

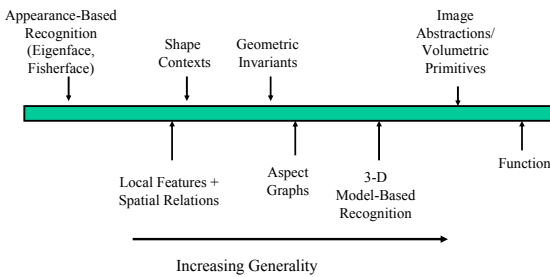
Recognition III

Introduction to Computer Vision
 CSE 152
 Lecture 20

Announcements

- Assignment 4: Due Friday
- Final Exam: Wed, 6/8/04, 3:00-6:00, in class room
- My last office hours: Tuesday, 6/7, 2:00-3:30
- I'll discuss briefly exam today

A Rough Recognition Spectrum



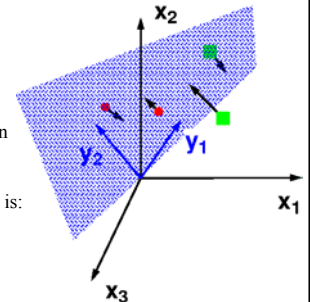
Projection, and reconstruction

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

$$y = Wx$$

- From $y \in \mathbb{R}^m$, the reconstruction of the point is $W^T y$

- The error of the reconstruction is: $\|x - W^T W x\|$



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 appendix of text to compute basis when $n < d$

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
 - **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:* $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_s$

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:

$$\Sigma = \frac{1}{n} \sum_{i=1}^n (\tilde{x}_i - \bar{\mu})(\tilde{x}_i - \bar{\mu})^T$$
- So, ignoring $1/n$ subtract mean image μ from each input image, create data matrix, and perform (thin) SVD on the data matrix.

CSE152, Spr 05

Intro Computer Vision

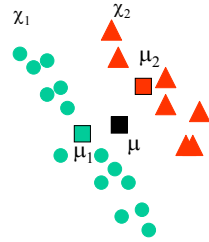
PCA & Fisher's Linear Discriminant

- **Between-class scatter**

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

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

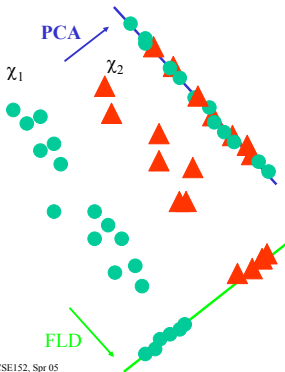


• If the data points are projected by $y=Wx$ and scatter of points is S , then the scatter of the projected points is $W^T S W$

CSE152, Spr 05

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 05

Intro Computer Vision

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

CSE152, Spr 05

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 05

Intro Computer Vision

Appearance manifold approach

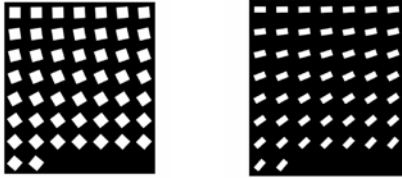
- for every object (Nayar et al. '96)
 1. sample the set of viewing conditions
 2. Crop & scale images to standard size
 3. Use as feature vector
- apply a PCA over all the images
- keep the dominant PCs
- Set of views for one object is represented as a manifold in the projected space
- Recognition: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



CSE152, Spr 05

Intro Computer Vision

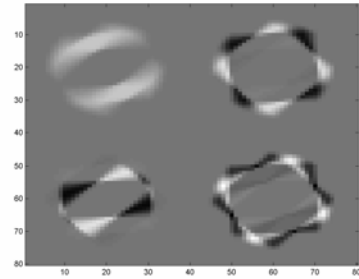
An example: input images



CSE152, Spr 05

Intro Computer Vision

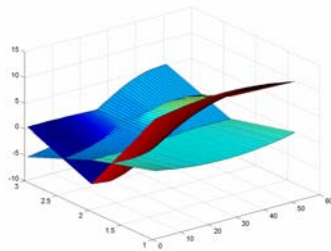
An example: basis images



CSE152, Spr 05

Intro Computer Vision

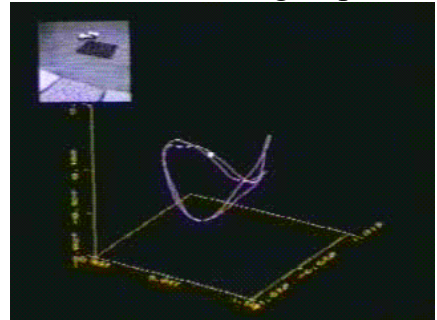
An example: surfaces of first 3 coefficients



CSE152, Spr 05

Intro Computer Vision

Parameterized Eigenspace



CSE152, Spr 05

Intro Computer Vision

Recognition



CSE152, Spr 05

Intro Computer Vision

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?

CSE152, Spr 05

Intro Computer Vision

Bayesian Classification

Discussed on blackboard, but slides may be helpful

Basic ideas in classifiers

- Loss
 - some errors may be more expensive than others
 - e.g. a fatal disease that is easily cured by a cheap medicine with no side-effects -> false positives in diagnosis are better than false negatives
 - We discuss two class classification: $L(1 \rightarrow 2)$ is the loss caused by calling 1 a 2
- Total risk of using classifier s

$$R(s) = Pr\{1 \rightarrow 2 | \text{using } s\} L(1 \rightarrow 2) + Pr\{2 \rightarrow 1 | \text{using } s\} L(2 \rightarrow 1)$$

Basic ideas in classifiers

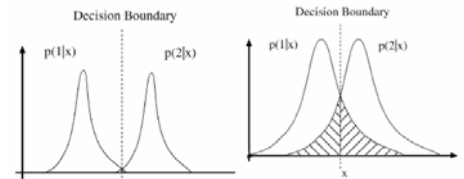
- Generally, we should classify as 1 if the expected loss of classifying as 1 is better than for 2
- gives

$$1 \text{ if } p(1|x)L(1 \rightarrow 2) > p(2|x)L(2 \rightarrow 1)$$

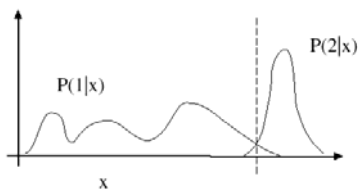
$$2 \text{ if } p(1|x)L(1 \rightarrow 2) < p(2|x)L(2 \rightarrow 1)$$

- Crucial notion: Decision boundary
 - points where the loss is the same for either case

Some loss may be inevitable: the minimum risk (shaded area) is called the Bayes risk



Finding a decision boundary is not the same as modelling a conditional density.



Example: known distributions

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

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

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

- Classifier boils down to: choose class that minimizes:

$$\delta(\mathbf{x}, \mu_k) - 2 \log \pi_k$$

where

Mahalanobis distance — $\delta(\mathbf{x}, \mu_k) = [(\mathbf{x} - \mu_k)^T \Sigma^{-1} (\mathbf{x} - \mu_k)]^{1/2}$

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



Finding skin

- Skin has a very small range of (intensity independent) colours, and little texture
 - Compute an intensity-independent colour measure, check if colour is in this range, check if there is little texture (median filter)
 - See this as a classifier - we can set up the tests by hand, or learn them.
 - get class conditional densities (histograms), priors from data (counting)
- Classify
 - if $p(\text{skin}|\mathbf{x}) > \theta$, classify as skin
 - if $p(\text{skin}|\mathbf{x}) < \theta$, classify as not skin
 - if $p(\text{skin}|\mathbf{x}) = \theta$, choose classes uniformly and at random

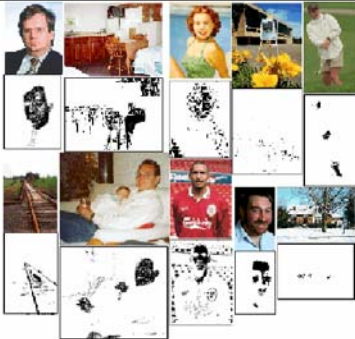


Figure from "Statistical color models with application to skin detection," M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE

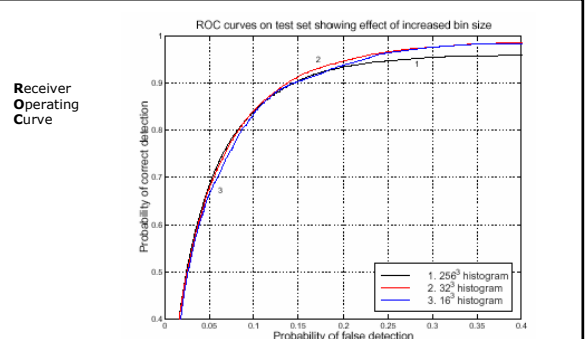


Figure from "Statistical color models with application to skin detection," M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE

Appearance-Based Vision: Lessons

Strengths

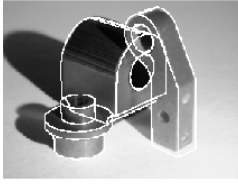
- Posing the recognition metric in the image space rather than a derived representation is more powerful than expected.
- Modeling objects from many images is not unreasonable given hardware developments.
- The data (images) may provide a better representations than abstractions for many tasks.

Appearance-Based Vision: Lessons

Weaknesses

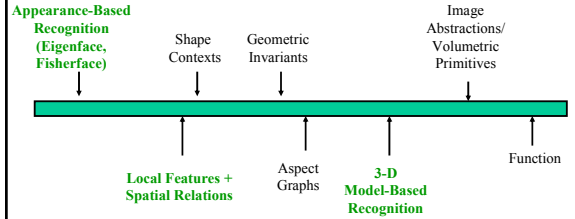
- Segmentation or object detection is still an issue.
- To train the method, objects have to be observed under a wide range of conditions (e.g. pose, lighting, shape deformation).
- Limited power to extrapolate or generalize (abstract) to novel conditions.

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.

A Rough Recognition Spectrum



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
- Issues
 - where do the hypotheses come from?
 - How do we compare to image (verification)?
- 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.

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

```

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
    
```

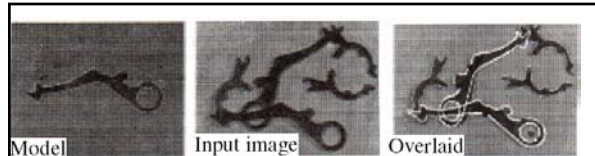


Figure from "Object recognition using alignment," D.P. Huttenlocher and S. Ullman, Proc. Int. Conf. Computer Vision, 1986, copyright IEEE, 1986

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.

CSE152, Spr 05

Intro Computer Vision

```

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
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
    
```

CSE152, Spr 05

Intro Computer Vision

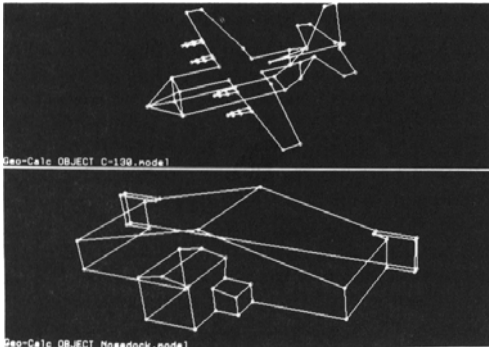


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

CSE152, Spr 05

Intro Computer Vision



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

CSE152, Spr 05

Intro Computer Vision

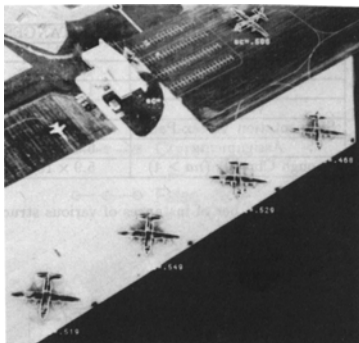


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

CSE152, Spr 05

Intro Computer Vision

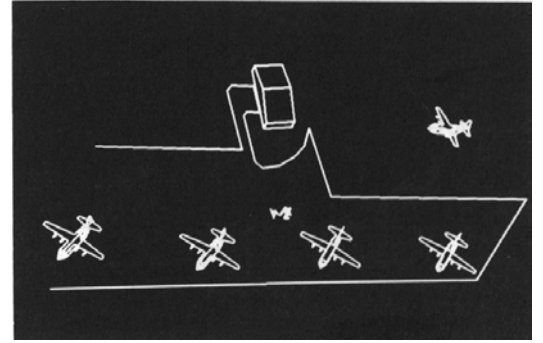


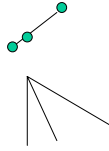
Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

CSE152, Spr 05

Intro Computer Vision

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



CSE152, Spr 05

Intro Computer Vision

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

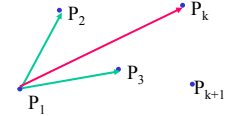
Now image points are obtained by multiplying by a plane affine transformation, so

$$p_k = AP_k$$

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

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

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



CSE152, Spr 05

Intro Computer Vision

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

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
    
```

CSE152, Spr 05

Intro Computer Vision

CSE152, Spr 05

Intro Computer Vision

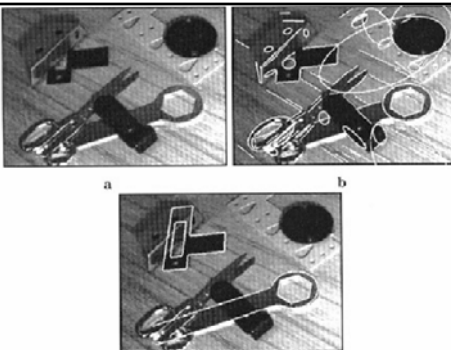


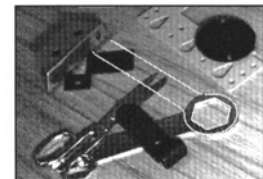
Figure from "Efficient model library access by projectively invariant indexing functions," by C.A. Rothwell et al., Proc. Computer Vision and Pattern Recognition, 1992, copyright 1992, IEEE

CSE152, Spr 05

Intro Computer Vision

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?



CSE152, Spr 05

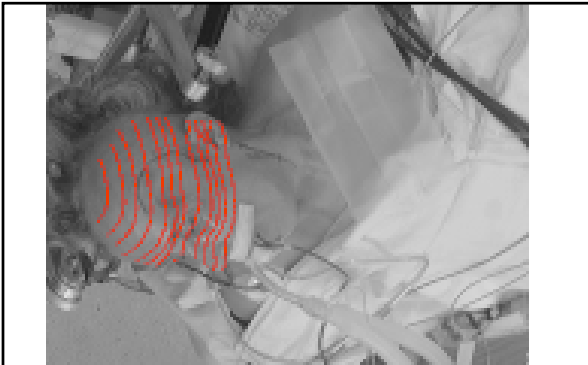
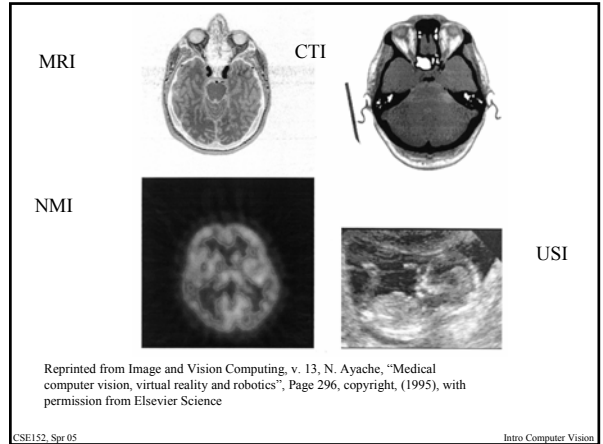
Intro Computer Vision

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

CSE152, Spr 05

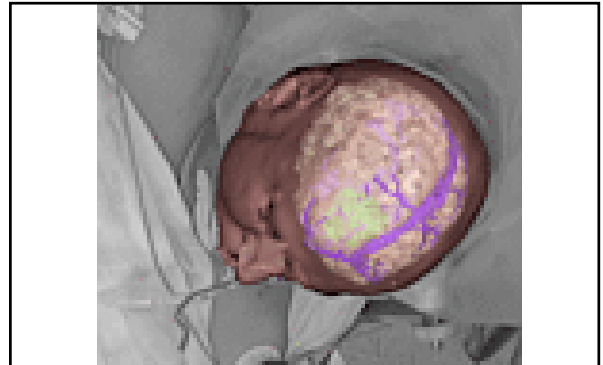
Intro Computer Vision



Figures by kind permission of Eric Grimson; further information can be obtained from his web site <http://www.ai.mit.edu/people/welg/welg.html>.

CSE152, Spr 05

Intro Computer Vision



Figures by kind permission of Eric Grimson; further information can be obtained from his web site <http://www.ai.mit.edu/people/welg/welg.html>.

CSE152, Spr 05

Intro Computer Vision

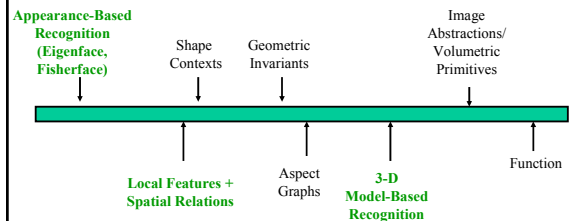


Figures by kind permission of Eric Grimson; further information can be obtained from his web site <http://www.ai.mit.edu/people/welg/welg.html>.

CSE152, Spr 05

Intro Computer Vision

A Rough Recognition Spectrum



CSE152, Spr 05

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, 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.

CSE152, Spr 05

Intro Computer Vision

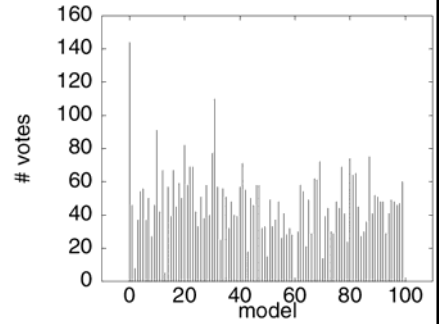


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

CSE152, Spr 05

Intro Computer Vision

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
 $p_{ix} = \lambda < \mu$ for all i
- Lik that n_p patches came from that pattern and $n_i - n_p$ patches come from noise, is $P(\text{interpretation} | \text{pattern}) = \lambda^{n_i} \mu^{(n_i - n_p)}$

CSE152, Spr 05

Intro Computer Vision

Employ spatial relations

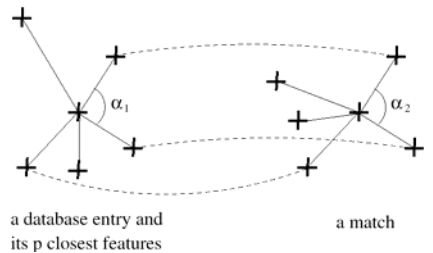


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

CSE152, Spr 05

Intro Computer Vision

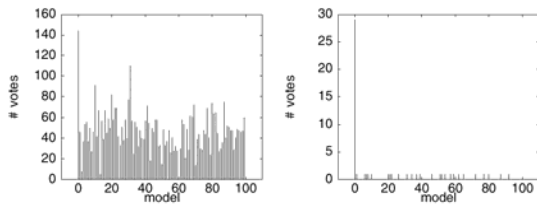


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

CSE152, Spr 05

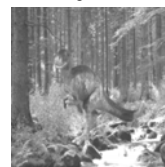
Intro Computer Vision

Example

Training examples



Test image



CSE152, Spr 05

Intro Computer Vision

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

CSE152, Spr 05

Intro Computer Vision

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

CSE152, Spr 05

Intro Computer Vision



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

CSE152, Spr 05

Intro Computer Vision

Even without shading, shape reveals a lot - line drawings



CSE152, Spr 05

Intro Computer Vision

Scene Interpretation



"The Swing"
Fragonard, 1766

CSE152, Spr 05

Intro Computer Vision

Final Exam

- Closed book
- One cheat sheet
 - Single piece of paper, handwritten, no photocopying, no physical cut & paste. – you can start with sheet from the midterm, if you want.
- 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.

CSE152, Spr 05

Intro Computer Vision

Further Studies

- CSE166: Image Processing
- AI (CSE150,151)
- CSE159: Projects in Computer Vision