
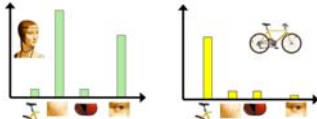

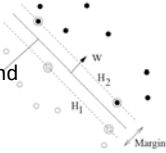


Recognition and learning

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

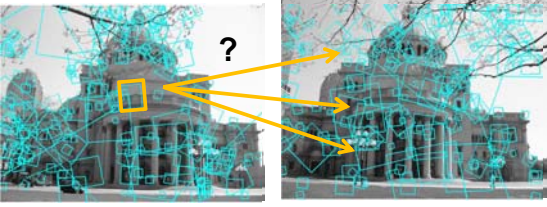
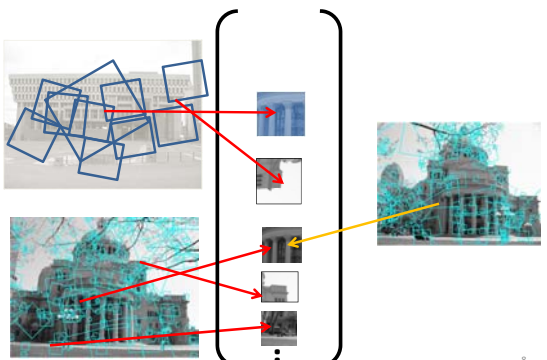


Image 1 Image 2

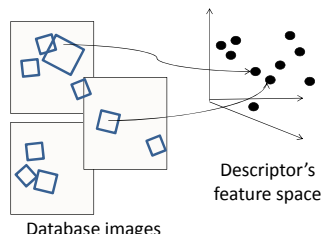
To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)
 Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

Indexing local features



Indexing local features

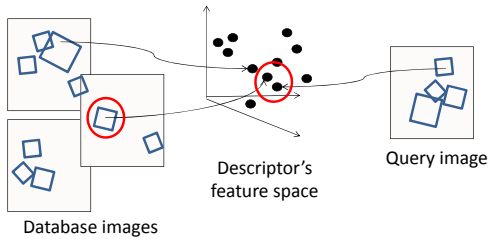
- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)



Database images Descriptor's feature space

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

Index	Value
Alabama	100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

- 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

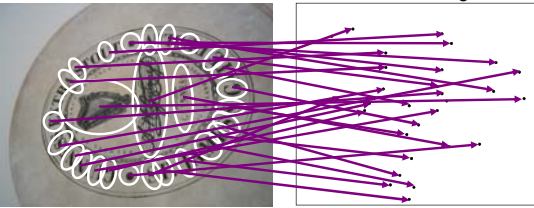
Text retrieval vs. image search

- 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

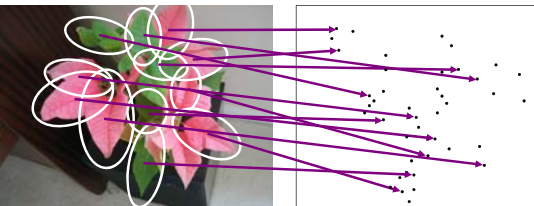
- Extract some local features from a number of images ...



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


14

Visual words: main idea



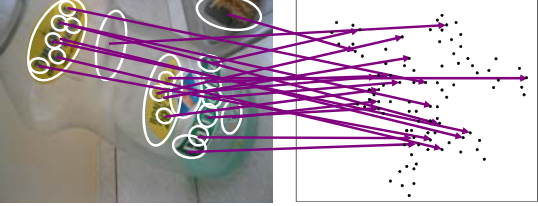
15

Visual words: main idea

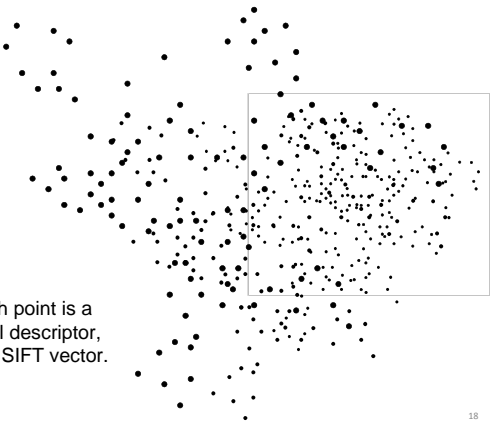


16

Visual words: main idea

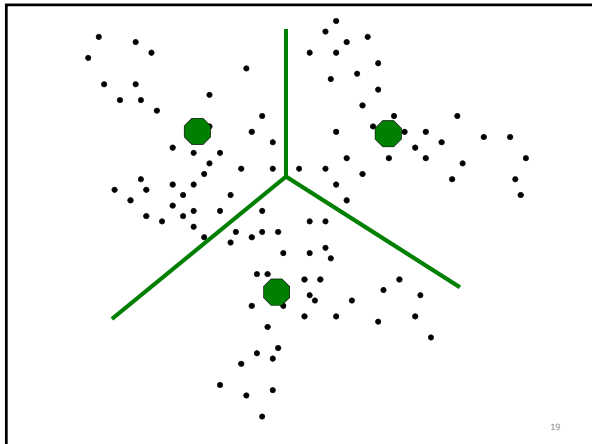


17



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

18



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.

Word #2

Descriptor's feature space

20

Visual words

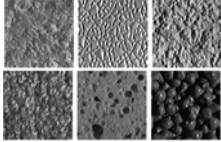
- Example: each group of patches belongs to the same visual word


Figure from Sivic & Zisserman, ICCV 2003

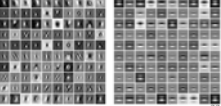
21

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.



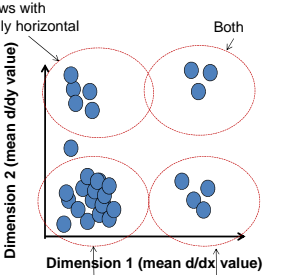




Leung & Malik 1999; Varma & Zisserman, 2002

Recall: Texture representation example

Windows with primarily horizontal edges



Dimension 1 (mean d/dx value)

Dimension 2 (mean d/dy value)

	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

Database images





Image #	Word #	Image #
Image #1	1	3
	2	
	...	
Image #2	7	1, 2
	8	3
	9	
Image #3	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?



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.

26

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

27

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 these impressions. The messages that reach the brain from the eyes are processed at a certain point by the cerebral cortex. Through this process, we now know more about the visual image. Various cell layers in the visual cortex. Hubel and Wiesel have been able to determine the message about the image falling on the retina. The message undergoes a step-wise 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.

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 a 18% jump in exports. Further, it predicted that China's exports will rise to \$1.2 trillion in 2007. The ministry deliberated the surplus in the first quarter. Xiaochuan said more to be stayed with the value of the yuan. The yuan rose in July and permitted to trade freely. However, Beijing has made it clear that it will take its time and tread carefully allowing the yuan to rise further in value.

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

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

The image illustrates the concept of visual words. At the top, three objects are shown: a woman's face, a bicycle, and a violin. Below each object is a histogram showing the distribution of visual words (represented by small icons) that occur in that object. For the face, the distribution is dominated by words related to facial features. For the bicycle, it's dominated by words related to wheels and frames. For the violin, it's dominated by words related to the instrument's body and strings. At the bottom, a grid of small image patches shows the individual visual words used in the histograms.

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.

The diagram illustrates the 'Bag of visual words' concept. It shows three examples: a woman's face, a bicycle, and a violin. For each object, a histogram shows the distribution of visual words (represented by small icons) that occur in that object. Below each histogram is a small grid of the visual words used in the histogram. This represents the entire image as a collection of words, analogous to a bag of words in text processing.


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


Find this clock*



Find this place*



"Groundhog Day" [Rammsis, 1993]



Sivic & Zisserman, ICCV 2003 32

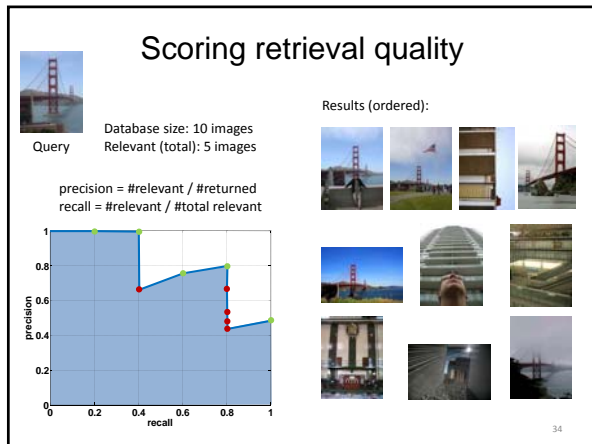
Example

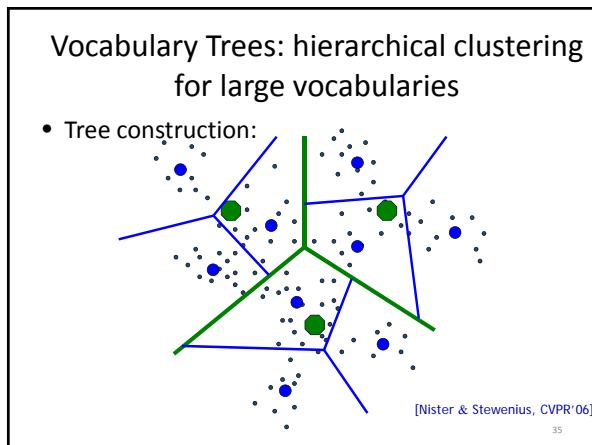


retrieved shots

Start frame 72707	End frame 73024	End frame 73024
Start frame 74242	End frame 74376	End frame 74488
Start frame 74770	End frame 75274	End frame 75340
Start frame 74879	End frame 74200	End frame 74200
Start frame 70709	End frame 70720	End frame 70709
Start frame 40700	End frame 40024	End frame 43040
Start frame 39301	End frame 39676	End frame 39770

Sivic & Zisserman, ICCV 2003 33





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

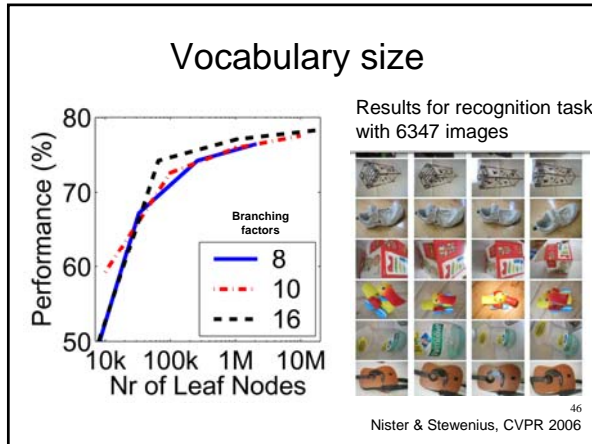
- **Matching local invariant features:** useful to provide matches to find objects and scenes.
- **Bag of words** representation: quantize feature space to make discrete set of visual words
- **Inverted index:** pre-compute index to enable faster search at query time

44

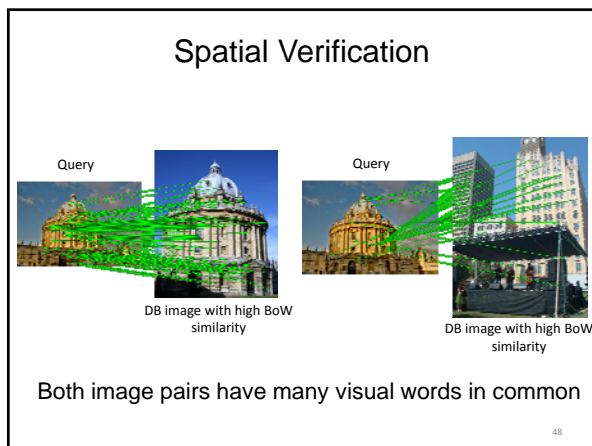
Instance recognition: remaining issues

- 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



- ### 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

Query DB image with high BoW similarity

Query DB image with high BoW similarity

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

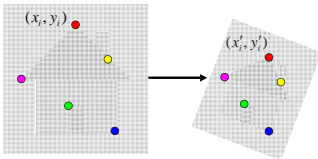
Query DB image with high BoW similarity

Query DB image with high BoW similarity

Only some of the matches are mutually consistent

51

Recall: Fitting an affine transformation

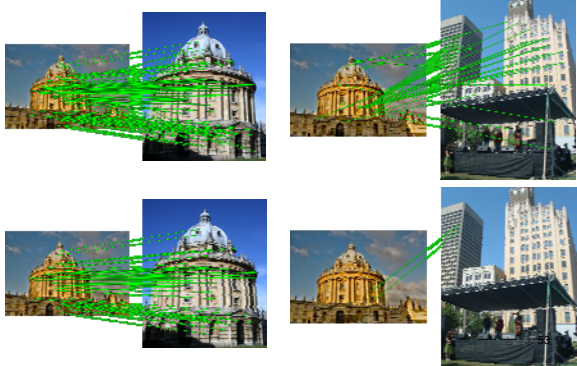


Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.

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

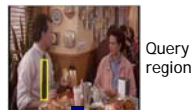
$$\begin{bmatrix} x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \dots \\ x'_i \\ y'_i \\ \dots \end{bmatrix}$$

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



Query region




Retrieved frames

Sivic & Zisserman, ICCV 2003

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

Visual Object Recognition Tutorial


Example Applications



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

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

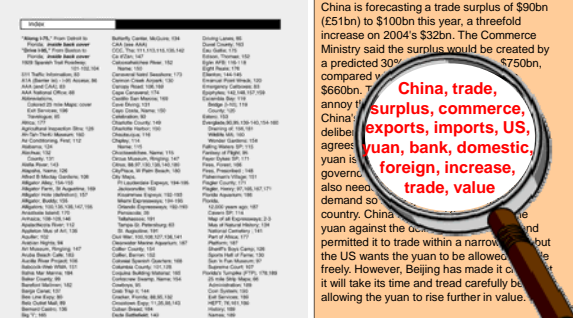
Application: Large-Scale Retrieval



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% increase in exports to \$750bn, compared with \$575bn in 2004. China's trade surplus, however, is also needed to cover the government's deficit. China's trade surplus, however, is also needed to cover the government's deficit.

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

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 **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':

- 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



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
