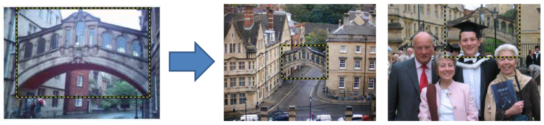
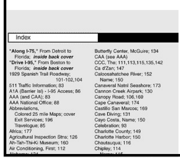
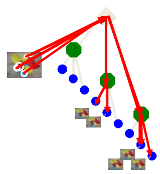


Instance recognition

Tues April 4
Kristen Grauman
UT Austin





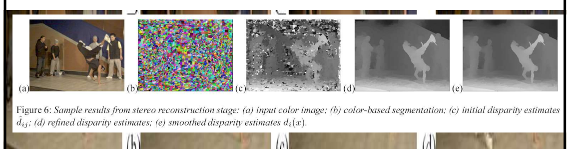
Last time

- Depth from stereo: main idea is to triangulate from corresponding image points.
- Epipolar geometry defined by two cameras
  - We've assumed known extrinsic parameters relating their poses
- Epipolar constraint limits where points from one view will be imaged in the other
  - Makes search for correspondences quicker
- To estimate depth
  - Limit search by epipolar constraint
  - Compute correspondences, incorporate matching preferences

Stereo error sources

- Low-contrast ; textureless image regions
- Occlusions
- Camera calibration errors
- Violations of *brightness constancy* (e.g., specular reflections)
- Large motions

Virtual viewpoint video



C. Zitnick et al, High-quality video view interpolation using a layered representation, SIGGRAPH 2004.

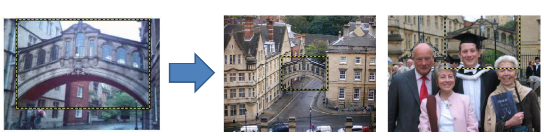
Review questions (on your own)

- When solving for stereo, when is it necessary to break the soft disparity gradient constraint?
- What can cause a disparity value to be undefined?
- Suppose we are given a disparity map indicating offset in the x direction for corresponding points. What does this imply about the layout of the epipolar lines in the two images?

Slide credit: Kristen Grauman

Today

- Instance recognition
  - Indexing local features efficiently
  - Spatial verification models



Recognizing or retrieving specific objects

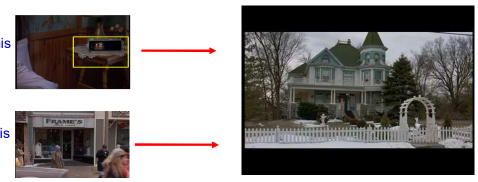
Example I: Visual search in feature films

Visually defined query

"Groundhog Day" [Rammis, 1993]

"Find this clock"

"Find this place"



Slide credit: J. Sivic

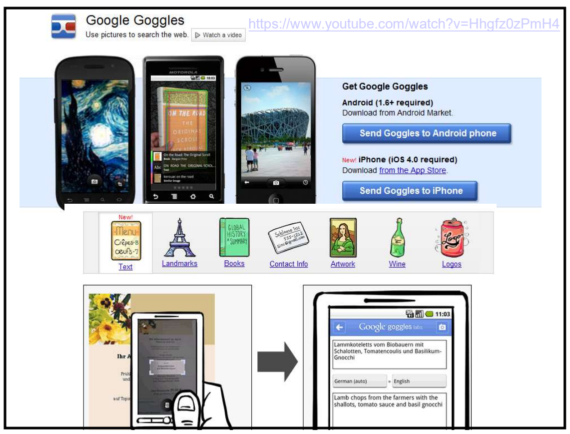
Recognizing or retrieving specific objects

Example II: Search photos on the web for particular places

Find these landmarks

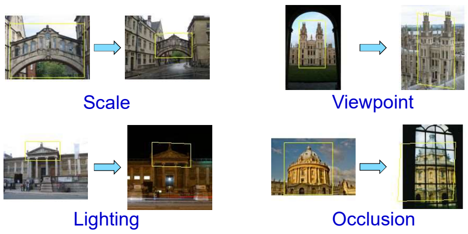
...in these images and 1M more

Slide credit: J. Sivic



Why is it difficult?

Want to find the object despite possibly large changes in scale, viewpoint, lighting and partial occlusion



Slide credit: J. Sivic

Recall: matching local features

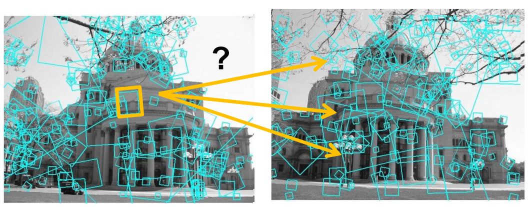


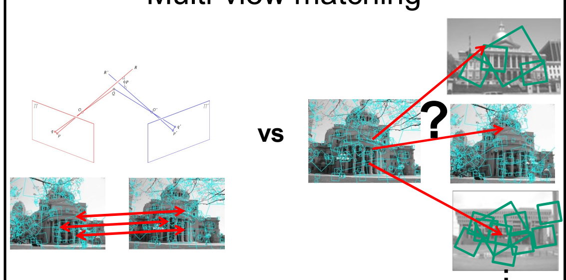
Image 1 Image 2

To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

Slide credit: Kristen Grauman

Multi-view matching

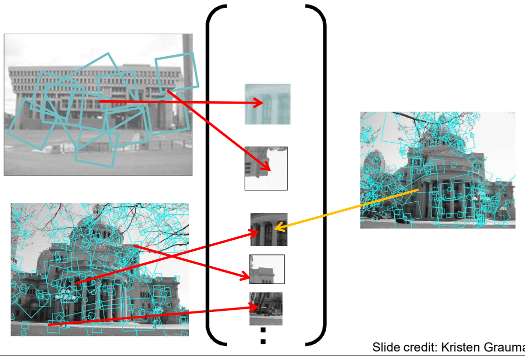


Matching two given views for depth

Search for a matching view for recognition

Slide credit: Kristen Grauman

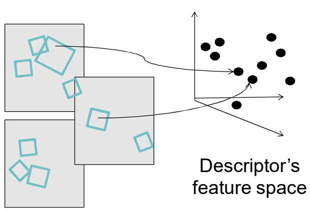
Indexing local features



Slide credit: Kristen Grauman

Indexing local features

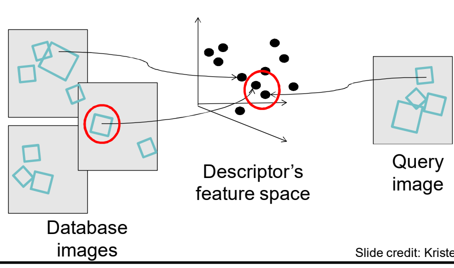
- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)



Slide credit: Kristen Grauman

Indexing local features

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



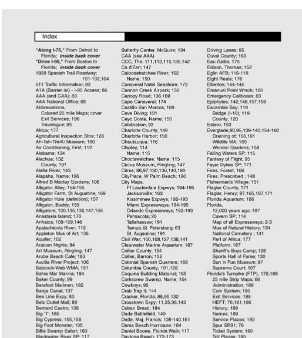
Slide credit: Kristen Grauman

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?
- Possible solutions:
  - Inverted file
  - Nearest neighbor data structures
    - Kd-trees
    - Hashing

Slide credit: Kristen Grauman

Indexing local features: inverted file index

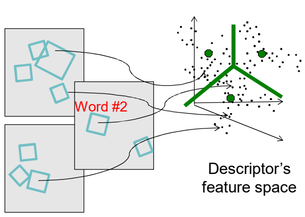


- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words".

Slide credit: Kristen Grauman

Visual words

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

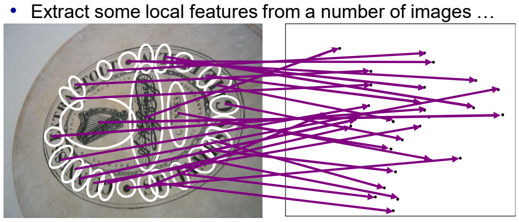


- Quantize via clustering, let cluster centers be the prototype "words"
- Determine which word to assign to each new image region by finding the closest cluster center.

Slide credit: Kristen Grauman

Visual words: main idea

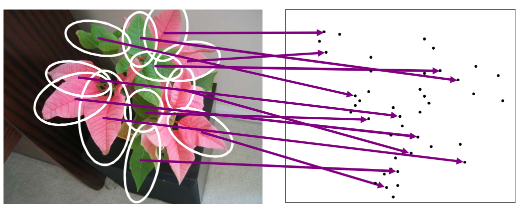
- Extract some local features from a number of images …



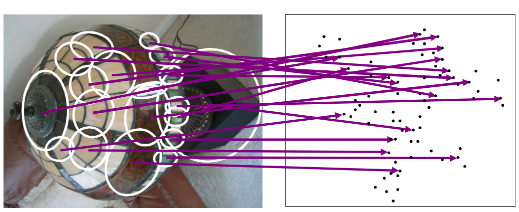
e.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister, CVPR 2006

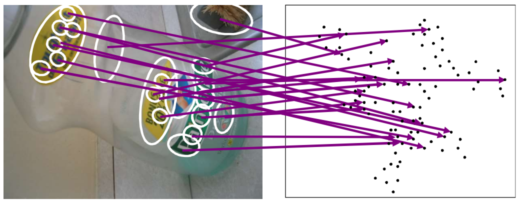
Visual words: main idea

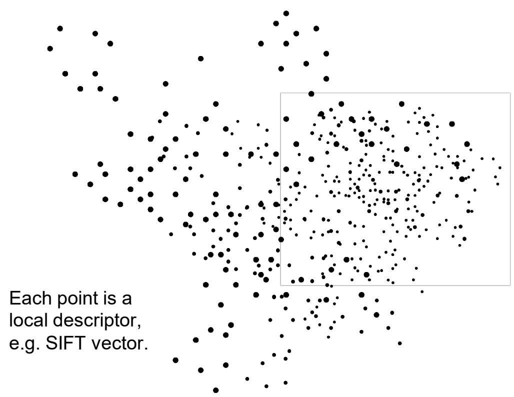


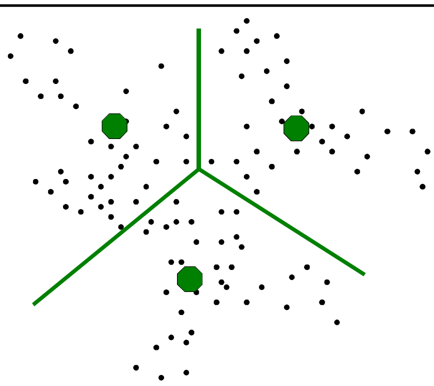
Visual words: main idea



Visual words: main idea

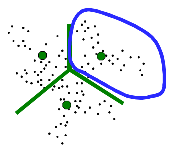






Visual words

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



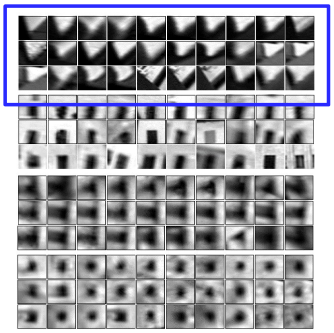
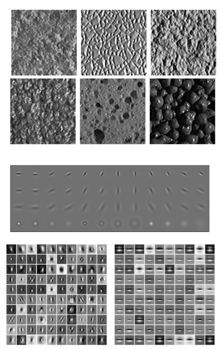


Figure from Sivic & Zisserman, ICCV 2003

Visual words and textons

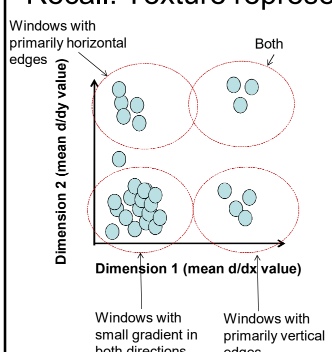
- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.



Leung & Malik 1999; Varma & Zisserman, 2002

Slide credit: Kristen Grauman

Recall: Texture representation example



	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

Slide credit: Kristen Grauman

Visual vocabulary formation

Issues:

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

Slide credit: Kristen Grauman

Inverted file index

Database images

Image #1

Image #2

Image #3

Word #	Image #
1	3
2	
…	
7	1, 2
8	3
9	
10	
…	
91	2

…

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

Slide credit: Kristen Grauman

Inverted file index

When will this give us a significant gain in efficiency?

New query image

Word #	Image #
1	3
2	
…	
7	1, 2
8	3
9	
10	
…	
91	2

…

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

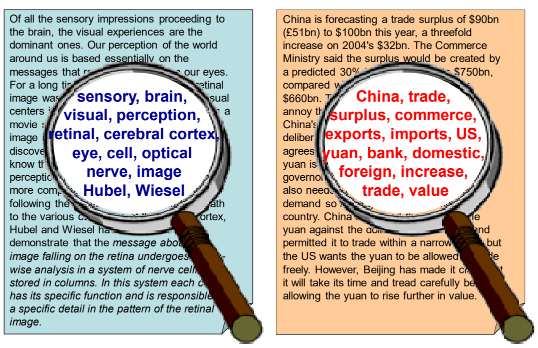
Slide credit: Kristen Grauman

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?

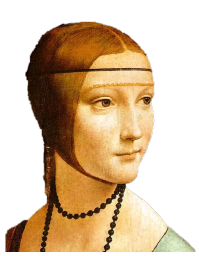
Slide credit: Kristen Grauman

Analogy to documents

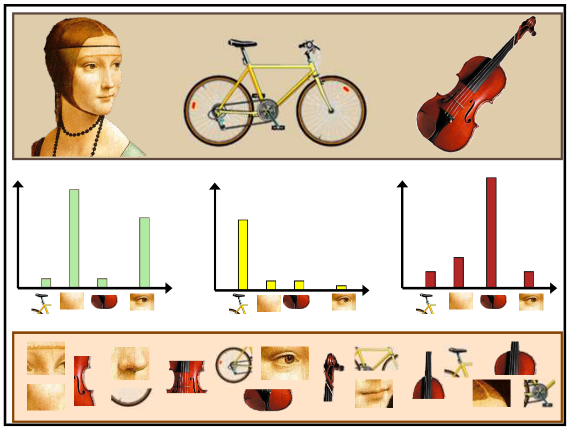


ICCV 2005 short course, L. Fei-Fei

Object → **Bag of ‘words’**

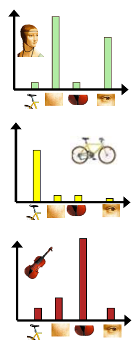


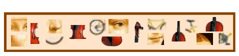
ICCV 2005 short course, L. Fei-Fei



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.

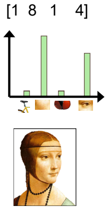


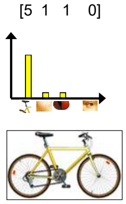


Comparing bags of words

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

[1 8 1 4] [5 1 1 0]





$\vec{d}_j$ $\vec{q}$

$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i)^2}}$$

for vocabulary of *V* words

Slide credit: Kristen Grauman

tf-idf weighting

- **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

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

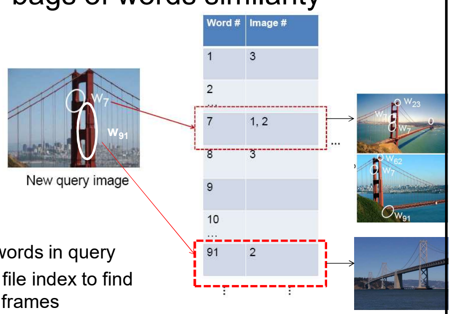
Number of occurrences of word i in document d

Number of words in document d

Total number of documents in database

Number of documents word i occurs in, in whole database

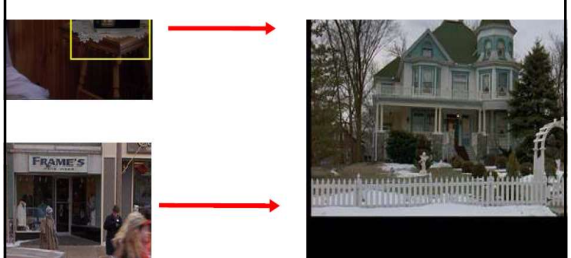
Inverted file index and bags of words similarity



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

Slide credit: Kristen Grauman

Bags of words for content-based image retrieval



Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003


Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

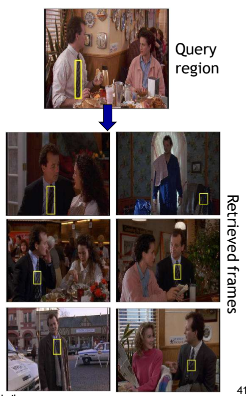
Visual Object Recognition Tutorial

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



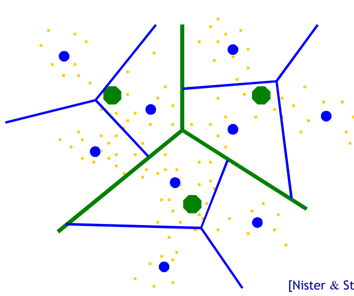
K. Grauman, B. Leibe

41

Visual Object Recognition Tutorial

Vocabulary Trees: hierarchical clustering for large vocabularies

- Tree construction:



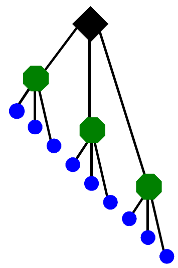
[Nister & Stewenius, CVPR'06]

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

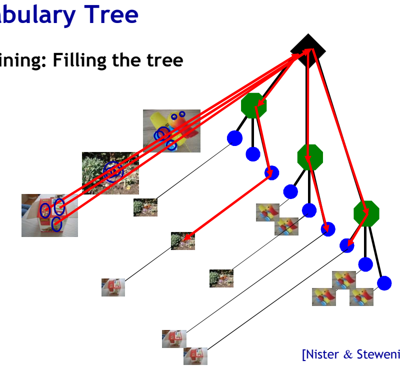
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

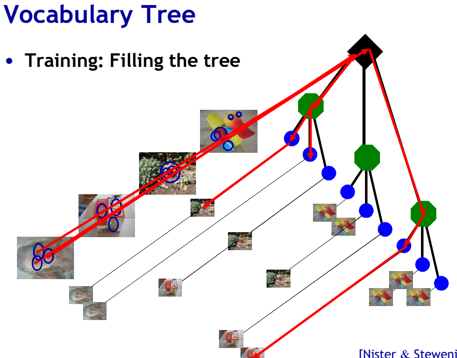
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree



[Nister & Stewenius, CVPR'06]

Visual Object Recognition Tutorial

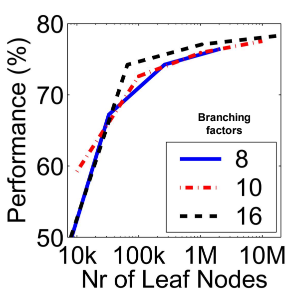
K. Grauman, B. Leibe

Slide credit: David Nister

45

What is the computational advantage of the hierarchical representation bag of words, vs. a flat vocabulary?

Vocabulary size



Results for recognition task with 6347 images

Influence on performance, sparsity?

Nister & Stewenius, CVPR 2006

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

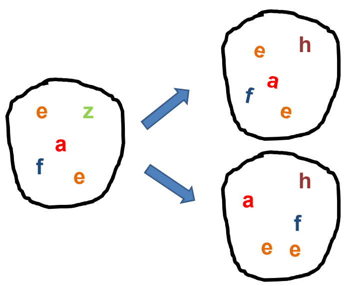
Slide credit: Kristen Grauman

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?

Slide credit: Kristen Grauman

Which matches better?

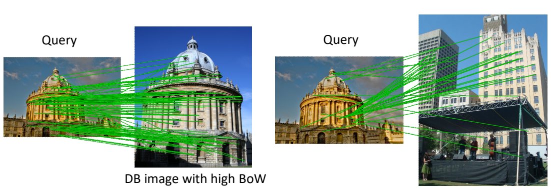


Derek Hoiem

Spatial Verification

Query

Query



DB image with high BoW similarity

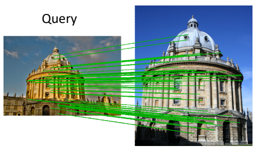
DB image with high BoW similarity

Both image pairs have many visual words in common.

Slide credit: Ondrej Chum

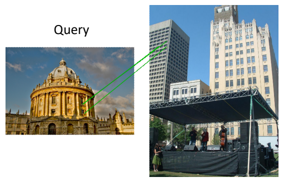
Spatial Verification

Query



DB image with high BoW similarity

Query



DB image with high BoW similarity

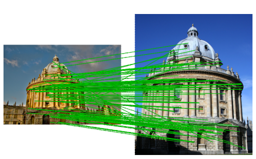
Only some of the matches are mutually consistent

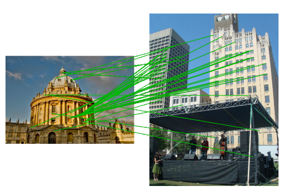
Slide credit: Ondrej Chum

Spatial Verification: two basic strategies

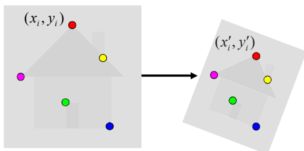
- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible transformations
    - e.g., "success" if find a transformation with > N inlier correspondences
- Generalized Hough Transform
  - Let each matched feature cast a vote on location, scale,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

RANSAC verification





Recall: Fitting an affine transformation

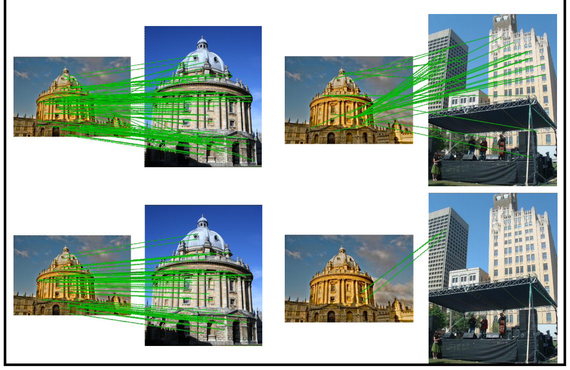


Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

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

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

RANSAC verification

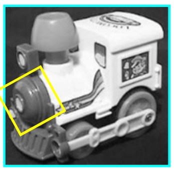


Spatial Verification: two basic strategies

- 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).



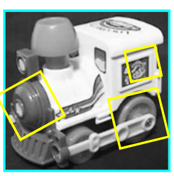
Model

Novel 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



Model

Novel 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 btwn 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

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Slide credit: Lana Lazebnik

Example result

Background subtract for model boundaries

Objects recognized,

Recognition in spite of occlusion

[Lowe]

Recall: 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.

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

Slide credit: Kristen Grauman

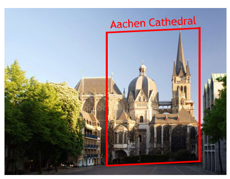
Example applications

- Snap, pick, pay

- https://www.usatoday.com/videos/tech/2014/10/31/18261641/

Slide credit: Kristen Grauman

Example Applications



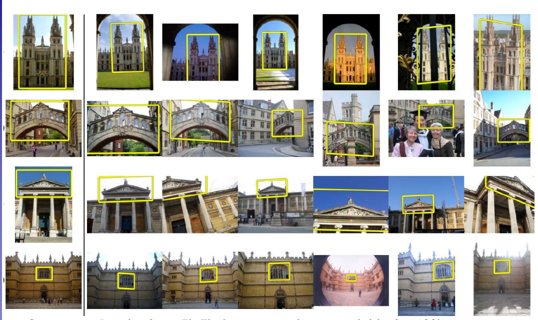
Mobile tourist guide
- Self-localization
- Object/building recognition
- Photo/video augmentation

Visual Object Recognition Tutorial

B. Leibe

[Quack, Leibe, Van Gool, CIVR'08]

Application: Large-Scale Retrieval



Query

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

[Philbin CVPR'07]

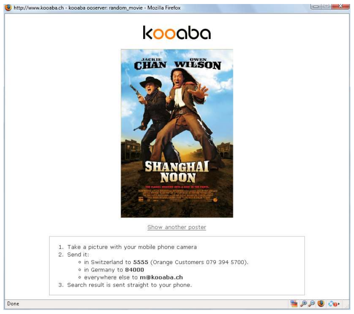
Visual Object Recognition Tutorial

Web Demo: Movie Poster Recognition

Visual Object Recognition Tutorial

50'000 movie posters indexed

Query-by-image from mobile phone available in Switzer-land



http://www.kooaba.com/en/products_engine.html#

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?

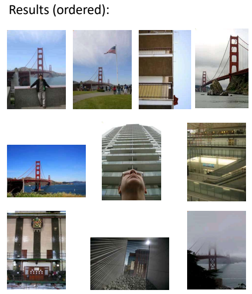
Kristen Grauman

Scoring retrieval quality

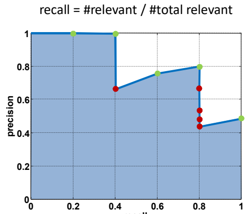
Query

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

Results (ordered):



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



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:
- 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

Kristen Grauman