Highlighted Project 2 Implementations
Vocabulary trees: complexity

Number of words given tree parameters: branching factor and number of levels
\[
\text{branching\_factor}^{\text{number\_of\_levels}}
\]

Word assignment cost vs. flat vocabulary
\[
O(k) \text{ for flat} \\
O(\log_{\text{branching\_factor}}(k) \times \text{branching\_factor})
\]

Is this like a kd-tree?
Yes, but with better partitioning and defeatist search.
This hierarchical data structure is lossy – you might not find your true nearest cluster.
110,000,000 Images in 5.8 Seconds
Higher branch factor works better (but slower)
Visual words/bags of words

+ flexible to geometry / deformations / viewpoint
+ compact summary of image content
+ provides fixed dimensional 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?
Can we be more accurate?

So far, we treat each image as containing a “bag of words”, with no spatial information.
Can we be more accurate?

So far, we treat each image as containing a “bag of words”, with no spatial information.

Real objects have consistent geometry.
Both image pairs have many visual words in common.
Spatial Verification

Only some of the matches are mutually consistent
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

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

Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.
RANSAC verification
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?
Scoring retrieval quality

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

Results (ordered):

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

Slide credit: Ondrej Chum
China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004’s $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump in exports to $750bn, compared with a 18% rise in imports to $660bn. This is likely to annoy the US, which has long argued that China’s exports are unfairly helped by a deliberately undervalued yuan. While Beijing agrees the surplus is too high, it says the yuan is only one factor. Bank of China governor Zhou Xiaochuan said the country also needed to do more to boost domestic demand so more goods stayed within the country. China increased the value of the yuan against the dollar by 2.1% in July and permitted it to trade within a narrower band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.
**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)

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

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

Query Expansion

Results

Spatial verification

New results

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

Slide credit: Ondrej Chum
Recognition via alignment

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

Cons:
• 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
Lessons from a Decade Later

• For Category recognition (project 4)
  – Bag of Feature models remained the state of the art until Deep Learning.
  – Spatial layout either isn't that important or its too difficult to encode.
  – Quantization error is, in fact, the bigger problem. Advanced feature encoding methods address this.
  – Bag of feature models are nearly obsolete. At best they seem to be inspiring tweaks to deep models e.g. NetVLAD.
Lessons from a Decade Later

• For *instance* retrieval (this lecture)
  – deep learning is taking over.
  – learn better local features (replace SIFT) e.g. MatchNet
  – or learn better image embeddings (replace the histograms of visual features) e.g. Vo and Hays 2016.
  – or learn to do spatial verification e.g. DeTone, Malisiewicz, and Rabinovich 2016.
  – or learn a monolithic deep network to recognition all locations e.g. Google’s PlaNet 2016.