Overview

• Image Features and Categorization
  – Histograms
  – Bags of features/visual words
  – Vocabulary trees
  – Spatial layout and context (preview)

– Based on slides by K. Grauman, D. Hoiem and S. Lazebnik
Image Features and Categorization
Training phase

Training Images

Training

Image Features

Classifier Training

Trained Classifier

Training Labels
Testing phase

Training phase:
- Training Images
- Training Labels
- Image Features
- Classifier Training
- Trained Classifier

Testing phase:
- Test Image
- Image Features
- Trained Classifier
- Prediction Outdoor
Testing phase

Training phase:
- Training Images
  - Image Features
    - Classifier Training
      - Trained Classifier

Testing phase:
- Test Image
  - Image Features
    - Trained Classifier
      - Prediction Outdoor
Q: What are good features for recognizing a beach?
Q: What are good features for...

- recognizing fabrics?
Q: What are good features for...  
• recognizing a mug?
What are the right features?

Depends on what we want to know!

• Object: shape
  – Local shape info, shading, shadows, texture
• Scene: geometric layout
  – Linear perspective, gradients, line segments
• Material properties: albedo, feel, hardness
  – Color, texture
• Action: motion
  – Optical flow, tracked points
General Principles of Representation

• Coverage
  – Ensure that all relevant information is captured

• Conciseness
  – Minimize number of features without sacrificing coverage

• Directness
  – Ideal features are independently useful for prediction
Image Representations

• Templates
  – Intensity, gradients, etc.

• Histograms
  – Color, texture, SIFT descriptors, etc.

• Average of features
Image representations: histograms

Global histogram
- Represent distribution of features
  – Color, texture, depth, ...

Images from Dave Kauchak
Image representations: histograms

- Data samples in 2D
Image representations: histograms

- Probability or count of data in each bin
- Marginal histogram on feature 1
Image representations: histograms

• Marginal histogram on feature 2
Image representations: histograms

- Joint histogram
Modeling multi-dimensional data

Joint histogram
- Requires lots of data
- Loss of resolution to avoid empty bins

Marginal histogram
- Requires independent features
- More data/bin than joint histogram
Computing histogram distance

• Histogram intersection

\[ \text{histint}(h_i, h_j) = 1 - \sum_{m=1}^{K} \min(h_i(m), h_j(m)) \]

• Chi-squared Histogram matching distance

\[ \chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^{K} \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)} \]

• Earth mover’s distance
  (Cross-bin similarity measure)
  – minimal cost paid to transform one distribution into the other

[Rubner et al. The Earth Mover's Distance as a Metric for Image Retrieval, IJCV 2000]
Histograms: implementation issues

- **Quantization**
  - Grids: fast but applicable only with few dimensions
  - Clustering: slower but can quantize data in higher dimensions (see next slides)

- **Matching**
  - Histogram intersection or Euclidean distance may be faster
  - Chi-squared distance often works better
  - Earth mover’s distance is good when nearby bins represent similar values

- Few Bins
  - Need less data
  - Coarser representation

- Many Bins
  - Need more data
  - Finer representation
What kind of things do we compute histograms of?

- Color
  - L*a*b* color space
  - HSV color space

- Texture (filter banks or descriptors)
Bags of Features/Visual Words
Bags of features
Origin 1: Texture recognition

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters
Origin 1: Texture recognition

Universal texton dictionary

histogram

Origin 2: Bag-of-words models

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin 2: Bag-of-words models

Origin 2: Bag-of-words models

Bag-of-features steps

1. Extract local features
2. Learn “visual vocabulary”
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of “visual words”
Local feature extraction

- Regular grid or interest regions
Local feature extraction

Detect patches

Normalize patch

Compute descriptor

Slide credit: Josef Sivic
Local feature extraction

Slide credit: Josef Sivic
Learning the visual vocabulary
Learning the visual vocabulary

Slide credit: Josef Sivic
Learning the visual vocabulary

Clustering

Visual vocabulary

Clustering

Slide credit: Josef Sivic
Example codebook

Source: B. Leibe
Another codebook

Source: B. Leibe
1. Extract local features
2. Learn “visual vocabulary”
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of “visual words”
Visual vocabularies: Details

• How to choose vocabulary size?
  – Too small: visual words not representative of all patches
  – Too large: quantization artifacts, overfitting
  – Right size is application-dependent

• Improving efficiency of quantization
  – Vocabulary trees (Nister and Stewenius, 2006)

• Improving vocabulary quality
  – Discriminative/supervised training of codebooks
  – Sparse coding, non-exclusive assignment to codewords

• More discriminative bag-of-words representations
  – Fisher Vectors (Perronnin et al., 2007), VLAD (Jegou et al., 2010)

• Incorporating spatial information
Bags of features for action recognition

Space-time interest points

Bags of features for action recognition

Indexing local features

Kristen Grauman
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.
Indexing local features

• With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?
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
Text retrieval vs. image search

• What makes the problems similar, different?
Visual words: main idea

• Extract some local features from a number of images …

Slide credit: D. Nister, CVPR 2006

e.g., SIFT descriptor space: each point is 128-dimensional
Visual words: main idea
Each point is a local descriptor, e.g. SIFT vector.
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.
Visual words

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

Figure from Sivic & Zisserman, ICCV 2003

Kristen Grauman
Inverted file index

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

<table>
<thead>
<tr>
<th>Word #</th>
<th>Image #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1, 2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>2</td>
</tr>
</tbody>
</table>
Inverted file index

When will this give us a significant gain in efficiency?

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

<table>
<thead>
<tr>
<th>Word #</th>
<th>Image #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1, 2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>2</td>
</tr>
</tbody>
</table>

Kristen Grauman
• If a local image region is a visual word, how can we summarize an image (the document)?
Comparing bags of words

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

\[
\text{sim}(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\|\|q\|}
\]

\[
= \frac{\sum_{i=1}^{V} d_j(i) \times q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} \times \sqrt{\sum_{i=1}^{V} q(i)^2}}
\]

for vocabulary of \( V \) words

Kristen Grauman
**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)

\[ t_i = \frac{n_{id}}{n_d} \log \left( \frac{N}{n_i} \right) \]

- Number of occurrences of word i in document d
- Total number of documents in database
- Number of words in document d
- Number of documents word i occurs in, in whole database

Kristen Grauman
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`:


Slide credit: Ondrej Chum
Query Expansion

Results

Query image

Spatial verification

New results

New query

Chum, Philbin, Sivic, Isard, Zisserman: Total Recall…, ICCV 2007

Slide credit: Ondrej Chum
Bags of words for content-based image retrieval

Visually defined query

“Find this clock”

“Groundhog Day” [Rammis, 1993]

“Find this place”
Example

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003
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

Query region

Retrieved frames
Is having the same set of visual words enough to identify the object or scene?

How to verify spatial agreement?
Spatial Verification

Both image pairs have many visual words in common.

Slide credit: Ondrej Chum
Spatial Verification

Only some of the matches are mutually consistent

Query

DB image with high BoW similarity

Query

DB image with high BoW similarity

Slide credit: Ondrej Chum
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 a transformation with > N inlier correspondences can be found

• 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.

\[
\begin{bmatrix}
    x'_i \\
    y'_i
\end{bmatrix} = \begin{bmatrix}
    m_1 & m_2 \\
    m_3 & m_4
\end{bmatrix}
\begin{bmatrix}
    x_i \\
    y_i
\end{bmatrix} + \begin{bmatrix}
    t_1 \\
    t_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
    m_1 \\
    m_2 \\
    m_3 \\
    m_4
\end{bmatrix}
\begin{bmatrix}
    x_i & y_i & 0 & 0 & 1 & 0 \\
    0 & 0 & x_i & y_i & 0 & 1 \\
    \ldots
\end{bmatrix}
= \begin{bmatrix}
    \ldots \\
    x'_i \\
    y'_i
\end{bmatrix}
\]
RANSAC verification
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).

Adapted from Lana Lazebnik
Voting: Generalized Hough Transform

- A hypothesis generated by a single match may be unreliable,
- So let each match vote for a hypothesis in Hough space
Generalized Hough Transform details

- **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.


Slide credit: Lana Lazebnik
Results

Background subtraction for model boundaries

Objects recognized,

Recognition in spite of occlusion
Difficulties of voting

- Noise/clutter can lead to as many votes as 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
Generalized 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
Vocabulary Trees: hierarchical clustering for large vocabularies

• Tree construction:

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree

• Training: Filling the tree

[Nister & Stewenius, CVPR’06]
Vocabulary Tree

• Training: Filling the tree

[Nister & Stewenius, CVPR’06]

K. Grauman, B. Leibe

Slide credit: David Nister
Vocabulary Tree

• Training: Filling the tree

[Nister & Stewenius, CVPR’06]
Vocabulary Tree

• Training: Filling the tree

[Nister & Stewenius, CVPR’06]
Vocabulary Tree

- Training: Filling the tree

K. Grauman, B. Leibe

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
What is the computational advantage of the hierarchical representation bag of words, vs. a flat vocabulary?
Vocabulary Tree

- Recognition

RANSAC verification

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Scoring retrieval quality

Query
Database size: 10 images
Relevant (total): 5 images

precision = #relevant / #returned
recall = #relevant / #total relevant

Results (ordered):
Bags of words: pros and cons

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

- basic model ignores geometry - must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear
Spatial Layout and Context
But what about spatial layout?

All of these images have the same color histogram
Spatial pyramid

Compute histogram in each spatial bin
Spatial pyramid

[LaNoek et al. CVPR 2006]
Results: Scene category dataset

Multi-class classification results
(100 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (vocabulary size: 16)</th>
<th>Strong features (vocabulary size: 200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0 (1 x 1)</td>
<td>45.3 ±0.5</td>
<td>56.2 ±0.6</td>
</tr>
<tr>
<td>1 (2 x 2)</td>
<td>53.6 ±0.3</td>
<td>56.2 ±0.6</td>
</tr>
<tr>
<td>2 (4 x 4)</td>
<td>61.7 ±0.6</td>
<td>64.7 ±0.7</td>
</tr>
<tr>
<td>3 (8 x 8)</td>
<td>63.3 ±0.8</td>
<td>66.8 ±0.6</td>
</tr>
</tbody>
</table>
## Results: Caltech101 dataset

Multi-class classification results (30 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (16)</th>
<th>Strong features (200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0</td>
<td>15.5 ±0.9</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>31.4 ±1.2</td>
<td>32.8 ±1.3</td>
</tr>
<tr>
<td>2</td>
<td>47.2 ±1.1</td>
<td>49.3 ±1.4</td>
</tr>
<tr>
<td>3</td>
<td>52.2 ±0.8</td>
<td><strong>54.0 ±1.1</strong></td>
</tr>
</tbody>
</table>
Region representation

- Segment the image into superpixels
- Use features to represent each image segment
Region representation

• Color, texture, BoW
  – Only computed within the local region

• Shape of regions

• Position in the image
Working with regions

- Spatial support is important - multiple segmentations

Geometric context [Hoiem et al. ICCV 2005]