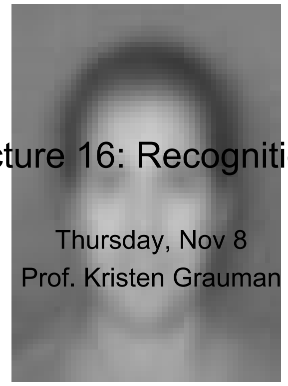


Lecture 16: Recognition II

Thursday, Nov 8

Prof. Kristen Grauman

Outline

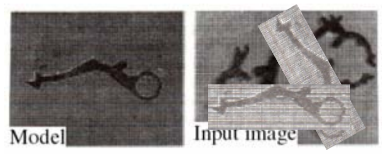
- Finish up model-based recognition:
  - Pose consistency / alignment
  - Pose clustering
- Recognition by classifying windows
  - Face detection/recognition algorithms
    - Eigenfaces for recognition
    - Viola and Jones detector

Model-based recognition

- Which image features correspond to which features on which object model in the "modelbase"?
- If enough match, *and* they match well with a particular transformation for given camera model, then
  - Identify the object as being there
  - Estimate pose relative to camera

Hypothesize and test

- Given model of object
- New image: hypothesize object identity and pose
- Render object in camera
- Compare rendering to actual image: if close, good hypothesis.

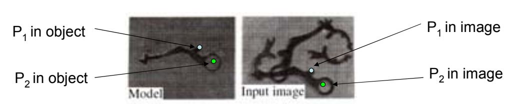


Alignment (pose consistency)

- Key idea:
  - If we find good correspondences for a small set of features, easy to obtain correspondences for a much larger set.
- Strategy:
  - Generate hypotheses using small numbers of correspondences (how many depends on camera type)
  - Backproject: transform all model features to image features
  - Verify

2d affine mappings

- Say camera is looking down perpendicularly on planar surface



- We have two coordinate systems (object and image), and they are related by some affine mapping (rotation, scale, translation, shear).

2d affine mappings

In non-homogenous coordinates

$$\underset{\text{[image point]}}{\begin{bmatrix} u \\ v \end{bmatrix}} = \underbrace{\begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix}}_{\text{[scale, rotation, shear]}} \overset{\text{[model point]}}{\begin{bmatrix} x \\ y \end{bmatrix}} + \underbrace{\begin{bmatrix} t_x \\ t_y \end{bmatrix}}_{\text{[translation]}}$$

In homogenous coordinates

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}}_{\text{[translation, scale, rotation, shear]}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Alignment: backprojection

Similar ideas for camera models (3d->2d)

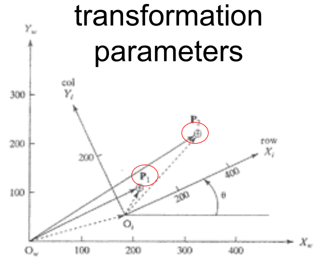
- Perspective camera

$$\underset{\text{image coordinates}}{\overline{\mathbf{p}}} = \mathbf{M}\underset{\text{model coordinates}}{\mathbf{P}_w}$$

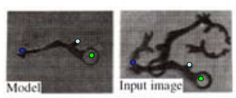
$$x_{im} = \frac{\mathbf{M}_1 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w} \qquad y_{im} = \frac{\mathbf{M}_2 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}$$

- Simpler calibration possible with simpler camera models (affine, projective)

Solving for the transformation parameters



$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$



$\boxed{\mathbf{P}_1^{(model)}} = [200,100]$ $\boxed{\mathbf{P}_1^{(image)}} = [100,60]$

$\mathbf{P}_2^{(model)} = [300,200]$ $\mathbf{P}_2^{(image)} = [380,120]$

$\vdots \qquad \vdots$

$$\begin{bmatrix} \boxed{\begin{matrix} x & y & 0 & 0 \\ 0 & 0 & x & y \end{matrix}} & \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \\ \cdots & \\ \cdots & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} u \\ v \end{matrix}} \\ \vdots \end{bmatrix}$$

Rewrite in terms of unknown parameters

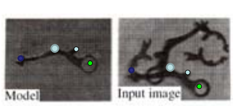
Alignment: verification

- Given the backprojected model in the image:
  - Check if image edges coincide with predicted model edges
  - May be more robust if also require edges to have the same orientation
  - Consider texture in corresponding regions?

Alignment: backprojection

- Having solved for this transformation from some number of detected matches (3+ here), can compute (hypothesized) location of any *other* model points in the image space.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$



Alignment: verification

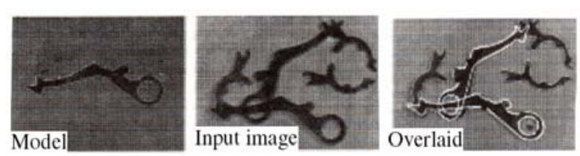
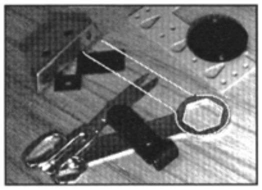


Figure from "Object recognition using alignment," D.P. Huttenlocher and S. Ullman, Proc. Int. Conf. Computer Vision, 1986, copyright IEEE, 1986

Alignment: verification



Edge-based verification can be brittle

Pose clustering (voting)

```
For all objects O
  For all object frame groups F(O)
    For all image frame groups F(I)
      For all correspondences C between
        elements of F(I) and elements
        of F(O)

        Use F(I), F(O) and C to infer object pose P(O)

        Add a vote to O's pose space at the bucket
        corresponding to P(O).
      end
    end
  end
end
For all objects O
  For all elements P(O) of O's pose space that have
    enough votes

    Use the P(O) and the
    camera model estimate to render the object

    If the rendering conforms to the image,
    the object is present
  end
end
```

Alignment: Matching object and image groups to infer a camera model

```
For all object frame groups O
  For all image frame groups F
    For all correspondences C between
      elements of F and elements
      of O

      Use F, C and O to infer the missing parameters
      in a camera model

      Use the camera model estimate to render the object

      If the rendering conforms to the image,
        the object is present
    end
  end
end
```

What hypotheses should be considered for verification?

Forsyth and Ponce

Recall: difficulties of voting

- Noise/clutter can lead to as many votes as true target
- Bin size for the accumulator array must be chosen carefully

Pose clustering (voting)

- Narrow down the number of hypotheses to verify: identify those model poses that a lot of features agree on.
  – Use each group's correspondence to estimate pose
  – Vote for that object pose in accumulator array (one array per object if we have multiple models)

Pose clustering and verification with SIFT [Lowe]

1) Index descriptors (distinctive features narrow possible matches)
2) Hough transform to vote for poses (keypoints have record of parameters relative to model coordinate system)
3) Affine fit to check for agreement between model and image (approximates perspective projection for planar objects)

Planar objects

Model images and their SIFT keypoints

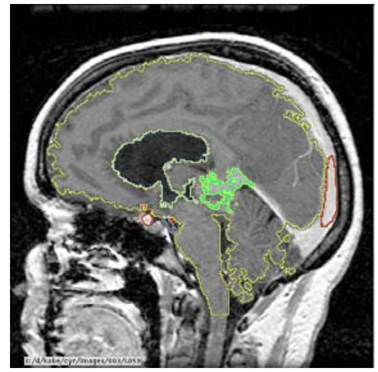
Input image

Model keypoints that were used to recognize, get least squares solution.

Recognition result

[Lowe]

Segmentation used to break single MRI slice into regions.

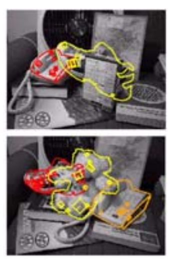


Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

3d objects

Background subtract for model boundaries

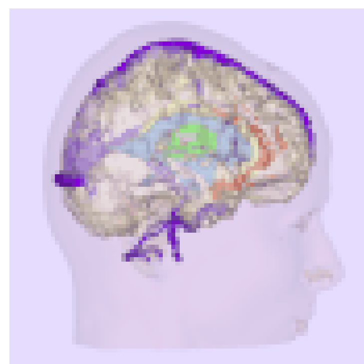
Objects recognized, though affine model not as accurate.



Recognition in spite of occlusion

[Lowe]

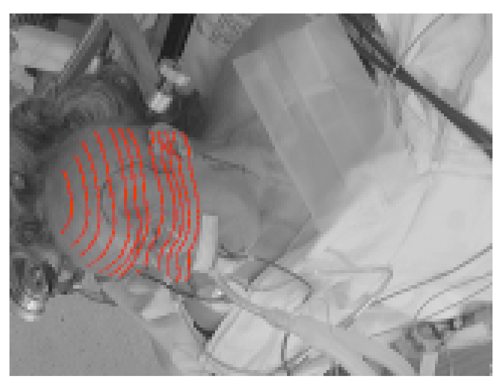
Regions assembled into 3d model



Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

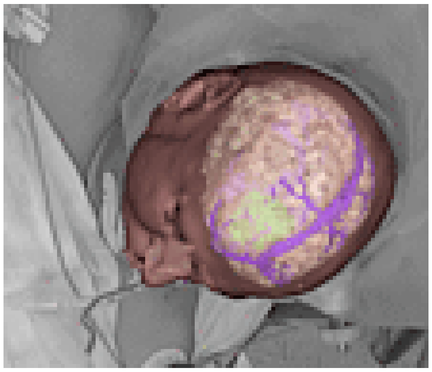
Application: Surgery

- To minimize damage by operation planning
- To reduce number of operations by planning surgery
- To remove only affected tissue
- Problem
  - ensure that the model with the operations planned on it and the information about the affected tissue lines up with the patient
  - display model information supervised on view of patient
  - **Big Issue**: coordinate alignment, as above



Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

Patient with model superimposed. Note that view of model is registered to patient's pose here.



Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

Outline

- Finish up model-based recognition:
  - Pose consistency / alignment
  - Pose clustering
- Recognition by classifying windows
  - Face detection/recognition algorithms

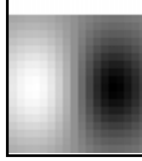
Figures by kind permission of Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html.

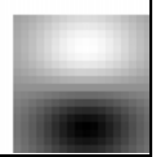
Recall:

Filters as templates

- Applying filter = taking a dot-product between image and some vector
- Filtering the image is a set of dot products

- Insight
  - filters look like the effects they are intended to find
  - filters find effects they look like



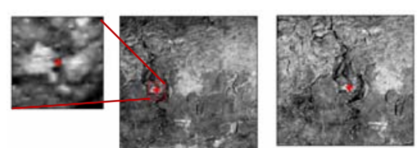


Summary: model-based recognition

- Hypothesize and test: looking for object and pose that fits well with image
  - Use good correspondences to designate hypotheses
  - Limit verifications performed by voting
- Requires model for the specific objects
  - Searching a modelbase
  - Registration tasks
- Requires camera model selection

Recall:

Normalized cross correlation



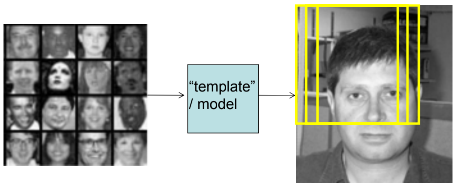
Template

Best match

- Normalized correlation: normalize for image region brightness
- Windowed correlation search: inexpensive way to find a fixed scale pattern
- (Convolution = correlation if filter is symmetric)

Recognition via template matching

- If the structure/shape of an object is regular enough, can consider this as approach to object recognition.



Supervised classification

- Want to minimize the expected misclassification
- Two general strategies
  - Use the training data to build representative probability model (generative)
  - Directly construct a good decision boundary (discriminative)

- At each window location, how to determine whether object template is present?
  - Representation of window and template
  - "Test" as function of these representations that returns present or absent.

Generative vs. Discriminative Models

- Generative approach: separately model class-conditional densities and priors

$$p(\mathbf{x}|\mathcal{C}_k), \qquad p(\mathcal{C}_k)$$

then evaluate posterior probabilities using Bayes' theorem

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}$$

- Discriminative approach: directly model posterior probabilities

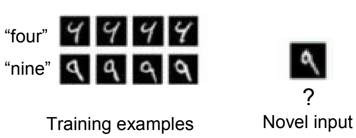
$$p(\mathcal{C}_k|\mathbf{x})$$

- In both cases usually work in a feature space

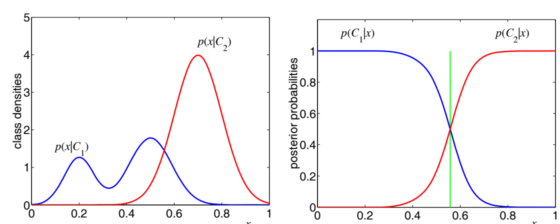
Slide from Christopher M. Bishop, MSR Cambridge

Supervised classification

- Given a collection of *labeled* examples, come up with a function that will predict the labels of new examples.



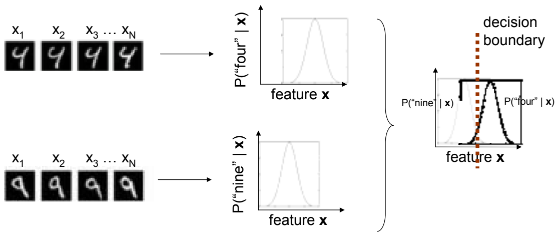
Generative vs. Discriminative



Slide from Christopher M. Bishop, MSR Cambridge

Supervised classification

- Use the training data to build representative probability model

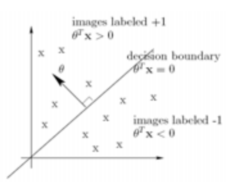


Supervised classification

- Use the training data to build representative probability model
  - Pros:
    - Model for each class means we can draw samples, interpret what is learned
  - Cons:
    - May be hard to get a good model with small number of parameters
    - Models variability that is not important for the task
    - Possible to get a good classifier with density model that doesn't accurately describe the data

Supervised classification

- Directly construct a good decision boundary



Supervised classification

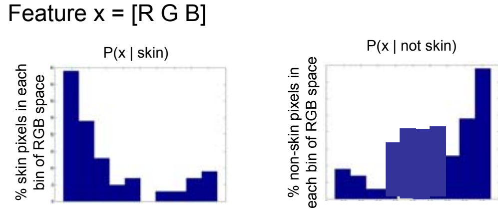
- Directly construct a good decision boundary
  - Pros:
    - Concentrates computational effort on problem we want to solve
    - Appealing when infeasible to model data itself
    - Excel in practice
  - Cons
    - Generally can't say what prediction uncertainty is
    - Cannot interpret class model or sample
    - Interpolate between training examples, can fail with novel inputs

Histogram-based classifiers

- Represent the class-conditional densities with discrete histograms
- P(x | class1), P(x | class2), …

Example: classifying skin pixels

Feature x = [R G B]



Apply Bayes' rule: P(skin | x) α P(x | skin) P(skin)

Example: classifying skin pixels

For every pixel in a new image, can estimate probability that it is generated by skin

Figure from G. Bradski

Classify pixels based on these probabilities

- if $p(\text{skin}|\boldsymbol{x}) > \theta$, classify as skin
- if $p(\text{skin}|\boldsymbol{x}) < \theta$, classify as not skin
- if $p(\text{skin}|\boldsymbol{x}) = \theta$, choose classes uniformly and at random

Example: classifying skin pixels

- Black=pixels classified as skin

Jones and Rehg, CVPR 1999.

Why faces?

- Natural applications in human-computer interfaces (teleconferencing, assistive technology), organizing personal photos, surveillance,…
- Well-studied category, special structure

Faces

- Detection: given an image, where is the face?
- Recognition: whose face is it?

Ann

Image credit: H. Rowley

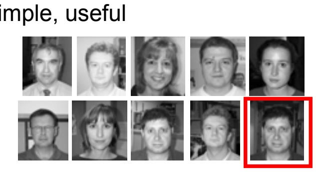
Challenges

- Face pose
- Occlusions
- Illumination
- Variable components (glasses, mustache, etc.)
- Differences in expression

Nearest neighbor classifiers

- Simple, useful

Labeled training set



Assign class label of nearest example in training set (or vote among top *k*)

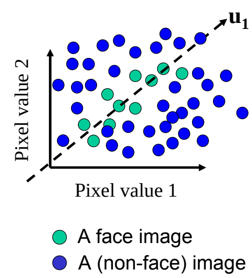
- In general, challenges:
  - Searching for exact neighbors in high-dimensional spaces is expensive
  - What distance is appropriate?

Recognition via template matching

- Templates are simple view-based representation
  - Maintain templates for every viewing condition, lighting, etc.
  - Nearest neighbor search with cross-correlation
- Issues
  - Storage, computation costs unreasonable as number of faces or variations handled is increased
  - Variations in the face images < num possible images represented with n x n values?

Intuition

- Construct lower dimensional linear subspace that best explains variation of the training examples



Eigenpictures/Eigenfaces

- **Main idea**: face images are highly correlated; low-d linear subspace captures most appearance variation

- Sirovitch and Kirby 1987: PCA to compress face images
- Turk and Pentland 1991: PCA + nearest neighbors to classify face images

Eigenfaces

- N data points: $\mathbf{x_1},\ldots,\mathbf{x_N}$ $\mathbf{x_i}$ in R^d
- Mean vector $\boldsymbol{\mu}$, covariance matrix Σ

Want new set of features that are linear combinations of original ones.

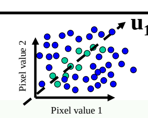
What unit vector $\mathbf{u}$ in R^d captures the most possible variance of the data?

Images as high-dimensional points

- Around d=80,000 pixels each
- To represent the space accurately, want num samples >> d

- But space of **face images** actually much smaller than space of all 80,000 dimensional images

PCA



We want new feature that captures most variance in original data

$$var(u) = \frac{1}{N}\sum_{i=1}^{N-1} \underbrace{\mathbf{u}^{\mathrm{T}}(\mathbf{x}_i - \mu)}_{\text{projection of data point}}(\mathbf{u}^{\mathrm{T}}(\mathbf{x}_i - \mu))^{\mathrm{T}}$$

$$= \mathbf{u}^{\mathrm{T}}\underbrace{\left[\sum_{i=1}^{N-1}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^{\mathrm{T}}\right]}_{\text{covariance of data points}}\mathbf{u}$$

$$= \mathbf{u}^{\mathrm{T}}\Sigma\mathbf{u}$$

Maximizing this is an eigenvalue problem→ use eigenvector(s) of Σ that correspond to the largest eigenvalue(s) as the new basis.

Eigenfaces

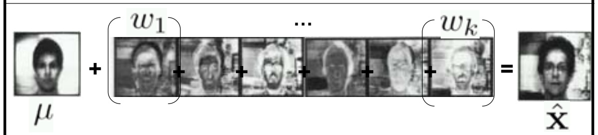
- Set of faces lie in a subspace of set of all images
- Use PCA to determine the *k* (*k*<*d*) vectors $\mathbf{u}_1,\ldots\mathbf{u}_k$ that span that subspace:

  $\mathbf{x} \approx \boldsymbol{\mu} + w_1\mathbf{u}_1 + w_2\mathbf{u}_2 + \ldots + w_k\mathbf{u}_k$
- Then use nearest neighbors in "face space" coordinates ($w_1,\ldots w_k$) to do recognition

Eigenfaces

Face **x** in "face space" coordinates:

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



Eigenfaces

Training images:

$\mathbf{x}_1,\ldots,\mathbf{x}_N$

Eigenface recognition

- Process labeled training images:
  - Run PCA
  - Project each training image onto subspace
- Given novel image:
  - Project onto subspace
  - If $||\hat{\mathbf{x}}-\mathbf{x}|| > \theta$

    Unknown, not face
  - Else

    Classify as closest training face in k-dimensional subspace

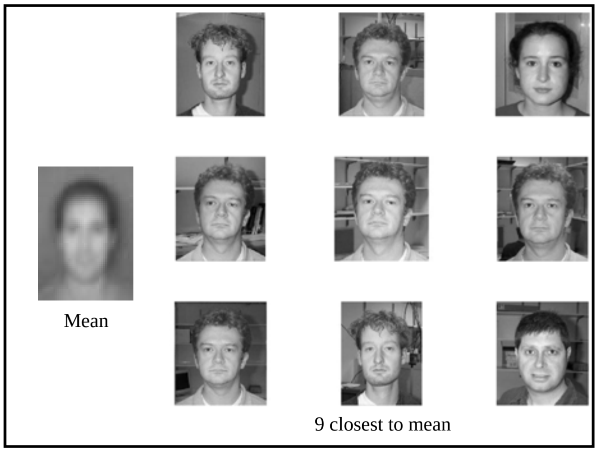
Eigenfaces

Top eigenvectors: $u_1,\ldots u_k$

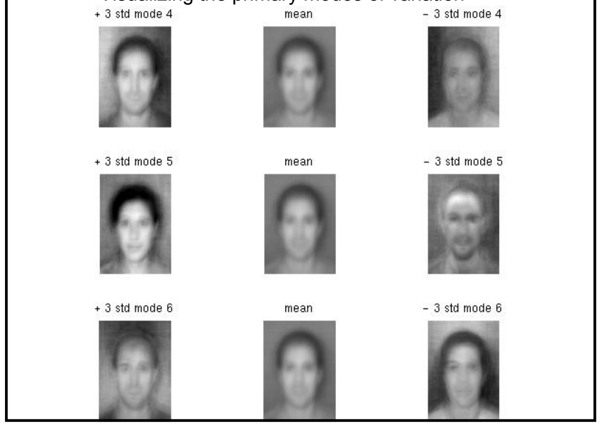
Mean: μ

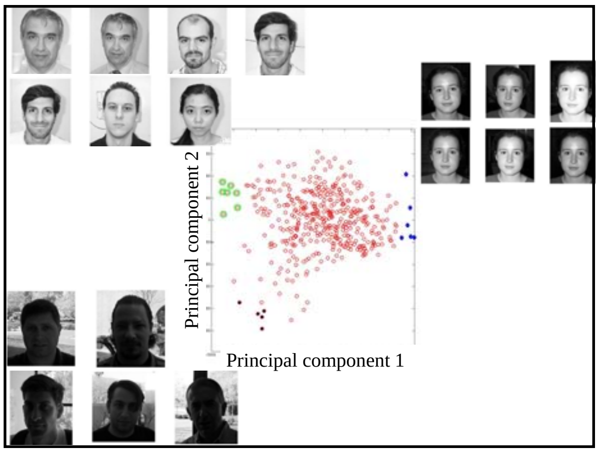
Small demo

- Eigenfaces on the face images in the Caltech-4 database
- 435 images, same scale, aligned



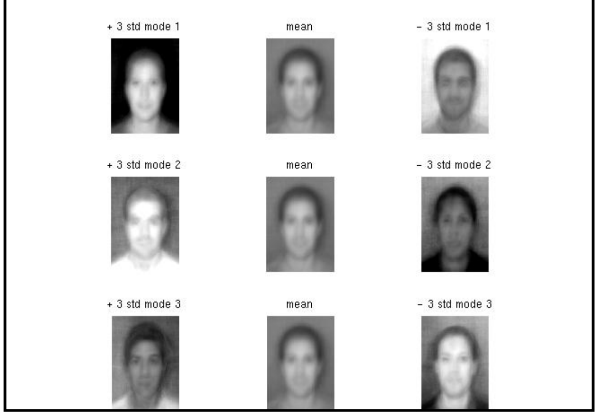
Visualizing the primary modes of variation





Clustering in the face subspace

Visualizing the primary modes of variation



Clustering in face subspace

Clustering in face subspace

Limitations

- PCA useful to *represent* data, but directions of most variance not necessarily useful for classification
- Not appropriate for all data: PCA is fitting Gaussian where Σ is covariance matrix
- Assumptions about pre-processing may be unrealistic, or demands good detector
- Suited for what kinds of categories?

Coming up

- Problem set 3 due on Tuesday 11/13
- Read FP 22.5, Ch 25, Viola & Jones paper