

Recognition III

Introduction to Computer Vision
CSE 152
Lecture 19

CSE152, Spr 12

Intro Computer Vision

Announcements

- HW 4 has been posted. Due 06/08
- Lecture plan – Wrap up Recognition and on tomotion.
- Final exam: June 12, 8AM-11AM, cumulative

CSE152, Spr 12

Intro Computer Vision

Linear Subspaces & Linear Projection

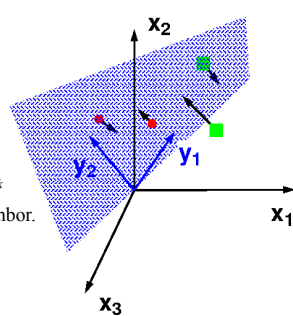
- An n -pixel image $x \in \mathbb{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbb{R}^k$ by

$$y = Wx$$

where W is an k by d matrix.

- Recognition is performed in $\mathbb{R}^k$ using, for example, nearest neighbor.
- How do we choose a good W ?

Example: Projecting from $\mathbb{R}^3$ to $\mathbb{R}^2$



CSE152, Spr 12

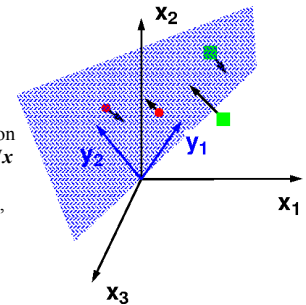
Intro Computer Vision

Distance to Linear Subspace

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

$$y = Wx$$

- From $y \in \mathbb{R}^k$, the reconstruction of the point in $\mathbb{R}^d$ is $W^T y = W^T W x$
- The error of the reconstruction, or the distance from x to the subspace spanned by W is:
 $\|x - W^T W x\|$



CSE152, Spr 12

Intro Computer Vision

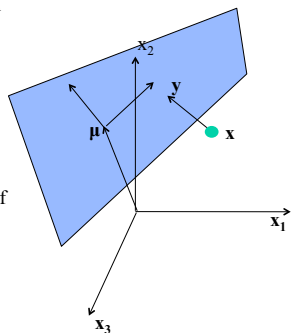
Distance to Affine Subspace (i.e., Distance to Face Space)

- Represented by mean vector μ and basis images W
- An n -pixel image $x \in \mathbb{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbb{R}^k$ by

$$y = W(x - \mu)$$

- From $y \in \mathbb{R}^k$, the reconstruction of the point in $\mathbb{R}^d$ is
 $W^T y + \mu = W^T W(x - \mu) + \mu$

- The error of the reconstruction, or the distance from x to the affine is:
 $\|x - W^T W(x - \mu) - \mu\| = \|(I - W^T W)(x - \mu)\|$



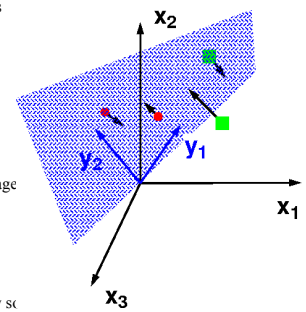
CSE152, Spr 12

Intro Computer Vision

Application 1:

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 threshold are taken as locations of faces.
 - Repeat at different scales.
 - Possibly normalize windows intensity so that $\|\omega\| = 1$.



CSE152, Spr 12

Intro Computer Vision

An important footnote:
We don't really implement PCA by constructing a covariance matrix!

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

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

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

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

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

CSE152, Spr 12

Intro Computer Vision

Singular Value Decomposition

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

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

$$[m \times n] = [m \times m][m \times n][n \times n]$$

- U : m by m , orthogonal matrix
 - Columns of U are the eigenvectors of AA^T
- V : n by n , orthogonal matrix,
 - columns are the eigenvectors of $A^T A$
- Σ : m by n , diagonal with non-negative entries ($\sigma_1, \sigma_2, \dots, \sigma_s$) with $s = \min(m, n)$ are called the singular values. SVD algorithm produces sorted singular values: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_s$

Important property

- Singular values are the square roots of Eigenvalues of both AA^T and $A^T A$ & Columns of U are corresponding Eigenvectors!!

CSE152, Spr 12

Intro Computer Vision

Performing PCA with SVD

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

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

- So, ignoring $1/n$ subtract mean image μ from each input image, create a $d \times n$ data matrix, and perform thin SVD on the data matrix. $D = [\mathbf{x}_1 - \mu \mid \mathbf{x}_2 - \mu \mid \dots \mid \mathbf{x}_n - \mu]$

CSE152, Spr 12

Intro Computer Vision

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

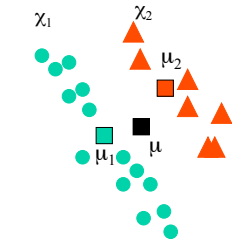
$$S_W = \sum_{i=1}^c \sum_{x_i \in \chi_i} (x_i - \mu_i)(x_i - \mu_i)^T$$

- Total scatter

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

- 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 .

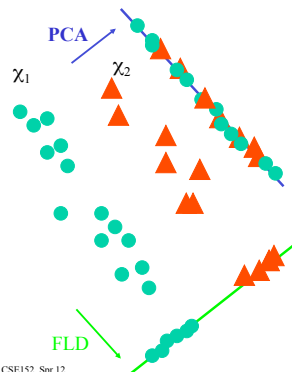


If the data points x_i are projected by $y_i = Wx_i$ and the scatter of x_i is S , then the scatter of the projected points y_i is $W^T S W$

CSE152, Spr 12

Intro Computer Vision

PCA & Fisher's Linear Discriminant



- PCA (Eigenfaces)

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

Maximizes projected total scatter

- Fisher's Linear Discriminant

$$W_{FLD} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

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

CSE152, Spr 12

Intro Computer Vision

Computing the Fisher Projection Matrix

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

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

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

- The $\mathbf{w}_i$ are orthonormal
- There are at most $c-1$ non-zero generalized Eigenvalues, so $m \leq c-1$
- Can be computed with *eig* in Matlab

CSE152, Spr 12

Intro Computer Vision

Fisherfaces

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

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

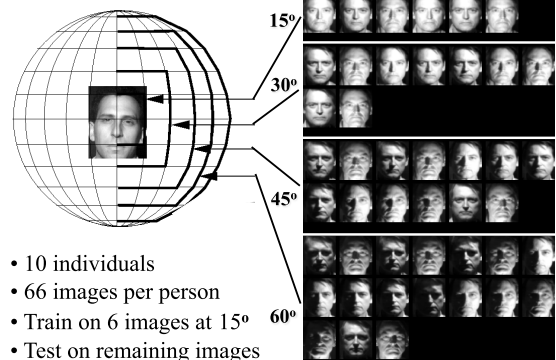
$$W_{fld} = \arg \max_W \frac{|W^T W_{PCA}^T S_B W_{PCA} W|}{|W^T W_{PCA}^T S_W W_{PCA} W|}$$

- 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.

CSE152, Spr 12

Intro Computer Vision

Harvard Face Database

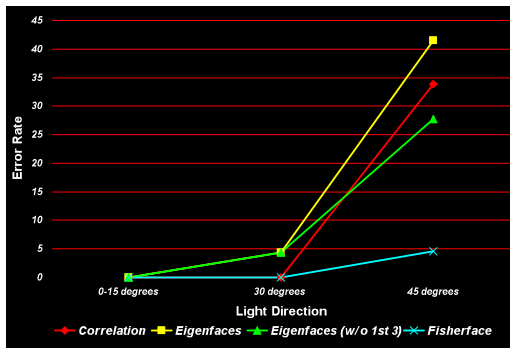


- 10 individuals
- 66 images per person
- Train on 6 images at 15°
- Test on remaining images

CSE152, Spr 12

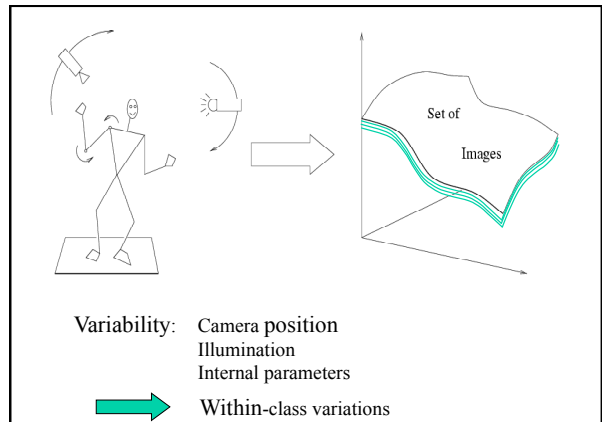
Intro Computer Vision

Recognition Results: Lighting Extrapolation



CSE152, Spr 12

Intro Computer Vision



CSE152, Spr 12

Intro Computer Vision

Appearance manifold approach

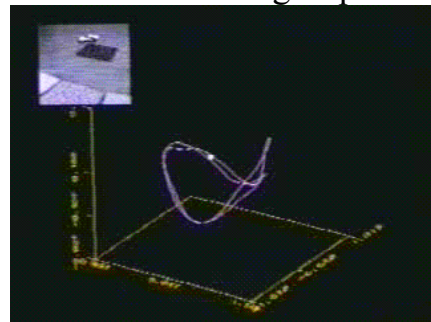
- for every object (Nayar et al. '96)
 1. sample the set of viewing conditions
 2. Crop & scale images to standard size
 3. Use as feature vector
- apply a PCA over all the images
- keep the dominant PCs
- Set of views for one object is represented as a manifold in the projected space
- Recognition: What is nearest manifold for a given test image?



CSE152, Spr 12

Intro Computer Vision

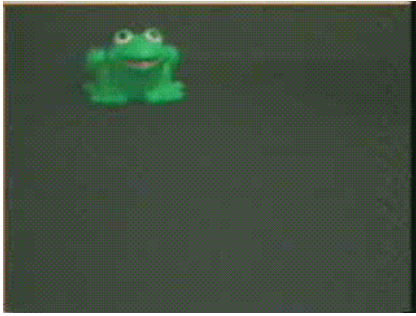
Parameterized Eigenspace



CSE152, Spr 12

Intro Computer Vision

Recognition



CSE152, Spr 12

Intro Computer Vision

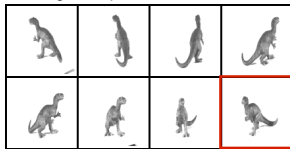
Recognition in Cluttered Scenes Interest Points + Feature Descriptors + Relations

CSE152, Spr 12

Intro Computer Vision

Example

Training examples



Test image



CSE152, Spr 12

Intro Computer Vision

Matching using Local Image features Simple approach

- Detect corners in image (e.g. Harris corner detector).
- Represent neighborhood of corner by a feature vector produced by Gabor Filters, K-jets, SIFT features, etc.
- Modeling: Given an training image of an object w/o clutter, detect corners, compute feature descriptors, store these.
- Recognition time: Given test image with possible clutter, detect corners and compute features. Find models with same feature descriptors (hashing) and vote.

CSE152, Spr 12

Intro Computer Vision

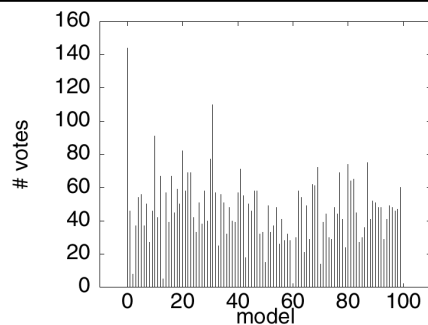


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

CSE152, Spr 12

Intro Computer Vision

Employ spatial relations

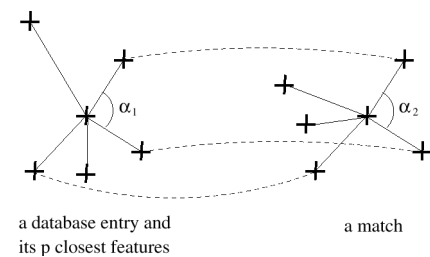


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

CSE152, Spr 12

Intro Computer Vision

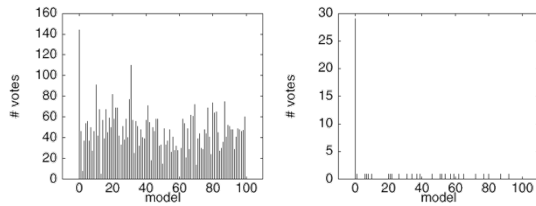


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

CSE152, Spr 12

Intro Computer Vision

Even without shading, shape reveals a lot



CSE152, Spr 12

Intro Computer Vision

Motion

Introduction to Computer Vision
CSE 152
Lecture 19-b

CSE152, Spr 12

Intro Computer Vision

Motion

Some problems of motion

1. Correspondence: Where have elements of the image moved between image frames
2. Reconstruction: Given correspondence, what is 3-D geometry of scene
3. Ego Motion: How has the camera moved.
4. Segmentation: What are regions of image corresponding to different moving objects
5. Tracking: Where have objects moved in the image? related to correspondence and segmentation.

Variations:

- Small motion (video),
- Wide-baseline (multi-view)

CSE152, Spr 12

Intro Computer Vision

Structure-from-Motion (SFM)

Goal: Take as input two or more images or video w/o any information on camera position/motion, and estimate camera position and 3-D structure of scene.

Two Approaches

1. Discrete motion (wide baseline)
 1. Orthographic (affine) vs. Perspective
 2. Two view vs. Multi-view
 3. Calibrated vs. Uncalibrated
2. Continuous (Infinitesimal) motion

CSE152, Spr 12

Intro Computer Vision

Discrete Motion: Some Counting

Consider M images of N points, how many unknowns?

1. Camera locations: Affix coordinate system to location of first camera location: $(M-1)*6$ Unknowns
2. 3-D Structure: $3*N$ Unknowns
3. Can only recover structure and motion up to scale. Why?

Total number of unknowns: $(M-1)*6+3*N-1$

Total number of measurements: $2*M*N$

Solution is possible when $(M-1)*6+3*N-1 \leq 2*M*N$

$M=2 \rightarrow N \geq 5$

$M=3 \rightarrow N \geq 4$

CSE152, Spr 12

Intro Computer Vision

Epipolar Constraint: Calibrated Case

$$\vec{Op} \cdot [\vec{OO'} \times \vec{Op'}] = 0 \Rightarrow \mathbf{p} \cdot [\mathbf{t} \times (\mathbf{R}\mathbf{p}')] = 0 \quad \text{with} \quad \begin{cases} \mathbf{p} = (u, v, 1)^T \\ \mathbf{p}' = (u', v', 1)^T \\ \mathbf{M} = (\mathbf{Id} \quad \mathbf{0}) \\ \mathbf{M}' = (\mathbf{R}^T, -\mathbf{R}^T \mathbf{t}) \end{cases}$$

Essential Matrix
(Longuet-Higgins, 1981) $\leftarrow \mathbf{p}^T \mathbf{E} \mathbf{p}' = 0$ with $\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$

where $[\mathbf{t}]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$

CSE152, Spr 12 Intro Computer Vision

The Eight-Point Algorithm (Longuet-Higgins, 1981)

Let F denote the Essential Matrix here

$$(u, v, 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \Rightarrow (uu', uv', u, vu', vv', v, u', v', 1) \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

Set F_{33} to 1

$$\begin{pmatrix} u_1 u'_1 & u_1 v'_1 & u_1 & v_1 u'_1 & v_1 v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2 u'_2 & u_2 v'_2 & u_2 & v_2 u'_2 & v_2 v'_2 & v_2 & u'_2 & v'_2 & 1 \\ u_3 u'_3 & u_3 v'_3 & u_3 & v_3 u'_3 & v_3 v'_3 & v_3 & u'_3 & v'_3 & 1 \\ u_4 u'_4 & u_4 v'_4 & u_4 & v_4 u'_4 & v_4 v'_4 & v_4 & u'_4 & v'_4 & 1 \\ u_5 u'_5 & u_5 v'_5 & u_5 & v_5 u'_5 & v_5 v'_5 & v_5 & u'_5 & v'_5 & 1 \\ u_6 u'_6 & u_6 v'_6 & u_6 & v_6 u'_6 & v_6 v'_6 & v_6 & u'_6 & v'_6 & 1 \\ u_7 u'_7 & u_7 v'_7 & u_7 & v_7 u'_7 & v_7 v'_7 & v_7 & u'_7 & v'_7 & 1 \\ u_8 u'_8 & u_8 v'_8 & u_8 & v_8 u'_8 & v_8 v'_8 & v_8 & u'_8 & v'_8 & 1 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Solve For F $\rightarrow$ **Solve For R and t**

CSE152, Spr 12 Intro Computer Vision

Sketch of Two View SFM Algorithm

Input: Two images

1. Detect feature points
2. Find 8 matching feature points (easier said than done, but usually done with RANSAC Algorithm)
3. Compute the Essential Matrix E using Normalized 8-point Algorithm
4. Compute R and T (recall that $\mathbf{E} = \mathbf{R}\mathbf{S}$ where S is skew symmetric matrix)
5. Perform stereo matching using recovered epipolar geometry expressed via E.
6. Reconstruct 3-D geometry of corresponding points.

CSE152, Spr 12 Intro Computer Vision

Feature points

Select strongest features (e.g. 1000/image)

CSE152, Spr 12 Intro Computer Vision

Feature matching

Evaluate normalized cross correlation (or sum of squared differences) for all features with similar coordinates

$$\text{e.g. } (x', y') \in \left[x - \frac{w}{10}, x + \frac{w}{10} \right] \times \left[y - \frac{h}{10}, y + \frac{h}{10} \right]$$

Keep mutual best matches
Still many wrong matches!

CSE152, Spr 12 Intro Computer Vision

Comments

- Greedy Algorithm:
 - Given feature in one image, find best match in second image irrespective of other matches.
 - OK for small motions, little rotation, small search window
- Otherwise
 - Must compare descriptor over rotation
 - Can't consider $O(n^8)$ potential pairings (way too many), so
 - Manual correspondence (e.g., façade, photogrammetry).
 - use random sampling (RANSAC)
 - More descriptive features (line segments, SIFT, larger regions, color).
 - Use video sequence to track, but perform SFM w/ first and last image.

CSE152, Spr 12 Intro Computer Vision