Recognition, SVD, and PCA

Recognition

- Suppose you want to find a face in an image
- One possibility: look for something that looks sort of like a face (oval, dark band near top, dark band near bottom)
- Another possibility: look for pieces of faces (eyes, mouth, etc.) in a specific arrangement

Templates

- Model of a “generic” or “average” face
  - Learn templates from example data
- For each location in image, look for template at that location
  - Optionally also search over scale, orientation

Templates

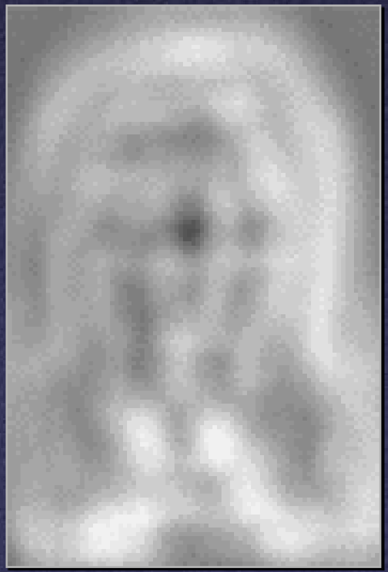
- In the simplest case, based on intensity
  - Template is average of all faces in training set
  - Comparison based on e.g. SSD
- More complex templates
  - Outputs of feature detectors
  - Color histograms
  - Both position and frequency information (wavelets)

Average Princetonian Face

- From 2005 BSE thesis project by Clay Bavor and Jesse Levinson

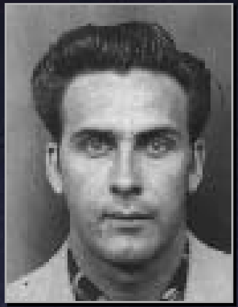
Detecting Princetonians

Matching response
(darker = better match)



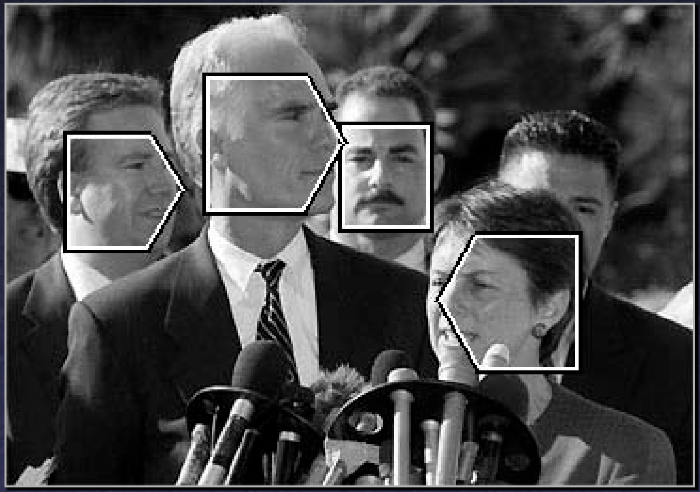
[Bavor & Levinson]

More Detection Results



Sample Images

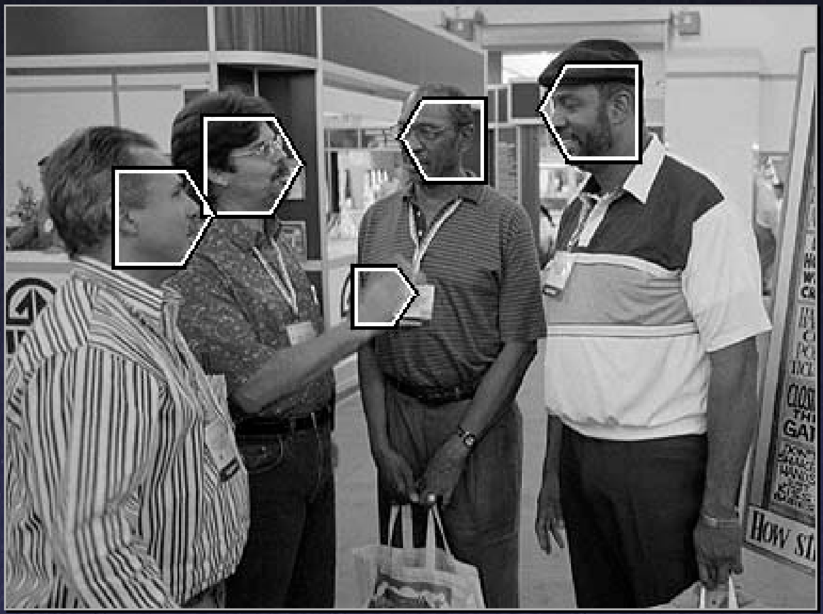
Wavelet Histogram Template

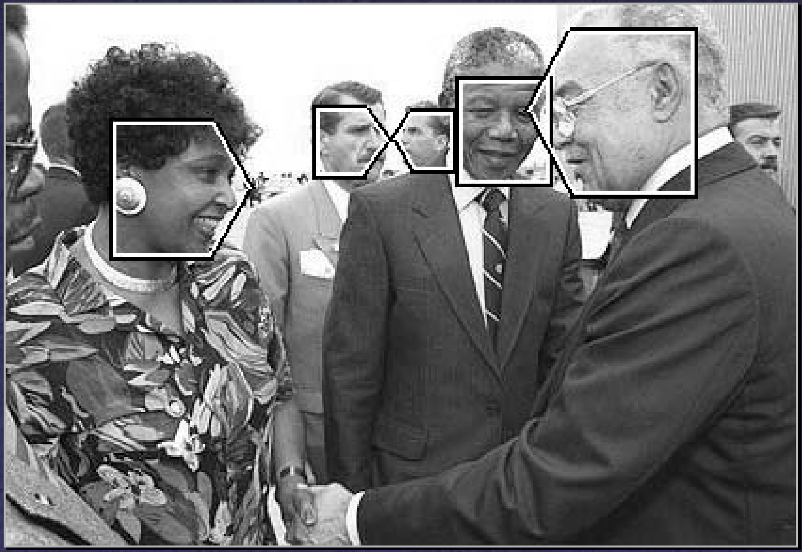


Detection of frontal / profile faces

[Schneiderman and Kanade]

More Face Detection Results



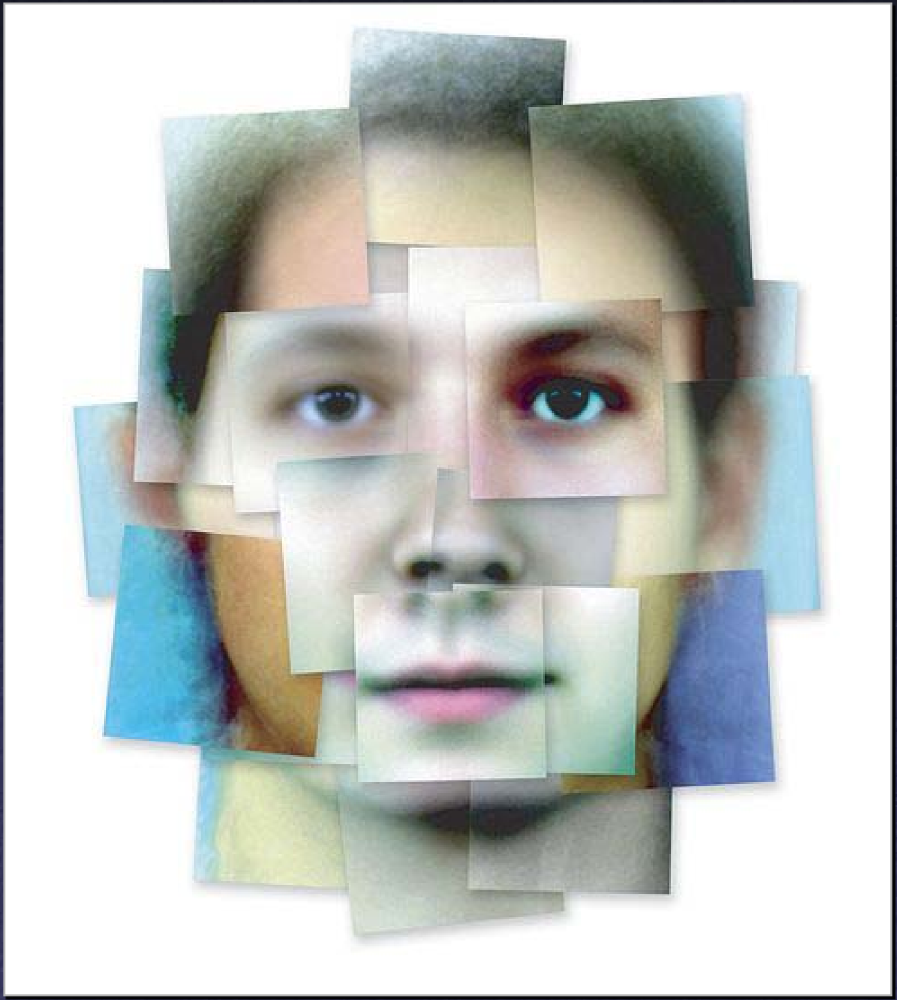


[Schneiderman and Kanade]

Recognition Using Relations Between Templates

- Often easier to recognize a small feature
  - e.g., lips easier to recognize than faces
  - For articulated objects (e.g. people), template for whole class usually complicated
- So, identify small pieces…

Pieces of Princetonians

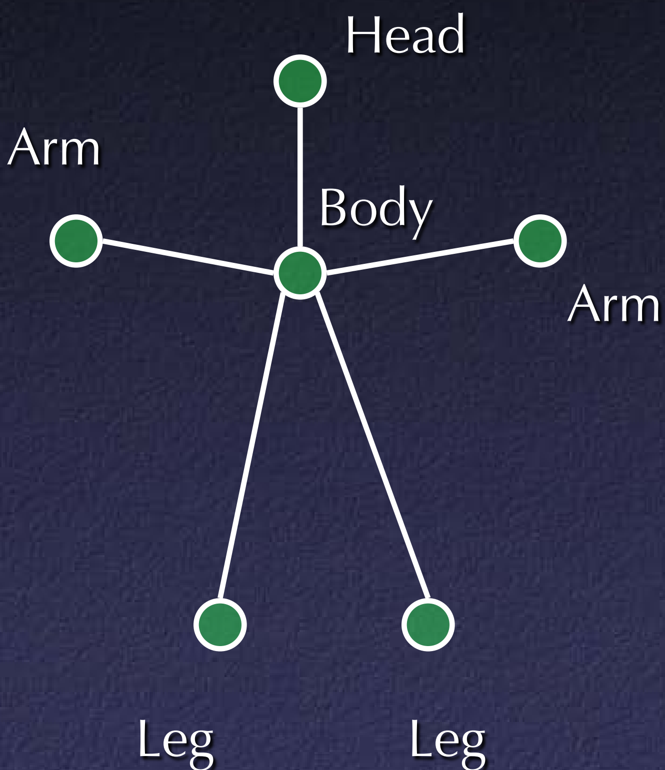


[Bavor & Levinson]

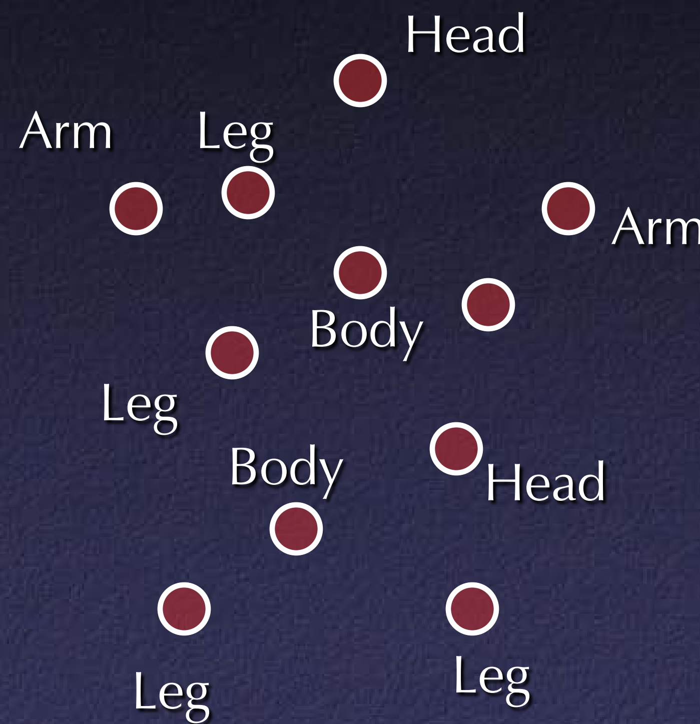
Recognition Using Relations Between Templates

- Often easier to recognize a small feature
  - e.g., lips easier to recognize than faces
  - For articulated objects (e.g. people), template for whole class usually complicated
- So, identify small pieces and look for spatial arrangements
  - Many false positives from identifying pieces

Graph Matching

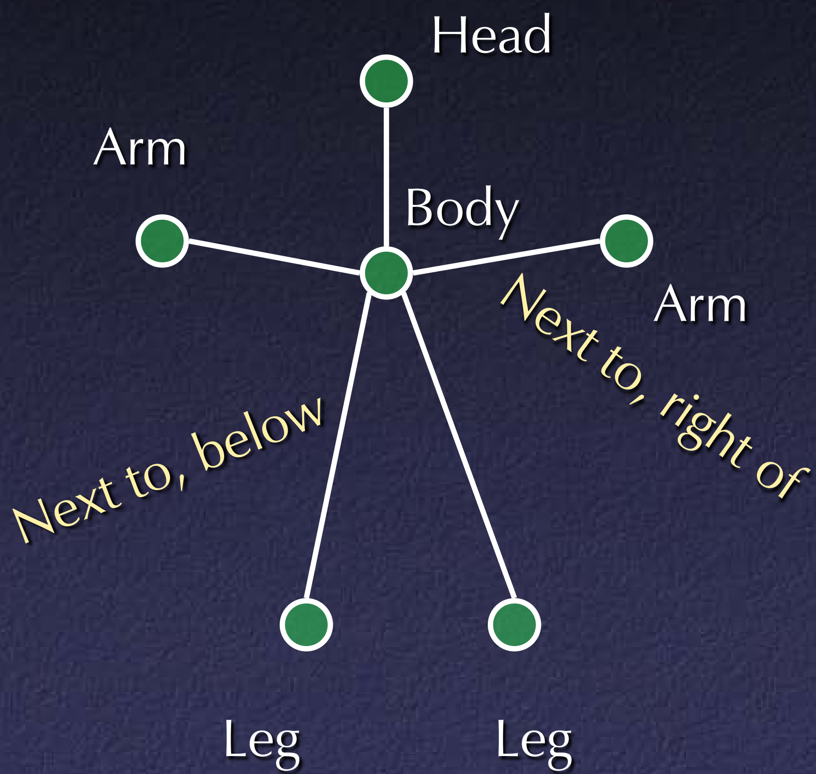


Model



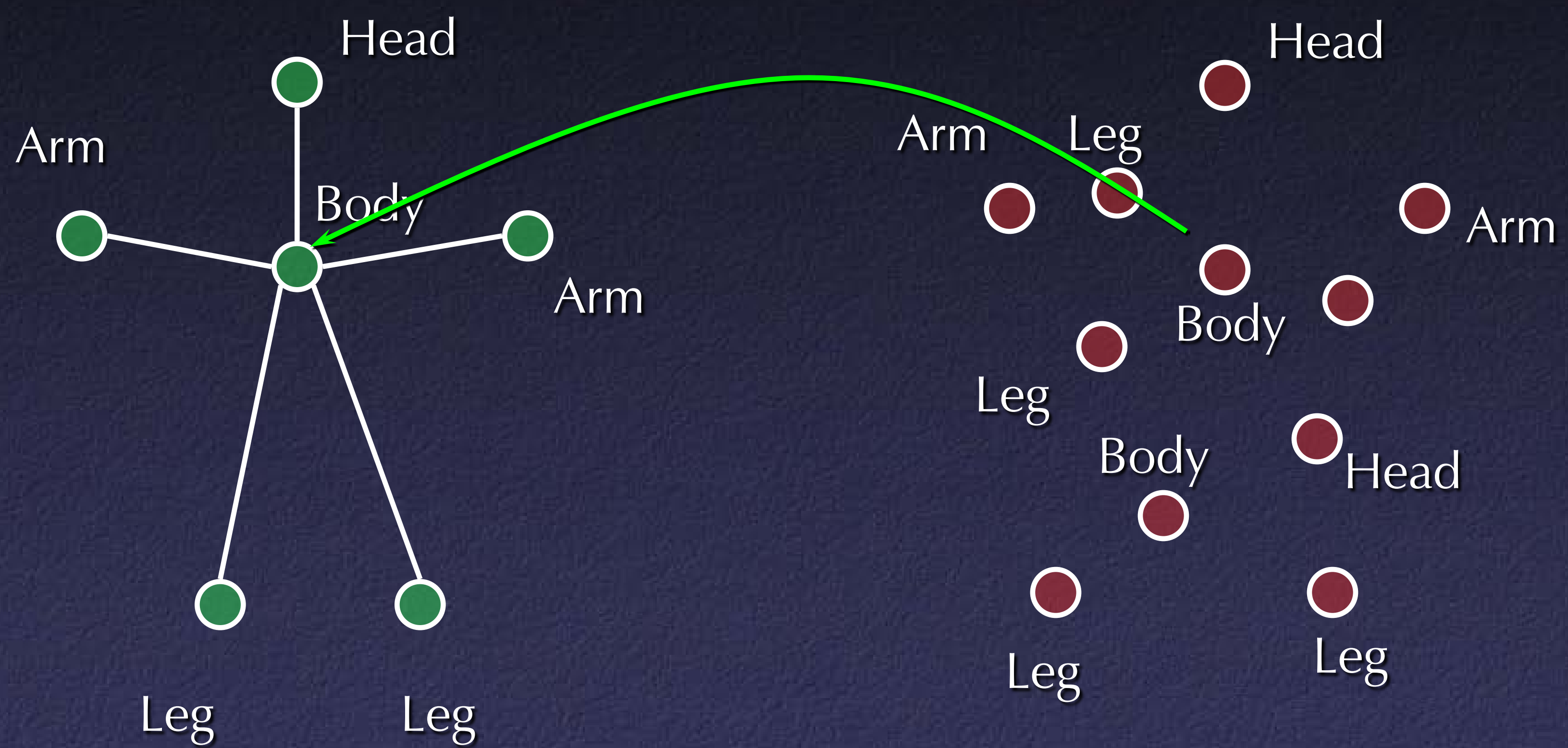
Feature detection results

Graph Matching



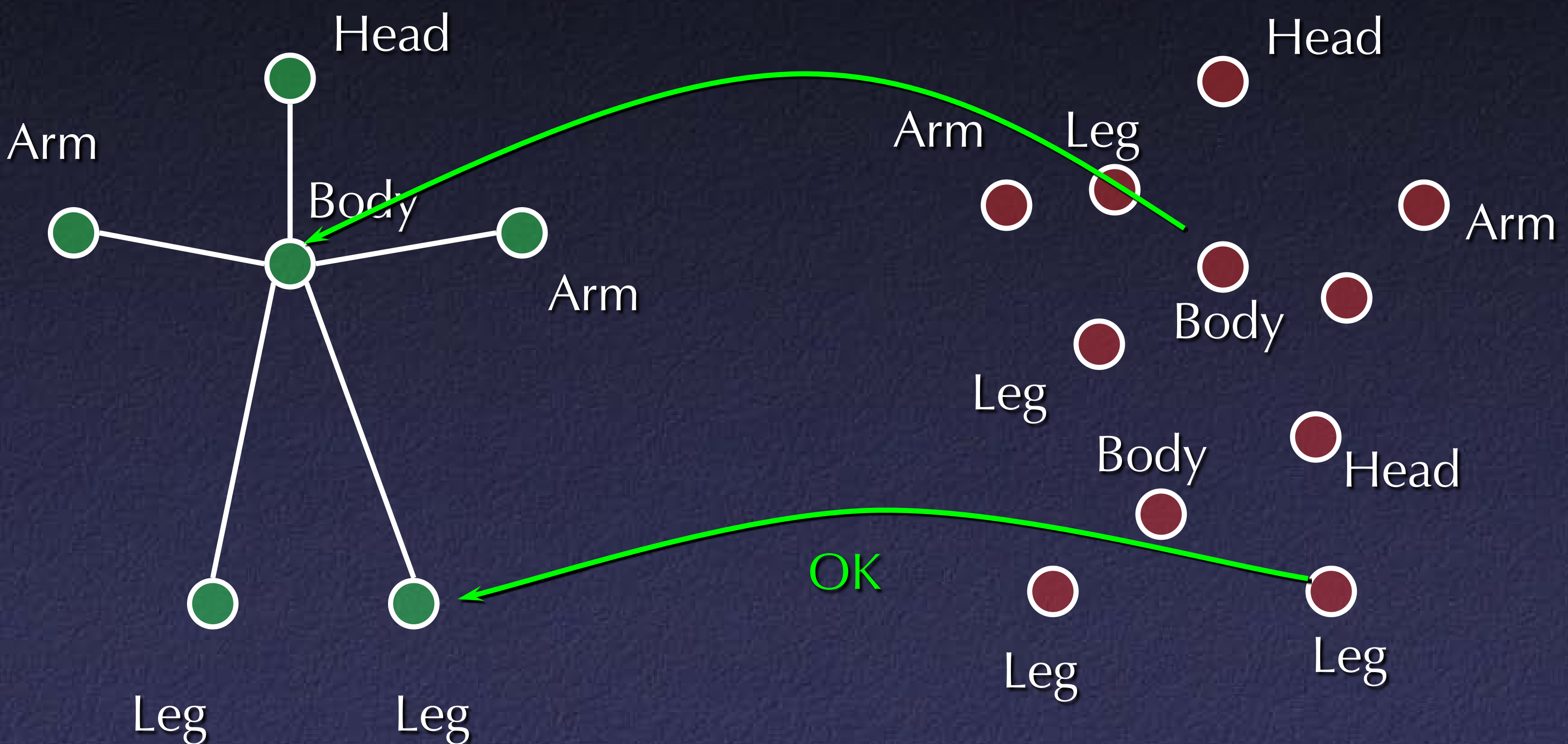
Constraints

Graph Matching



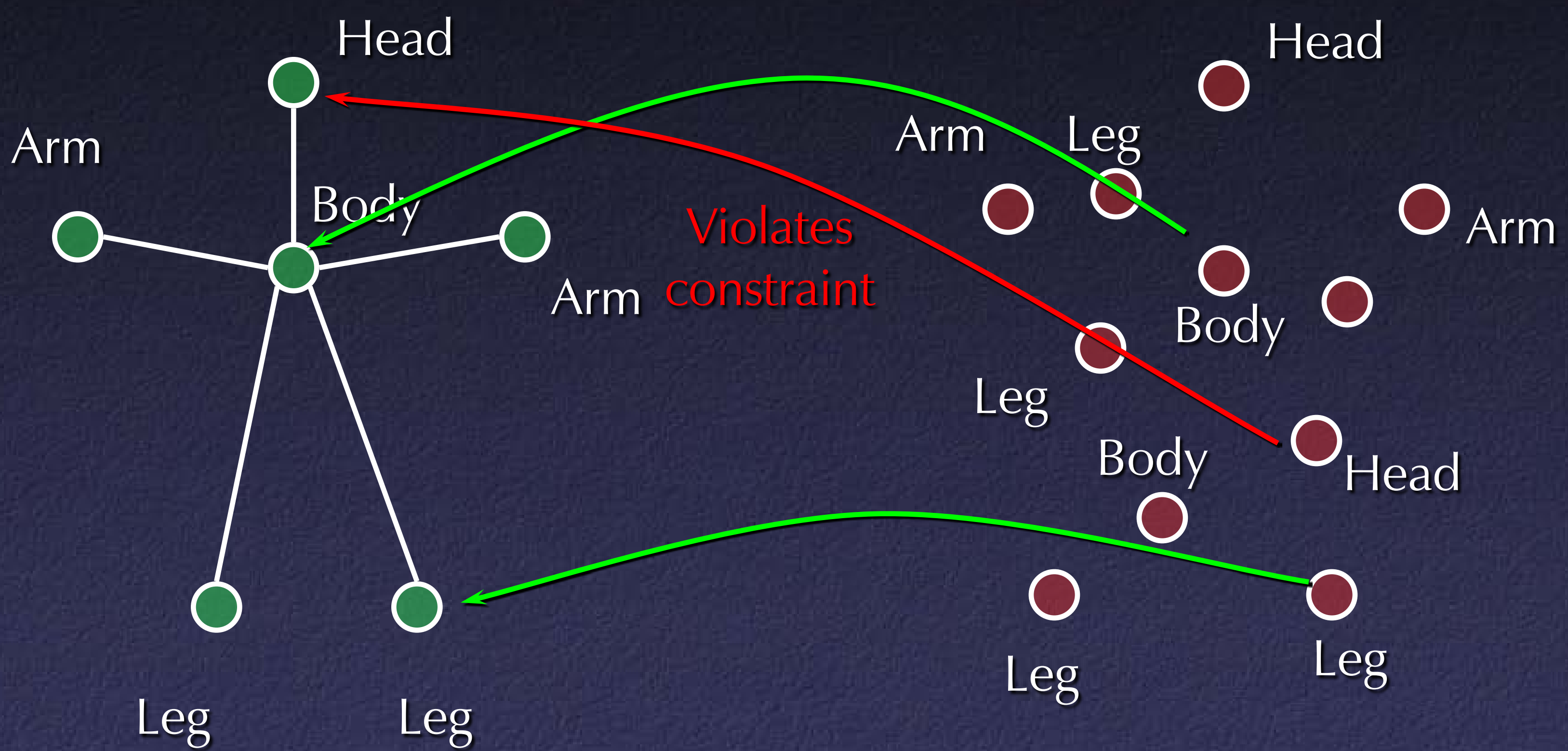
Combinatorial search

Graph Matching



Combinatorial search

Graph Matching



Combinatorial search

Graph Matching

- Large search space
  - Heuristics for pruning
- Missing features
  - Look for maximal consistent assignment
- Noise, spurious features
- Incomplete constraints
  - Verification step at end

Recognition

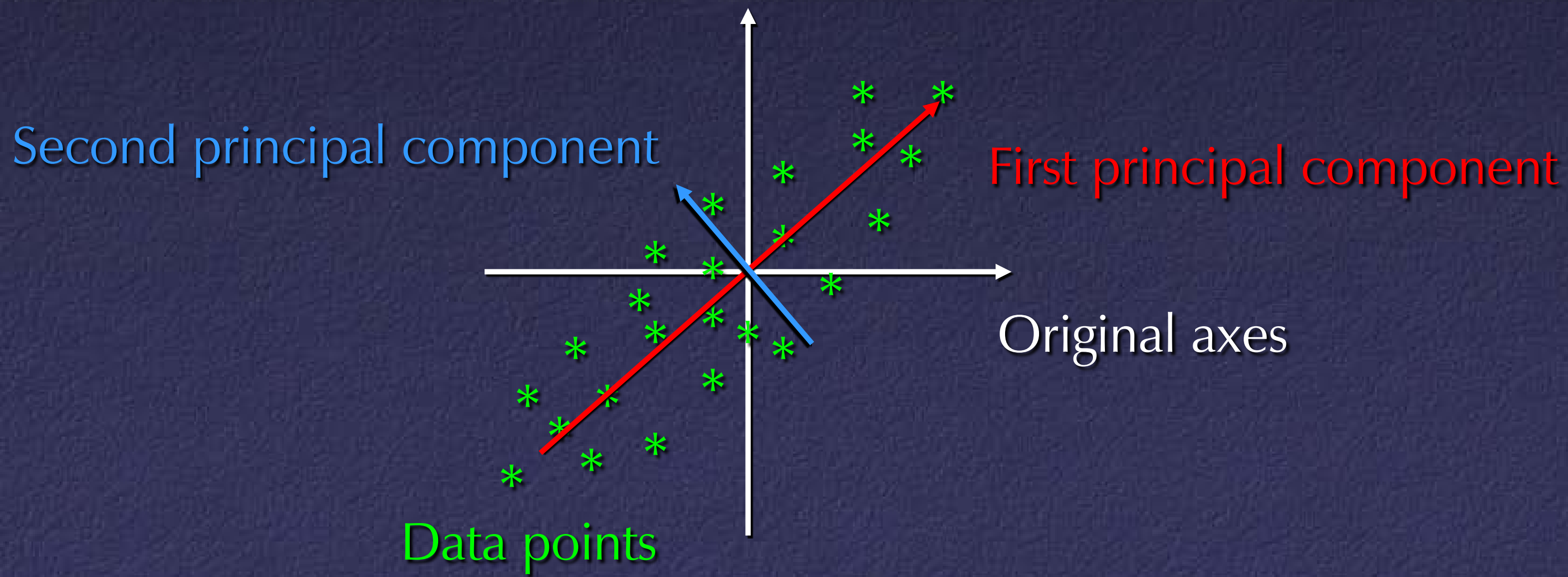
- Suppose you want to recognize a *particular* face
- How does *this* face differ from average face

How to Recognize Specific People?

- Consider variation from average face
- Not all variations equally important
  - Variation in a single pixel relatively unimportant
- If image is high-dimensional vector, want to find directions in this space with high variation

Principal Components Analaysis

- Principal Components Analysis (PCA): approximating a high-dimensional data set with a lower-dimensional subspace



Digression: Singular Value Decomposition (SVD)

- Handy mathematical technique that has application to many problems
- Given any $m \times n$ matrix $\mathbf{A}$, algorithm to find matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ such that

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T$$

$\mathbf{U}$ is $m \times n$ and orthonormal

$\mathbf{V}$ is $n \times n$ and orthonormal

$\mathbf{W}$ is $n \times n$ and diagonal

SVD

$$\begin{pmatrix} \\ \\ & \mathbf{A} & \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \\ & \mathbf{U} & \\ \\ \\ \end{pmatrix} \begin{pmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n \end{pmatrix} \begin{pmatrix} & & \\ & \mathbf{V} & \\ & & \end{pmatrix}^{\mathrm{T}}$$

- Treat as black box: code widely available (`svd(A,0)` in Matlab)

SVD

- The w_i are called the singular values of **A**
- If **A** is singular, some of the w_i will be 0
- In general $rank(\mathbf{A})$ = number of nonzero w_i
- SVD is mostly unique (up to permutation of singular values, or if some w_i are equal)

SVD and Inverses

- Why is SVD so useful?
- Application #1: inverses
- $\mathbf{A}^{-1} = (\mathbf{V}^T)^{-1}\,\mathbf{W}^{-1}\,\mathbf{U}^{-1} = \mathbf{V}\,\mathbf{W}^{-1}\,\mathbf{U}^T$
- This fails when some w_i are 0
  - It's *supposed* to fail – singular matrix
- Pseudoinverse: if $w_i = 0$, set $1/w_i$ to 0 (!)
  - "Closest" matrix to inverse
  - Defined for all (even non-square) matrices

SVD and Least Squares

- Solving **Ax**=**b** by least squares
- **x**=pseudoinverse(**A**) times **b**
- Compute pseudoinverse using SVD
  - Lets you see if data is singular
  - Even if not singular, ratio of max to min singular values (condition number) tells you how stable the solution will be
  - Set $1/w_i$ to 0 if w_i is small (even if not exactly 0)

SVD and Eigenvectors

- Let $\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T$, and let x_i be i^{th} column of $\mathbf{V}$
- Consider $\mathbf{A}^T\mathbf{A}\, x_i$:

$$\mathbf{A}^T\mathbf{A}x_i = \mathbf{V}\mathbf{W}^T\mathbf{U}^T\mathbf{U}\mathbf{W}\mathbf{V}^T x_i = \mathbf{V}\mathbf{W}^2\mathbf{V}^T x_i = \mathbf{V}\mathbf{W}^2\begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{V}\begin{pmatrix} 0 \\ \vdots \\ w_i^2 \\ \vdots \\ 0 \end{pmatrix} = w_i^2 x_i$$

- So elements of $\mathbf{W}$ are squared eigenvalues and columns of $\mathbf{V}$ are eigenvectors of $\mathbf{A}^T\mathbf{A}$

SVD and Matrix Similarity

- One common definition for the norm of a matrix is the Frobenius norm:

$$\|\mathbf{A}\|_{\mathrm{F}} = \sum_i \sum_j a_{ij}^{\ 2}$$

- Frobenius norm can be computed from SVD

$$\|\mathbf{A}\|_{\mathrm{F}} = \sum_i w_i^{\ 2}$$

- So changes to a matrix can be evaluated by looking at changes to singular values

SVD and Matrix Similarity

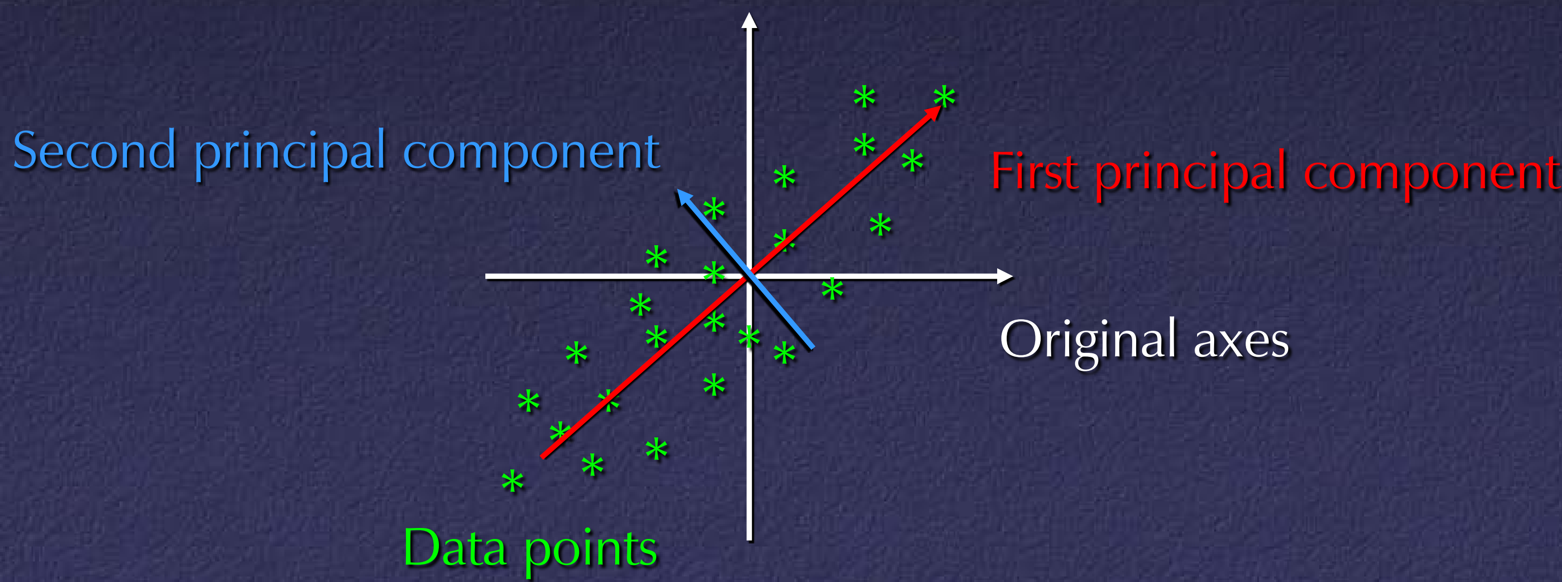
- Suppose you want to find best rank-k approximation to $\mathbf{A}$
- Answer: set all but the largest k singular values to zero
- Can form compact representation by eliminating columns of $\mathbf{U}$ and $\mathbf{V}$ corresponding to zeroed w_i

SVD and Orthogonalization

- The matrix **U** is the "closest" orthonormal matrix to **A**
- Yet another useful application of the matrix-approximation properties of SVD
- Much more stable numerically than Graham-Schmidt orthogonalization
- Find rotation given general affine matrix

SVD and PCA

- Principal Components Analysis (PCA): approximating a high-dimensional data set with a lower-dimensional subspace



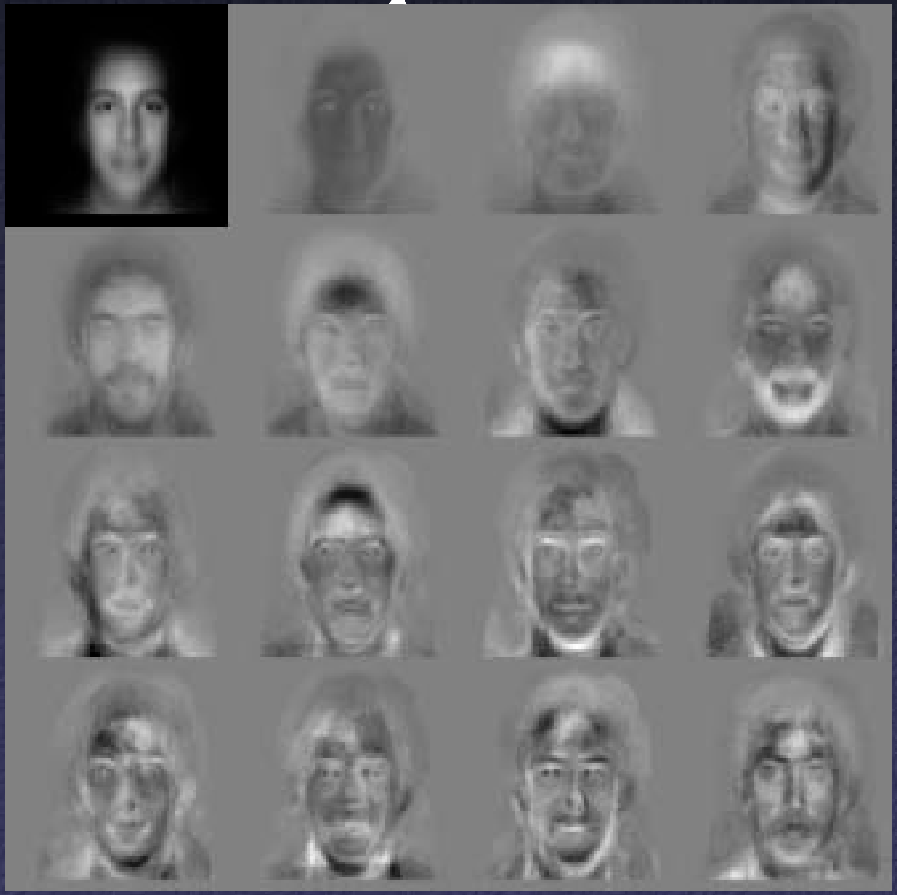
SVD and PCA

- Data matrix with points as rows, take SVD
  - Subtract out mean ("whitening")
- Columns of $\mathbf{V}_k$ are principal components
- Value of w_i gives importance of each component

PCA on Faces: “Eigenfaces”

First principal component

Average face



Other components

For all except average,
“gray” = 0,
“white” > 0,
“black” < 0

Using PCA for Recognition

- Store each person as coefficients of projection onto first few principal components

$$\text{image} = \sum_{i=0}^{i_{\max}} a_i \text{Eigenface}_{\text{i}}$$

- Compute projections of target image, compare to database ("nearest neighbor classifier")