Lecture 22
Object recognition III
Edge based descriptors
Edge based descriptors

What makes an image memorable?

Gavrila, Philomin, ICCV 1999

Papageorgiou & Poggio (2000)

J. Shotton, A. Blake, R. Cipolla. PAMI 2008.

Opelt, Pinz, Zisserman, ECCV 2006
Edges and chamfer distance

Gavrila, Philomin, ICCV 1999
Edges and chamfer distance

Template

Gavrila, Philomin, ICCV 1999
Chamfer distance

\[ d_{\text{chamfer}}(x) = \sum_{u \in F} \min_{v \in E} \| (u + x) - v \|_2 \]

Sum over pixels on the edge template \( F \)

Find closest edge location after displacement \( x \)

\( E = \) edge map of the image
Chamfer distance

DT(E) = Function that assigns to each pixel the distance to the nearest edge.

Using the distance transform, the Chamfer distance can be written as a convolution
Edges and chamfer distance
Distance transform

Edges
Distance transform

Edges

Distance transform (with Manhattan distance)
Efficient computation of DT

\[ P = \text{set of edge pixels.} \]

Two pass \( O(n) \) algorithm for 1D \( L_1 \) norm

1. **Initialize:** For all \( j \)
   \[ D[j] \leftarrow 1_p[j] \]
   // 0 if \( j \) is in \( P \), infinity otherwise

2. **Forward:** For \( j \) from 1 up to \( n-1 \)
   \[ D[j] \leftarrow \min(D[j], D[j-1]+1) \]

3. **Backward:** For \( j \) from \( n-2 \) down to 0
   \[ D[j] \leftarrow \min(D[j], D[j+1]+1) \]

Adapted from D. Huttenlocher
Chamfer distance

\[ d_{chamfer}(x) = \sum_{u \in F} \min_{v \in E} \| (u + x) - v \|_2 \]

\[ = F \ast DT(E) \]

Sum over pixels on the edge template \( F \)

Find closest edge location after displacement \( x \)

\( E = \) edge map of the image
REAL-TIME OBJECT DETECTION FOR "SMART" VEHICLES

D.M. Gavrila

Image Understanding Systems
DaimlerChrysler Research
Ulm 89081, Germany
dariu.gavrila@DaimlerChrysler.com

V. Philomin

Computer Vision Laboratory
University of Maryland
College Park, MD 20742, U.S.A.
vasi@cs.umd.edu
To deal with multiple appearances…
Issues

Global templates are sensitive to:
• Partial occlusions
• Non-rigid deformations

Constellation of local edge fragments
Building a Fragment Dictionary

Masks
(~10 images)

Contour Fragments $T_n$
(~1000 fragments)
Matching Features

- Gaussian weighted oriented chamfer matching
  - aligns features to image

Opelt, Pinz, Zisserman, ECCV 2006
J. Shotton, A. Blake, R. Cipolla. PAMI 2008.
Matching Features

• Gaussian weighted oriented chamfer matching
  – aligns features to image

\[ u(F_m, E | c) \] feature match score at optimal position

\[ r(F_m, E | c) \] optimal position
Location Sensitive Classification

- Feature match scores make detection simple
- Detection uses a boosted classification function \( K(c) \):

\[
K(c) = \sum_{m=1}^{M} a_m \delta(\nu(F_m, E|c) > \theta_m) + b_m
\]

<table>
<thead>
<tr>
<th>( M )</th>
<th>number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_m )</td>
<td>feature ( m )</td>
</tr>
<tr>
<td>( E )</td>
<td>canny edge map</td>
</tr>
<tr>
<td>( c )</td>
<td>object centroid</td>
</tr>
<tr>
<td>( \theta_m )</td>
<td>weak learner threshold</td>
</tr>
<tr>
<td>( a_m )</td>
<td>weak learner confidence</td>
</tr>
<tr>
<td>( b_m )</td>
<td>weak learner confidence</td>
</tr>
<tr>
<td>( \delta )</td>
<td>0-1 indicator function</td>
</tr>
</tbody>
</table>
Object Detection

- Evaluate $K(c)$ for all $c$ gives a classification map
  - confidence as function of position

- Globally thresholded local maxima give final detections
Learning System

Segmented Training Data

Boosting Algorithm

K(c)

Detection

Object Detections
Training Data

Class

Segmented (10)

Unsegmented (40)

Background (50)
Boosting as Feature Selection

1. Fragment Selection

2. Model Parameter Estimation
   Select $\sigma, \lambda$ for each feature

3. Weak-Learner Estimation
   Select $\theta, a, b$ for each feature
Contour Results
Histograms of oriented gradients
Histograms of oriented gradients

SIFT, D. Lowe, ICCV 1999

Shape context
Belongie, Malik, Puzicha, NIPS 2000

Count the number of points inside each bin, e.g.:
- Count = 4
- Count = 10

Compact representation of distribution of points relative to each point
Image features:

Histograms of oriented gradients (HOG)

Bin gradients from 8x8 pixel neighborhoods into 9 orientations

(Dalal & Triggs CVPR 05)

Source: Deva Ramanan
Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.
Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l’Europe, Montbonnot 38334, France

Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.
HOG

a) Input image

dx

r = dx^2 + dy^2

b) Oriented edges

b = \text{round}(\theta/9 \pi) \mod 9

\theta = \arctan(dx/dy)

c) Subsampling edge maps

\[ F_1(x,y) = r \quad \text{if} \quad b = 1 \]

\[ C_1(x,y) \]

\[ N_1(x,y) = \left[ C(x,y)^2 + C(x+1,y)^2 + C(x,y+1)^2 + C(x+1,y+1)^2 \right]^{0.5} \]

\[ \frac{C}{N_1} \]

\[ N_2(x,y) = \left[ C(x,y)^2 + C(x-1,y)^2 + C(x,y-1)^2 + C(x-1,y-1)^2 \right]^{0.5} \]

\[ \frac{C}{N_2} \]

\[ N_3(x,y) = \left[ C(x,y)^2 + C(x+1,y)^2 + C(x,y+1)^2 + C(x+1,y+1)^2 \right]^{0.5} \]

\[ \frac{C}{N_3} \]

\[ N_4(x,y) = \left[ C(x,y)^2 + C(x-1,y)^2 + C(x,y-1)^2 + C(x-1,y-1)^2 \right]^{0.5} \]

\[ \frac{C}{N_4} \]

d) Local contrast normalization

HOG = 36 dimensions/pixel
A Support Vector Machine (SVM) learns a classifier with the form:

\[ H(x) = \sum_{m=1}^{M} a_m y_m k(x, x_m) \]

Where \{x_m, y_m\}, for \( m = 1 \ldots M \), are the training data with \( x_m \) being the input feature vector and \( y_m = +1, -1 \) the class label. \( k(x, x_m) \) is the kernel and it can be any symmetric function satisfying the Mercer Theorem.

The classification is obtained by thresholding the value of \( H(x) \).

There is a large number of possible kernels, each yielding a different family of decision boundaries:

- Linear kernel: \( k(x, x_m) = x^T x_m \)
- Radial basis function: \( k(x, x_m) = \exp(-|x - x_m|^2/\sigma^2) \).
- Histogram intersection: \( k(x,x_m) = \sum_i (\min(x(i), x_m(i))) \)
Linear SVM

\[ f(x) = (w \cdot x + b) \]
Scanning-window templates
Dalal and Triggs CVPR05 (HOG)
Papageorgiou and Poggio ICIP99 (wavelets)

pos

neg

\[ w = \text{weights for orientation and spatial bins} \]
\[ w \cdot x > 0 \]

Train with a linear classifier (perceptron, logistic regression, SVMs...)

Source: Deva Ramanan
How to interpret positive and negative weights?

\[ w \cdot x > 0 \]

\[ (w_{pos} - w_{neg}) \cdot x > 0 \]

\[ w_{pos} \cdot x > w_{neg} \cdot x \]

Pedestrian template > Pedestrian background template

\[ w_{pos}, w_{neg} = \text{weighted average of positive, negative support vectors} \]

Right approach is to compete pedestrian, pillar, doorway... models

Background class is hard to model - easier to penalize particular vertical edges

Source: Deva Ramanan
Histograms of oriented gradients

Dalal & Trigs, 2006
Figure 3. The performance of selected detectors on (left) MIT and (right) INRIA data sets. See the text for details.
Constellation models

Source: short course on object recognition. Fergus, Fei-fei, Torralba
Representation

- Object as set of parts
  - Generative representation

- Model:
  - Relative locations between parts
  - Appearance of part

- Issues:
  - How to model location
  - How to represent appearance
  - Sparse or dense (pixels or regions)
  - How to handle occlusion/clutter
History of Parts and Structure approaches

- Fischler & Elschlager 1973
- Yuille ‘91
- Brunelli & Poggio ‘93
- Lades, v.d. Malsburg et al. ‘93
- Cootes, Lanitis, Taylor et al. ‘95
- Amit & Geman ‘95, ‘99
- Perona et al. ‘95, ‘96, ‘98, ’00, ’03, ’04, ’05
- Felzenszwalb & Huttenlocher ’00, ’04
- Crandall & Huttenlocher ’05, ’06
- Leibe & Schiele ’03, ’04
- Many papers since 2000
The Representation and Matching of Pictorial Structures

MARTIN A. FISCHLER AND ROBERT A. ELSCHLAGER

Abstract—The primary problem dealt with in this paper is the following. Given some description of a visual object, find that object in an actual photograph. Part of the solution to this problem is the specification of a descriptive scheme, and a metric on which to base the decision of "goodness" of matching or detection.

We offer a combined descriptive scheme and decision metric which is general, intuitively satisfying, and which has led to promising experimental results. We also present an algorithm which takes the above descriptions, together with a matrix representing the intensities of the actual photograph, and then finds the described object in the matrix. The algorithm uses a procedure similar to dynamic programming in order to cut down on the vast amount of computation otherwise necessary.

One desirable feature of the approach is its generality. A new programming system does not need to be written for every new description; instead, one just specifies descriptions in terms of a certain set of primitives and parameters.

There are many areas of application: scene analysis and description, map matching for navigation and guidance, optical tracking, stereo compilation, and image change detection. In fact, the ability to describe, match, and register scenes is basic for almost any image processing task.

Index Terms—Dynamic programming, heuristic optimization, picture description, picture representation.

THE PRIMARY paper is the following: a visual object, find graph. The object might be complicated, such as an object which can be linguistic, pictorial, or actual. A photograph will be called a two-dimensional array of gray levels and the object being sought is called the pattern.

This ability to find a pattern, equivalently, to match patterns, scenes, is basic for almost any image processing. Application to such an area as registration, map matching for land vehicles, navigation, and automatic scene analysis is a major goal.

Manuscript received November 30, 1971; revised May 22, 1972, and August 21, 1972.

The authors are with the Lockheed Palo Alto Research Laboratory, Lockheed Missiles & Space Company, Inc., Palo Alto, Calif. 94304.

Martin A. Fischer (S'57-M'58) was born in New York, N. Y., on February 15, 1932. He received the B.S. degree from the City College of New York, New York, in 1954 and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, Calif., in 1958 and 1962, respectively.

He served in the U.S. Army for two years and held positions at the National Bureau of Standards and at Hughes Aircraft Corporation during the period 1954 to 1958. In 1958 he joined the technical staff of the Lockheed Missiles & Space Company, Inc., at the Lockheed Palo Alto Research Laboratory, Palo Alto, Calif., and currently holds the title of Staff Scientist. He has conducted research and published in the areas of artificial intelligence, picture processing, switching theory, computer organization, and information theory.

Dr. Fischer is a member of the Association for Computing Machinery, the Pattern Recognition Society, the Mathematical Association of America, Tau Beta Pi, and Eta Kappa Nu. He is currently an Associate Editor of the journal Pattern Recognition and is a past Chairman of the San Francisco Chapter of the IEEE Society on Systems, Man, and Cybernetics.

Robert A. Elschlager was born in Chicago, Ill., on May 25, 1943. He received the B.S. degree in mathematics from the University of Illinois, Urbana, in 1964, and the M.S. degree in mathematics from the University of California, Berkeley, in 1969.

Since then he has been an Associate Scientist with the Lockheed Missiles & Space Company, Inc., at the Lockheed Palo Alto Research Center, Palo Alto, Calif. His current interests are picture processing, operating systems, computer languages, and computer understanding.

Mr. Elschlager is a member of the American Mathematical Society, the Mathematical Association of America, and the Association for Symbolic Logic.
(4)

Fig. 4. Examples of image-matching experiments using faces. (a) Successful embedding under coherent noise.
Sparse representation

+ Computationally tractable ($10^5$ pixels $\rightarrow 10^1 -- 10^2$ parts)
+ Generative representation of class
+ Avoid modeling global variability
+ Success in specific object recognition

- Throw away most image information
- Parts need to be distinctive to separate from other classes
Structure models

Voting models
- Many patches (>100)

Constellation models
- Few parts (~6)

Deformable models
- No parts
Region operators

– Local maxima of interest operator function

– Can give scale/orientation invariance

Figures from [Kadir, Zisserman and Brady 04]
The correspondence problem

- Model with $P$ parts
- Image with $N$ possible assignments for each part
- Consider mapping to be 1-1

$N^P$ combinations!!!
Different connectivity structures

- O(N^6)
  - a) Constellation [13]
  - Csurka '04
  - Vasconcelos '00

- O(N^2)
  - b) Star shape [9, 14]
  - Bouchard & Triggs '05

- O(N^3)
  - c) k-fan (k = 2) [9]
  - Carneiro & Lowe '06

- O(N^2)
  - d) Tree [12]
  - Felzenszwalb & Huttenlocher '00

- O(N^2)
  - e) Bag of features [10, 21]
How much does shape help?

- Crandall, Felzenszwalb, Huttenlocher CVPR’05
- Shape variance increases with increasing model complexity
- Do get some benefit from shape
Some class-specific graphs

- Articulated motion
  - People
  - Animals

- Special parameterisations
  - Limb angles

Images from [Kumar, Torr and Zisserman 05, Felzenszwalb & Huttenlocher 05]
Layout CRF: Winn & Shotton, CVPR ‘06

Part labels (color-coded)
How to model location?

• Explicit: Probability density functions
• Implicit: Voting scheme

• Invariance
  – Translation
  – Scaling
  – Similarity/affine
  – Viewpoint
Explicit shape model

- **Cartesian**
  - E.g. Gaussian distribution
  - Parameters of model, $\mu$ and $\Sigma$
  - Independence corresponds to zeros in $\Sigma$
  - Burl et al. ’96, Weber et al. ‘00, Fergus et al. ’03

\[
\mu = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \quad \Sigma = \begin{pmatrix} x_1 x_1 & x_1 x_2 & x_1 x_3 & x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 x_1 & x_2 x_2 & x_2 x_3 & x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 x_1 & x_3 x_2 & x_3 x_3 & x_3 y_1 & x_3 y_2 & x_3 y_3 \\ y_1 x_1 & y_1 x_2 & y_1 x_3 & y_1 y_1 & y_1 y_2 & y_1 y_3 \\ y_2 x_1 & y_2 x_2 & y_2 x_3 & y_2 y_1 & y_2 y_2 & y_2 y_3 \\ y_3 x_1 & y_3 x_2 & y_3 x_3 & y_3 y_1 & y_3 y_2 & y_3 y_3 \end{pmatrix}
\]

- **Polar**
  - Convenient for invariance to rotation

Mikolajczyk et al., CVPR ‘06
Implicit shape model

- Use Hough space voting to find object
- Leibe and Schiele ’03,’05

**Learning**

- Learn appearance codebook
  - Cluster over interest points on training images

- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object
  - Centroid is given

**Recognition**

**Interest Points**

**Matched Codebook Entries**

**Probabilistic Voting**
Deformable Template Matching

Berg, Berg and Malik CVPR 2005

- Formulate problem as Integer Quadratic Programming
- $O(N^P)$ in general
- Use approximations that allow $P=50$ and $N=2550$ in <2 secs
Multiple view points

Hoiem, Rother, Winn, 3D Layout CRF for Multi-View Object Class Recognition and Segmentation, CVPR ‘07

Thomas, Ferrari, Leibe, Tuytelaars, Schiele, and L. Van Gool. Towards Multi-View Object Class Detection, CVPR 06
Representation of appearance

• Needs to handle intra-class variation
  – Task is no longer matching of descriptors
  – Implicit variation (VQ to get discrete appearance)
  – Explicit model of appearance (e.g. Gaussians in SIFT space)

• Dependency structure
  – Often assume each part’s appearance is independent
  – Common to assume independence with location
Representation of appearance

- Invariance needs to match that of shape model
- Insensitive to small shifts in translation/scale
  - Compensate for jitter of features
  - e.g. SIFT
- Illumination invariance
  - Normalize out
Appearance representation

• SIFT

• Decision trees

[Figure from Winn & Shotton, CVPR ‘06]

• PCA

[Figure from Lepetit and Fua, CVPR 2005]
Background clutter

- **Explicit model**
  - Generative model for clutter as well as foreground object

- **Use a sub-window**
  - At correct position, no clutter is present
A simple parts and structure object detector

ICCV 2005 short courses on Recognizing and Learning Object Categories

An intuitive way to represent objects is as a collection of distinctive parts. Such schemes model both the relative positions of the parts as well as their appearances, giving a sparse representation that captures the essence of the object.

This simple demo illustrates the concepts behind many such “parts and structure” approaches. For simplicity, training is manually guided with the user hand-clicking on the distinctive parts of a few training images. A simple model is then built for use in recognition. Two different recognition approaches are provided: one relying on feature points [1], the other using the efficient methods of Felzenszwalb and Huttenlocher [2].

The code consists of Matlab scripts (which should run under both Windows and Linux). The Image Processing toolbox is required. The code is for teaching/research purposes only. If you find a bug, please email me at frossin@eecs.berkeley.edu.

Download

Download the code and datasets (24 Mbytes)

Operation of code

To run the demos:

1. Unpack the zip file into a new directory to add to your path environment
Demo (2)
Demo (3)
Demo (4)
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan

Abstract—We describe an object detection system based on mixtures of multiscale deformable part models. Our system is able to represent highly variable object classes and achieves state-of-the-art results in the PASCAL object detection challenges. While deformable part models have become quite popular, their value had not been demonstrated on difficult benchmarks such as the PASCAL datasets. Our system relies on new methods for discriminative training with partially labeled data. We combine a margin-sensitive approach for data-mining hard negative examples with a formalism we call latent SVM. A latent SVM is a reformulation of MI-SVM in terms of latent variables. A latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples. This leads to an iterative training algorithm that alternates between fixing latent values for positive examples and optimizing the latent SVM objective function.

Index Terms—Object Recognition, Deformable Models, Pictorial Structures, Discriminative Training, Latent SVM

----------------------------------
PASCAL Visual Object Challenge

5000 training images

5000 testing images

20 everyday object categories

aeroplane bike bird boat bottle bus car cat chair cow table
dog horse motorbike person plant sheep sofa train tv

Source: Deva Ramanan
5 years of PASCAL people detection

Discriminative mixtures of star models 2007-2010 Felzenszwalb, McAllester, Ramanan CVPR 2008
Felzenszwalb, Girshick, McAllester, and Ramanan PAMI 2009

Source: Deva Ramanan
Deformable part models

Model encodes **local appearance** + **pairwise geometry**

Source: Deva Ramanan
Deformable part models

Model encodes **local appearance** + **pairwise geometry**

Source: Deva Ramanan
Scoring function

\[
\text{score}(x,z) = \sum_i w_i \phi(x, z_i) + \sum_{i,j} w_{ij} \Psi(z_i, z_j)
\]

\[
\text{score}(x,z) = w \cdot \Phi(x, z)
\]

Score is linear in local templates \( w_i \) and spring parameters \( w_{ij} \)

Source: Deva Ramanan
Inference: $\max_z \text{score}(x, z)$

Felzenszwalb & Huttenlocher 05

Star model: the location of the root filter is the anchor point.
Given the root location, all part locations are independent.
Classification

\[ f_w(x) = \mathbf{w} \cdot \Phi(x) \]

\[ f_w(x) > 0 \]
Latent-variable classification

\[ f_w(x) = w \cdot \Phi(x) \]

\[ f_w(x) > 0 \]

\[ f_w(x) = \max_z S(x, z) = \max_z w \cdot \Phi(x, z) \]

Source: Deva Ramanan
Learning Initialization

Learn root filter with SVM
Initialize part filters to regions in root filter with lots of energy

Source: Deva Ramanan
1) Given positive part locations, learn $w$ with a convex program

$$w = \arg\min_w L(w) \quad \text{with fixed} \quad \{z_n : n \in \text{pos}\}$$

2) Given $w$, estimate part locations on positives

$$z_n = \arg\max_z w \cdot \Phi(x_n, z) \quad \forall n \in \text{pos}$$

The above steps perform coordinate descent on a joint loss

Source: Deva Ramanan
Example models

Source: Deva Ramanan
Example models

Source: Deva Ramanan
Example models

False positive due to imprecise bounding box

Source: Deva Ramanan
Other tricks:
• Mining hard negative examples
• Noisy annotations
Class experiment
Experiment 1: draw a horse (the entire body, not just the head) in a white piece of paper.

Do not look at your neighbor! You already know how a horse looks like... no need to cheat.
Experiment 2: draw a horse (the entire body, not just the head) but this time chose a viewpoint as weird as possible.
Structure models

- Voting models
  - Many parts (>100)
  - Lecture 20

- Constellation models
  - Few parts (~6)
  - Lecture 21

- Deformable models
  - No parts
  - Lecture 22
Structure models

Voting models
- Many parts (>100)

Constellation models
- Few parts (~6)

Deformable models
- No parts

Lecture 20
Lecture 21
Lecture 22
From wikipedia: Perhaps the most famous part of the work is chapter XVII, "The Comparison of Related Forms," where Thompson explored the degree to which differences in the forms of related animals could be described by means of relatively simple mathematical transformations.
Shape Matching and Object Recognition Using Shape Contexts

Serge Belongie, Member, IEEE, Jitendra Malik, Member, IEEE, and Jan Puzicha

Abstract—We present a novel approach to measuring similarity between shapes and exploit it for object recognition. In our framework, the measurement of similarity is preceded by 1) solving for correspondences between points on the two shapes, 2) using the correspondences to estimate an aligning transform. In order to solve the correspondence problem, we attach a descriptor, the shape context, to each point. The shape context at a reference point captures the distribution of the remaining points relative to it, thus offering a globally discriminative characterization. Corresponding points on two similar shapes will have similar shape contexts, enabling us to solve for correspondences as an optimal assignment problem. Given the point correspondences, we estimate the transformation that best aligns the two shapes; regularized thin-plate splines provide a flexible class of transformation maps for this purpose. The dissimilarity between the two shapes is computed as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transform. We treat recognition in a nearest-neighbor classification framework as the problem of finding the stored prototype shape that is maximally similar to that in the image. Results are presented for silhouettes, trademarks, handwritten digits, and the COIL data set.

Index Terms—Shape, object recognition, digit recognition, correspondence problem, MPEG7, image registration, deformable templates.
Matching Framework

- Find correspondences between points on shape
- Fast pruning
- Estimate transformation & measure similarity
Comparing Pointsets
Shape Context

Count the number of points inside each bin, e.g.:

- Count = 4
- Count = 10

✦ Compact representation of distribution of points relative to each point
Shape Context
Comparing Shape Contexts

Compute matching costs using Chi Squared distance:

\[ C_{ij} = \frac{1}{2} \sum_{k=1}^{K} \left[ \frac{h_i(k) - h_j(k)}{h_i(k) + h_j(k)} \right]^2 \]

Recover correspondences by solving linear assignment problem with costs \(C_{ij}\)

[Jonker & Volgenant 1987]
Matching Framework

- Find correspondences between points on shape
- Fast pruning
- Estimate transformation & measure similarity
Fast pruning

- Find best match for the shape context at only a few random points and add up cost
Matching Framework

- Find correspondences between points on shape
- Fast pruning
- **Estimate transformation & measure similarity**
Matching
Example

model
target
Outlier Test Example
The spaces of faces is not convex

The average of two faces is …
The spaces of faces is not convex

The average of two faces is not another face
A shape-texture face model

Blanz, V. and Vetter, T., A morphable model for the synthesis of 3D faces, 1999
Image warping
Image warping

Original image

Background
Face database
Appearance Model (AppModel.m)

- Each image is represented as (1) a collection of correspondence points (shape) and (2) a texture image normalized in shape.

1 - Shape information (texture free)

2 - Texture information (shape free)

Original image

Original image = ImageWarp(

• 2 - Shape normalization is obtained by warping the image into the mean shape of the training database.

Shape free texture

Mean shape

zeros

shape

Original image
Shape model

• PCA of shape information for the training database:

  PC1  PC2  PC3
  + s1 + s2 + s3 + ...
  PC4  PC5  PC6
  Mean shape

• Each shape can be decomposed as:
Texture model

- PCA of texture information for the training database:
  The PCA is done on the shape free images

PC1  PC2  PC3
PC4  PC5  PC6

Each texture (shape free) can be decomposed as:

Shape free texture = Mean texture + t1 + t2 + t3
Summary of Appearance Model of one image

A set of model parameters encode shape and gray level variation learned from a training set.
Active Appearance Model Search

Given a new “face” the model has to build an appearance model (shape + texture) that reproduces the original image:

![Novel face](image)

\[
\text{Shape} = ? \\
\text{Texture} = ?
\]

This is done in an iterative procedure that tries to minimize the reconstruction error.
Results

<table>
<thead>
<tr>
<th>Iter</th>
<th>Input image</th>
<th>Model</th>
<th>Shape</th>
<th>Residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1.png" alt="Input image" /></td>
<td><img src="image2.png" alt="Model" /></td>
<td><img src="image3.png" alt="Shape" /></td>
<td><img src="image4.png" alt="Residual" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="image5.png" alt="Input image" /></td>
<td><img src="image6.png" alt="Model" /></td>
<td><img src="image7.png" alt="Shape" /></td>
<td><img src="image8.png" alt="Residual" /></td>
</tr>
<tr>
<td>10</td>
<td><img src="image10.png" alt="Input image" /></td>
<td><img src="image11.png" alt="Model" /></td>
<td><img src="image12.png" alt="Shape" /></td>
<td><img src="image13.png" alt="Residual" /></td>
</tr>
<tr>
<td>100</td>
<td><img src="image100.png" alt="Input image" /></td>
<td><img src="image101.png" alt="Model" /></td>
<td><img src="image102.png" alt="Shape" /></td>
<td><img src="image103.png" alt="Residual" /></td>
</tr>
</tbody>
</table>
Active Appearance Model Search (Results)

Slide: Dhruv Batra
Essence of the Idea: Recognition by Synthesis

Explain a new example in terms of the model parameters

Initial  3 its  8 its  11 its  Converged  Original

Slide: Dhruv Batra
Enhancing gender

more same  original  androgynous  more opposite
Changing age

Face becomes “rounder” and “more textured” and “grayer”

- **original**
- **shape**
- **color**
- **both**
Structure models

- **Voting models**
  - Many parts (>100)

- **Constellation models**
  - Few parts (~6)

- **Deformable models**
  - No parts
Structure models

Voting models
• Many parts (>100)

Constellation models
• Few parts (~6)

Deformable models
• No parts

Bag of words
• No structure!
Object → Bag of ‘words’
Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach us from our eyes. For a long time it was thought that the retinal image was transmitted point by point to visual centers in the cerebral cortex; the cerebral cortex was a movie screen, so to speak, upon which the image in the eye was projected. Through the discoveries of Hubel and Wiesel we now know that behind the origin of the visual perception in the brain there is a considerably more complicated course of events. By following the visual impulses along their path to the various cell layers of the optical cortex, Hubel and Wiesel have been able to demonstrate that the message about the image falling on the retina undergoes a step-wise analysis in a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.

China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump in exports to $750bn, compared with a 18% rise in imports to $660bn. This is likely to annoy the US, which has long argued that China's exports are unfairly helped by a deliberately undervalued yuan. Beijing agrees the surplus is too high, but says the yuan is only one factor. Bank of China governor Zhou Xiaochuan said the country also needs to do more to boost domestic demand so more goods stay within the country. China increased the value of the yuan against the dollar by 2.1% in July and permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.
Related works

• Early “bag of words” models: mostly texture recognition

• Hierarchical Bayesian models for documents (pLSA, LDA, etc.)
  – Hoffman 1999; Blei, Ng & Jordan, 2004; Teh, Jordan, Beal & Blei, 2004

• Object categorization
  – Csurka, Bray, Dance & Fan, 2004; Sivic, Russell, Efros, Freeman & Zisserman, 2005; Sudderth, Torralba, Freeman & Willsky, 2005;

• Natural scene categorization
  – Vogel & Schiele, 2004; Fei-Fei & Perona, 2005; Bosch, Zisserman & Munoz, 2006
Discovering topics in text collections

Text document

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Discovered topics

<table>
<thead>
<tr>
<th>“Arts”</th>
<th>“Budgets”</th>
<th>“Children”</th>
<th>“Education”</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW</td>
<td>MILLION</td>
<td>CHILDREN</td>
<td>SCHOOL</td>
</tr>
<tr>
<td>FILM</td>
<td>TAX</td>
<td>WOMEN</td>
<td>STUDENTS</td>
</tr>
<tr>
<td>SHOW</td>
<td>PROGRAM</td>
<td>PEOPLE</td>
<td>SCHOOLS</td>
</tr>
<tr>
<td>MUSIC</td>
<td>BUDGET</td>
<td>CHILD</td>
<td>EDUCATION</td>
</tr>
<tr>
<td>MOVIE</td>
<td>BILLION</td>
<td>YEARS</td>
<td>TEACHERS</td>
</tr>
<tr>
<td>PLAY</td>
<td>FEDERAL</td>
<td>FAMILIES</td>
<td>HIGH</td>
</tr>
<tr>
<td>MUSICAL</td>
<td>YEAR</td>
<td>WORK</td>
<td>PUBLIC</td>
</tr>
<tr>
<td>BEST</td>
<td>SPENDING</td>
<td>PARENTS</td>
<td>TEACHER</td>
</tr>
<tr>
<td>ACTOR</td>
<td>NEW</td>
<td>SAYS</td>
<td>BENNETT</td>
</tr>
<tr>
<td>FIRST</td>
<td>STATE</td>
<td>FAMILY</td>
<td>MANIGAT</td>
</tr>
<tr>
<td>YORK</td>
<td>PLAN</td>
<td>WELFARE</td>
<td>NAMPHY</td>
</tr>
<tr>
<td>OPERA</td>
<td>MONEY</td>
<td>MEN</td>
<td>STATE</td>
</tr>
<tr>
<td>THEATER</td>
<td>PROGRAMS</td>
<td>PERCENT</td>
<td>PRESIDENT</td>
</tr>
<tr>
<td>ACTRESS</td>
<td>GOVERNMENT</td>
<td>CARE</td>
<td>ELEMENTARY</td>
</tr>
<tr>
<td>LOVE</td>
<td>CONGRESS</td>
<td>LIFE</td>
<td>HAITI</td>
</tr>
</tbody>
</table>
Visual analogy

document - image

word - visual word

topics - objects
Two bag-of-words classifiers

ICCV 2005 short courses on
Recognizing and Learning Object Categories

A simple approach to classifying images is to treat them as a collection of regions, describing only their appearance and ignoring their spatial relationships. This description has been successfully used in the text community for analyzing documents and are known as "bag-of-words" models, since each document may be represented as a distribution over fixed vocabulary(s). Using such a representation, methods such as probabilistic latent semantic analysis (pLSA) [1] or Latent Dirichlet Allocation (LDA) [2] are able to extract coherent topics within document collections in an unsupervised manner.

Recently, Fei-Fei et al. [3] and Sivic et al. [4] have applied such methods to the visual domain. The demo code implements pLSA, including a graphical user interface. For comparison, a Naive Bayes classifier is also provided which requires labelled training data, unlike pLSA.

The code consists of Matlab scripts (which should run under both Windows and Linux) and a couple of 32-bit Linux binaries for doing the actual processing. Hence the whole system will need to be run on Linux. The code is for teaching/research purposes only. If you find a bug or have questions, please send an email to the author.

Download

Download the code and datasets (32 Mbytes)

Operation of code

To run the demos:
From Images to Features

• Pixels are very sensitive to changes in lighting & pose
• Instead represent image as affine covariant regions:
  – Harris affine invariant regions (corners & edges)
  – Maximally stable extremal regions (segmentation)

Software provided by Oxford Visual Geometry Group
Sample Detected Features
Describing Feature Appearance

- **SIFT**: Scale Invariant Feature Transform
- Normalized histogram of orientation energy in each affinely adapted region (128-dim.)
A Discrete Feature Vocabulary

- Using all training images, build a dictionary via K-means clustering (~1000 words)
- Map each SIFT descriptor to nearest word

\[ w_{ji} \quad \text{appearance of feature } i \text{ in image } j \]

\[ y_{ji} \quad \text{2D position of feature } i \text{ in image } j \]
Form dictionary

Build visual vocabulary by k-means clustering
SIFT descriptors (K≈2,000)

Slide credit: Bryan Russell & Josef Sivic
Example regions assigned to the same dictionary cluster

Cluster 1

Cluster 2

Slide credit: Bryan Russell & Josef Sivic
Representing an image with visual words

Sivic & Zisserman '03

Interest regions

Visual words

Slide credit: Bryan Russell & Josef Sivic
System overview

Input image    Compute visual words    Discover visual topics

Slide credit: Bryan Russell & Josef Sivic
Bag of words

Stack visual word histograms as columns in matrix

Throw away spatial information!

Slide credit: Bryan Russell & Josef Sivic
Documents collection

Co-occurrence table:

Number of times word i appears on document/image j
Latent Dirichlet Allocation (LDA)
Blei, et al. 2003

- LDA model assumes exchangeability
- Order of words does not matter

\[ w_{ij} \] - words
\[ z_{ij} \] - topic assignments
\[ \theta_i \] - topic mixing weights
\[ \Phi_k \] - word mixing weights

\[ z_{ij}|\theta_i \sim \theta_i \]
\[ \theta_i|\alpha \sim \text{Dirichlet}(\alpha) \]
\[ w_{ij}|z_{ij} = k, \phi \sim \phi_k \]
\[ \phi_k|\beta \sim \text{Dirichlet}(\beta) \]

\[ p(w_{ij}) \propto \sum_{k=1}^{K} p(w_{ij}|z_{ij} = k, \phi_k) p(z_{ij} = k|\theta_i) \]
Inference

- $w_{ij}$ - words
- $z_{ij}$ - topic assignments
- $\theta_i$ - topic mixing weights
- $\phi_k$ - word mixing weights

Use Gibbs sampler to sample topic assignments

$$z_{ij} \sim p(z_{ij} = k| w_{ij} = v, w_{(i)}, z_{(i)}, \beta, \alpha)$$

- Only need to maintain counts of topic assignments
- Sampler typically converges in less than 50 iterations
- Run time is less than an hour

[Griffiths & Steyvers 2004]
Apply to Caltech 4 + background images

Faces 435
Motorbikes 800
Airplanes 800
Cars (rear) 1155
Background 900
Total: 4090

Slide credit: Bryan Russell & Josef Sivic
\[ p(w_{ij} \mid d_i) \]

\[ w \]

\[ d \]

\[ p(z_{ij} \mid z_{ij}) \]

\[ z \]

\[ d \]

\[ p(z_{ij} \mid d_i) \]
\begin{align*}
  p(w_{ij} | d_i) & \quad \sim \quad p(z_{ij} | d_i) \\
  p(w_{ij} | z_{ij}) & \quad = \quad p(z_{ij} | d_i)
\end{align*}
Most likely words given topic

**Topic 1**

Word 1

Word 2

**Topic 2**

Word 1

Word 2

Slide credit: Bryan Russell & Josef Sivic
Most likely words given topic

Topic 3

Topic 4

Slide credit: Bryan Russell & Josef Sivic
\[ d \rightarrow p(w_{ij} | d_i) \]

\[ w \rightarrow p(w_{ij} | z_{ij}) \]

\[ z \rightarrow p(z_{ij} | d_i) \]
Image clustering

Confusion matrices:

Average confusion:

<table>
<thead>
<tr>
<th>Expt.</th>
<th>Categories</th>
<th>T</th>
<th>LDA %</th>
<th>LDA #</th>
<th>pLSA %</th>
<th>pLSA #</th>
<th>KM baseline %</th>
<th>KM baseline #</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>4</td>
<td>4</td>
<td>97</td>
<td>86</td>
<td>98</td>
<td>70</td>
<td>72</td>
<td>908</td>
</tr>
<tr>
<td>(2)</td>
<td>4 + bg</td>
<td>5</td>
<td>78</td>
<td>931</td>
<td>78</td>
<td>931</td>
<td>56</td>
<td>1820</td>
</tr>
<tr>
<td>(2)*</td>
<td>4 + bg</td>
<td>6</td>
<td>84</td>
<td>656</td>
<td>76</td>
<td>1072</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(2)*</td>
<td>4 + bg</td>
<td>7</td>
<td>78</td>
<td>1007</td>
<td>83</td>
<td>768</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(2)*</td>
<td>4 + bg-fxd</td>
<td>7</td>
<td>90</td>
<td>330</td>
<td>93</td>
<td>238</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Image as a mixture of topics (objects)
Beyond single classes

- Multiclass
- Multiview
- Datasets
Beyond single classes

• Multiclass
• Multiview
• Datasets
Shared features

- Is learning the object class 1000 easier than learning the first?
- Can we transfer knowledge from one object to another?
- Are the shared properties interesting by themselves?
“MTL improves generalization by leveraging the domain-specific information contained in the training signals of related tasks. It does this by training tasks in parallel while using a shared representation”.

Sejnowski & Rosenberg 1986; Hinton 1986; Le Cun et al. 1989; Suddarth & Kergosien 1990; Pratt et al. 1991; Sharkey & Sharkey 1992; ...
Sharing in constellation models
(next Wednesday)

Pictorial Structures
Fischler & Elschlager, IEEE Trans. Comp. 1973

SVM Detectors
Heisele, Poggio, et. al., NIPS 2001

Constellation Model
Fergus, Perona, & Zisserman, CVPR 2003

Model-Guided Segmentation
Mori, Ren, Efros, & Malik, CVPR 2004
Reusable Parts

Krempp, Geman, & Amit “Sequential Learning of Reusable Parts for Object Detection”. TR 2002

Goal: Look for a vocabulary of edges that reduces the number of features.
Additive models and boosting

• Independent binary classifiers:

• Binary classifiers that share features:

Torralba, Murphy, Freeman. CVPR 2004. PAMI 2007
Generalization as a function of object similarities

12 unrelated object classes

12 viewpoints

Area under ROC

Number of training samples per class

Area under ROC

Number of training samples per class

K = 2.1

K = 4.8

Torralba, Murphy, Freeman. CVPR 2004. PAMI 2007
Beyond single classes

• Multiclass
• Multiview
• Datasets
3D object categorization

Wait: object categorization in humans is **not** invariant to 3D pose
3D object categorization

Despite we can categorize all three pictures as being views of a horse, the three pictures do not look as being equally typical views of horses. And they do not seem to be recognizable with the same easiness.
**Experiment** (Palmer, Rosch & Chase 81): participants are shown views of an object and are asked to rate “how much each one looked like the objects they depict” (scale: 1=very much like, 7=very unlike)

In a recognition task, reaction time correlated with the ratings.

Canonical views are recognized faster at the entry level.
Canonical Viewpoint

Clocks are preferred as purely frontal
Object representations

Explicit 3D models: use volumetric representation. Have an explicit model of the 3D geometry of the object.

Appealing but hard to get it to work…
Object representations

**Implicit 3D models**: matching the input 2D view to view-specific representations.

(b) For cars, classifiers are trained on 8 viewpoints

Not very appealing but somewhat easy to get it to work…
Beyond single classes

- Multiclass
- Multiview
- Datasets
In 2007, the twenty object classes that have been selected are:

*Person*: person

*Animal*: bird, cat, cow, dog, horse, sheep

*Vehicle*: aeroplane, bicycle, boat, bus, car, motorbike, train

*Indoor*: bottle, chair, dining table, potted plant, sofa, tv/monitor
Tool went online July 1st, 2005
530,000 object annotations collected
Labelme.csail.mit.edu
80,000,000 images

75,000 non-abstract nouns from WordNet

7 Online image search engines

And after 1 year downloading images

Google: 80 million images

A. Torralba, R. Fergus, W.T. Freeman. PAMI 2008
• An ontology of images based on WordNet

• ImageNet currently has
  – 13,000+ categories of visual concepts
  – 10 million human-cleaned images (~700im/categ)
  – 1/3+ is released online @ www.image-net.org

Deng, Dong, Socher, Li & Fei-Fei, CVPR 2009
Dataset biases

Torralba, Efros. Unbiased Look at Dataset Bias. CVPR 2011