Recognition

- Scene understanding / visual object categorization
- Pose clustering
- Object recognition by local features
- Image categorization
- Bag-of-features models
- Large-scale image search
Scene understanding

Scene categorization
- outdoor/indoor
- city/forest/factory/etc.
Scene understanding

Object detection
- find pedestrians
Scene understanding

- sky
- mountain
- building
- tree
- banner
- street lamp
- people
- market
Visual object categorization

~10,000 to 30,000
Visual object categorization

- **objects**
  - **animals**
    - ..... vertebrate
      - **mammals**
        - **tapir**
      - **birds**
        - **boar**
  - **plants**
  - **inanimate**
    - **natural**
    - **man-made**
      - **grouse**
      - **camera**
Visual object categorization

- Motorbikes
- Airplanes
- Faces
- Cars (Side)
- Cars (Rear)
- Spotted Cats
- Background
Visual object categorization

Recognition is all about modeling variability

- camera position
- illumination
- shape variations
- within-class variations
Visual object categorization

Within-class variations (why are they chairs?)
Pose clustering

Working in transformation space - main ideas:

- Generate many hypotheses of transformation image vs. model, each built by a tuple of image and model features
- Correct transformation hypotheses appear many times

Main steps:

- Quantize the space of possible transformations
- For each tuple of image and model features, solve for the optimal transformation that aligns the matched features
- Record a “vote” in the corresponding transformation space bin
- Find "peak" in transformation space
Pose clustering

Example: Rotation only. A pair of one scene segment and one model segment suffices to generate a transformation hypothesis.
Pose clustering


2-tuples of image and model corner points are used to generate hypotheses. (a) The corners found in an image. (b) The four best hypotheses found with the edges drawn in. The nose of the plane and the head of the person do not appear because they were not in the models.
Object recognition by local features

D.G. Lowe: Distinctive image features from scale-invariant keypoints. IJCV, 60(2): 91-110, 2004

- The SIFT features of training images are extracted and stored
- For a query image
  ✓ Extract SIFT features
  ✓ Efficient nearest neighbor indexing
  ✓ **Pose clustering** by Hough transform
  ✓ For clusters with >2 keypoints (object hypotheses): determine the optimal affine transformation parameters by least squares method; geometry **verification**
Object recognition by local features

Robust feature-based alignment (panorama)
Object recognition by local features

- Extract features
Object recognition by local features

- Extract features
- Compute *putative matches*
Object recognition by local features

- Extract features
- Compute *putative matches*
- Loop:
  - Hypothesize transformation $T$ (small group of putative matches that are related by $T$)
Object recognition by local features

- Extract features
- Compute *putative matches*
- Loop:
  - Hypothesize transformation \( T \) (small group of putative matches that are related by \( T \))
  - Verify transformation (search for other matches consistent with \( T \))
Object recognition by local features

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
  - *Verify* transformation (search for other matches consistent with $T$)
Object recognition by local features

Pose clustering by Hough transform \((\text{Hypothesize transformation } T)\):

- Each of the SIFT keypoints of input image specifies 4 parameters: 2D location, scale, and orientation.

- Each matched SIFT keypoint in the database has a record of the keypoint’s parameters relative to the training image in which it was found.

- We can create a Hough transform entry predicting the model location, orientation, and scale from the match hypothesis.
Object recognition by local features

Recognition: Cluster of 3 corresponding feature pairs
Object recognition by local features
Object recognition by local features
Image categorization

Training

Training Images → Image Features → Classifier Training → Trained Classifier

Testing

Test Image → Image Features → Trained Classifier → Prediction

Outdoor
Global histogram (distribution):
- color, texture, motion, …
- histogram matching distance
Image categorization

Cars found by color histogram matching

See Chapter “Inhaltsbasierte Suche in Bilddatenbanken”
Bag-of-features models

Object → Bag of ‘words’
Bag-of-features models

Origin 1: **Text recognition** by orderless document representation (frequencies of words from a dictionary), Salton & McGill (1983)
Bag-of-features models

Origin 2: 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*
Bag-of-features models
Bag-of-features models

We need to build a “visual” dictionary!

Bag-of-features models

Main step 1: Extract features (e.g. SIFT)
Bag-of-features models

Main step 2: Learn visual vocabulary (e.g. using clustering)
Bag-of-features models

visual vocabulary (cluster centers)
Bag-of-features models

Example: learned codebook

Source: B. Leibe
Bag-of-features models

Main step 3: Quantize features using “visual vocabulary”
Bag-of-features models

Main step 4: Represent images by frequencies of “visual words” (i.e., bags of features)
Bag-of-features models

Example: representation based on learned codebook
Main step 5: Apply any classifier to the histogram feature vectors

**category models**

Model space

Class 1

Class N
Bag-of-features models

Example:
Bag-of-features models

Table 2. Confusion matrix and mean rank for SVM ($k=1000$, linear kernel).

<table>
<thead>
<tr>
<th>True classes $\rightarrow$</th>
<th>faces</th>
<th>buildings</th>
<th>trees</th>
<th>cars</th>
<th>phones</th>
<th>bikes</th>
<th>books</th>
</tr>
</thead>
<tbody>
<tr>
<td>faces</td>
<td>98</td>
<td>14</td>
<td>10</td>
<td>10</td>
<td>34</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>buildings</td>
<td>1</td>
<td>63</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>trees</td>
<td>1</td>
<td>10</td>
<td>81</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>cars</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>85</td>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>phones</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>55</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>bikes</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>91</td>
<td>0</td>
</tr>
<tr>
<td>books</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>73</td>
</tr>
<tr>
<td>Mean ranks</td>
<td>1.04</td>
<td>1.77</td>
<td>1.28</td>
<td>1.30</td>
<td>1.83</td>
<td>1.09</td>
<td>1.39</td>
</tr>
</tbody>
</table>

Dictionary quality and size are very important parameters!
Bag-of-features models

Action recognition

Large-scale image search

Large-scale image search

Mobile tourist guide
- self-localization
- object/building recognition
- photo/video augmentation

Aachen Cathedral

[Quack, Leibe, Van Gool, CIVR’08]
Large-scale image search

Google Goggles
Use pictures to search the web.

Get Google Goggles
Android (2.1+ required)
Download from Android Market

Send Goggles to Android phone

New! iPhone (iOS 4.0 required)
Download from the App Store

Send Goggles to iPhone

Google Goggles in action
Click the icons below to see the different kinds of objects and places you can search for using Google Goggles.

Menu
Landmarks
Books
Contact Info
Artwork
Wine

Google goggles
Golden Gate Bridge
San Francisco, United States

Kapitel 14 “Recognition” – p. 45
Large-scale image search

Application: Image auto-annotation

Left: Wikipedia image
Right: closest match from Flickr
Sources

- Course materials from others (G. Bebis, J. Hays, S. Lazebnik, …)