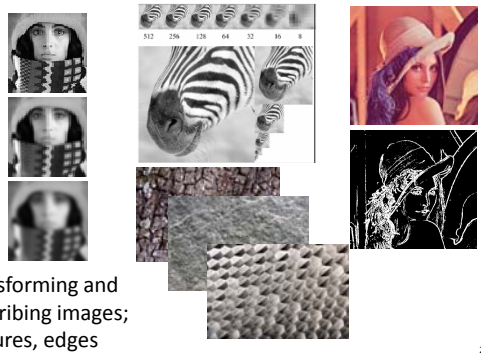
Indexing local features and instance recognition

May 14th, 2019

Yong Jae Lee
UC Davis

Many slides borrowed from Kristen Grauman, David Lowe, Fei-Fei Li, Andrew Zisserman, Ondrej Chum

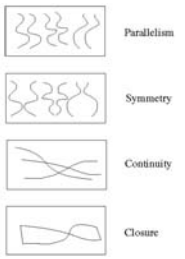
Recap: Features and filters

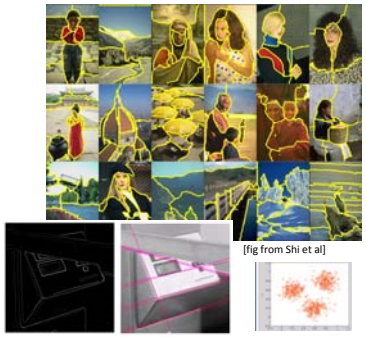


Transforming and describing images; textures, edges

2

Recap: Grouping & fitting



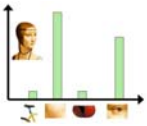


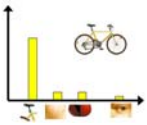
[fig from Shi et al]

Clustering, segmentation, fitting; what parts belong together?

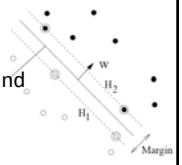
3

Recognition and learning





Recognizing objects and categories, learning techniques



4

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

5

Matching local features

6

Matching local features

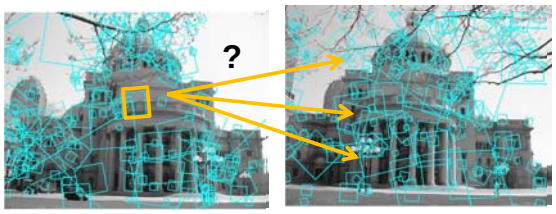


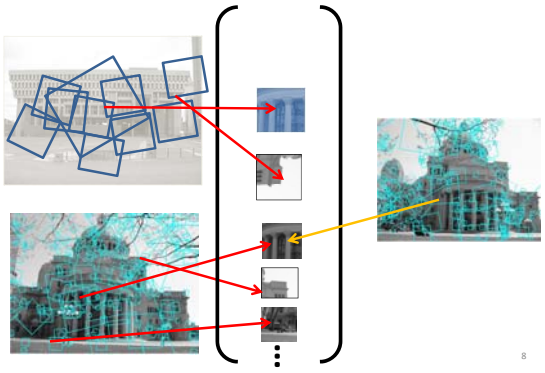
Image 1 Image 2

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)

7

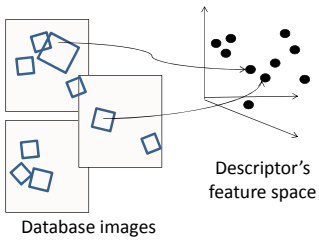
Indexing local features



8

Indexing local features

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



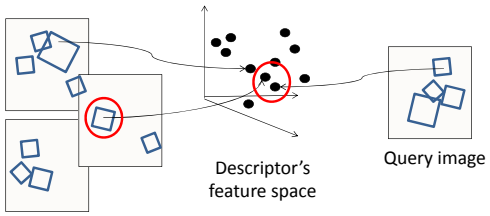
Descriptor's feature space

Database images

9

Indexing local features

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



10

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?

11

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".

12

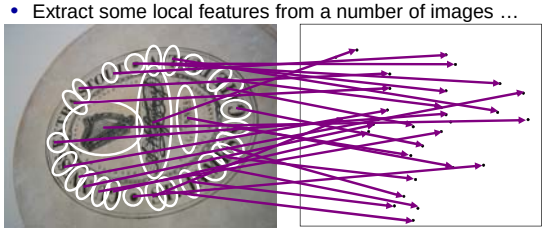
Text retrieval vs. image search

- What makes the problems similar, different?
- Text words are discrete “tokens”, whereas local image descriptors are high-dimensional, real-valued feature points.
- Need to *quantize* the visual features into discrete visual words.

13

Visual words: main idea

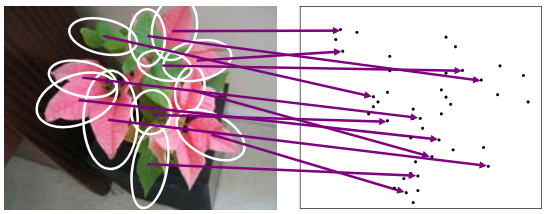
- Extract some local features from a number of images …



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

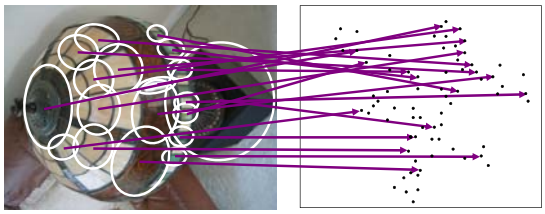
14

Visual words: main idea



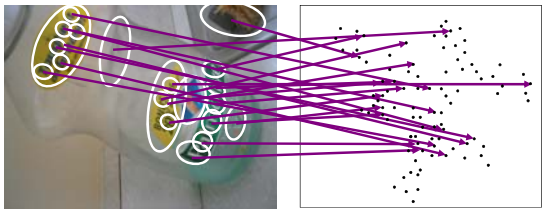
15

Visual words: main idea

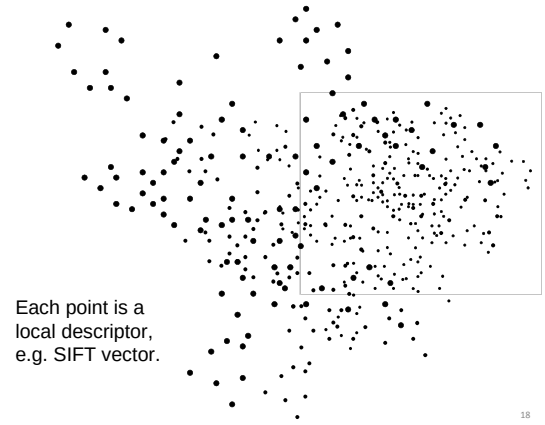


16

Visual words: main idea

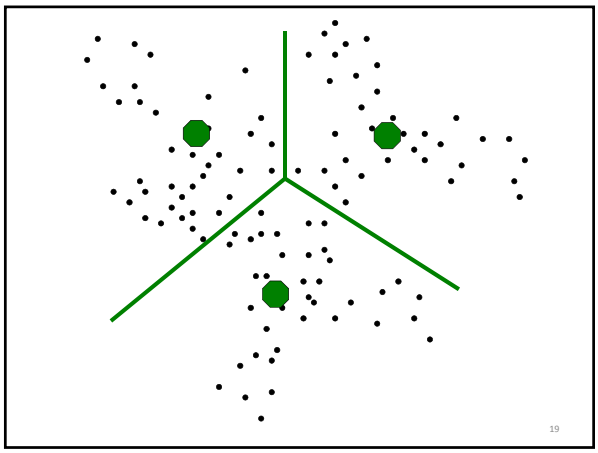


17



Each point is a local descriptor, e.g. SIFT vector.

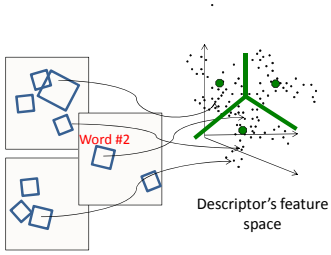
18



19

Visual words

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

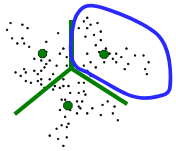


- Quantize via clustering, let cluster centers be the prototype "words"
- Determine which word to assign to each new image region by finding the closest cluster center.

20

Visual words

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



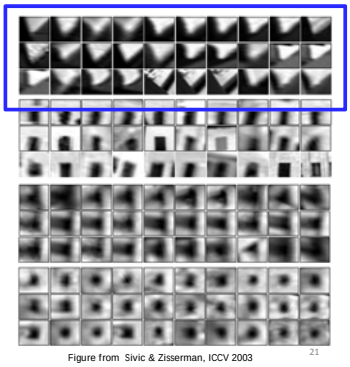


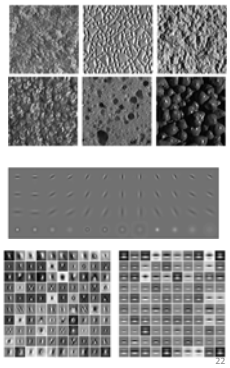
Figure from Sivic & Zisserman, ICCV 2003

21

Visual words and textons

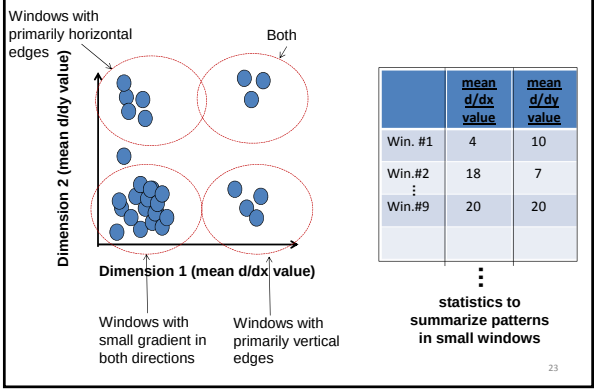
- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.

Leung & Malik 1999; Varma & Zisserman, 2002



22

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

23

Visual vocabulary formation

Issues:

- Sampling strategy: where in image to extract features?
- Clustering / quantization algorithm?
- What corpus/dataset provides features (universal vocabulary?)
- Vocabulary size (number of words)?

24

Inverted file index

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

25

Inverted file index

When will this give us a significant gain in efficiency?

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

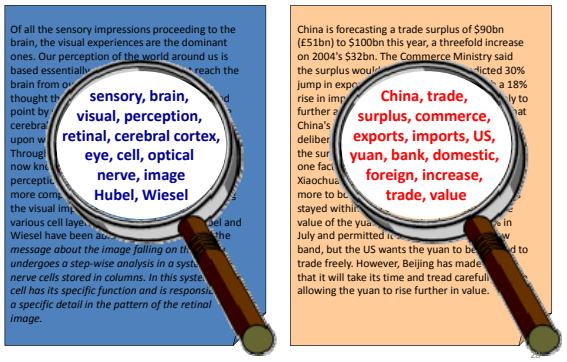
- New query image is mapped to indices of database images that share a word.

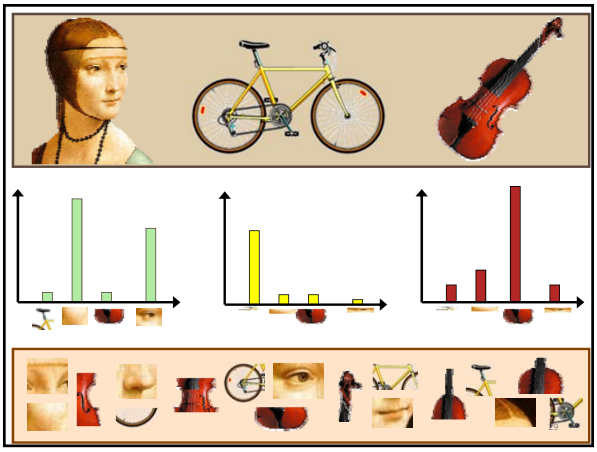
26

- If a local image region is a visual word, how can we summarize an image (the document)?

27

Analogy to documents





Bag of visual words

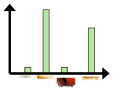
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for text documents.

30

Comparing bags of words

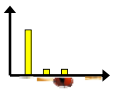
- Rank frames by normalized scalar product between their occurrence counts---*nearest neighbor* search for similar images.

[1 8 1 4]



$\vec{d}_j$

[5 1 1 0]



$\vec{q}$

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

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

for vocabulary of *V* words

31

Bags of words for content-based image retrieval

Visually defined query

"Groundhog Day" [Rammis, 1993]

"Find this clock"

"Find this place"

Sivic & Zisserman, ICCV 2003

32

Example

retrieved shots

Sivic & Zisserman, ICCV 2003

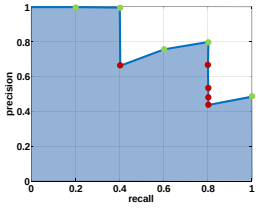
33

Scoring retrieval quality

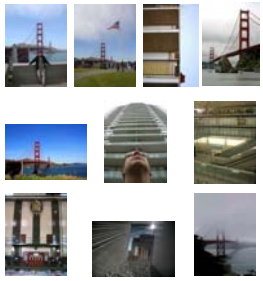
Query

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

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



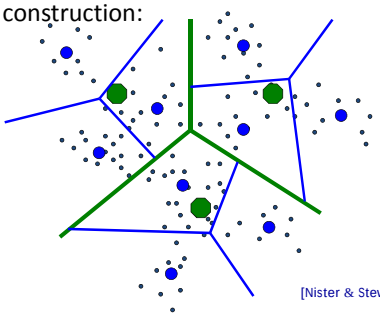
Results (ordered):



34

Vocabulary Trees: hierarchical clustering for large vocabularies

- Tree construction:



[Nister & Stewenius, CVPR'06]

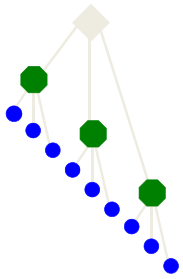
35

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

36

Vocabulary Tree

- Training: Filling the tree

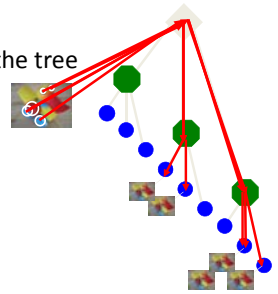


[Nister & Stewenius, CVPR'06]

37

Vocabulary Tree

- Training: Filling the tree

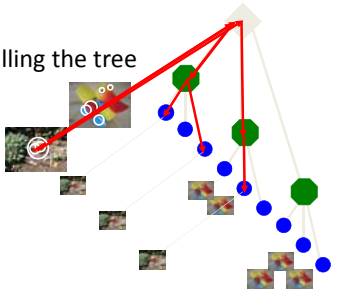


[Nister & Stewenius, CVPR'06]

38

Vocabulary Tree

- Training: Filling the tree

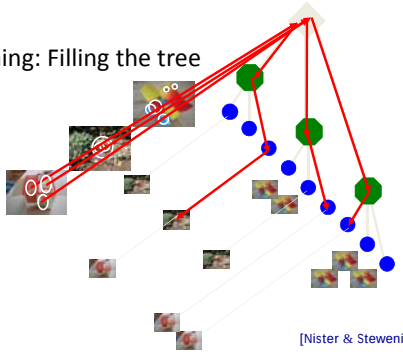


[Nister & Stewenius, CVPR'06]

39

Vocabulary Tree

- Training: Filling the tree

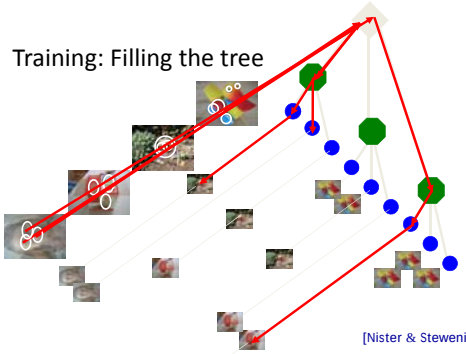


[Nister & Stewenius, CVPR'06]

40

Vocabulary Tree

- Training: Filling the tree

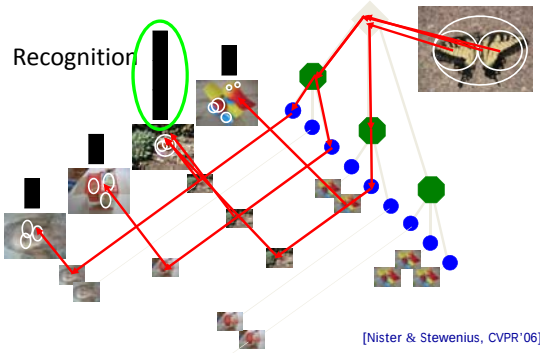


[Nister & Stewenius, CVPR'06]

41

Vocabulary Tree

- Recognition



[Nister & Stewenius, CVPR'06]

42

Bags of words: pros and cons

+ flexible to local deformations / viewpoint
+ compact summary of image content
+ provides vector representation for sets
+ decent results in practice

- basic model ignores global geometry – must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- optimal vocabulary formation unclear

43

Summary So Far

- **Matching local invariant features**: useful to provide matches to find objects and scenes.
- **Bag of words** representation: quantize feature space to make discrete set of visual words
- **Inverted index**: pre-compute index to enable faster search at query time

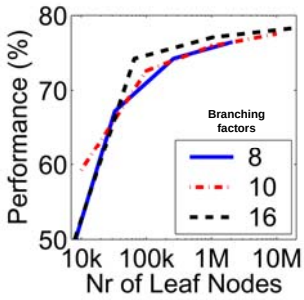
44

Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How to score the retrieval results?
- How large should the vocabulary be? How to perform quantization efficiently?

45

Vocabulary size



Results for recognition task with 6347 images

46

Nister & Stewenius, CVPR 2006

Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How to score the retrieval results?
- 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?

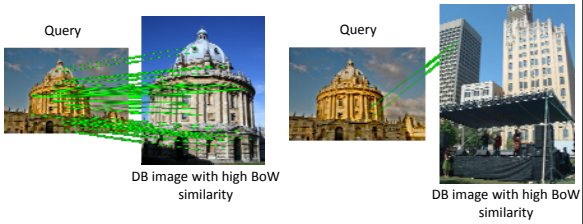
47

Spatial Verification

Both image pairs have many visual words in common

48

Spatial Verification



Only some of the matches are mutually consistent

49

Spatial Verification

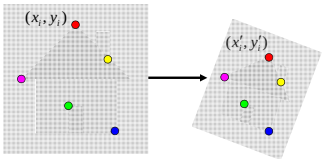
- RANSAC
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible transformations
    - e.g., "success" if find a transformation with > N inlier correspondences

50

RANSAC verification

51

Recall: Fitting an affine transformation



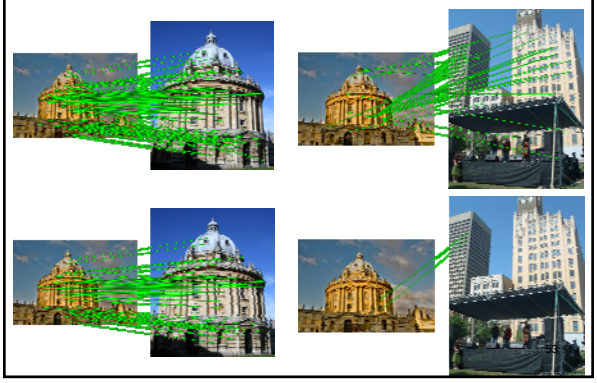
Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

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

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

52

RANSAC verification

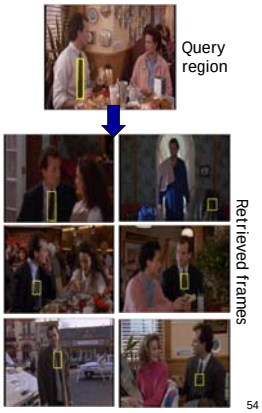


Video Google System

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



Visual Object Recognition Tutorial

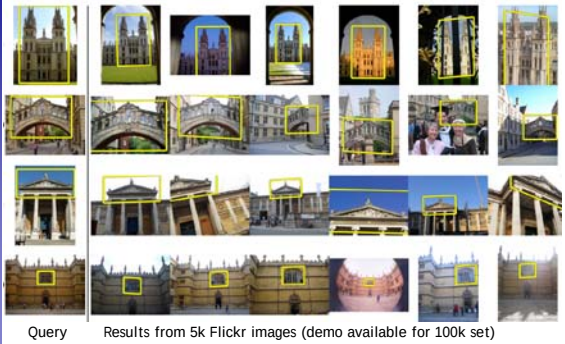
54

Example Applications

Mobile tourist guide
- **Self-localization**
- **Object/building recognition**

[Quack, Leibe, Van Gool, CIVR'08] 55

Application: Large-Scale Retrieval



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

[Philbin CVPR'07] 56

What else can we borrow from text retrieval?

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% [illegible] $750bn, compared [illegible] $660bn. [illegible] annoy [illegible] China's [illegible] deliber[illegible] agrees [illegible] yuan is [illegible] governo[illegible] also need[illegible] demand so [illegible] country. China [illegible] yuan against the [illegible] and permitted it to trade within a narrow [illegible] but the US wants the yuan to be allowed [illegible] freely. However, Beijing has made it c[illegible] it will take its time and tread carefully be[illegible] allowing the yuan to rise further in value.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

57

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)

Number of occurrences of word i in document d

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

Number of words in document d

Total number of documents in database

Number of documents word i occurs in, in whole database

58

Query expansion

Query: ***golf green***

Results:

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

Irrelevant result can cause a `topic drift':

- Volkswagen ***Golf***, 1999, ***Green***, 2000cc, petrol, manual, , hatchback, 94000miles, 2.0 GTi, 2 Registered Keepers, HPI Checked, Air-Conditioning, Front and Rear Parking Sensors, ABS, Alarm, Alloy

59

Query expansion

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

60

Recognition via local-feature based 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 generic category recognition

61

Summary

- **Matching local invariant features**
  - Useful 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

62

Questions?

See you Thursday!

63