

Recognition

Computer Vision I

CSE252A

Lecture 20

CS252A, Winter 2006

Computer Vision I

Announcements

- HW 4 Due Date extended to Friday
- Final Exam – next Friday, 3:00-6:00 here.

Q & A session on wed, 1:00-2:30. Place TBA.

CS252A, Winter 2006

Computer Vision I

Final Exam

- Closed book
- One cheat sheet
 - Single piece of paper, handwritten, no photocopying, no physical cut & paste.
- What to study
 - Basically material presented in class, and supporting material from text
 - If it was in text, but NEVER mentioned in class, it is very unlikely to be on the exam
- Question style:
 - Short answer
 - Some longer problems to be worked out.

CS252A, Winter 2006

Computer Vision I

Example: Face Detection

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



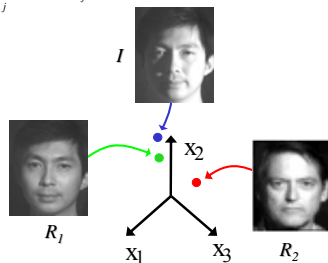
CS252A, Winter 2006

Computer Vision I

Nearest Neighbor Classifier

$\{ R_j \}$ are set of training images.

$$ID = \arg \min_j \text{dist}(R_j, I)$$



CS252A, Winter 2006

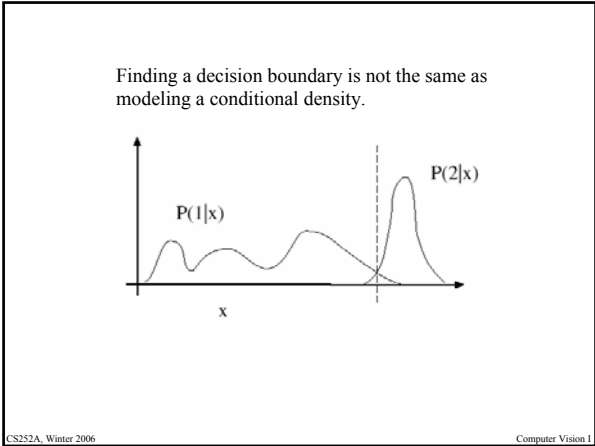
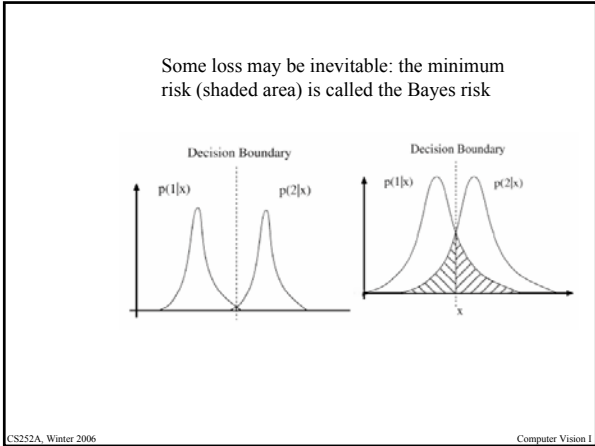
Computer Vision I

Bayesian Classification

Blackboard

CS252A, Winter 2006

Computer Vision I



Example: known distributions

$$p(\underline{x}|k) = \left(\frac{1}{2\pi}\right)^{-D/2} |\underline{\Sigma}|^{-1/2} \exp\left[-\frac{1}{2}(\underline{x} - \underline{\mu}_k)^T \underline{\Sigma}^{-1}(\underline{x} - \underline{\mu}_k)\right]$$

- Assume normal class densities, p-dimensional measurements with common (known) covariance and different (known) means
- Class priors are π_k
- Can ignore a common factor in posteriors - important; posteriors are then:

$$p(k | x) \propto (\pi_k) \left(\frac{1}{2\pi}\right)^{-D/2} |\underline{\Sigma}|^{-1/2} \exp\left[-\frac{1}{2}(\underline{x} - \underline{\mu}_k)^T \underline{\Sigma}^{-1}(\underline{x} - \underline{\mu}_k)\right]$$

CS252A, Winter 2006 Computer Vision I

- Classifier boils down to: choose class that minimizes:

$$\delta(\underline{x}, \underline{\mu}_k) - 2 \log \pi_k$$

where

Mahalanobis distance $\delta(\underline{x}, \underline{\mu}_k) = [(\underline{x} - \underline{\mu}_k)^T \underline{\Sigma}^{-1}(\underline{x} - \underline{\mu}_k)]^{1/2}$

because covariance is common, this simplifies to sign of a linear expression (i.e. Voronoi diagram in 2D for $\Sigma=I$)

CS252A, Winter 2006 Computer Vision I

Plug-in classifiers

- Assume that distributions have some parametric form - now estimate the parameters from the data.
- Common:
 - assume a normal distribution with shared covariance, different means; use usual estimates
 - ditto, but different covariances; ditto
- Issue: parameter estimates that are “good” may not give optimal classifiers.

CS252A, Winter 2006 Computer Vision I

Example: Face Detection

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

CS252A, Winter 2006 Computer Vision I

Eigenfaces: linear projection

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

$$y = Wx$$
 where W is an n by m matrix.
- Recognition is performed using nearest neighbor in \mathbb{R}^m .
- How do we choose a good W ?

CS252A, Winter 2006 Computer Vision I

Eigenfaces: Principal Component Analysis (PCA)

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

$$\mu = \frac{1}{n} \sum_i x_i$$

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

The unit eigenvectors of Σ — which we write as v_1, v_2, \dots, v_d , where the order is given by the size of the eigenvalue and v_1 has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{v_1, \dots, v_k\}$ 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 text to compute basis when $n \ll d$

CS252A, Winter 2006 Computer Vision I

Singular Value Decomposition

[Repeated from Lecture 7]

- 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 called the singular values
 - Singular values are the square roots of eigenvalues of both AA^T and $A^T A$ & Columns of U are corresponding Eigenvectors
 - Result of SVD algorithm: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_s$

CS252A, Winter 2006 Computer Vision I

SVD Properties

- In Matlab $[u \ s \ v] = \text{svd}(A)$, and you can verify that: $A = u * s * v'$
- $r = \text{Rank}(A) = \#$ 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 $d \leq r$, the first d column of U provide the best d -dimensional basis for columns of A in least squares sense.

CS252A, Winter 2006 Computer Vision I

Thin SVD

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

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

$$[m \times n] = [m \times m][m \times n][n \times n]$$
- If $m > n$, then one can view Σ as:

$$\begin{bmatrix} \Sigma' \\ 0 \end{bmatrix}$$
- Where $\Sigma' = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_s)$ 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: $[U \ S \ V] = \text{svds}(A)$

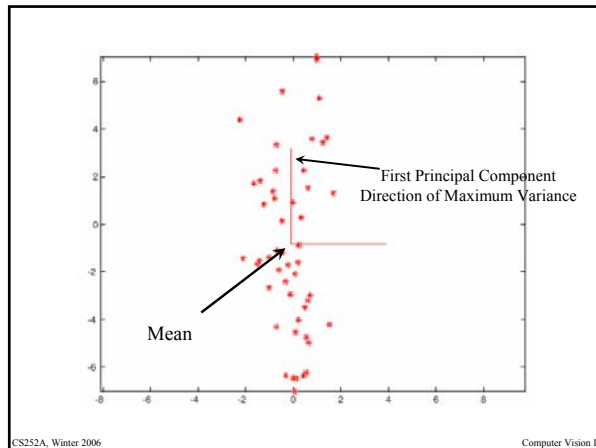
CS252A, Winter 2006 Computer Vision I

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 $\sum_{i=1}^n a_i a_i^T = [a_1 \ a_2 \ \dots \ a_n] [a_1 \ a_2 \ \dots \ a_n]^T = AA^T$
- 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 data matrix, and perform thin SVD on the data matrix.

CS252A, Winter 2006 Computer Vision I



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.

CS252A, Winter 2006 Computer Vision I

Eigenfaces: Training Images

[Turk, Pentland 01

CS252A, Winter 2006 Computer Vision I

Eigenfaces

Mean Image

Basis Images

CS252A, Winter 2006 Computer Vision I

Eigenfaces for a single individual, observed under variable lighting

CS252A, Winter 2006 Computer Vision I

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 $|\omega| = 1$.

CS252A, Winter 2006 Computer Vision I

Underlying assumptions

- Background is not cluttered (or else only looking at interior of object)
- Lighting in test image is similar to that in training image.
- No occlusion
- Size of training image (window) same as window in test image.

CS252A, Winter 2006

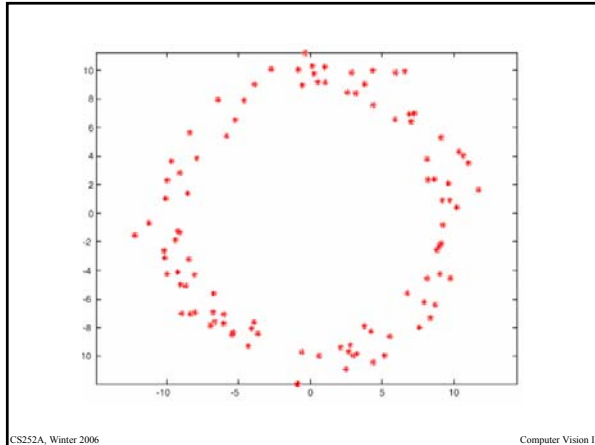
Computer Vision I

Difficulties with PCA

- Projection may suppress important detail
 - smallest variance directions may not be unimportant
- Method does not take discriminative task into account
 - typically, we wish to compute features that allow good discrimination
 - not the same as largest variance or minimizing reconstruction error.

CS252A, Winter 2006

Computer Vision I



CS252A, Winter 2006

Computer Vision I

Fisherfaces: Class specific linear projection

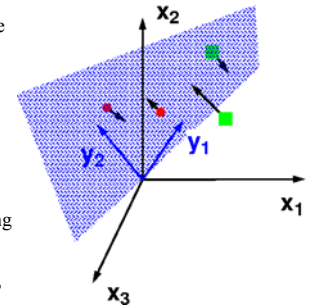
P. Belhumeur, J. Hespanha, D. Kriegman, *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*, PAMI, July 1997, pp. 711–720.

- 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 ?



CS252A, Winter 2006

Computer Vision I

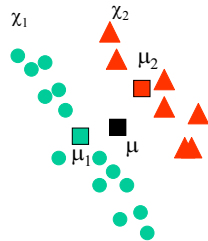
PCA & Fisher's Linear Discriminant

- 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 are projected by $y=Wx$ and scatter of points is S , then the scatter of the projected points is $W^T S W$

CS252A, Winter 2006

Computer Vision I

PCA & Fisher's Linear Discriminant

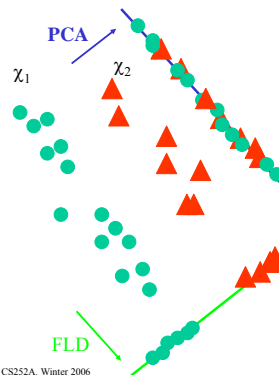
- PCA (Eigenfaces)

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

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



CS252A, Winter 2006

Computer Vision I

Computing the Fisher Projection Matrix

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

$$= [w_1 \ w_2 \ \dots \ w_m] \quad (4)$$

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

$$S_B w_i = \lambda_i S_W w_i, \quad i = 1, 2, \dots, m$$

- The 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

CS252A, Winter 2006

Computer Vision I

Fisherfaces

$$W = W_{fd} W_{PCA}$$

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

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

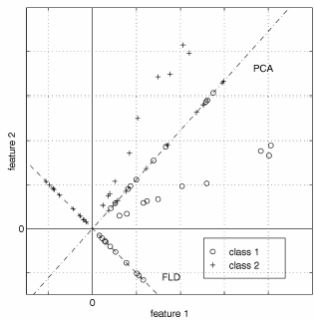
- Since S_W is rank $N-c$, project training set to subspace spanned by first $N-c$ principal components of the training set.
- Apply FLD to $N-c$ dimensional subspace yielding $c-1$ dimensional feature space.

- 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.

CS252A, Winter 2006

Computer Vision I

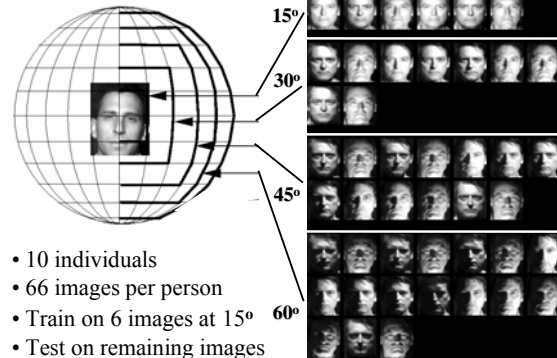
PCA vs. FLD



CS252A, Winter 2006

Computer Vision I

Harvard Face Database

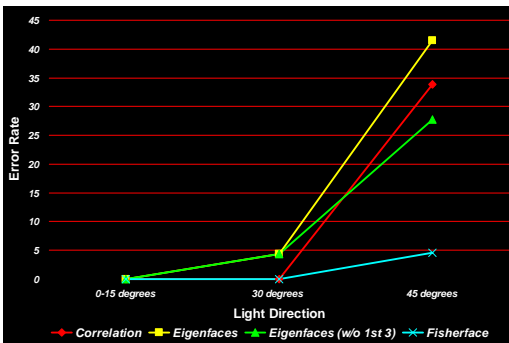


- 10 individuals
- 66 images per person
- Train on 6 images at 15°
- Test on remaining images

CS252A, Winter 2006

Computer Vision I

Recognition Results: Lighting Extrapolation



CS252A, Winter 2006

Computer Vision I

Support Vector Machines

- Bayes classifiers and generative approaches in general try to model of the posterior, $p(\omega | \mathbf{x})$
- Instead, try to obtain the decision boundary directly
 - potentially easier, because we need to encode only the geometry of the boundary, not any irrelevant wiggles in the posterior.
 - Not all points affect the decision boundary

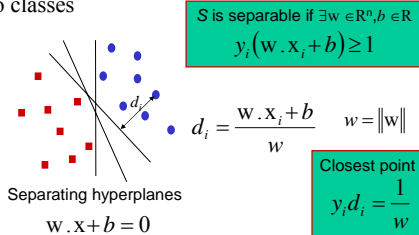
CS252A, Winter 2006

[From Marc Pollefeys]

Computer Vision I

Support Vector Machines

- Set S of points $x_i \in \mathbb{R}^n$, each x_i belongs to one of two classes $y_i \in \{-1, 1\}$
- The goal is to find a hyperplane that divides S in these two classes

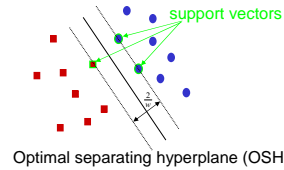


CS252A, Winter 2006 [From Marc Pollefyfes] Computer Vision I

Support Vector Machines

- Optimal separating hyperplane maximizes $\frac{1}{w}$

Problem 1:
 Minimize $\frac{1}{2} w \cdot w$
 Subject to $y_i(w \cdot x_i + b) \geq 1, i = 1, 2, \dots, N$



CS252A, Winter 2006 [From Marc Pollefyfes] Computer Vision I

Solve using Lagrange multipliers

- Lagrangian

$$L(w, b, \alpha) = \frac{1}{2} w \cdot w - \sum_{i=1}^N \alpha_i \{y_i(w \cdot x_i + b) - 1\}$$

– at solution

$$\frac{\partial L}{\partial b} = \sum_{i=1}^N y_i \alpha_i = 0$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^N \alpha_i y_i x_i = 0$$

$$L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

– therefore

$$\alpha \geq 0$$

CS252A, Winter 2006 [From Marc Pollefyfes] Computer Vision I

Decision function

- Once w and b have been computed the classification decision for input x is given by

$$f(x) = \text{sign}(w \cdot x + b)$$

- Note that the globally optimal solution can always be obtained (convex problem)

CS252A, Winter 2006 [From Marc Pollefyfes] Computer Vision I

Non-linear SVMs

- Non-linear separation surfaces can be obtained by non-linearly mapping the data to a high dimensional space and then applying the linear SVM technique
- Note that data only appears through vector product
- Need for vector product in high-dimension can be avoided by using Mercer kernels:

$$K(x, x_i) = \Phi(x_i) \Phi(x_i)$$

e.g. $K(x, y) = (x \cdot y)^p$ (Polynomial kernel)

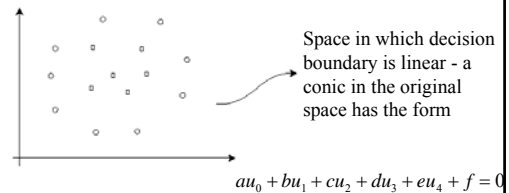
$$K(x, y) = (x \cdot y)^2 = x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_2^2 + x_2^2 y_1^2$$

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$
 (Radial Basis Function)

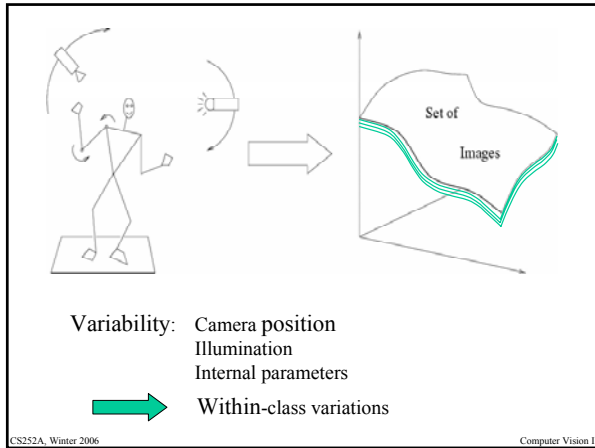
$$K(x, y) = \tanh(\kappa x \cdot y - \delta)$$
 (Sigmoidal function)

CS252A, Winter 2006 [From Marc Pollefyfes] Computer Vision I

$$(x, y) \rightarrow (x^2, xy, y^2, x, y) = (u_0, u_1, u_2, u_3, u_4)$$



CS252A, Winter 2006 [From Marc Pollefyfes] Computer Vision I

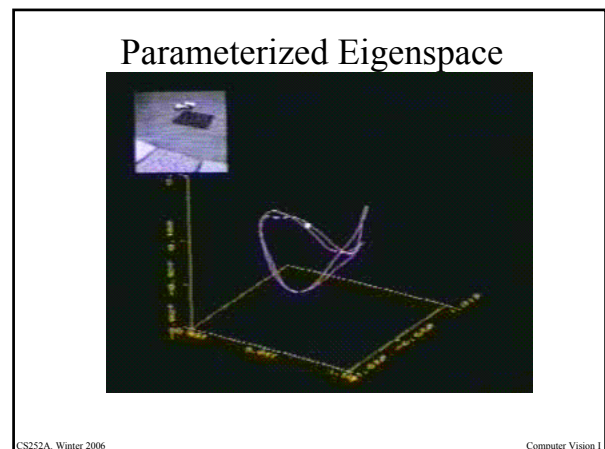
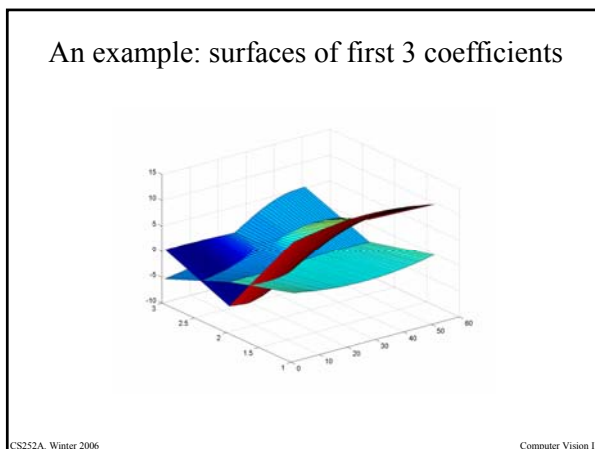
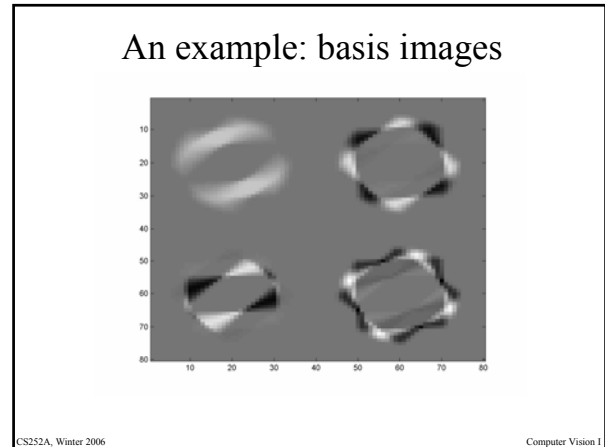
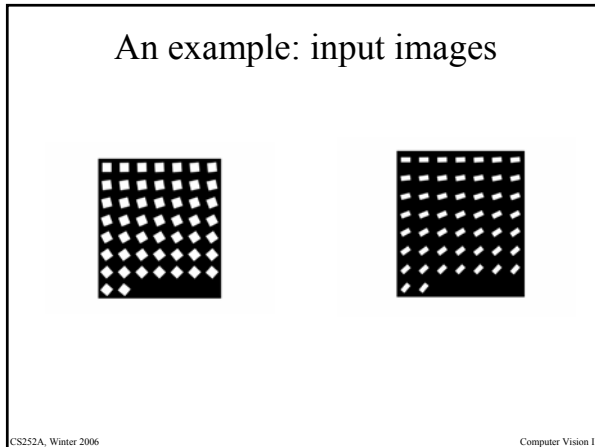


Appearance manifold approach

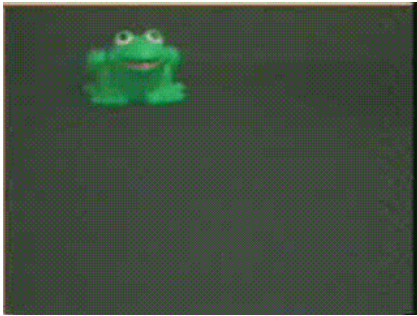
(Nayar et al. '96)

- for every object
 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?

CS252A, Winter 2006 Computer Vision I



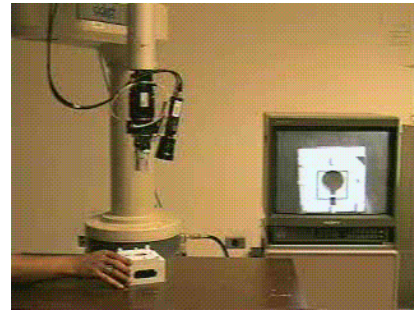
Recognition



CS252A, Winter 2006

Computer Vision I

Appearance-based vision for robot control



[Nayar, Nene, Murase 1994]

CS252A, Winter 2006

Computer Vision I

Limitations of these approaches

- Object must be segmented from background (How would one do this in non-trivial situations?)
- Occlusion?
- The variability (dimension) in images is large, so is sampling feasible?
- How can one generalize to classes of objects?

CS252A, Winter 2006

Computer Vision I

Appearance-Based Vision: Lessons

Strengths

- Posing the recognition metric in the image space rather than a derived representation is more powerful than expected.
- Modeling objects from many images is not unreasonable given hardware developments.
- The data (images) may provide a better representations than abstractions for many tasks.

CS252A, Winter 2006

Computer Vision I

Appearance-Based Vision: Lessons

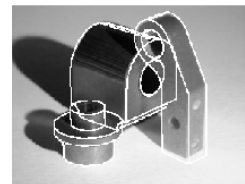
Weaknesses

- Segmentation or object detection is still an issue.
- To train the method, objects have to be observed under a wide range of conditions (e.g. pose, lighting, shape deformation).
- Limited power to extrapolate or generalize (abstract) to novel conditions.

CS252A, Winter 2006

Computer Vision I

Model-Based Vision

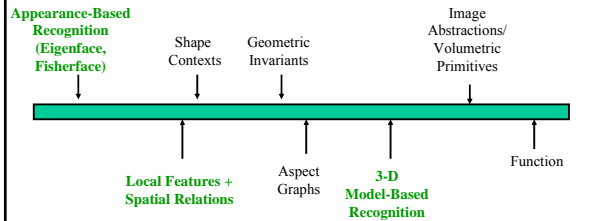


- Given 3-D models of each object
- Detect image features (often edges, line segments, conic sections)
- Establish correspondence between model & image features
- Estimate pose
- Consistency of projected model with image.

CS252A, Winter 2006

Computer Vision I

A Rough Recognition Spectrum



CS252A, Winter 2006

Computer Vision I

Recognition by Hypothesize and Test

- General idea
 - Hypothesize object identity and pose
 - Recover camera parameters (widely known as backprojection)
 - Render object using camera parameters
 - Compare to image
- Simplest approach
 - Construct a correspondence for all object features to every correctly sized subset of image points
 - These are the hypotheses
 - Expensive search, which is also redundant.
- Issues
 - where do the hypotheses come from?
 - How do we compare to image (verification)?

CS252A, Winter 2006

Computer Vision I

Pose consistency

- Correspondences between image features and model features are not independent.
- A small number of correspondences yields a camera matrix --- the others correspondences must be consistent with this.
- Strategy:
 - Generate hypotheses using small numbers of correspondences (e.g. triples of points for a calibrated perspective camera, etc., etc.)
 - Backproject and verify

CS252A, Winter 2006

Computer Vision I

```

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
    
```

CS252A, Winter 2006

Computer Vision I

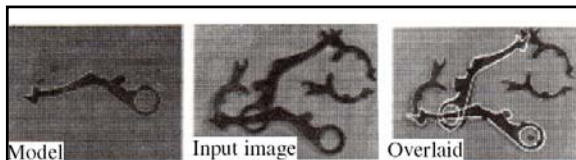


Figure from "Object recognition using alignment," D.P. Huttenlocher and S. Ullman, Proc. Int. Conf. Computer Vision, 1986, copyright IEEE, 1986

CS252A, Winter 2006

Computer Vision I

Voting on Pose

- Each model leads to many correct sets of correspondences, each of which has the same pose
 - Vote on pose, in an accumulator array
 - This is a hough transform, with all it's issues.

CS252A, Winter 2006

Computer Vision I

```

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

```

CS252A, Winter 2006

Computer Vision I

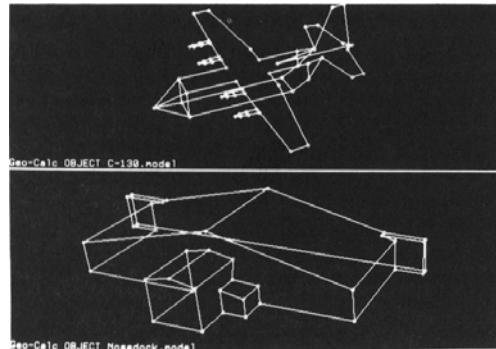


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

CS252A, Winter 2006

Computer Vision I

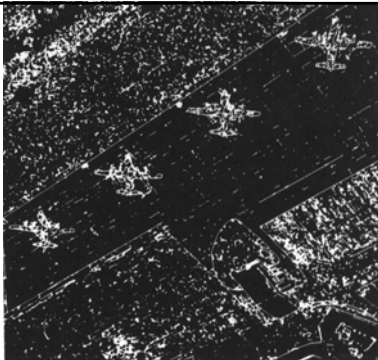


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

CS252A, Winter 2006

Computer Vision I

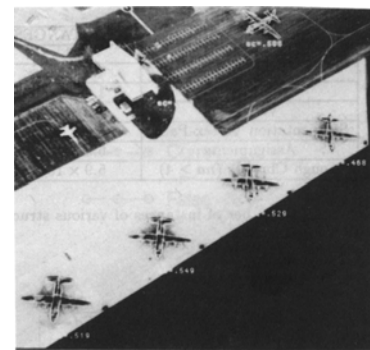


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

CS252A, Winter 2006

Computer Vision I

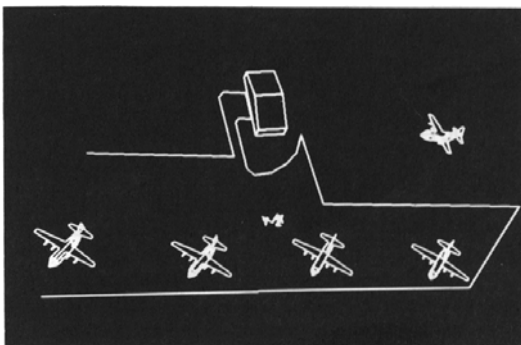


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

CS252A, Winter 2006

Computer Vision I

Invariance

- Properties or measures that are independent of some group of transformation (e.g., rigid, affine, projective, etc.)
- For example, under affine transformations:
 - Collinearity
 - Parallelism
 - Intersection
 - Distance ratio along a line
 - Angle ratios of tree intersecting lines
 - Affine coordinates

CS252A, Winter 2006

Computer Vision I

Invariance - 1

- There are geometric properties that are invariant to camera transformations
- Easiest case: view a plane object in scaled orthography.
- Assume we have three base points P_i ($i=1..3$) on the object
 - then any other point on the object can be written as

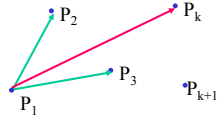
$$P_k = P_1 + \mu_{ka}(P_2 - P_1) + \mu_{kb}(P_3 - P_1)$$

- Now image points are obtained by multiplying by a plane affine transformation, so

$$p_k = AP_k$$

$$= A(P_1 + \mu_{ka}(P_2 - P_1) + \mu_{kb}(P_3 - P_1))$$

$$= p_1 + \mu_{ka}(p_2 - p_1) + \mu_{kb}(p_3 - p_1)$$



CS252A, Winter 2006

Computer Vision I

Geometric hashing

- Vote on identity and correspondence using invariants
 - Take hypotheses with large enough votes
- Building a table:
 - Take all triplets of points in on model image to be base points P_1, P_2, P_3 .
 - Take ever fourth point and compute μ 's
 - Fill up a table, indexed by μ 's, with
 - the base points and fourth point that yield those μ 's
 - the object identity

CS252A, Winter 2006

Computer Vision I

Algorithm 18.3: Geometric hashing: voting on identity and point labels

```

For all groups of three image points  $T(I)$ 
  For every other image point  $p$ 
    Compute the  $\mu$ 's from  $p$  and  $T(I)$ 
    Obtain the table entry at these values
    if there is one, it will label the three points in  $T(I)$ 
    with the name of the object
    and the names of these particular points.
    Cluster these labels;
    if there are enough labels, backproject and verify
  end
end
end
    
```

CS252A, Winter 2006

Computer Vision I

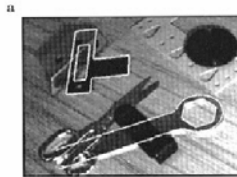
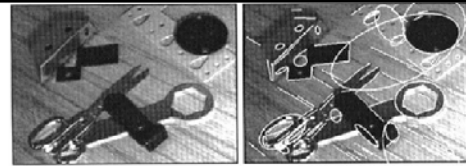


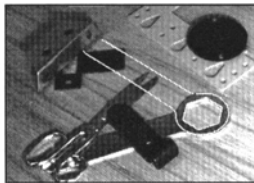
Figure from "Efficient model library access by projectively invariant indexing functions," by C.A. Rothwell et al., Proc. Computer Vision and Pattern Recognition, 1992, copyright 1992, IEEE

CS252A, Winter 2006

Computer Vision I

Verification

- Edge score
 - are there image edges near predicted object edges?
 - very unreliable; in texture, answer is usually yes
- Oriented edge score
 - are there image edges near predicted object edges with the right orientation?
 - better, but still hard to do well (see next slide)
- Texture
 - e.g. does the spanner have the same texture as the wood?



CS252A, Winter 2006

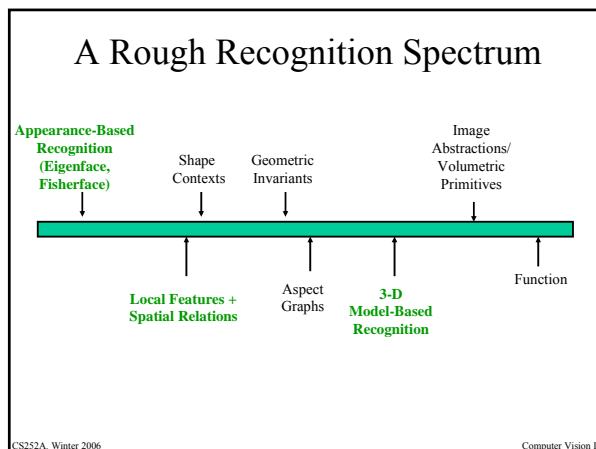
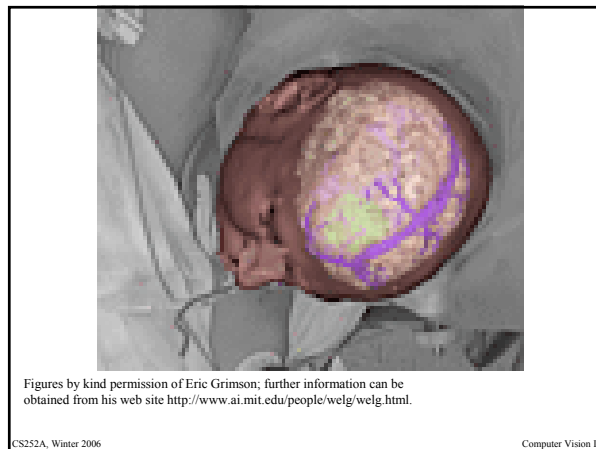
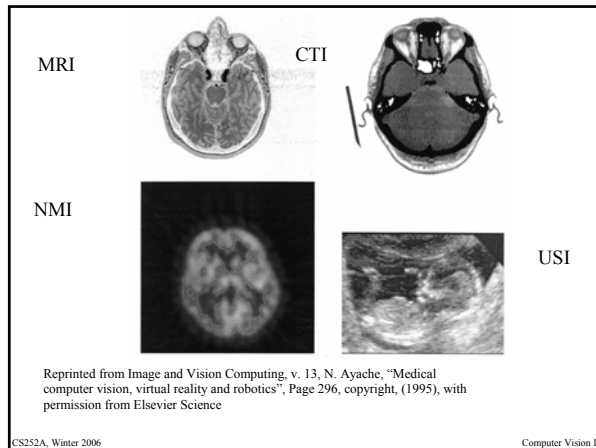
Computer Vision I

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

CS252A, Winter 2006

Computer Vision I

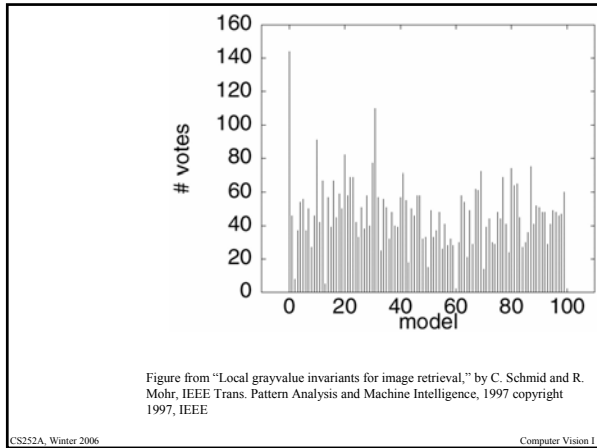


Matching using Local Image features

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, affine-invariant 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.

CS252A, Winter 2006 Computer Vision I

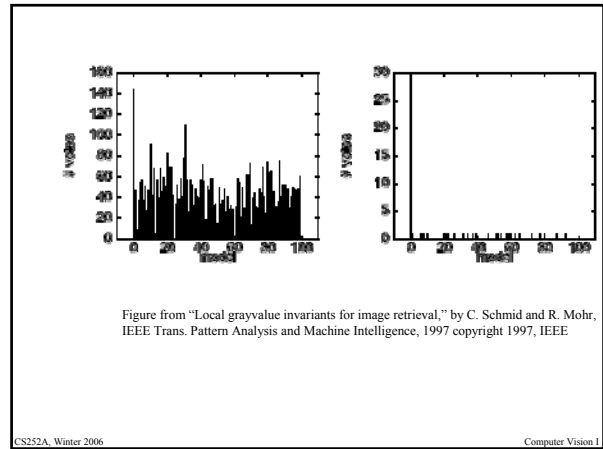
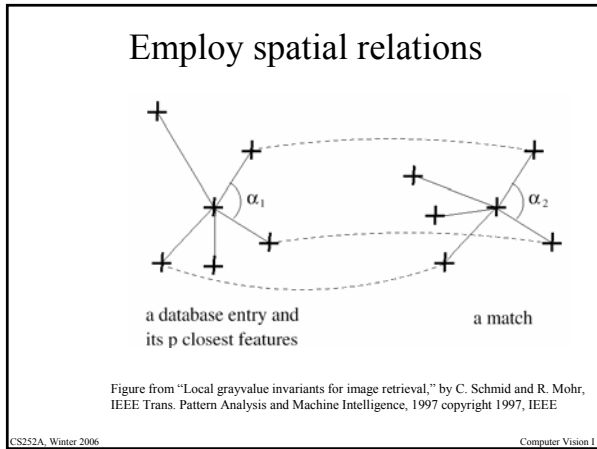


Probabilistic interpretation

- Write $P(\text{patch of type } i \text{ appears in image } | j\text{th pattern is present}) = p_{ij}$
- Assume $p_{ij} = \mu$ if the pattern can produce this patch and 0 otherwise
- Likelihood that n_p patches came from that pattern and $n_i - n_p$ patches come from noise, is

$$P(\text{interpretation} | \text{pattern}) = \lambda^{n_p} \mu^{(n_i - n_p)}$$

CS252A, Winter 2006 Computer Vision I



Example

Training examples

Test image

CS252A, Winter 2006 Computer Vision I

Finding faces using relations

- Strategy:
 - Face is eyes, nose, mouth, etc. with appropriate relations between them
 - build a specialised detector for each of these (template matching) and look for groups with the right internal structure
 - Once we've found enough of a face, there is little uncertainty about where the other bits could be

CS252A, Winter 2006 Computer Vision I

Finding faces using relations

- Strategy: compare

$P(\text{one face at } F | X_{le} = x_1, X_{re} = x_2, X_{lm} = x_3, X_{ln} = x_4, \text{all other responses})$
with

$P(\text{no face} | X_{le} = x_1, X_{re} = x_2, X_{lm} = x_3, X_{ln} = x_4, \text{all other responses})$



Notice that once some facial features have been found, the position of the rest is quite strongly constrained.

Figure from, "Finding faces in cluttered scenes using random labelled graph matching," by Leung, T.; Burl, M and Perona, P., Proc. Int. Conf. on Computer Vision, 1995 copyright 1995, IEEE

CS252A, Winter 2006

Computer Vision I

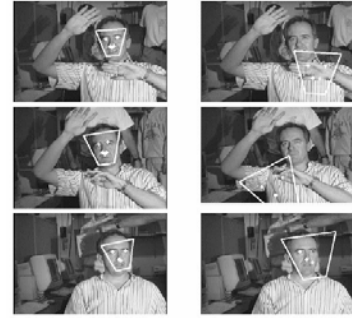


Figure from, "Finding faces in cluttered scenes using random labelled graph matching," by Leung, T.; Burl, M and Perona, P., Proc. Int. Conf. on Computer Vision, 1995 copyright 1995, IEEE

CS252A, Winter 2006

Computer Vision I

Even without shading, shape reveals a lot - line drawings



CS252A, Winter 2006

Computer Vision I

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

CS252A, Winter 2006

Computer Vision I

Scene Interpretation

"The Swing"
Fragonard, 1766



CS252A, Winter 2006

Computer Vision I