Recognizing object instances

Monday, April 4
Prof. Kristen Grauman
UT-Austin

Some pset 3 results!

Brian Bates

Christopher Tosh

Brian Nguyen

Che-Chun Su
Today: instance recognition

- Motivation – visual search
- Visual words
  - quantization, index, bags of words
- Spatial verification
  - affine; RANSAC, Hough
- Other text retrieval tools
  - tf-idf, query expansion
- Example applications

Multi-view matching

Matching two given views for depth

Search for a matching view for recognition

Indexing local features

- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)
Indexing local features

- When we see close points in feature space, we have similar descriptors, which indicates similar local content.

Database images

Descriptor's feature space

Query image

Easily can have millions of features to search!

Kristen Grauman

Indexing local features: inverted file index

- For text documents, an efficient way to find all pages on which a word occurs is to use an index...

- We want to find all images in which a feature occurs.

- To use this idea, we'll need to map our features to “visual words”.

Kristen Grauman

Visual words

- Map high-dimensional descriptors to tokens/words by quantizing the feature space

  - Quantize via clustering, let cluster centers be the prototype “words”
  
  - Determine which word to assign to each new image region by finding the closest cluster center.

Word #2

Descriptor's feature space

Kristen Grauman

Inverted file index

- Database images are loaded into the index mapping words to image numbers

Kristen Grauman

Visual vocabulary formation

Issues:

- Vocabulary size, number of words
- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- Unsupervised vs. supervised
- What corpus provides features (universal vocabulary?)

Kristen Grauman

Visual words

- Example: each group of patches belongs to the same visual word

Figure from: Sivic & Zisserman, ICCV 2003

Kristen Grauman
Inverted file index

New query image

- New query image is mapped to indices of database images that share a word.

Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant one. Our perception of the world around us is based essentially on messages that reach the brain from our eyes. For a long time it was thought that the retinal image was transmitted point by point to visual centers in the brain; 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.

Bags of visual words

- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.

Comparing bags of words

- Rank frames by normalized scalar product between their (possibly weighted) occurrence counts—nearest neighbor search for similar images.

\[
\text{sim}(d_j,q) = \frac{d_j \cdot q}{||d_j||_2 \cdot ||q||_2}
\]

\[
= \frac{\sum_{i=1}^{V} d_{j}(i) \cdot q(i)}{\sqrt{\sum_{i=1}^{V} d_{j}(i)^2} \cdot \sqrt{\sum_{i=1}^{V} q(i)^2}}
\]

for vocabulary of \(V\) words
Inverted file index and bags of words similarity

1. Extract words in query
2. Inverted file index to find relevant frames
3. Compare word counts

Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Vocabulary size

Results for recognition task with 6347 images

Vocabulary Trees: hierarchical clustering for large vocabularies

- Tree construction:
Vocabulary trees: complexity

Number of words given tree parameters:
- branching factor and number of levels
- Word assignment cost vs. flat vocabulary

Visual words/bags of words

+ flexible to geometry / deformations / viewpoint
+ compact summary of image content
+ provides vector representation for sets
+ very good results in practice

- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear
- basic model ignores geometry – must verify afterwards, or encode via features
Instance recognition:
remaining issues
• How to summarize the content of an entire image? And gauge overall similarity?
• How large should the vocabulary be? How to perform quantization efficiently?
• Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
• How to score the retrieval results?

Spatial Verification

Both image pairs have many visual words in common.

Spatial Verification: two basic strategies
• RANSAC
  – Typically sort by BoW similarity as initial filter
  – Verify by checking support (inliers) for possible transformations
    • e.g., “success” if find a transformation with > N inlier correspondences
• Generalized Hough Transform
  – Let each matched feature cast a vote on location, scale, orientation of the model object
  – Verify parameters with enough votes

RANSAC verification

Recall: Fitting an affine transformation
Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  m_1 & m_2 & x_i \\
  m_3 & m_4 & y_i
\end{bmatrix} + \begin{bmatrix}
  t_x \\
  t_y
\end{bmatrix}
\begin{bmatrix}
  x_i & y_i & 0 & 0 & 1 & 0 \\
  0 & 0 & x_i & y_i & 1 & 0
\end{bmatrix}\]
RANSAC verification

Video Google System
1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

• Demo online at:
  http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html

Example Applications

Mobile tourist guide
• Self-localization
• Object/building recognition
• Photo/video augmentation

Application: Large-Scale Retrieval

Query Results from 5k Flickr images (demo available for 100k set)

Web Demo: Movie Poster Recognition

50'000 movie posters indexed
Query-by-image from mobile phone available in Switzerland

Scoring retrieval quality

Spatial Verification: two basic strategies

• RANSAC
  – Typically sort by BoW similarity as initial filter
  – Verify by checking support (inliers) for possible transformations
    • e.g., “success” if find a transformation with > N inlier correspondences

• Generalized Hough Transform
  – Let each matched feature cast a vote on location, scale, orientation of the model object
  – Verify parameters with enough votes

Voting: Generalized Hough Transform

• If we use scale, rotation, and translation invariant local features, then each feature match gives an alignment hypothesis (for scale, translation, and orientation of model in image).

Gen Hough Transform details (Lowe’s system)

  • Training phase: For each model feature, record 2D location, scale, and orientation of model (relative to normalized feature frame)
  • Test phase: Let each match between a test SIFT feature and a model feature vote in a 4D Hough space
    • Use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times image size for location
    • Vote for two closest bins in each dimension
    • Find all bins with at least three votes and perform geometric verification
      • Estimate least squares affine transformation
      • Search for additional features that agree with the alignment

Example result

Recall: difficulties of voting

- Noise/clutter can lead to as many votes as the true target
- Bin size for the accumulator array must be chosen carefully
- In practice, good idea to make broad bins and spread votes to nearby bins, since verification stage can prune bad vote peaks.

Gen Hough vs RANSAC

**GHT**
- Single correspondence -> vote for all consistent parameters
- Represents uncertainty in the model parameter space
- Linear complexity in number of correspondences and number of voting cells; beyond 4D vote space impractical
- Can handle high outlier ratio

**RANSAC**
- Minimal subset of correspondences to estimate model -> count inliers
- Represents uncertainty in image space
- Must search all data points to check for inliers each iteration
- Scales better to high-d parameter spaces

What else can we borrow from text retrieval?

- **tf-idf weighting**
  - Term frequency – Inverse document frequency
  - Describe frame by frequency of each word within it, downweight words that appear often in the database
  - (Standard weighting for text retrieval)
  
  $t_i = \frac{n_{id} \cdot \log \frac{N}{n_i}}{n_d}$

Query expansion

Query: *golf green*

Results:
- How can the grass on the greens at a golf course be so perfect?
- For example, a skilled golfer expects to reach the green on a par-four hole in ...
- Manufactures and sells synthetic golf putting greens and mats.

Irrelevant result can cause a 'topic drift':
Recognition via alignment

**Pros:**
- Effective when we are able to find reliable features within clutter
- Great results for matching specific instances

**Cons:**
- Scaling with number of models
- Spatial verification as post-processing – not seamless, expensive for large-scale problems
- Not suited for category recognition.

Summary

- **Matching local invariant features**
  - Useful not only to provide matches for multi-view geometry, but also to find objects and scenes.
- **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
- **Recognition of instances via alignment**: matching local features followed by spatial verification
  - Robust fitting: RANSAC, GHT

Example

As an amateur shot of M31, by Filippe Ciferri (c.2007) from flickr.com
http://astrometry.net/gallery.html

Example

A beautiful image of Bode’s nebula (c.2007) by Peter Bresseler, from starlightfriend.de
http://astrometry.net/gallery.html

Example

A shot of the Great Nebula by Jerry Lodriguss (c.2006), from astropix.com
http://astrometry.net/gallery.html