Computer Vision and Object Recognition

Prof. Daniel Huttenlocher

Computer Vision

- Extraction of scene content from images and video
- Traditional applications in robotics and control
  - E.g., driver safety
- More recently in film and television
  - E.g., ad insertion
- Digital images now being used in many fields

Computer Vision Research Areas

- Commonly broken down according to degree of abstraction from image
  - Low-level: mapping from pixels to pixels
    - Edge detection, feature detection, stereopsis, optical flow
  - Mid-level: mapping from pixels to regions
    - Segmentation, recovering 3d structure from motion
  - High-level: mapping from pixels and regions to abstract categories
    - Recognition, classification, localization

Today's Overview

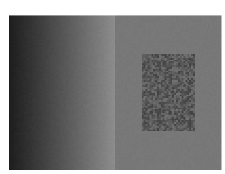
- Focus on some mid- and high-level vision problems and techniques
- Illustrate some computer vision algorithms and applications
- Segmentation and recognition because of potential utility for analyzing images gathered in the laboratory or the field
  - Cover basic techniques rather than particular applications

Image Segmentation

- Find regions of image that are "coherent"
- "Dual" of edge detection
  - Regions vs. boundaries
- Related to clustering problems
  - Early work in image processing and clustering
- Many approaches
  - Graph-based
    - Cuts, spanning trees, MRF methods
  - Feature space clustering
  - Mean shift

A Motivating Example

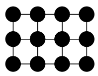
- Image segmentation plays a powerful role in human visual perception
  - Independent of particular objects or recognition



This image has three perceptually distinct regions

Graph Based Formulation

- $G=(V,E)$ with vertices corresponding to pixels and edges connecting neighboring pixels



4-connected or 8-conneted

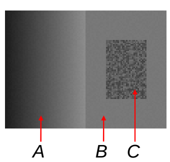
- Weight of edge is magnitude of intensity difference between connected pixels
- A *segmentation*, S, is a partition of V such that each $C \in S$ is connected

Important Characteristics

- Efficiency
  - Run in time essentially linear in the number of image pixels
    - With low constant factors
    - E.g., compared to edge detection
- Understandable output
  - Way to describe what algorithm does
    - E.g., Canny edge operator and step edge plus noise
- Not purely local
  - Perceptually important

Motivating Example

- Purely local criteria are inadequate
  - Difference along border between A and B is less than differences within C



- Criteria based on piecewise constant regions are inadequate
  - Will arbitrarily split A into subparts

MST Based Approaches

- Graph-based representation
  - Nodes corresponding to pixels, edge weights are intensity difference between connected pixels
- Compute minimum spanning tree (MST)
  - Cheapest way to connect all pixels into single component or "region"
- Selection criterion
  - Remove certain MST edges to form components
    - Fixed threshold
    - Threshold based on neighborhood
      - How to find neighborhood

Component Measure

- Don't consider just local edge weights in constructing MST
  - Consider properties of two components being merged when adding an edge
- Kruskal's MST algorithm adds edges from lowest to highest weight
  - Only if edges connect distinct components
- Apply criterion based on components to further filter added edges
  - Form of criterion limited by considering edges weight ordered

Measuring Component Difference

- Let *internal difference* of a component be maximum edge weight in its MST

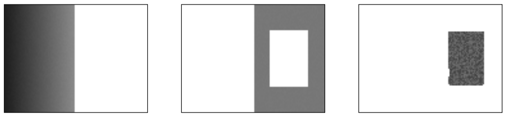
$$Int(C) = \max_{e \in MST(C,E)} w(e)$$

  - Smallest weight such that all pixels of C are connected by edges of at most that weight
- Let *difference* between two components be minimum edge weight connecting them

$$Dif(C_1,C_2) = \min_{v_i \in C_1, v_j \in C_2} w((v_i,v_j))$$

  - Note: infinite if there is no such edge

Regions Found by this Approach



A B C

- Three main regions plus a few small ones
- Why the algorithm stops growing these
  - Weight of edges between A and B large wrt max weight MST edges of A and of B
  - Weight of edges between B and C large wrt max weight MST edge of B (but not of C)

Closely Related Problems Hard

- What appears to be a slight change
  - Make *Dif* be quantile instead of min

    $$\text{k-th}_{\, v_i \in C_1,\, v_j \in C_2}\ w((v_i,v_j))$$

  - Desirable for addressing “cheap path” problem of merging based on one low cost edge
- Makes problem NP hard
  - Reduction from min ratio cut
    - Ratio of “capacity” to “demand” between nodes
- Other methods that we will see are also NP hard and approximated in various ways

Some Example Segmentations

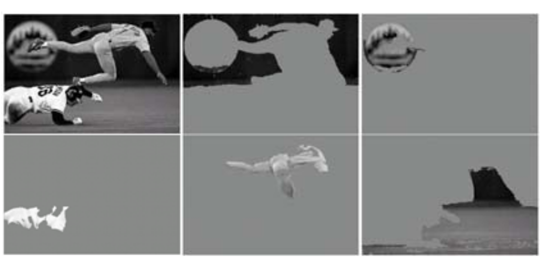
k=300
320 components larger than 10

k=200
323 components larger than 10

Simple Object Examples

Monochrome Example

- Components locally connected (grid graph)
  - Sometimes not desirable



Beyond Grid Graphs

- Image segmentation methods using affinity (or cost) matrices
  - For each pair of vertices v_i, v_j an associated weight w_{ij}
    - Affinity if larger when vertices more related
    - Cost if larger when vertices less related
  - Matrix $W=[\ w_{ij}\ ]$ of affinities or costs
    - W is large, avoid constructing explicitly
    - For images affinities tend to be near zero except for pixels that are nearby
      - E.g., decrease exponentially with distance
    - W is sparse

Cut Based Techniques

- For costs, natural to consider minimum cost cuts
  - Removing edges with smallest total cost, that cut graph in two parts
  - Graph only has non-infinite-weight edges
- For segmentation, recursively cut resulting components
  - Question of when to stop
- Problem is that cuts tend to split off small components

Normalized Cuts

- A number of normalization criteria have been proposed
- One that is commonly used

$$\text{Ncut}(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)}$$

- Where cut(A,B) is standard definition

$$\sum_{i\in A, j\in B} w_{ij}$$

- And $\text{assoc}(A,V) = \sum_j \sum_{i\in A} w_{ij}$

Computing Normalized Cuts

- Has been shown this is equivalent to an integer programming problem, minimize

$$\frac{y^T (D-W)y}{y^T D y}$$

- Subject to the constraint that $y_i \in \{1,b\}$ and $y^T D 1 = 0$
  - Where 1 vector of all 1's
- W is the affinity matrix
- D is the degree matrix (diagonal)

$$D(i,i) = \sum_j w_{ij}$$

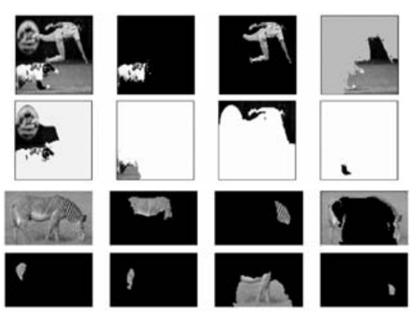
Approximating Normalized Cuts

- Integer programming problem NP hard
  - Instead simply solve continuous (real-valued) version – relaxation method
  - This corresponds to finding second smallest eigenvector of

$$(D-W)y_i = \lambda_i D y_i$$

- Widely used method
  - Works well in practice
    - Large eigenvector problem, but sparse matrices
    - Often resolution reduce images, e.g, 100x100
  - But no longer clearly related to cut problem

Normalized Cut Examples



Spectral Methods

- Eigenvectors of affinity and normalized affinity matrices
- Widely used outside computer vision for graph-based clustering
  - Link structure of web pages, citation structure of scientific papers
  - Often directed rather than undirected graphs

Segmentation

- Many other methods
  - Graph-based techniques such as the ones illustrated here have been most widely used and successful
  - Techniques based on Markov Random Field (MRF) models have underlying statistical model
    - Relatively widespread use for medical image segmentation problems
  - Perhaps most widely used non-graph-based method is simple local iterative update procedure called Mean Shift

Some Segmentation References

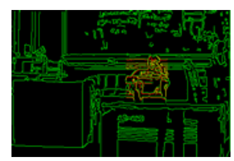
- J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* ,vol. 22, no. 8, pp. 888-905, 2000.
- P. Felzenszwalb and D. Huttenlocher, "Efficient Graph Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167-181, 2004.
- D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 603-619, 2002.

Recognition

- Specific objects
  - Much of the history of object recognition has been focused on recognizing specific objects in images
    - E.g., a particular building, painting, etc.
- Generic categories
  - More recently focus has been on generic categories of objects rather than specific individuals
    - E.g., faces, cars, motorbikes, etc.

Recognizing Specific Objects

- Approaches tend to be based on geometric properties of the objects
  - Comparing edge maps: Hausdorff matching
  - Comparing sparse features extracted from images: SIFT-based matching



Hausdorff Distance

- Classical definition
  - Directed distance (not symmetric)
    - $h(A,B) = \max_{a \in A} \min_{b \in B} \|a-b\|$
  - Distance (symmetry)
    - $H(A,B) = \max(h(A,B), h(B,A))$
- Minimization term is simply a distance transform of B
  - $h(A,B) = \max_{a \in A} D_B(a)$
  - Maximize over selected values of DT
- Not robust, single "bad match" dominates

Distance Transform Definition

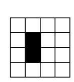
- Set of points, P, some distance $\|\bullet\|$

$$D_P(x) = \min_{y \in P} \|x - y\|$$

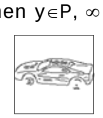
  - For each location x distance to nearest y in P
  - Think of as cones rooted at each point of P
- Commonly computed on a grid Γ using

$$D_P(x) = \min_{y \in \Gamma} (\|x - y\| + 1_P(y))$$

  - Where $1_P(y) = 0$ when $y \in P$, ∞ otherwise



2	1	2	3
1	0	1	2
1	0	1	2
2	1	2	3

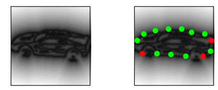


Hausdorff Matching

- Best match
  - Minimum fractional Hausdorff distance over given space of transformations
- Good matches
  - Above some fraction (rank) and/or below some distance
- Each point in (quantized) transformation space defines a distance
  - Search over transformation space
    - Efficient branch-and-bound “pruning” to skip transformations that cannot be good

Hausdorff Matching

- Partial (or fractional) Hausdorff distance to address robustness to outliers
  - Rank rather than maximum
    - $h_k(A,B) = kth_{a\in A} \min_{b\in B} \|a-b\| = kth_{a\in A} D_B(a)$
  - K-th largest value of D_B at locations given by A
  - Often specify as fraction f rather than rank
    - 0.5, median of distances; 0.75, 75th percentile



1,1,2,2,3,3,3,3,4,4,5,12,14,15

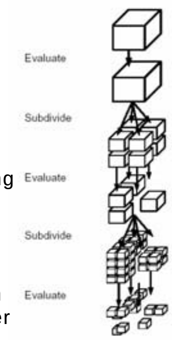
.25 .5 .75 1.0

Fast Hausdorff Search

- Branch and bound hierarchical search of transformation space
- Consider 2D transformation space of translation in x and y
  - (Fractional) Hausdorff distance cannot change faster than linearly with translation
    - Similar constraints for other transformations
  - Quad-tree decomposition, compute distance for transform at center of each cell
    - If larger than cell half-width, rule out cell
    - Otherwise subdivide cell and consider children

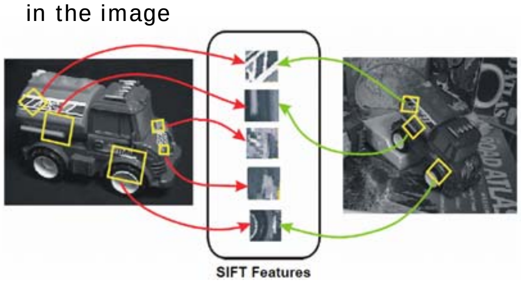
Branch and Bound Illustration

- Guaranteed (or admissible) search heuristic
  - Bound on how good answer could be in unexplored region
    - Cannot miss an answer
  - In worst case won’t rule anything out
- In practice rule out vast majority of transformations
  - Can use even simpler tests than computing distance at cell center



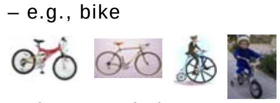
SIFT Feature Matching

- Sparse local features, invariant to changes in the image

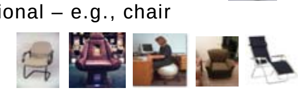


Object Category Recognition

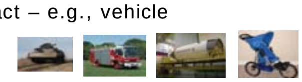
- Generic classes rather than specific objects
  - Visual – e.g., bike



  - Functional – e.g., chair



  - Abstract – e.g., vehicle

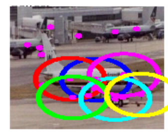


Recognition Cues

- Appearance
  - Patterns of intensity or color, e.g., tiger fur
  - Sometimes measured locally, sometimes over entire object
- Geometry
  - Spatial configuration of parts or local features
    - E.g., face has eyes above nose above mouth
- Early approaches relied on geometry (1960-80) later ones on appearance (1985-95), more recently using both

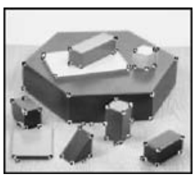
Using Appearance and Geometry

- Constellations of parts [FPZ03]
  - Detect affine-invariant features
    - E.g., corners without preserving angle
  - Use Gaussian spatial model of how feature locations vary within category (n x n covariance)
  - Match the detected features to spatial model



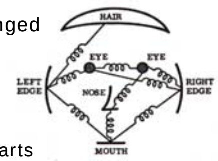
Problems With Feature Detection

- Local decisions about presence or absence of features are difficult and error prone
  - E.g., often hard to determine whether a corner is present without more context



Spatial Models Without Feature Detection

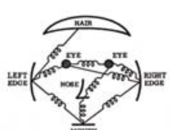
- Pictorial structures [FE73]
  - Model consists of parts arranged in deformable configuration
    - Match cost function for each part
    - Deformation cost function for each connected pair of parts



- Intuitively natural notion of parts connected by springs
  - "Wiggle around until fits" – no feature detection
  - Abandoned due to computational difficulty

Formal Definition of Model

- Object modeled by graph, $M=(V,E)$
  - Parts $V=(v_1, \ldots, v_m)$
  - Spatial relations $E=\{e_{ij}\}$
    - Gaussian on relative locations for pair of parts i,j
- Spatial prior $P_M(L)$ on configurations of parts $L=(\ell_1, \ldots, \ell_m)$
  - Where ℓ_i over discrete configuration space
    - E.g., translation, rotation, scale

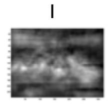


7 nodes
9 edges
(out of 21)

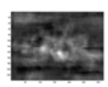
Single Overall Estimation Problem

- Likelihood of image given parts at specific configuration
  - E.g., under translation
- Degree to which configuration fits prior spatial model
- No error-prone local feature detection step
- Tractability depends on graph structure
  - E.g., for trees

I



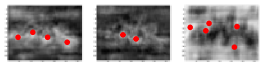
$P_M(I|\ell_1)$



$P_M(I|\ell_2)$

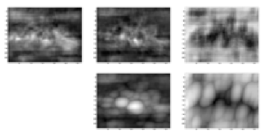
Single Estimation vs. Feature Detection

- Feature based
  - Local feature detection (threshold likelihood)
  - "Matching" techniques that handle missing and extra features



Detected Locations of Individual Features

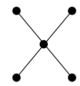
- Single estimation
  - Determine feature responses (likelihood)
  - Dynamic programming techniques to combine with spatial model (prior)



Transform Feature Maps Using Spatial Model

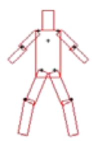
Graphical Models

- Probabilistic model
  - Collection of random variables with explicit dependencies between certain pairs
- Undirected edges – dependencies not causality
  - Markov random field (MRF)
- Reachability corresponds to (conditional) independence
  - E.g., case of star graph

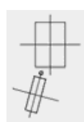


Tree Structured Models

- Kinematic structure of animate objects
  - Skeleton forms tree
  - Parts as nodes, joints as edges



- 2D image of joint
  - Spatial configuration for pair of parts
  - Relative orientation, position and scale (foreshortening)

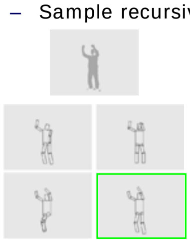


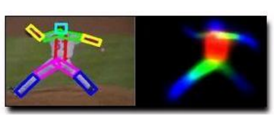
Best Match (MAP Estimate)

- All possible spatial configurations "considered" – most eliminated implicitly
  - Dynamic programming for efficiency
- Example using simple binary silhouette for appearance
  - Model error, min cost match not always "best"

Sampling (Total Evidence)

- Compute (factored) posterior distribution
- Efficiently generate sample configurations
  - Sample recursively from a "root part"





Used by best 2D human pose detection techniques, e.g. [RFZ05]

Single Estimation Approach

- Single estimation more accurate (and faster) than using feature detection
  - Optimization approach [CFH05,FPZ05] for star or k-fan vs. feature detection for full joint Gaussian [FPZ03]
  - 6 parts under translation, Caltech-4 dataset
  - Single class, equal ROC error

	Airplane	Motorbike	Faces	Cars
Feat. Det. [FPZ03]	90.2%	92.5%	96.4%	90.3%
Est.-Star [FPZ05]	93.6%	97.3%	90.3%	87.7%
Est.-Fan [CFH05]	93.3%	97.0%	98.2%	92.2%

Learning the Models

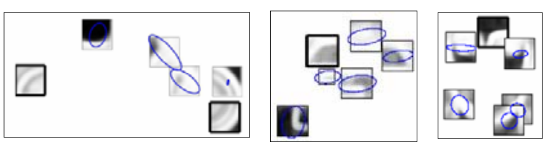
- [FPZ05] uses feature detection to learn models under weakly supervised regime
  - Know only which training images contain instances of the class, no location information
- [CFH05] does not use feature detection but requires extensive supervision
  - Know locations of all the parts in all the positive training images
- Investigate weak supervision but without relying on feature detection

Weakly Supervised Learning

- Consider large number of initial patch models to generate possible parts
- Generate all pairwise models formed by two initial patches – compute likelihoods
- Consider all sets of reference parts for fixed k
- Greedily add parts based on likelihood to produce initial model
- EM-style hill climbing to improve model

Example Learned Models

- Six part models, weak supervision
  - Black borders illustrate reference parts
  - Ellipses illustrate spatial uncertainty with respect to reference parts



Motorbike 2-fan Car (rear) 1-fan Face 1-fan

Detection Examples

Some Recognition References

- D.P. Huttenlocher, G.A. Klanderman, W.A. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence* ,vol. 15, no. 9, pp. 850-863, 1993.
- D.G. Lowe, "Object recognition from local scale-invariant features," *IEEE Conference on Computer Vision and Pattenr Recognition*, pp. 1150-1157, 1999.
- D. Crandall, P. Felzenszwalb and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," *IEEE Conference on Computer Vision and Pattenr Recognition*, pp. 10-17, 2005.