Recognition II

Introduction to Computer Vision

CSE 152

Lecture 19

Announcements

- Assignment 4: Due Friday
- Read: Trucco & Verri, Chapter 10 on recognition
- Final Exam:

Mathematical formulation

[Note change of notation: image coordinates now (x,y), not (u,v)]

I (x,y,t) = brightness at image point *(x,y)* at time *t*

Consider scene (or camera) to be moving, so x(t), y(t)

Brightness constancy assumption:

$$I(x+\frac{dx}{dt}\delta t, y+\frac{dy}{dt}\delta t, t+\delta t) = I(x,y,t) \Rightarrow \frac{dI}{dt} = 0$$

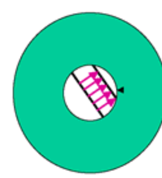
Optical flow constraint equation :

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

Aperture Problem and Normal Flow

Measurements

$I_x = \frac{\partial I}{\partial y}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$



The gradient constraint:

$$I_x u + I_y v + I_t = 0$$
$$\nabla I \bullet \vec{U} = 0$$

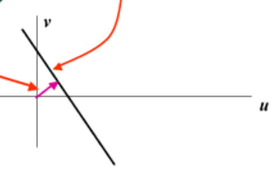
Defines a line in the *(u,v)* space

Flow vector

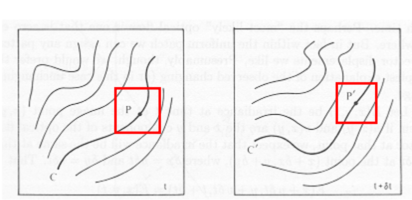
$u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$

Normal Flow:

$$u_\perp = -\frac{I_t}{|\nabla I|}\frac{\nabla I}{|\nabla I|}$$



Two ways to get flow



1. Think globally, and regularize over image
2. Look over window and assume constant motion in the window

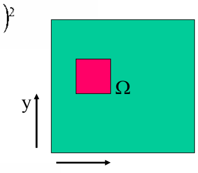
Lucas-Kanade: Integrate over a Patch

Assume a single velocity for all pixels within an image patch

$$E(u,v) = \sum_{x,y\in\Omega}\left(I_x(x,y)u + I_y(x,y)v + I_t\right)^2$$

$$\frac{dE(u,v)}{du} = \sum 2I_x\left(I_x u + I_y v + I_t\right) = 0$$

$$\frac{dE(u,v)}{dv} = \sum 2I_y\left(I_x u + I_y v + I_t\right) = 0$$



Solve with:

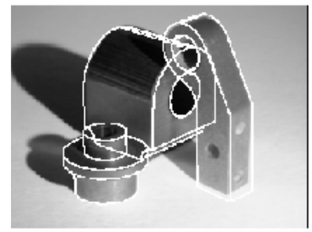
$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

On the LHS: sum of the 2x2 outer product tensor of the gradient vector

$$\left(\sum \nabla I \nabla I^T\right)\vec{U} = -\sum \nabla I I_t$$

Recognition

Recognition

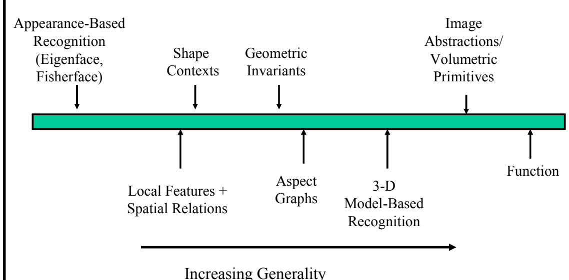


Given a database of objects and an image determine what, if any of the objects are present in the image.

Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional
- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability (perspective, orthographic projection)
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial
- Clutter in background -> false positives

A Rough Recognition Spectrum

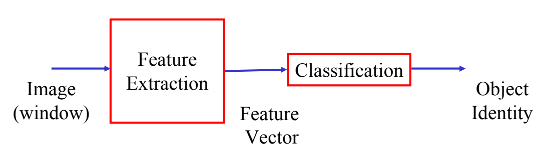


Appearance-Based Vision:

A Pattern Classification Viewpoint

1. Feature Space + Nearest Neighbor
2. Dimensionality Reduction
3. Bayesian Classification
4. Appearance Manifolds

Sketch of a Pattern Recognition Architecture



Example: Face Detection

- Scan window over image.
- Classify window as either:
  - Face
  - Non-face

Sketch of a Pattern Recognition Architecture

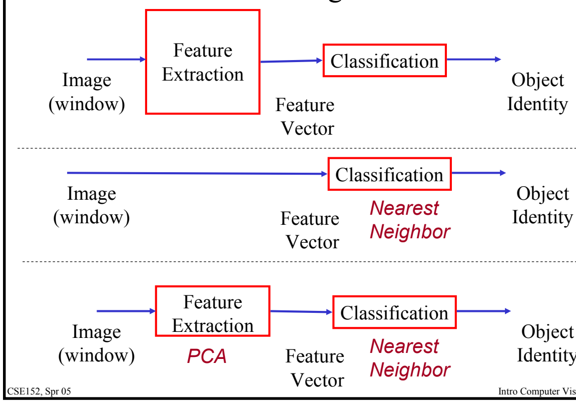
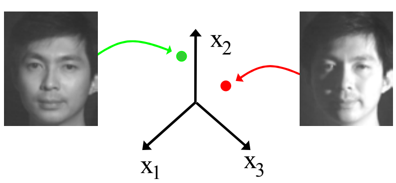


Image as a Feature Vector

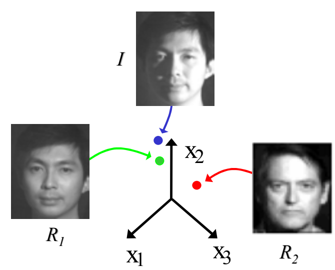


- Consider an n-pixel image (window) to be a point in an n-dimensional space, $\mathbf{x} \in \mathbf{R}^n$.
- Each pixel value is a coordinate of **x**.

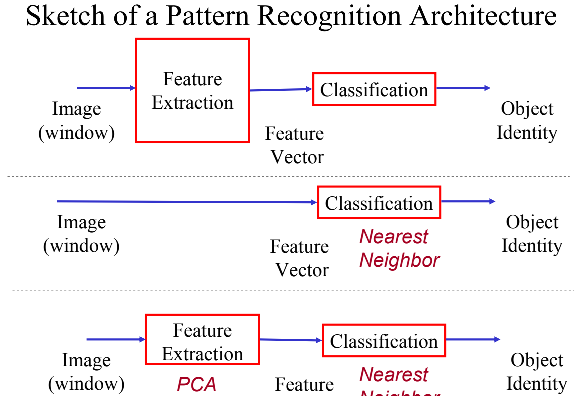
Simplest Recognition Scheme

- R_j is an image.
- $c(R_j, I)$ is Euclidean distance.

⇨ **Nearest Neighbor Classifier**



Sketch of a Pattern Recognition Architecture



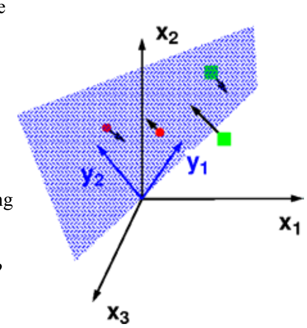
Dimensionality Redcution: Linear Projection

- An n-pixel image $x \in \mathbf{R}^n$ can be projected to a low-dimensional feature space $y \in \mathbf{R}^m$ by

$$y = Wx$$

where W is an m by n matrix.

- Recognition is performed using nearest neighbor in $\mathbf{R}^m$.
- How do we choose a good W?



Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of n feature vectors $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ in $\mathbb{R}^d$. Write

$$\boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{x}_i$$

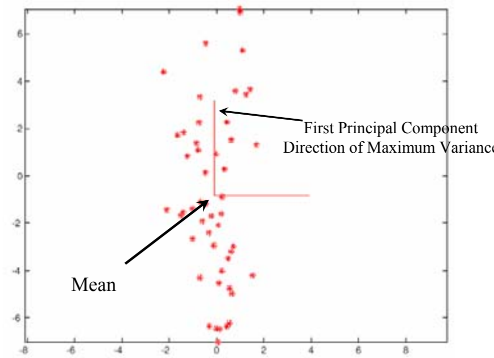
$$\Sigma = \frac{1}{n-1}\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

The unit eigenvectors of Σ — which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the k-dimensional set of linear features that preserves the most variance.

Algorithm 22.5: *Principal components analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.*

Some details: Use Singular value decomposition, "trick" described in appendix of text to compute basis when n<<d



Eigenfaces

- Modeling
  1. Given a collection of n labeled training images,
  2. Compute mean image and covariance matrix.
  3. Compute k Eigenvectors (note that these are images) of covariance matrix corresponding to k largest Eigenvalues. (Or perform using SVD!!)
  4. Project the training images to the k-dimensional Eigenspace.
- Recognition
  1. Given a test image, project to Eigenspace.
  2. Perform classification to the projected training images.

Eigenfaces: Training Images

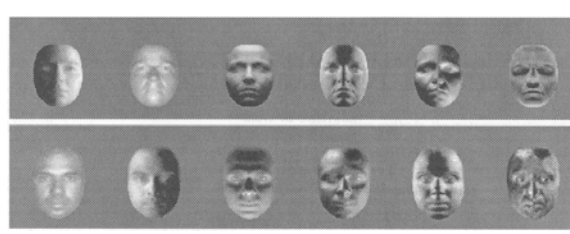
[Turk, Pentland 91

Eigenfaces

Mean Image

Basis Images

Basis Images for Variable Lighting



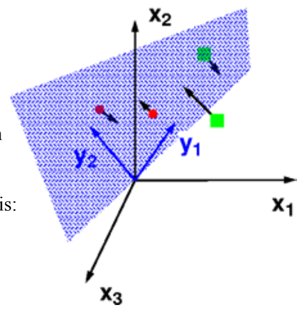
Projection, and reconstruction

• An n-pixel image $\boldsymbol{x} \in \mathbf{R}^n$ can be projected to a low-dimensional feature space $\boldsymbol{y} \in \mathbf{R}^m$ by

$$\boldsymbol{y} = W\boldsymbol{x}$$

• From $\boldsymbol{y} \in \mathbf{R}^m$, the reconstruction of the point is $\mathbf{W}^T\mathbf{y}$

• The error of the reconstruction is: $\|\mathbf{x}-\mathbf{W}^T\mathbf{W}\mathbf{x}\|$



Reconstruction using Eigenfaces

• Given image on left, project to Eigenspace, then reconstruct an image (right).

Face detection using "distance to face space"

• Scan a window ω across the image, and classify the window as face/not face as follows:

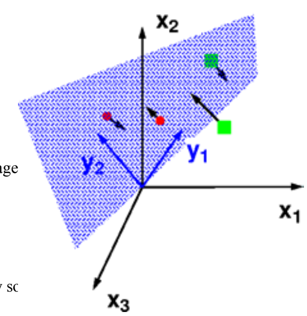
• Project window to subspace, and reconstruct as described earlier.

•Compute distance between ω and reconstruction.

•Local minima of distance over all image locations less than some treshold are taken as locations of faces.

•Repeat at different scales.

•Possibly normalize windows intensity sc that $|\omega| = 1$.



And important footnote: Don't really implement PCA this way!

Why?

1. How big is Σ?
   - n by n where n is the number of pixels in an image!!
2. You only need the first k Eigenvectors

Singular Value Decomposition

- Any m by n matrix A may be factored such that

$$A = U\Sigma V^T$$

[m x n] = [m x m][m x n][n x n]

- U: m by m, orthogonal matrix
  – Columns of $\mathbf{U}$ are the eigenvectors of AA^T
- V: n by n, orthogonal matrix,
  – columns are the eigenvectors of A^TA
- **Σ:** m by n, diagonal with non-negative entries (σ_1, σ_2, …, σ_s) with s=min(m,n) are called the called the singular values
  – **Singular values are the square roots of eigenvalues of both AA^T and A^TA *& Columns of U are corresponding Eigenvectors!!***
  – *Result of SVD algorithm:* $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$

SVD Properties

- In Matlab [u s v] = svd(A), and you can verify that: A=u*s*v'
- r=Rank(A) = # of non-zero singular values.
- U, V give us orthonormal bases for the subspaces of A:
  – 1st r columns of U: Column space of A
  – Last m - r columns of U: Left nullspace of A
  – 1st r columns of V: Row space of A
  – 1st n - r columns of V: Nullspace of A
- *For d<= r, the first d column of U provide the best d-dimensional basis for columns of A in least squares sense.*

Performing PCA with SVD

- Singular values of A are the square roots of eigenvalues of both AA^T and A^TA & Columns of U are corresponding Eigenvectors
- And $\sum_{i=1}^{n} a_i a_i^T = [a_1 \; a_2 \; \cdots \; a_n][a_1 \; a_2 \; \cdots \; a_n]^T = AA^T$
- Covariance matrix is:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T$$

- So, ignoring 1/n subtract mean image μ from each input image, create data matrix, and perform (thin) SVD on the data matrix.

Thin SVD

- Any m by n matrix A may be factored such that

$$A = U\Sigma V^T$$

[m x n] = [m x m][m x n][n x n]

- If m>n, then one can view Σ as:

$$\begin{bmatrix} \Sigma' \\ 0 \end{bmatrix}$$

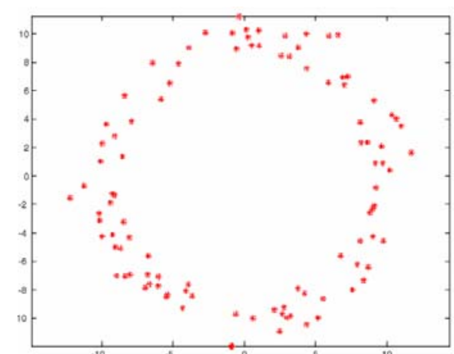
- Where $\Sigma'=\text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$ with s=min(m,n), and lower matrix is (n-m by m) of zeros.
- Alternatively, you can write:

$$A = U'\Sigma' V^T$$

- In Matlab, thin SVD is:[U S V] = svds(A)

This is what you should use!!

Alternative projections



Fisherfaces: Class specific linear projection

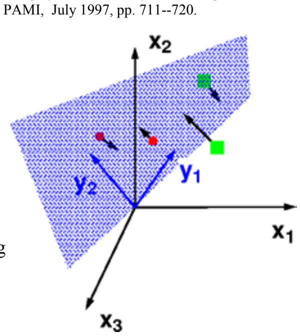
P. Belhumeur, J. Hespanha, D. Kriegman, *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*, PAMI, July 1997, pp. 711--720.

- An n-pixel image $x \in \mathbf{R}^n$ can be projected to a low-dimensional feature space $y \in \mathbf{R}^m$ by

$$y = Wx$$

where W is an n by m matrix.

- Recognition is performed using nearest neighbor in $\mathbf{R}^m$.
- How do we choose a good W?



PCA & Fisher's Linear Discriminant

- Between-class scatter

$$S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$

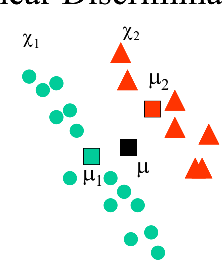
- Within-class scatter

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

- Total scatter

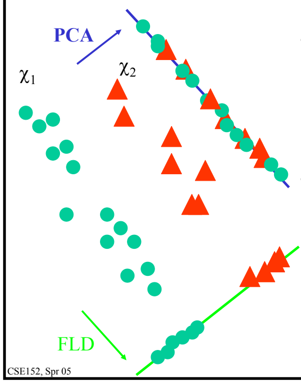
$$S_T = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu)(x_k - \mu)^T = S_B + S_W$$

- Where
  - c is the number of classes
  - μ_i is the mean of class χ_i
  - $|\chi_i|$ is number of samples of χ_i.



- If the data points are projected by y=Wx and scatter of points is S, then the scatter of the projected points is W^TSW

PCA & Fisher's Linear Discriminant



- PCA (Eigenfaces)

$$W_{PCA} = \arg\max_W \left| W^T S_T W \right|$$

Maximizes projected total scatter

- Fisher's Linear Discriminant

$$W_{fld} = \arg\max_W \frac{\left| W^T S_B W \right|}{\left| W^T S_W W \right|}$$

Maximizes ratio of projected between-class to projected within-class scatter

Computing the Fisher Projection Matrix

$$W_{opt} = \arg\max_W \frac{\left| W^T S_B W \right|}{\left| W^T S_W W \right|} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \ldots \ \mathbf{w}_m] \quad (4)$$

where $\{\mathbf{w}_i \mid i = 1, 2, \ldots, m\}$ is the set of generalized eigenvectors of S_B and S_W corresponding to the m largest generalized eigenvalues $\{\lambda_i \mid i = 1, 2, \ldots, m\}$, i.e.,

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \quad i = 1, 2, \ldots, m$$

- The w_i are orthonormal
- There are at most *c*-1 non-zero generalized Eigenvalues, so m <= c-1
- Can be computed with *eig* function in Matlab

Computing the Fisher Projection Matrix

$$W_{opt} = \arg\max_W \frac{\left| W^T S_B W \right|}{\left| W^T S_W W \right|} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \ldots \ \mathbf{w}_m] \quad (4)$$

where $\{\mathbf{w}_i \mid i = 1, 2, \ldots, m\}$ is the set of generalized eigenvectors of S_B and S_W corresponding to the m largest generalized eigenvalues $\{\lambda_i \mid i = 1, 2, \ldots, m\}$, i.e.,

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \quad i = 1, 2, \ldots, m$$

- The w_i are orthonormal
- There are at most *c*-1 non-zero generalized Eigenvalues, so m <= c-1
- Can be computed with *eig* in Matlab

Fisherfaces

$$W = W_{fld} W_{PCA}$$

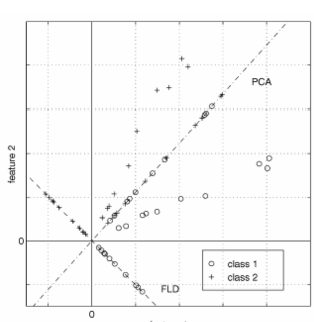
$$W_{PCA} = \arg\max_W \left| W^T S_T W \right|$$

$$W_{fld} = \arg\max_W \frac{\left| W^T W_{PCA}^T S_B W_{PCA} W \right|}{\left| W^T W_{PCA}^T S_W W_{PCA} W \right|}$$

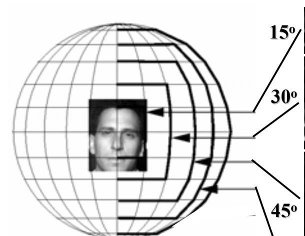
- Since S_W is rank N-c, project training set to subspace spanned by first *N-c* principal components of the training set.
- Apply FLD to *N-c* dimensional subspace yielding *c-1* dimensional feature space.

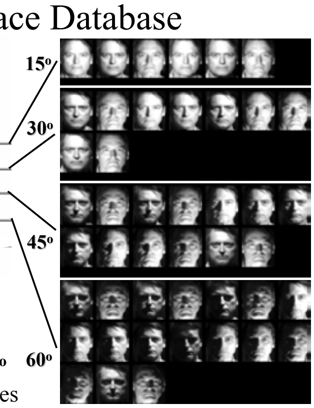
- Fisher's Linear Discriminant projects away the within-class variation (lighting, expressions) found in training set.
- Fisher's Linear Discriminant preserves the separability of the classes.

PCA vs. FLD



Harvard Face Database





- 10 individuals
- 66 images per person
- Train on 6 images at 15°
- Test on remaining images

Recognition Results: Lighting Extrapolation