Lecture 15: Model-based recognition

Tuesday, Nov 6

Prof. Kristen Grauman

Graduate student extension ideas

- Estimate fundamental matrix from image correspondences
- Use disparity/depth cues to aid segmentation
- Add geometry verification steps to SIFT matching

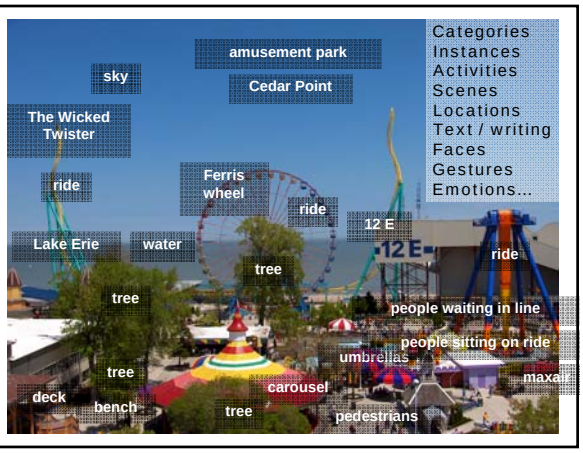
Last time

- Invariant features: distinctive matches possible in spite of significant view change, useful for wide baseline stereo
- Bag of words representation: quantize feature space to make discrete set of visual words
  - Summarize image by distribution of words
  - Index individual words
- Inverted index: pre-compute index to enable faster search at query time

Note: so far, we've only considered the *indexing* problem, and have not incorporated the geometry among the features we match.

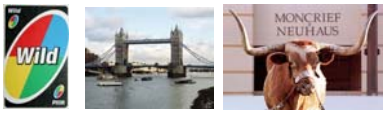
Today

- Overview of the recognition problem
- Model-based recognition
  - Hypothesize and test
    - Interpretation trees
    - Alignment, pose consistency
    - Pose clustering
    - Verification



Possible levels of recognition

Categories



Specific objects

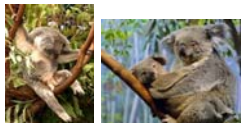
Functional

Challenges

Geometric, photometric transformations for different views of the same object.

Challenges

Illumination



Object pose, articulations

Clutter

Occlusions

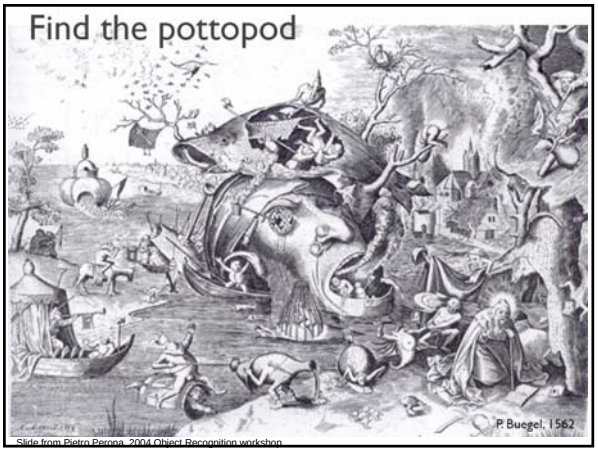
Intra-class appearance

Viewpoint

Scale: how many things need to be recognized?

This is a pottopod

Slide from Pietro Perona, 2004 Object Recognition workshop



Slide from Pietro Perona, 2004 Object Recognition workshop

Scope of the recognition problem

- In some cases, want to engineer solution to particular practical problem; constraints can make it manageable.
- In general, want understanding of human object recognition, and/or system that can mimic it; much more difficult.

Inputs/outputs/assumptions

- What **input** is available?
  - Static grayscale image
  - 3D range data
  - Video sequence
  - Multiple calibrated cameras
  - Segmented data, unsegmented data
  - CAD model
  - Labeled data, unlabeled data, partially labeled data

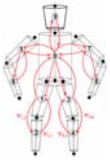
Inputs/outputs/assumptions

- What is the **goal**?
  - Say yes/no as to whether an object present in image
  - Determine pose of an object, e.g. for robot to grasp it
  - Categorize all objects
  - Forced choice from pool of categories
  - Bounding box on object
  - Full segmentation
  - Build a model of an object category

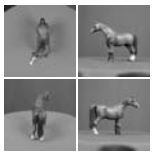
Primary issues

- How to **represent** a category or object
- How to perform **recognition** (classification, detection) with that representation
- How to **learn** models, new categories/objects

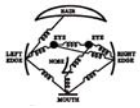
Representation



3-D models



View-based

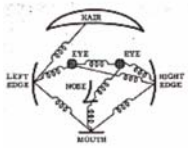


Parts + structure

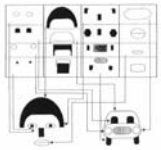
Appearance-based

Bag of features

Models: appearance+shape



Fischler & Elschlager, 1973



Perrett & Oram, 1993

Perona et al. '95

Schmid '99, Lowe '99, Moreels '04

Belongie et al. '02

Learning

- What defines a category/class?
- What distinguishes classes from one another?
- How to understand the connection between the real world and what we observe?
- What features are most informative?
- What can we do without human intervention?
- Does previous learning experience help learn the next category?

Analyze or learn?

Viewpoint

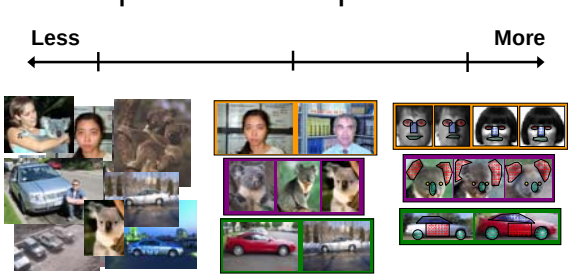
Feature

Lighting

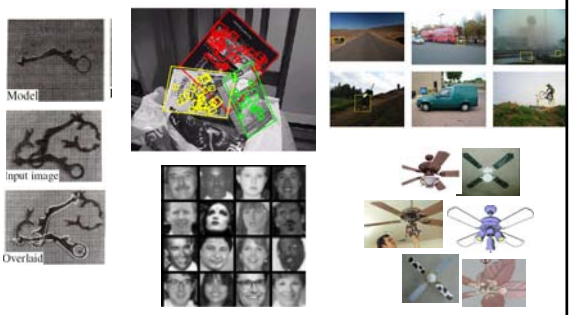
Deformations

Slide from Pietro Perona, 2004 Object Recognition workshop

Spectrum of supervision



Evolution of recognition focus



1980s | 1990s to early 2000s | Currently

What works

- Barcode readers
- Fingerprint recognition (Belongie & Bjorn, Digital Persona)
- Flat-textured object recognition (D. Lowe, Evolution Robotics)
- Face detection(?) (Viola-Jones, INTEL open vision library)

Slide from Pietro Perona, 2004 Object Recognition workshop

Key challenges today

- Scaling to large numbers of categories, large image databases
- Descriptors for categories: flexibility vs. discrimination
- Descriptors for objects: scaling
- Learning with cluttered examples, “weak” supervision
- Incremental learning of categories
- Unsupervised learning
- Multi-modal data

Today

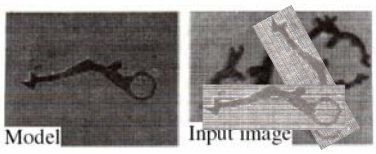
- Overview of the recognition problem
- Model-based recognition
  - Hypothesize and test
    - Interpretation trees
    - Alignment, pose consistency
    - Pose clustering
    - Verification

Model-based recognition

- Which image features correspond to which features on which object model in the “modelbase”?
- If enough match, *and* they match well with a particular transformation for given camera model, then
  - Identify the object as being there
  - Estimate pose relative to camera

Hypothesize and test: main idea

- Given model of object
- New image: hypothesize object identity and pose
- Render object in camera
- Compare rendering to actual image: if close, good hypothesis.



Issues

- How to form a hypothesis on object identity and pose?
- How to verify the hypothesis?

How to form a hypothesis?

Given a particular model object, we can estimate the *correspondences* between image and model features

Use correspondence to estimate camera pose relative to object coordinate frame

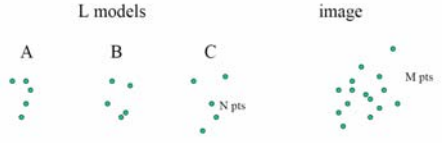
Generating hypotheses

We want a good correspondence between model features and image features.

– Brute force?

Brute force hypothesis generation

- For every possible model, try every possible subset of image points as matches for that model's points.
- Say we have L objects with N features, M features in image



What is the computational complexity?

Generating hypotheses

We want a good correspondence between model features and image features.

– Brute force?

– Prune search via geometric or relational constraints: interpretation tree

– Pose consistency: use subsets of features to estimate larger correspondence

– Voting, pose clustering

Interpretation tree

- Represents search space of assignments between model parts and image parts

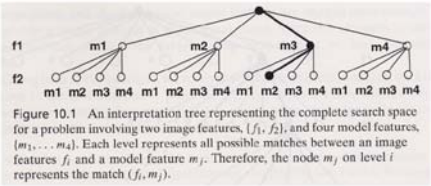


Figure 10.1 An interpretation tree representing the complete search space for a problem involving two image features, $\{f_1, f_2\}$, and four model features, $\{m_1, \ldots m_4\}$. Each level represents all possible matches between an image features f_i and a model feature m_j. Therefore, the node m_j on level i represents the match (f_i, m_j).

- Classic AI type of approach

Figure from Trucco & Verri

Interpretation tree for pruning

Given
- object model features
- image features
- way to compare features symbolically
- list of constraints that model features must satisfy

- Goal: find a mapping between model features and image features such that the features match correctly *and* satisfy the geometric constraints, without requiring brute force search

Interpretation tree: example

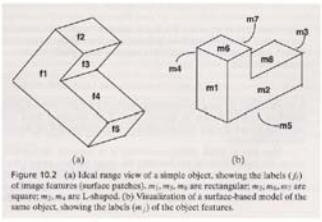


Figure 10.2 (a) Ideal range view of a simple object, showing the labels (f_i) of image features (surface patches). m_1, m_2, m_3 are rectangular; m_5, m_6, m_7 are square; m_4, m_8 are L-shaped. (b) Visualization of a surface-based model of the same object, showing the labels (m_j) of the object features.

Each feature is a rectangle, square, or L
- Get list of features for model
- Get list of features in image
- Constraint : features match only if they are the same type

Figure from Trucco & Verri

Interpretation tree: example

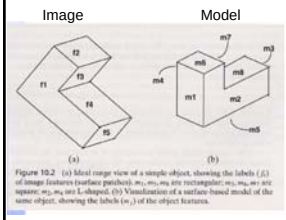


Figure 10.2 (a) Ideal range view of a simple object, showing the labels (f_i) of image features (surface patches). m_1, m_2, m_3 are rectangular; m_5, m_6, m_7 are square; m_4, m_8 are L-shaped. (b) Visualization of a surface-based model of the same object, showing the labels (m_j) of the object features.

Depth-first search for assignment that does not violate constraints

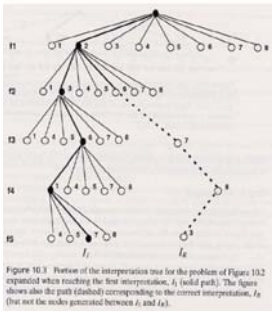


Figure 10.3 Portion of the interpretation tree for the problem of Figure 10.2 expanded when reaching the first interpretation, I_1 (solid path). The figure shows also the path (dashed) corresponding to the correct interpretation, I_R (but not the nodes generated between I_1 and I_R).

Figure from Trucco & Verri

Interpretation tree for pruning

- Tree gives all possible model-image feature assignments
- Depth-first search, recursive back-track
- Prune/terminate when constraints violated

  (Note: constraints could be relational, geometric; e.g., adjacency between parts)

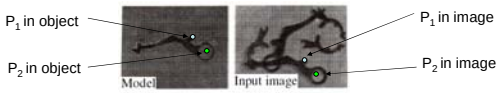
- Intent: search time reduced from brute force because many possible assignments can terminate early

Pose consistency / alignment

- Key idea:
  - If we find good correspondences for a small set of features, it is easy to obtain correspondences for a much larger set.
- Strategy:
  - Generate hypotheses using small numbers of correspondences (how many depends on camera type)
  - Backproject: transform all model features to image features
  - Verify

2d affine mappings

- Say camera is looking down perpendicularly on planar surface



- We have two coordinate systems (object and image), and they are related by some affine mapping (rotation, scale, translation, shear).

We left off here on Tuesday, to be continued Thursday.

Coming up

- Appearance based recognition, faces
- Read FP 22.1-22.3