a_1}, \underbrace{(\mathbf{x}-\overline{\mathbf{x}}) \cdot \mathbf{v_2}}_{a_2}, \ldots, \underbrace{(\mathbf{x}-\overline{\mathbf{x}}) \cdot \mathbf{v_K}}_{a_K})$$

$$\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$$

$\mathbf{x}$ $\quad a_1\mathbf{v_1}$ $\quad a_2\mathbf{v_2}$ $\quad a_3\mathbf{v_3}$ $\quad a_4\mathbf{v_4}$ $\quad a_5\mathbf{v_5}$ $\quad a_6\mathbf{v_6}$ $\quad a_7\mathbf{v_7}$ $\quad a_8\mathbf{v_8}$

Recognition with eigenfaces

- Algorithm
  1. Process the image database (set of images with labels)
     - Run PCA—compute eigenfaces
     - Calculate the K coefficients for each image
  2. Given a new image (to be recognized) x, calculate K coefficients

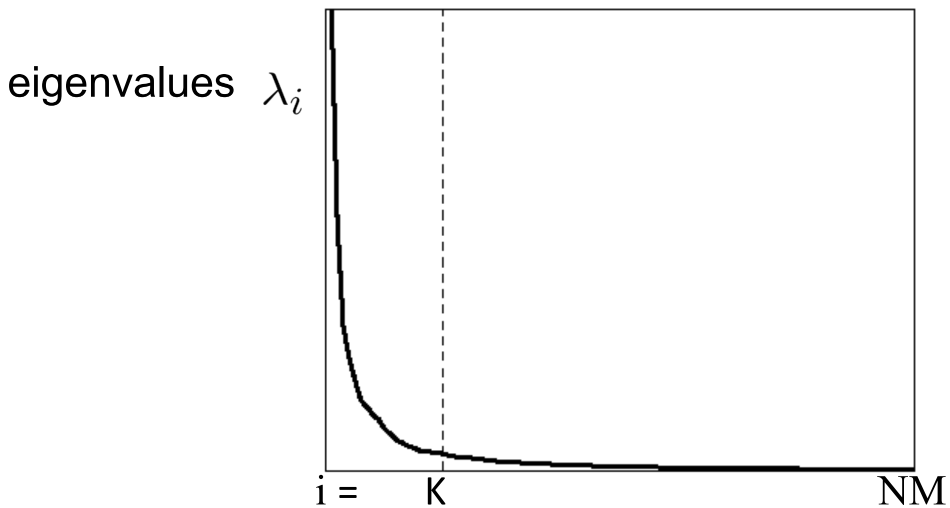
$$\mathbf{x} \rightarrow (a_1, a_2, \ldots, a_K)$$

  3. Detect if x is a face

$$\|\mathbf{x} - (\overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K})\| < \text{threshold}$$

  4. If it is a face, who is it?
     - Find closest labeled face in database
     - nearest-neighbor in K-dimensional space

Choosing the dimension K



- How many eigenfaces to use?
- Look at the decay of the eigenvalues
  - the eigenvalue tells you the amount of variance "in the direction" of that eigenface
  - ignore eigenfaces with low variance

PCA

- General dimensionality reduction technique
- Preserves most of variance with a much more compact representation
  - Lower storage requirements (eigenvectors + a few numbers per face)
  - Faster matching

Enhancing gender

more same **original** androgynous more opposite

D. Rowland and D. Perrett,
"Manipulating Facial Appearance through Shape and Color," IEEE CG&A,
September 1995

Slide credit: A. Efros

Changing age

•Face becomes "rounder" and "more textured" and "grayer"

•original

• color

D. Rowland and D. Perrett,
"Manipulating Facial Appearance through Shape and Color," IEEE CG&A, September 1995

Slide credit: A. Efros

Which face is more attractive?

http://www.beautycheck.de

Which face is more attractive?

Which face is more attractive?

0.5(attractive + average)

attractive

Which face is more attractive?

http://www.beautycheck.de

Which face is more attractive?

Which face is more attractive?


0.5(adult+child)


adult