

CDS6334

Visual Information Processing

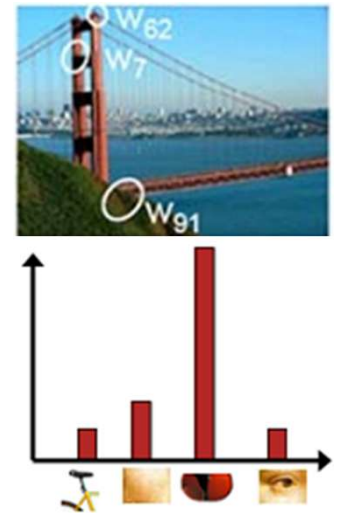
Lecture 10

Visual Words: Feature Indexing

Faculty of Computing and Informatics

Prepared by John See;

Modified by Lai-Kuan Wong; Yuen Peng Loh



SOME GRAPHICS AND MATERIALS ARE TAKEN FROM UTEXAS AT AUSTIN CS376 & CS381V NOTES, STANFORD UNI'S CS131 NOTES
AND GONZALEZ & WOOD'S DIP TEXTBOOK

PREVIOUSLY

- › **Local invariant features**
 - › Why local not global?
 - › Why invariant?
- › **Detection:** Corners as good distinctive features
 - › Harris corner detector
 - › Scale-space extrema (blob) detector
- › **Description:** Describing features in local “patches”
 - › SIFT

$d(f_A, f_B) < T$

Why subpatches?
Why does SIFT have some illumination invariance?



Lecture Outline

- › Feature indexing – Why do it?
- › “Visual words” concept
 - Bag of visual words
 - Inverted file index
 - Retrieval scoring
- › Application for image retrieval

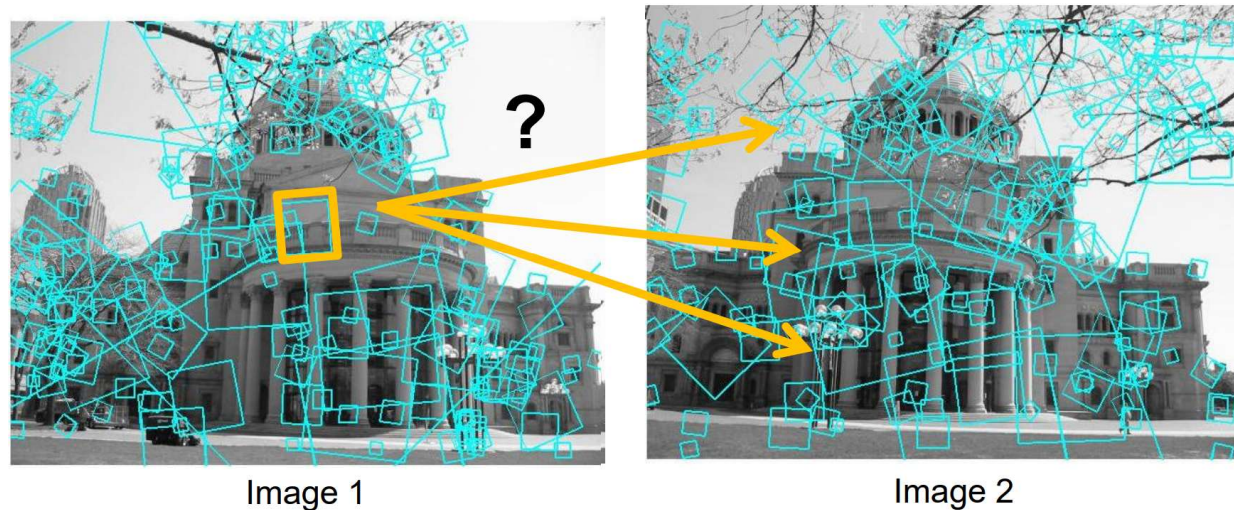


Feature Indexing





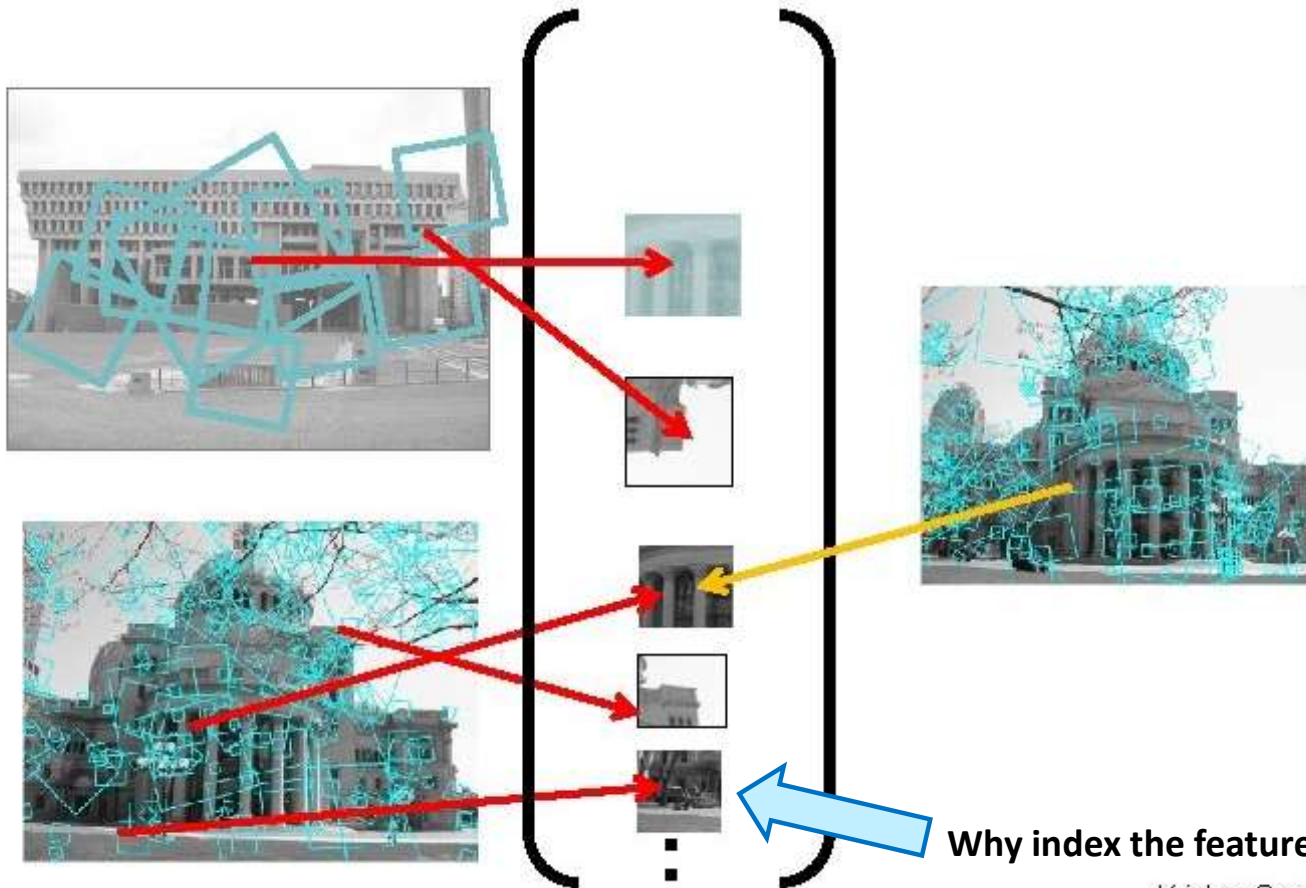
Matching local features



- › To generate candidate matches, find patches that have the most similar appearance (e.g. lowest SSD)
- › Simplest approach: Compare ALL, take the closest (or closest k , or within a threshold distance)



Indexing local features



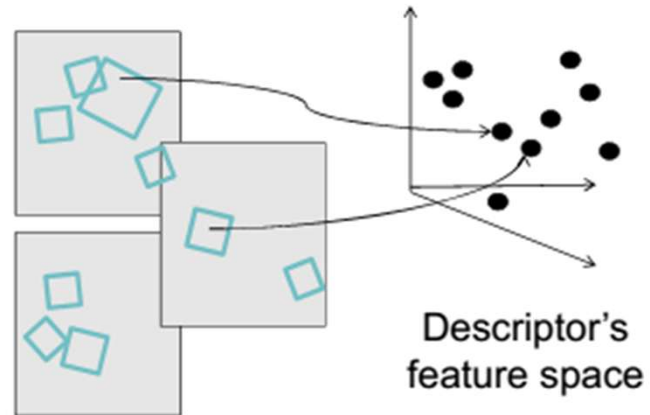
Why index the features?

Kristen Grauman



Indexing Local Features

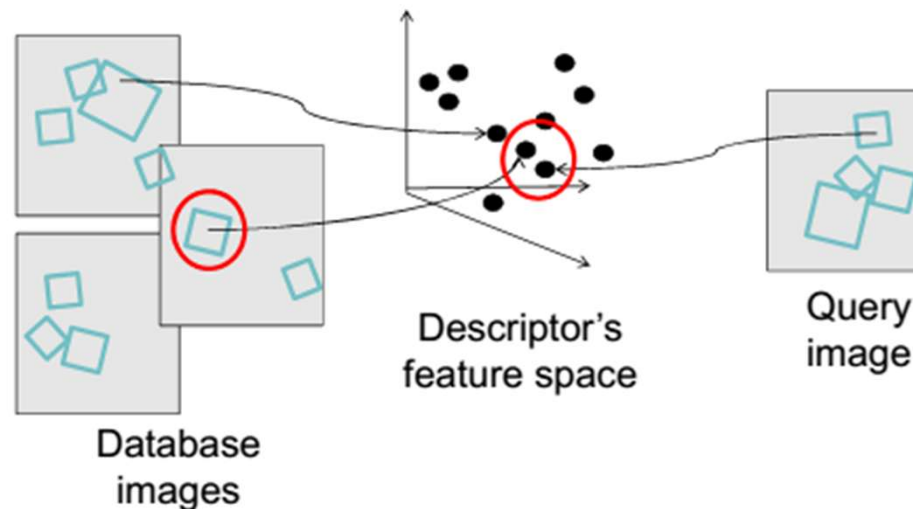
- › Each patch/local region has a descriptor, which is a point in some high-dimensional feature space (e.g. SIFT)





Indexing Local Features

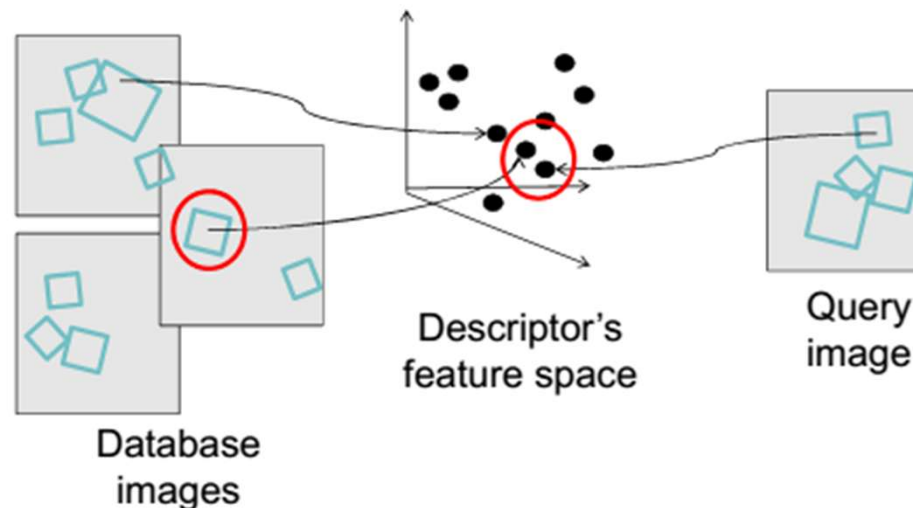
- › When we see **close points in feature space**, we have similar descriptors, which indicates **similar local content**





Indexing Local Features

- › With potentially thousands of features per image, and hundreds to millions of images to search and match, how to efficiently find those that are relevant to a new image?





Indexing Local Features

- › For text documents, an efficient way to find all pages on which a word occurs is to use an **index**...
- › Here, we want to find all images in which a **feature** occurs.
- › To use this idea, we need to map our features to this **“index”** of visual words.

Index		
Along I-75, From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuirk; 134	Driving Lanes; 85
Drive I-95, From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)	Duval County; 153
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142	Eau Gallie; 175
511 Traffic Information; 83	Ca (Zan); 147	Edson, Thomas; 152
A1A (Barrier Is) - I-95 Access; 86	Caloosahatchee River; 152	Eggs APB; 115-118
AAA (and CAA); 83	Name; 150	Eight Reels; 176
AAA National Office; 88	Canaveral Natri Seashore; 173	Eleuterio; 144-145
Abbreviations;	Cannon Creek Airport; 130	Emerald Point Wharf; 120
Colored 25-mile Maps; cover	Canopy Road; 106,189	Emergency Callboxes; 83
Exit Services; 196	Cape Canaveral; 174	EggyPhix; 142,148,157,159
Travelogue; 85	Castillo San Marcos; 169	Escambia Bay; 119
Africa; 177	Cave Diving; 131	Bridge 2-10; 119
Agricultural Inspection Struc; 126	Cayo Costa, Name; 150	County; 120
Ah-Tah-Tha-Ki Museum; 160	Celebration; 90	Etern; 153
Air Conditioning, First; 112	Charlotte County; 149	Everglades; 80,85,139-140,154-160
Alabama; 124	Charlotte Harbor; 150	Draining of; 156,181
Alachua; 132	Chautauque; 116	Wildlife MA; 160
County; 131	Chapley; 114	Wonder Gardens; 134
Alafia River; 143	Name; 115	Falling Waters SP; 115
Alapaha, Name; 126	Choctawhatchee, Name; 115	Factory of Flight; 95
Alfred B. Mackey Gardens; 106	Circuit Museum, Ringing; 147	Fayer Dykes SP; 171
Aligator Alley; 154-155	Citra; 88,97,130,136,140,180	Fees, Forest; 166
Aligator Farm, St. Augustine; 169	CityPlace, W Palm Beach; 180	Fees, Prescribed; 148
Aligator Hole (parish); 157	City Maps;	Fishermen's Village; 151
Aligator, Buddy; 155	FL Landerline Express; 194-195	Flagler County; 171
Aligators; 100,135,138,147,156	Jacksonville; 163	Flagler, Henry; 97,165,167,171
Anastasia Island; 170	Kissimmee Express; 192-193	Florida Aquarium; 186
Anchoa; 126-129,148	Miami Expressways; 194-195	Florida;
Apalachicola River; 112	Orlando Expressways; 192-193	12,000 years ago; 187
Appleton Max of Art; 136	Panoramas; 26	Caverns SP; 114
Aquifer; 102	Tallahassee; 191	Map of all Expressways; 2-3
Arabian Nights; 94	Tampa St. Petersburg; 63	Map of Natural History; 134
Art Museum, Ringing; 147	St. Augustine; 191	National Cemetery; 141
Aruba Beach Cafe; 163	Cul War; 100,108,127,138,141	Part of Africa; 177
Aucilla River Project; 106	Clearwater Marine Aquarium; 187	Perform; 187
Beachside Hwy 95SA; 151	Collier County; 154	Sheriff's Boys Camp; 126
Bahia Mar Marina; 184	Collier, Barron; 152	Sports Hall of Fame; 130
Baker County; 99	Colonial Spanish Quarters; 168	Sun 'n Fun Museum; 97
Barfoot Mallman; 162	Columbia County; 101,128	Supreme Court; 127
Barge Canal; 137	Coquina Building Material; 165	Florida's Tompkins (TP7); 178,189
Bea Line Expy; 80	Corkscrew Swamp, Name; 154	25-mile Strip Map; 66
Beiz Outlet Mall; 89	Cowboys; 95	Administration; 189
Bernard Castro; 136	Crab Trap II; 144	Coin System; 190
Big 'Y'; 165	Cracker, Florida; 88,95,132	Exit Services; 189
Big Cypress; 155,158	Crossroads Expy; 11,25,98,143	HEFT; 76,161,190
Big Foot Monster; 105	Cuban Bread; 104	History; 189
Billie Swamp Safari; 160	Dade Battlefield; 140	Name; 189
Blackwater River SP; 117	Dade, Maj. Francis; 139-140,161	Service Plaza; 190
	Dania Beach Hurricane; 184	Spur SP91; 76
	Daniel Boone, Florida Wash; 117	Ticket System; 190
	Daytona Beach; 172-173	Toll Plaza; 190



Text retrieval vs. image search

- › So, what makes the problems similar, or different?
 - A. Different because images and texts have different dimensions.
 - B. Different because images describe visual details while text describe high level concepts.
 - C. Similar because we can have a dictionary for image features like a dictionary for text words.
 - D. Similar because image features are the same vectors as text word vectors.

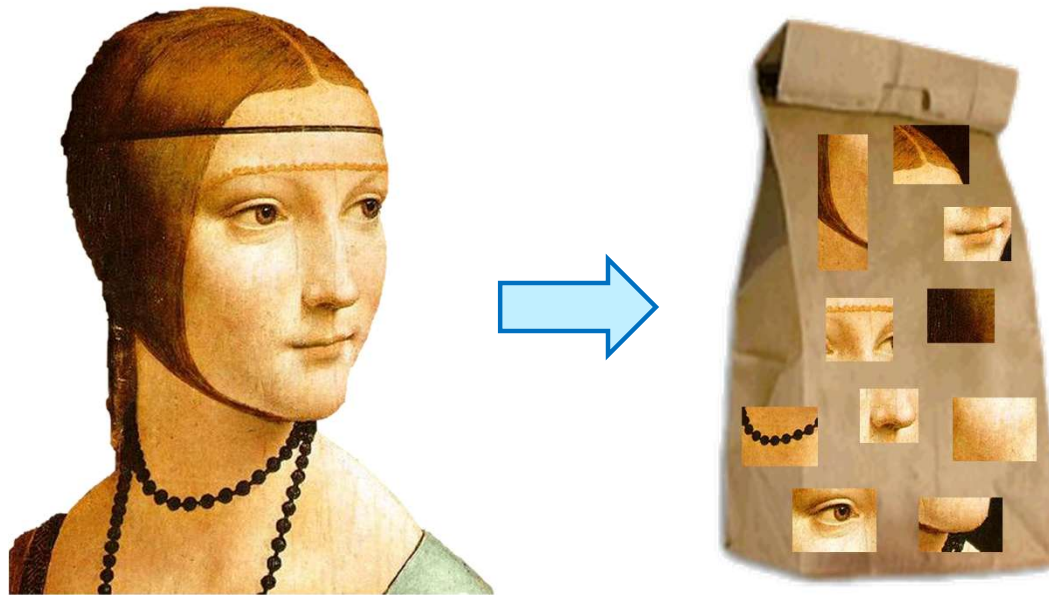


Concept of Visual Words





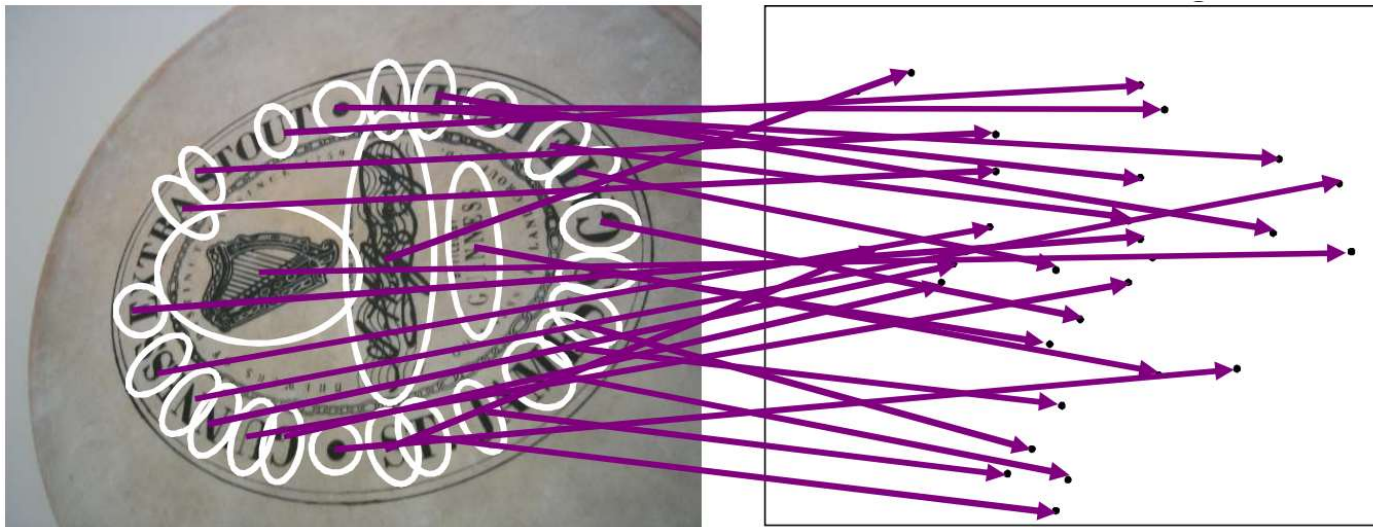
Visual Words





Visual Words

- › Extract some local features from a number of images...

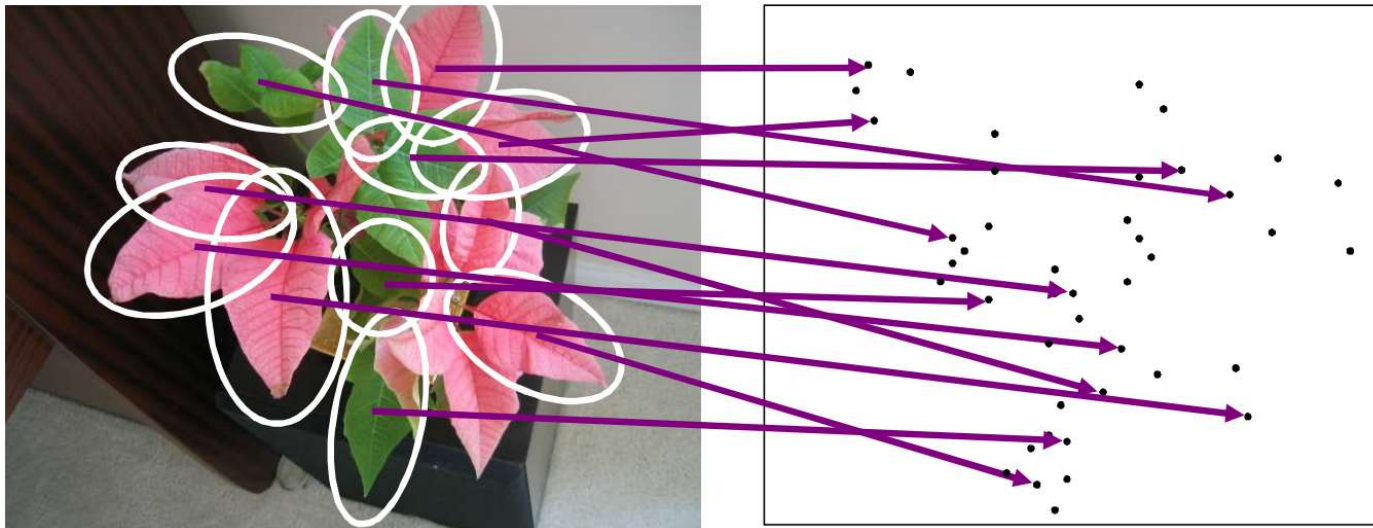


- e.g. SIFT descriptor space: each point is 128-dimensional



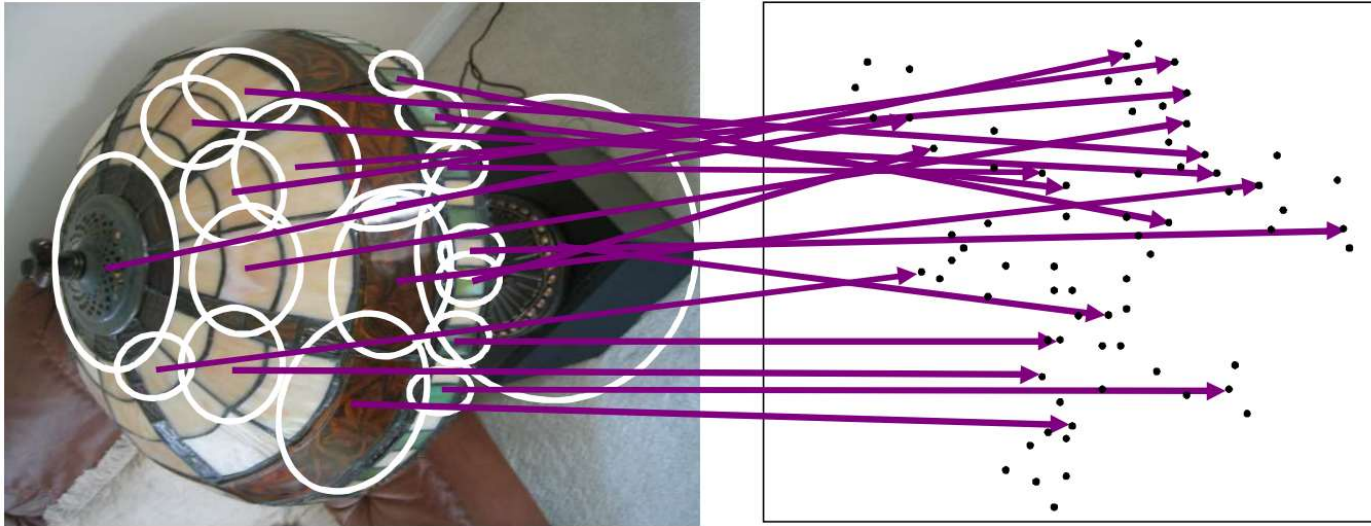
Visual Words

- › It is impossible to visualize 128-dimensions.
- › Most of the time we show it in 2-D or 3-D only





Visual Words





Visual Words

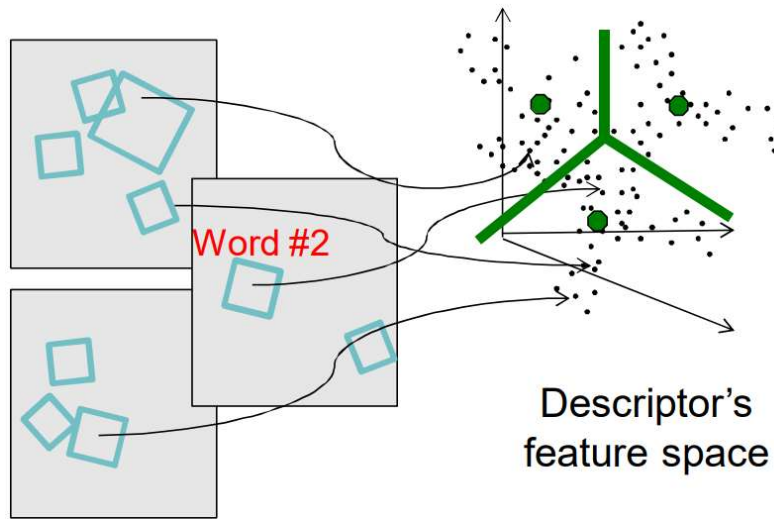
- › Next, we want to try group these points (each 128-dimensions) into groups which will reflect distinctive characteristics
- › Solution: Use a clustering technique such as k-means





Quantizing the feature space

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



- › **Clustering:** Let **cluster centers** be the representative of the “**words**”
- › **Quantization:** Determine which word to assign to each image descriptor by finding the closest cluster center



Quantizing the feature space

- › Example: Each group of patches belongs to the same visual word.
 - Look how similar they are after performing clustering

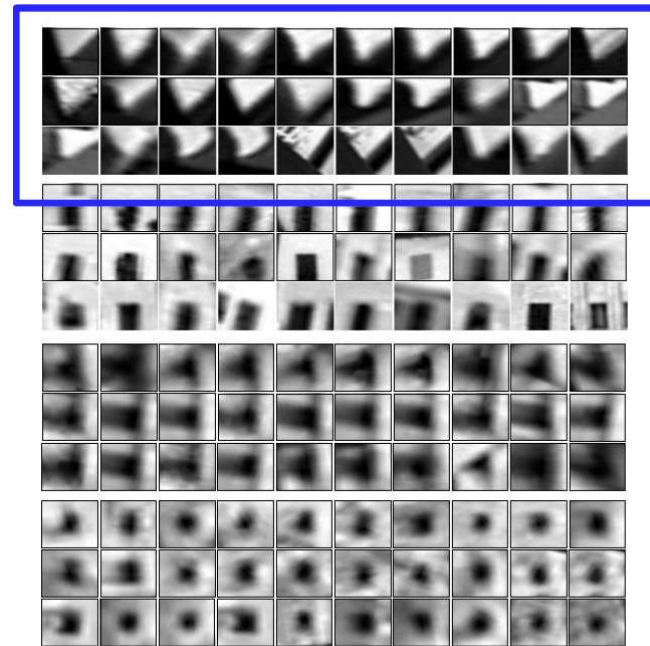
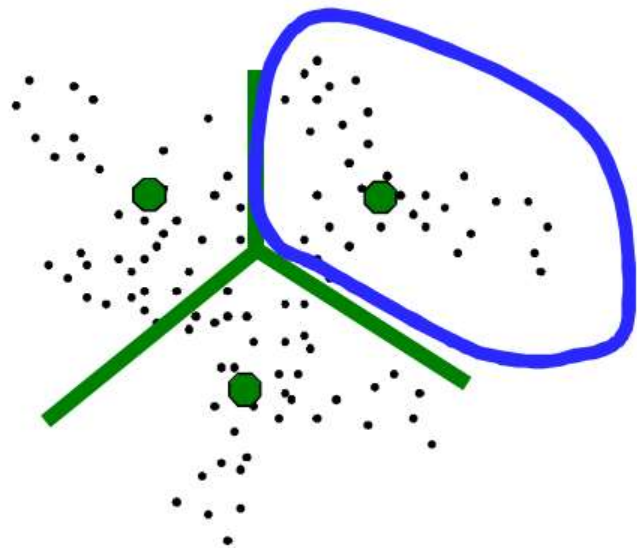
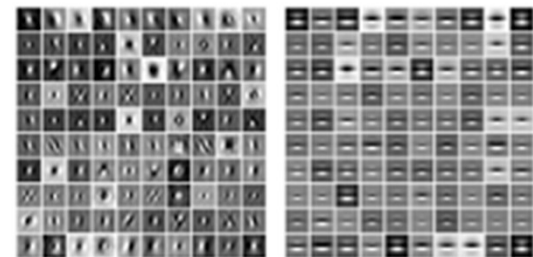
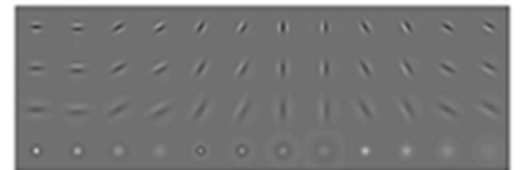
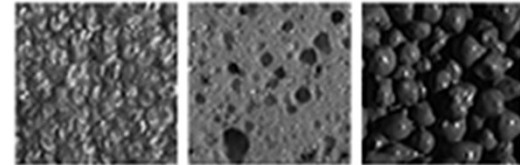
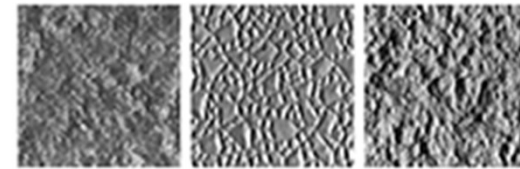


Figure from Sivic & Zisserman, ICCV 2003



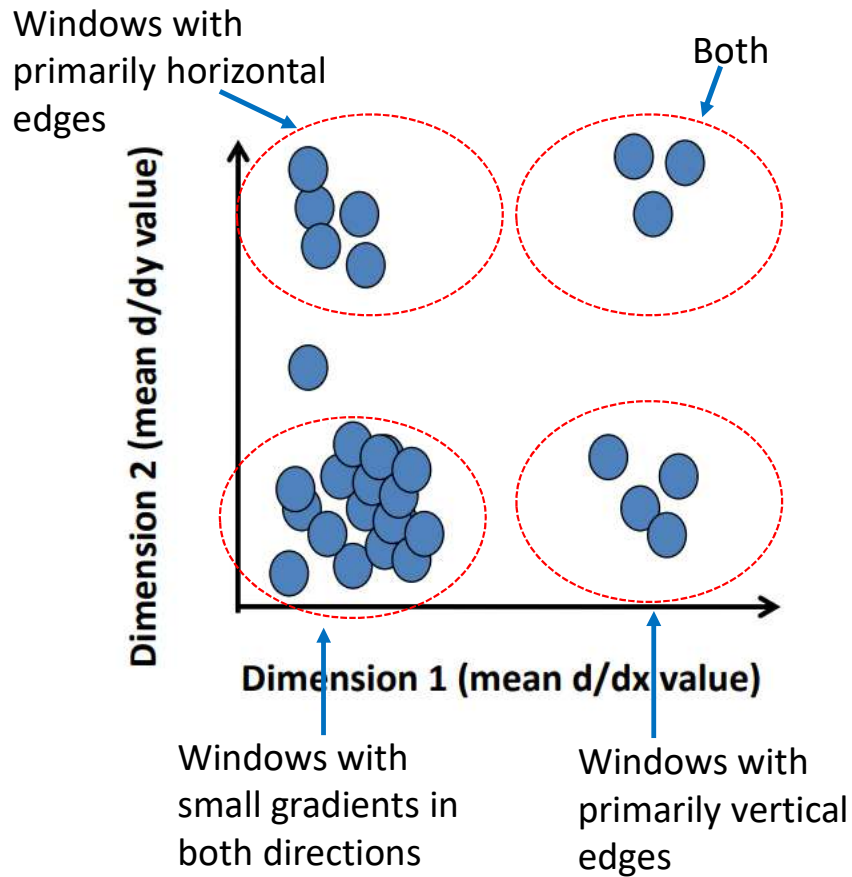
Visual words and “textons”

- › First explored in 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





Recall: Texture Representation



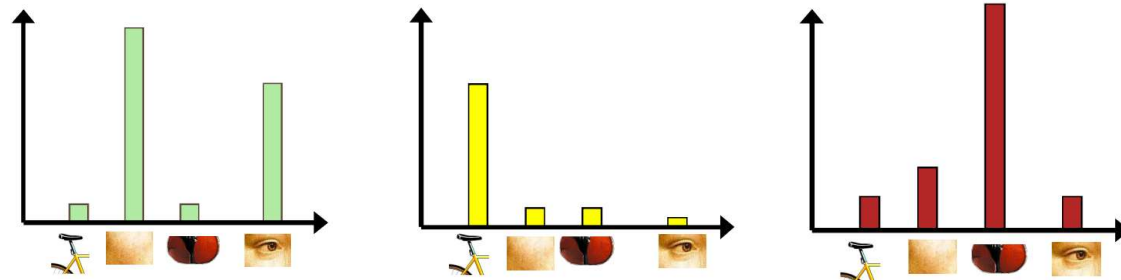
	<u>Mean d/dx value</u>	<u>Mean d/dy value</u>
Win. #1	4	10
Win. #2	18	7
⋮	⋮	⋮
Win. #9	20	20

•
•
•

statistics to summarize patterns in small windows



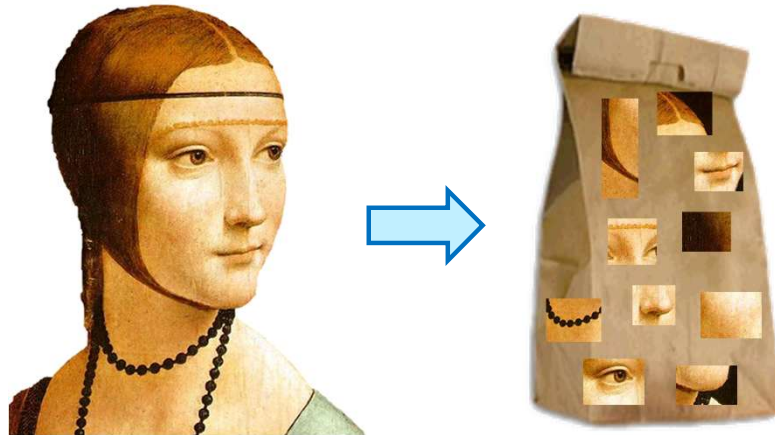
Whole image in terms of its “parts”





Bag of Visual Words

- › Summarize entire image based on its distribution (histogram) of visual word occurrences
- › Analogous to “bag of words” concept commonly used in documents



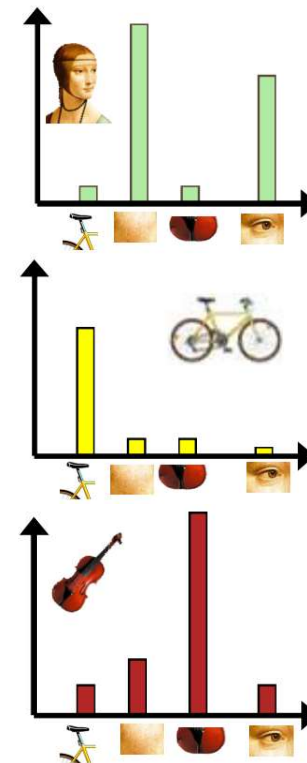


Bag of Visual Words

- › Summarize entire image based on its distribution (histogram) of visual word occurrences
- › Analogous to “bag of words” concept commonly used in documents



Visual words

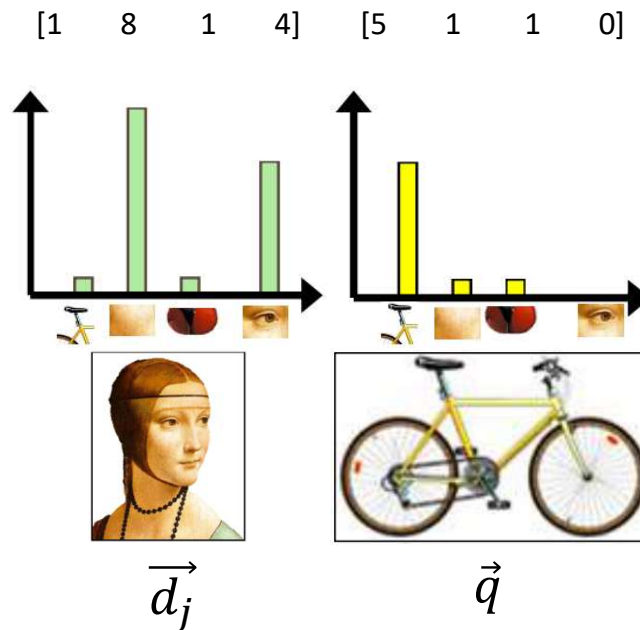


Bag of visual words



Comparing Bag of Words

- › Rank frames by normalized scalar (dot) product between their (possibly weighted) occurrence counts
 - nearest neighbour search for similar images



Cosine similarity

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

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

for vocabulary of V words



Visual Vocabulary Formation

> **Issues:**

- Sampling strategy: where to extract features?
 - > A complex image can contain features in irrelevant portions of the image
- Clustering / quantization algorithm
 - > What are some problems of k-means?
- Vocabulary size – number of words
 - > What's a good number of features to be used for representation

Inverted File Indexing





Representing Image with Visual Words



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



Analogy to Documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach our eyes. For a long time, the retinal image was considered as a simple picture. However, the discovery of the visual cortex in the monkey brain and the work of Hubel and Wiesel demonstrate that the message about the image falling on the retina undergoes a complex analysis in a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.



**sensory, brain,
visual, perception,
retinal, cerebral cortex,
eye, cell, optical
nerve, image
Hubel, Wiesel**

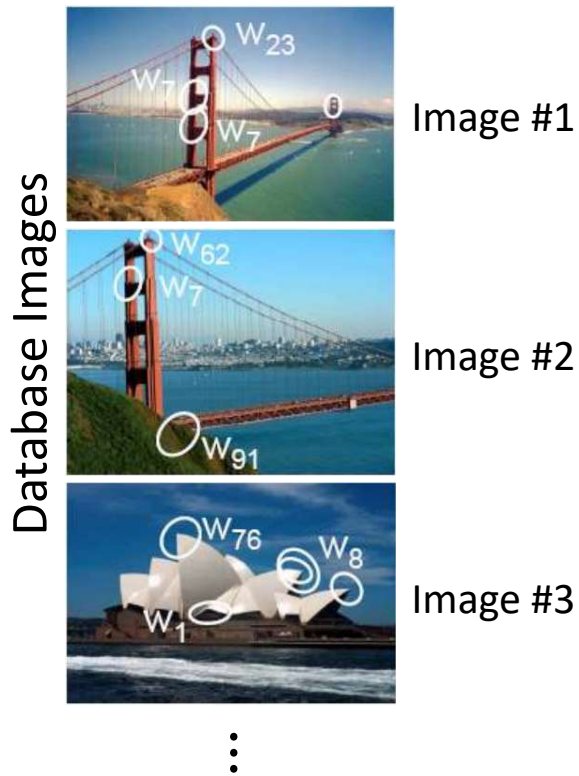
China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% increase in exports to \$750bn, compared with \$570bn in 2004. The surplus of \$660bn. The surplus would also annoy the US. China's deliberate policy of keeping the yuan undervalued against the dollar has also annoyed the US government. The US government also needs to increase demand so that it can remain a major country. China's policy of keeping the yuan against the dollar undervalued and permitted it to trade within a narrow range but the US wants the yuan to be allowed to float 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.



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



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



Inverted File Index



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

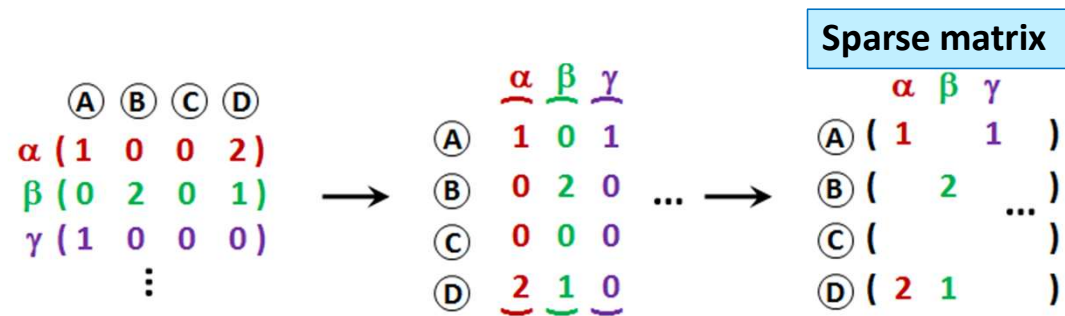


- › A new query image is mapped to indices of database images that share a particular word



Inverted File Index

- › The visual vocabulary can be very big (thousands to millions of words)
- › For quick searching, use an **inverted file index** in sparse form to reduce the size of a matrix that has many zero elements



- › Logically, the **weight of each word** computed from its frequency divided by length of vector of words could be useful...



tf-idf Weighting

- › **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- › Describe by frequency of each word within it, **downweight words** that appear often in database
- › Standard weighting for text retrieval – can be applied to image search too

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

Number of occurrences of word i in document d → n_{id}

Number of words in document d → n_d

Total number of documents in database → N

Number of documents word i occurs in, in whole database → n_i

Common to perform normalization later, using just the raw frequency

Denominator can be 0 if the word does not occur. So normally, we can use $\log \frac{1+N}{1+n_i} + 1$ to solve this problem



tf-idf Example

› **Remember:** document \Rightarrow image

$$t_{id} = n_{id} \left(\log \frac{1 + N}{1 + n_i} + 1 \right)$$

N = number of images in database

n_i = number of images with word i

n_{id} = number of occurrences of word i in image d

For Image 0:

$$n_0 = 6, n_{0,0} = 3$$

$$t_{0,0} = 3 \left(\log \frac{1 + 6}{1 + 3} + 1 \right) = 3$$

$$t_{1,0} = 0 \left(\log \frac{1 + 6}{1 + 1} + 1 \right) = 0$$

$$t_{2,0} = 1 \left(\log \frac{1 + 6}{1 + 2} + 1 \right) = 1.368$$

	3 words			$N = 6$
Word, i	0	1	2	Image, d
Counts =	[3,	0,	1],	0
...	[2,	0,	0],	1
...	[3,	0,	0],	2
...	[4,	0,	0],	3
...	[3,	2,	0],	4
...	[3,	0,	2]]	5

[3, 0, 1.368]

[0.9099, 0, 0.4149]

We see an increase in this value.... Why?

Due to these changes, **normalization MUST** be performed after that, by normalizing the *tf-idf* value by its magnitude (**Euclidean norm**: $\|n_{id}\|_2$)



tf-idf Example

Word, i	0	1	2	Image, d
Counts =	[3,	0,	1],	0
...	[2,	0,	0],	1
...	[3,	0,	0],	2
...	[4,	0,	0],	3
...	[3,	2,	0],	4
...	[3,	0,	2]]	5

[3, 0, 1.368]

We see an increase in this value. Why?

- A. The visual word has the least number of occurrence in the database, so it is most important.
- B. The visual word has less number of occurrence but in various images, so it is useful as an index.
- C. The visual word frequency needs to be normalized hence the change in value.
- D. The frequency properly reflects the general count of words, not only in this database.



BOW: Order-less Representation

- › **Bag-of-Words** \Rightarrow orderless representation (spatial relationships between features are gone)
- › **But we can use the following ideas to help:**
 - **Visual “phrases”** – frequently co-occurring words
Descriptive visual words and visual phrases for image applications
 - Let **position** be part of each feature
 - **Localize it further:** Perform BOW only within sub-grids or blocks of an image
 - After matching, **verify spatial consistency** (look at neighbours, are they same too?)

Application and Scoring





Application of BOW for Image Retrieval

- › Retrieve an object from video that matches the query region

Visually defined query

"Find this clock"



"Find this place"



"Groundhog Day" [Rammis, 1993]



Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003



Video Google System

> “Object matching” in videos



Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

retrieved shots





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

<http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html>





Scoring Retrieval Quality

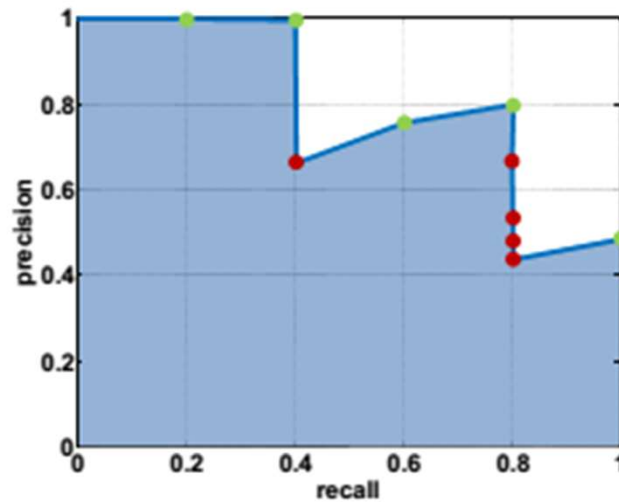
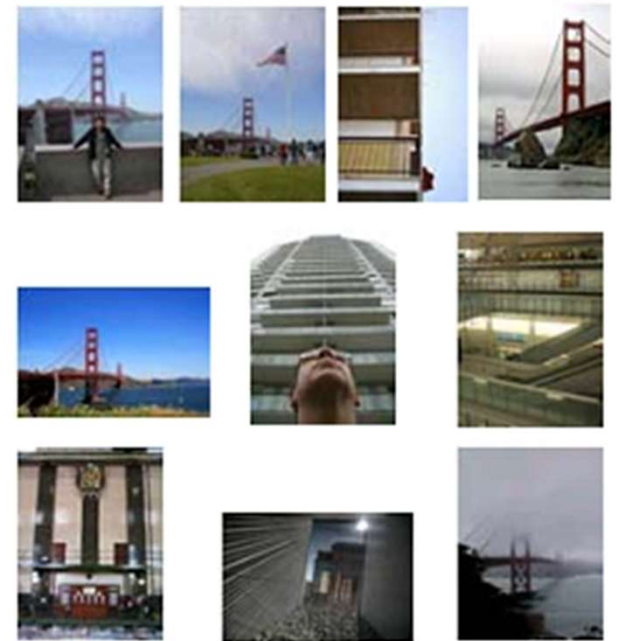
› Example:

- Database size: 10 images
- Relevant (total): 5 images
- Precision = # relevant / # returned
- Recall = # relevant / # total relevant



Query

Results (ordered):



Precision-Recall curve



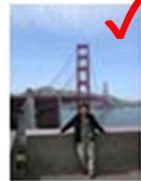
Scoring Retrieval Quality

Example

Results (ordered):



Query



Recall = 5/8

Precision = 5/10

- › Total returned images: 10
 - This can be tuned by setting a threshold on the distance/similarity score
- › Relevant images retrieved: 5
 - Based on ground truth, the number of bridge images returned.
- › Total relevant images: 8
 - Based on ground truth, the number of the bridge images that should have been returned



Bag of Words: Pros and Cons

> Pros

- Flexible to geometry / deformations / viewpoint
- Compact summary of image content
- Provides vector representation for sets
- Very good results in practice

> Cons

- Basic model ignores geometry – must verify or encode via features
- Background and foreground mixed when bag covers whole image
- Optimal vocabulary formation remains unclear (how many words?)

SUMMARY

- › **Matching local invariant features**
 - › Useful not only to provide matches for multi-view geometry, but also to find objects and scenes in an image retrieval task
- › **Bag of words (BOW) representation**
 - › Quantize feature space to make discrete set of visual words
 - › summarize image by distribution (or histogram) of words
 - › then index these words
- › **Inverted index**
 - › Pre-compute index to enable faster search at query time
- › **Next**
 - › Deep Learning for Computer Vision

