EE 6882
Visual Search Engine

Feb. 27th, 2012
Lecture #6

- Object Search Using Local Features
- Applications of Mobile Visual Search
- Mid-Term Project

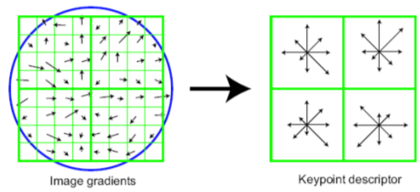
Reading List

- Sivic, J. and A. Zisserman, *Video Google: A text retrieval approach to object matching in videos, in ICCV. 2003.*
- Nister, D. and H. Stewenius. *Scalable recognition with a vocabulary tree. in CVPR. 2006.*
- Chum, O., et al. *Total recall: Automatic query expansion with a generative feature model for object retrieval. in ICCV. 2007.*
- Felix X. Yu, Rongrong Ji, Tongtao Zhang, Shih-Fu Chang. *Active Query Sensing for mobile location search*. In Proceeding of ACM International Conference on Multimedia (ACM MM), 2011.
- Junfeng He, Tai-Hsu Lin, Jinyuan Feng, Shih-Fu Chang. *Mobile product search with bag of hash bits*. In Proceeding of ACM International Conference on Multimedia (ACM MM), demo paper, 2011.
- Nokia. *Nokia Point and Find. 2006; Available from: http://www.pointandfind.nokia.com.*
- Kooaba. *Available from: http://www.kooaba.com.*

Local Appearance Descriptor (SIFT)

[Lowe, ICCV 1999]

Compute gradient
in a local patch



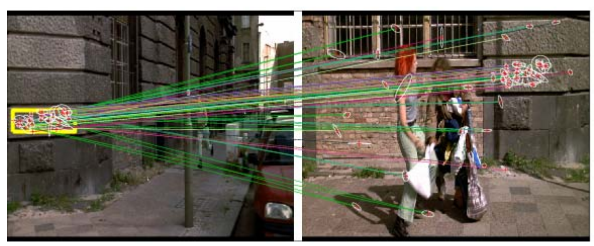
Histogram of oriented gradients over local grids

- e.g., 4x4 grids and 8 directions
  -> 4x4x8=128 dimensions
- Rotation-aligned, Scale invariant

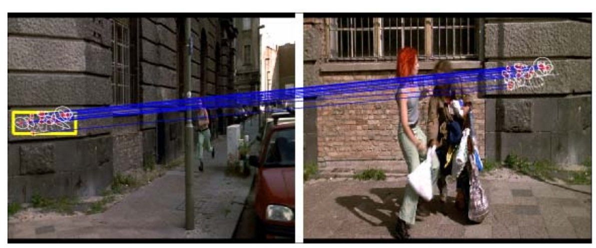
There are many other local features, e.g., SUFR, HOG, BRIEF, MSER. STIP

Example of Local Feature Matching

Initial matches



Spatial consistency required

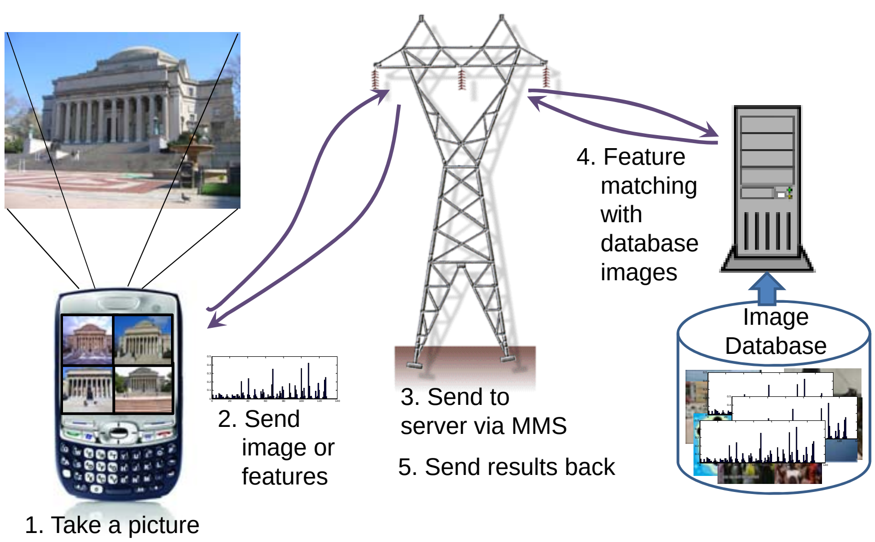


Slide credit: J. Sivic

Application:
Large Scale Mobile Visual Search

digital video | multimedia lab

Mobile Visual Search



Application: particular object retrieval

Example I: Visual search in feature films

Visually defined query

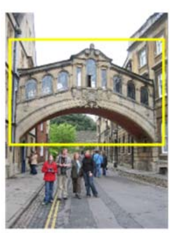
"Groundhog Day" [Rammis, 1993]

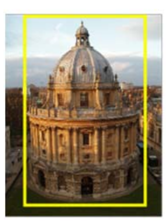
"Find this clock"

"Find this place"

Slide credit: J. Sivic

Example II: Search photos on the web for particular places





Find these landmarks

...in these images and 1M more

Slide credit: J. Sivic

Global vs. Local Feature Matching

- Global
  - Convert query and database images to global representations such as Bags of Words
  - Perform global matching
- Local
  - Use each local feature as query
  - Search matched local features in the database
  - Rank images
  - Perform spatial verification

Outline of a local feature retrieval strategy

Local Features

frames

invariant descriptor vectors

invariant descriptor vectors

1. Compute affine covariant regions in each frame independently
2. “Represent” each region by a vector of descriptors
3. Finding corresponding regions is transformed to finding nearest neighbour vectors
4. Rank retrieved frames by number of corresponding regions
5. Verify retrieved frame based on spatial consistency

Slide credit: J. Sivic

Bottleneck:
nearest-neighbor matching over gigantic database

Solve following problem for all feature vectors, $\mathbf{x_j}$, in the query image:

$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||$$

where $\mathbf{x_i}$ are features in database images.

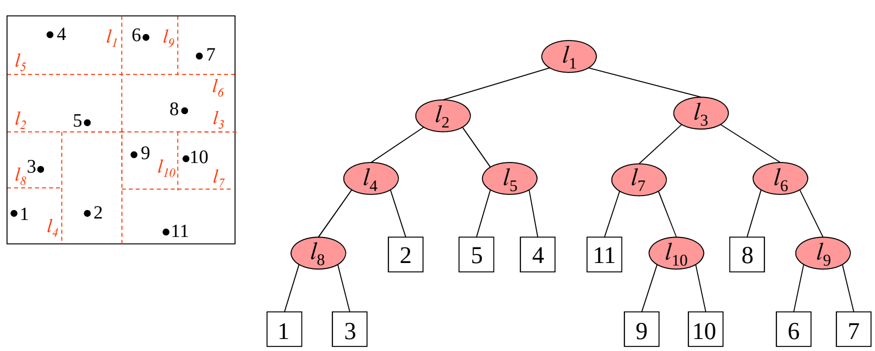
Nearest-neighbour matching is the major computational bottleneck

- Linear search performs *dxn* operations for *n* features in the database and *d* dimensions
- n may be as high as billions
- No exact methods are faster than linear search for d>10
- Explore approximate methods (e.g., tree based indexing)

Slide credit: J. Sivic

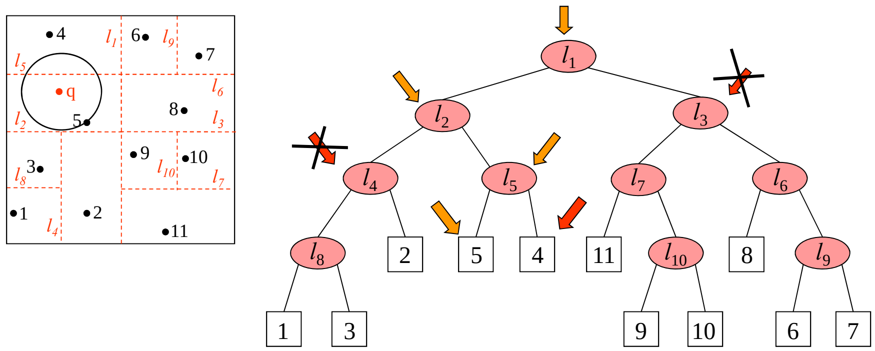
K-d tree construction

Simple 2D example



Slide credit: Anna Atramentov

K-d tree query

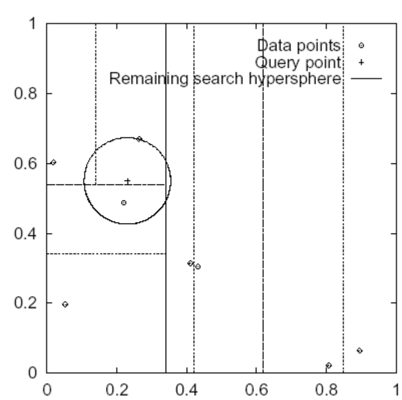


Slide credit: Anna Atramentov

Approximate nearest neighbour K-d tree search

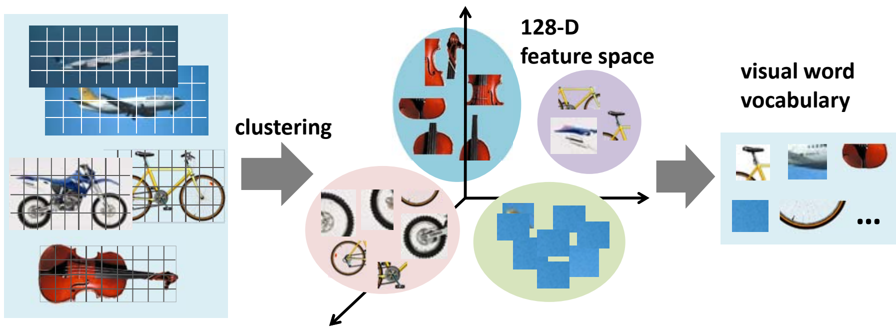
Issues

- Need backtracing to find exact NN
- Exponential cost when dimension grows
- Remedy: limit the number of neighboring bins to explore
- Search k-d tree bins in order of distance from query



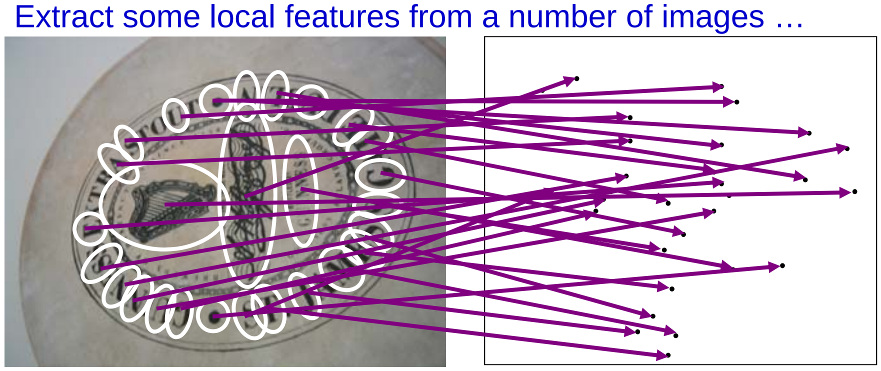
Slide credit: J. Sivic

Alternative method: mapping local features to *Visual Words*



Visual words: main idea

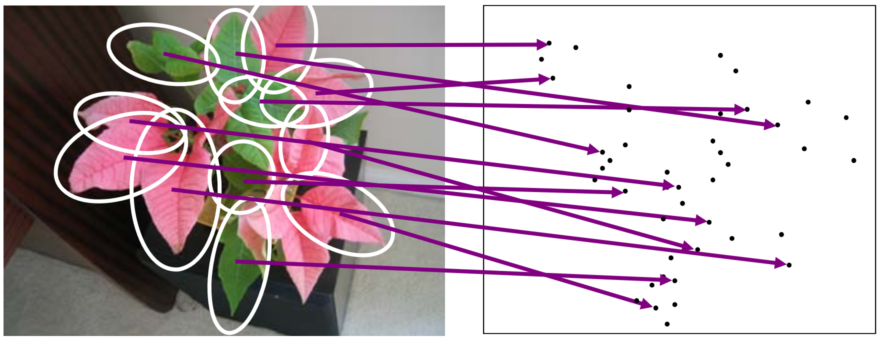
Extract some local features from a number of images …



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

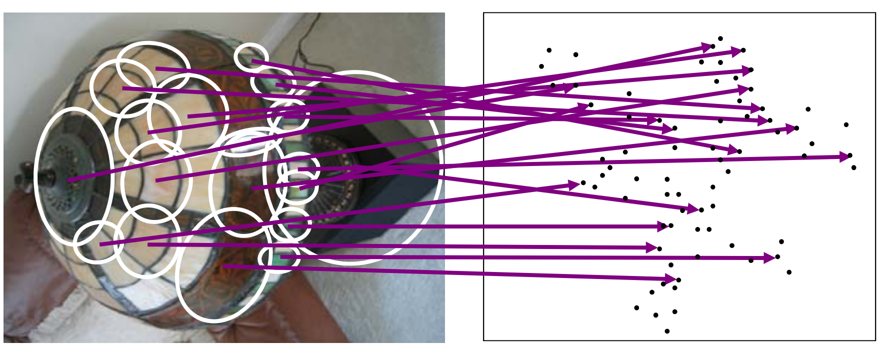
Slide credit: D. Nister

Visual words: main idea



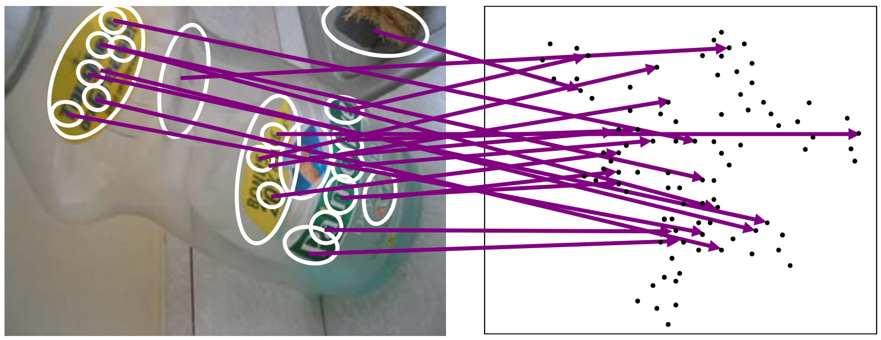
Slide credit: D. Nister

Visual words: main idea

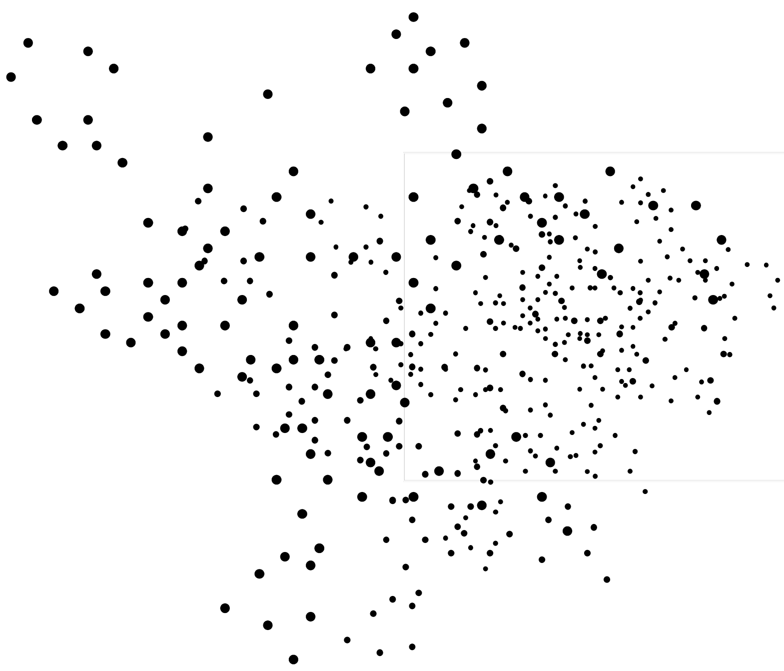


Slide credit: D. Nister

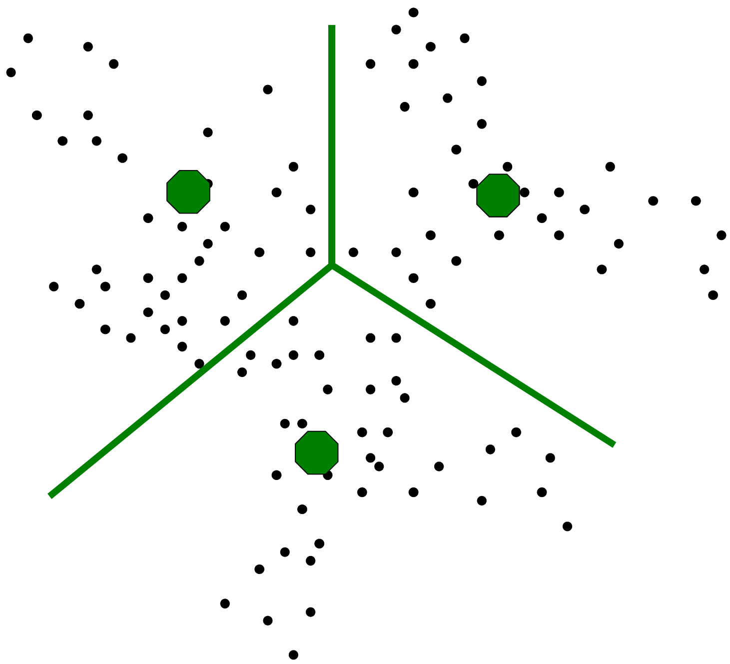
Visual words: main idea



Slide credit: D. Nister



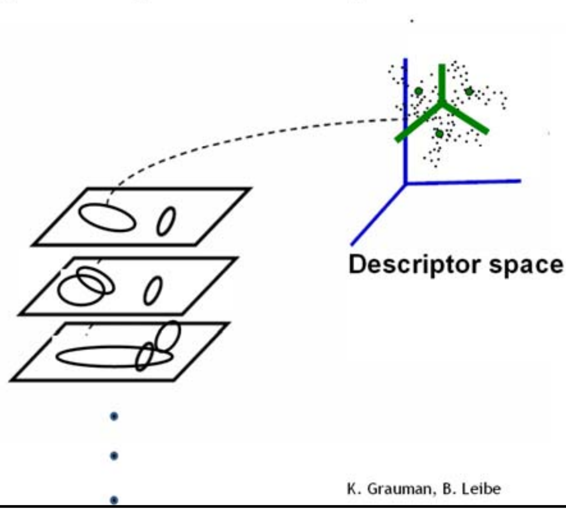
Slide credit: D. Nister



Slide credit: D. Nister

Visual words: main idea

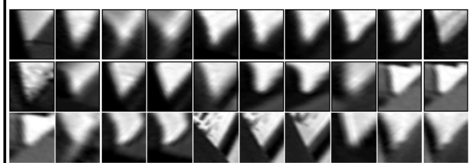
Map high-dimensional descriptors to tokens/words by quantizing the feature space



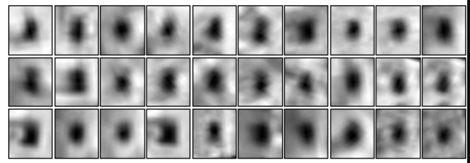
- Quantize via clustering, let cluster centers be the prototype "words"

K. Grauman, B. Leibe

Visual Words: Image Patch Patterns

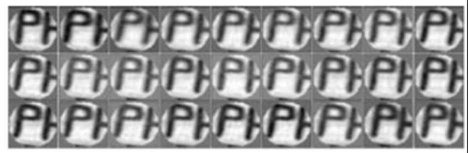


Corners



Blobs

eyes



letters

Sivic and Zisserman, "Video Google", 2006

Inverted file index for images comprised of visual words

frame #5

frame #10

Word number	List of image numbers
1	→ 5,10, ...
2	→ 10,...
...	...

- Score each image by the number of common visual words (tentative correspondences)
- But: does not take into account spatial layout of regions

Slide credit: J. Sivic

How to create visual words?
Clustering / quantization methods

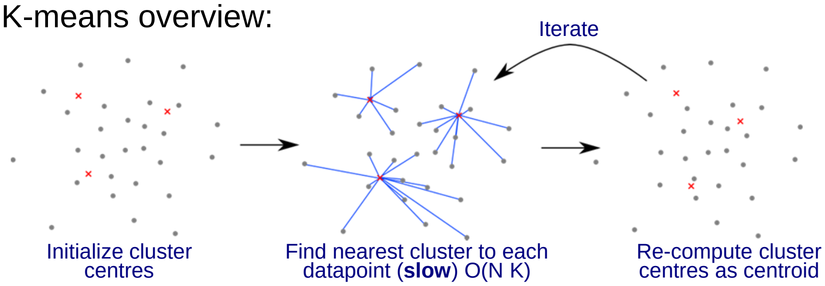
- k-means (typical choice), agglomerative clustering, mean-shift,…

- Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies
  - Vocabulary tree [Nister & Stewenius, CVPR 2006]

Slide credit: J. Sivic

Quantization using K-means

- K-means overview:



- K-means provably locally minimizes the sum of squared errors (SSE) between a cluster centre and its points

- But: The quantizer depends on the initialization.

- The nearest neighbour search is the bottleneck

Slide credit: J. Sivic

Approximate K-means

- Use the approximate nearest neighbour search (randomized forest of kd-trees) to determine the closest cluster centre for each data point.

- Original K-means complexity = O(N K)
- Approximate K-means complexity = O(N log K)

- Can be scaled to very large K.

Slide credit: J. Sivic

Clustering / quantization methods

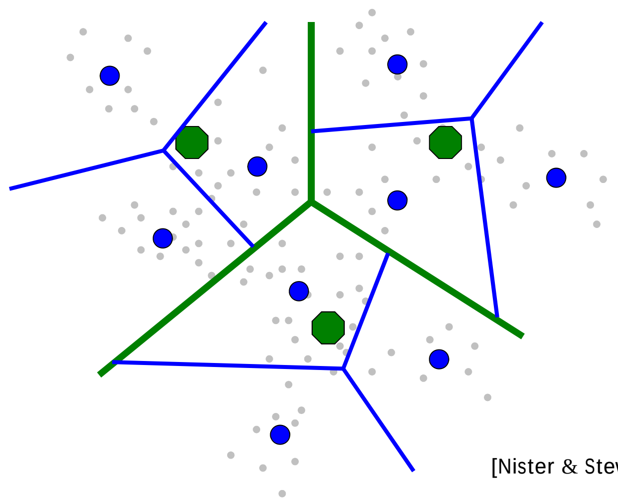
- k-means (typical choice), agglomerative clustering, mean-shift,…

- **Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies**
  - **Vocabulary tree [Nister & Stewenius, CVPR 2006]**

Slide credit: J. Sivic

Example: Recognition with Vocabulary Tree

Tree construction:

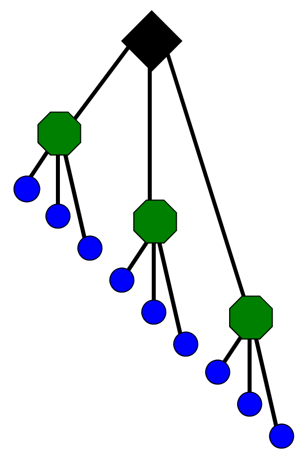


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

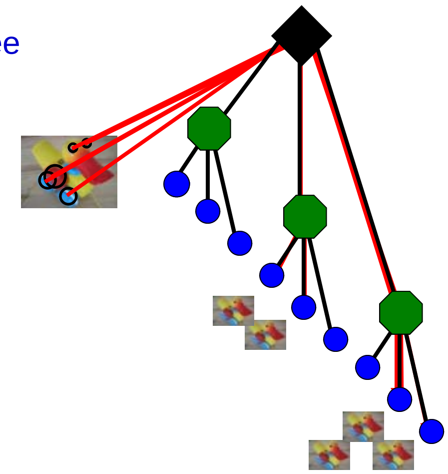


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

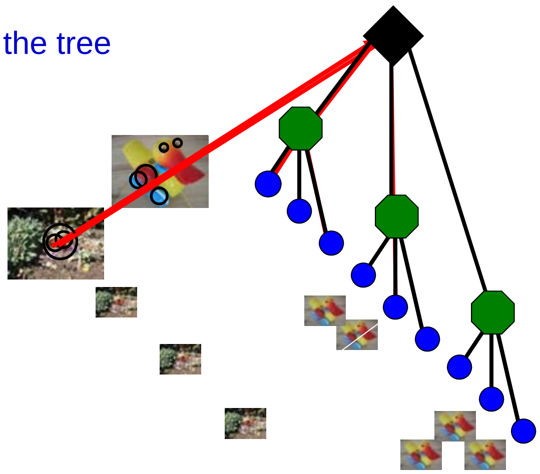


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

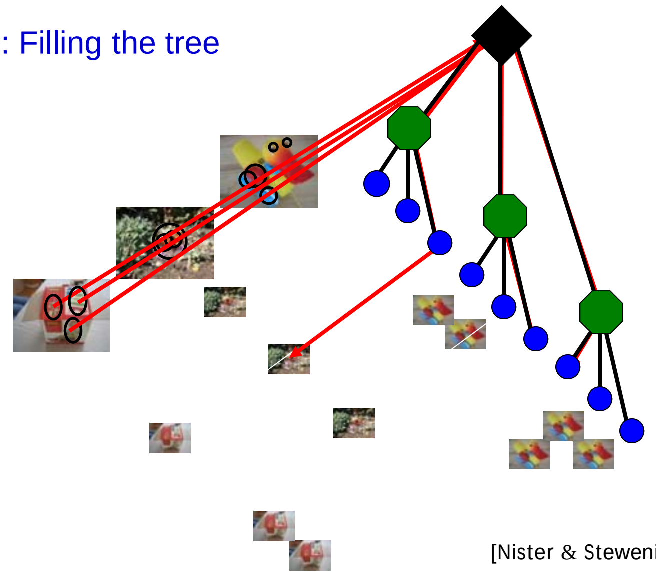


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

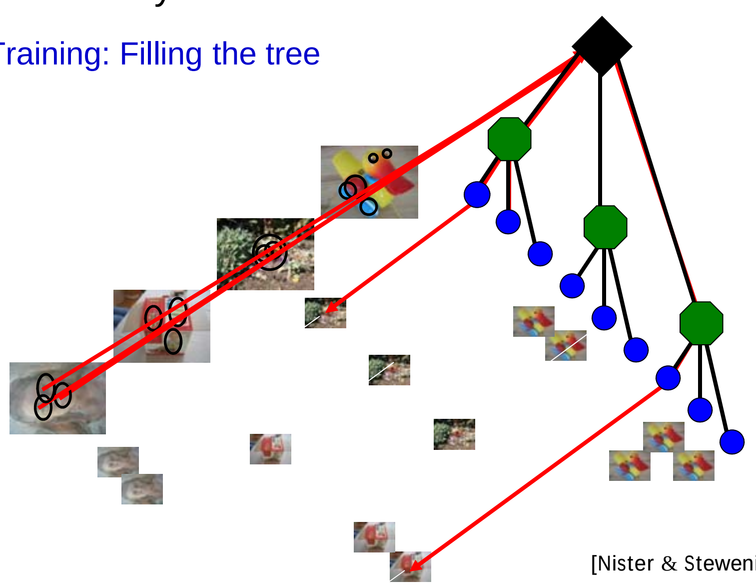


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Training: Filling the tree

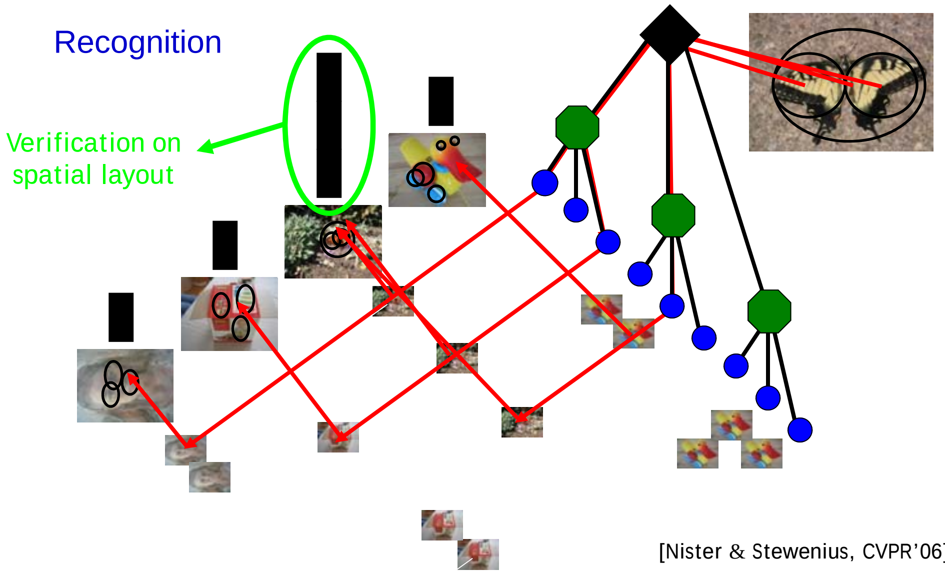


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

Recognition

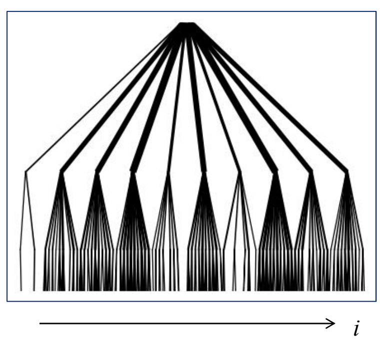


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Voc Tree can also be used
to score images efficiently

q: query feature vector
d: database feature vector



$$
\begin{aligned}
\| \mathbf{q} - \mathbf{d} \|_p^p &= \sum_i |q_i - d_i|^p \\
&= \sum_{i|d_i=0} |q_i|^p + \sum_{i|q_i=0} |d_i|^p + \sum_{i|q_i\neq 0, d_i \neq 0} |q_i - d_i|^p \\
&= \|\mathbf{q}\|_p^p + \|\mathbf{d}\|_p^p + \sum_{i|q_i\neq 0, d_i \neq 0} (|q_i - d_i|^p - |q_i|^p - |d_i|^p) \\
&= 2 + \sum_{i|q_i\neq 0, d_i \neq 0} (|q_i - d_i|^p - |q_i|^p - |d_i|^p),
\end{aligned}
$$

$$
\boxed{\| \mathbf{q} - \mathbf{d} \|_2^2 = 2 - 2 \sum_{i|q_i\neq 0, d_i \neq 0} q_i d_i}
$$

Incremental update of scores for every query visual word

Vocabulary Tree: Performance

Evaluated on large databases

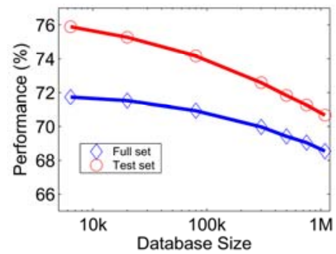
- Indexing with up to 1M images

Online recognition for database of 50,000 CD covers

- Retrieval in ~1s

Find experimentally that large vocabularies can be beneficial for recognition

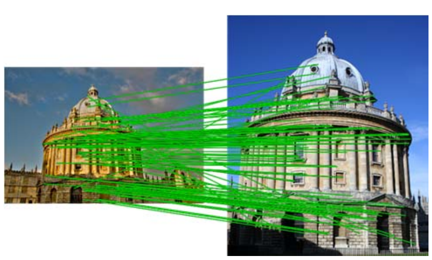
[Nister & Stewenius, CVPR'06]

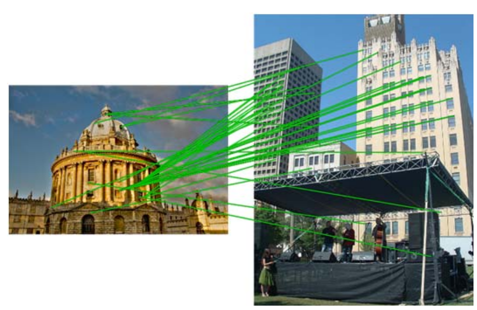


Slide credit: J. Sivic

Beyond Bag of Words

- Use the **position** and **shape** of the underlying features to improve retrieval quality



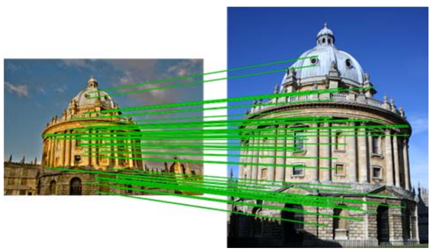


- Both images have many matches – which is correct?

Slide credit: J. Sivic

Beyond Bag of Words

- We can measure **spatial consistency** between the query and each result to improve retrieval quality



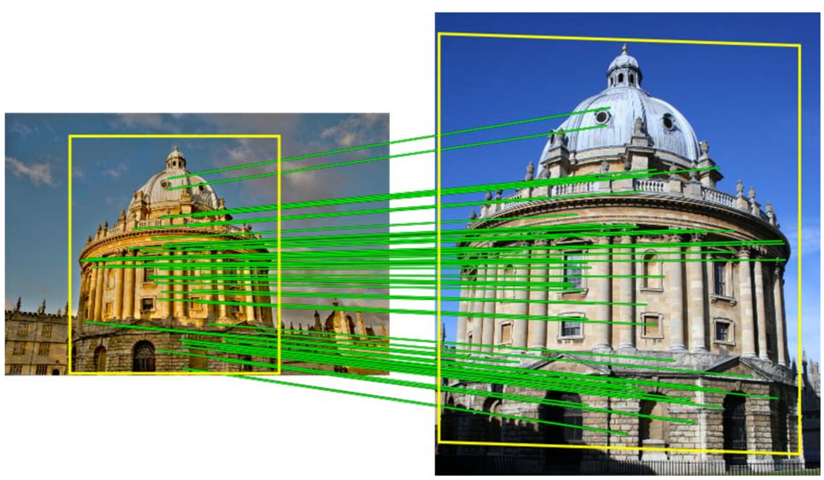
Many spatially consistent matches – **correct result**

Few spatially consistent matches – **incorrect result**

Slide credit: J. Sivic

Beyond Bag of Words

- Extra bonus – gives **localization** of the object

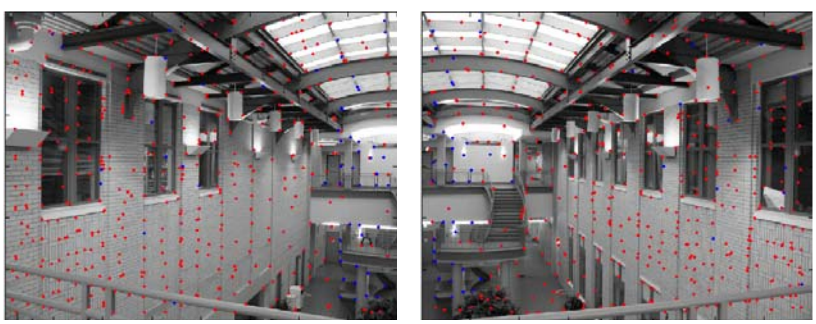


Slide credit: J. Sivic

Spatial Verification

- Check consistency of relative distance (shift)
- Check consistency of scale change
- Check consistency of transformation (RANSAC)

Feature-space outliner rejection



Can we now compute H from the blue points?

- No! Still too many outliers…
- What can we do?

Slide of A. Efros

RANSAC for estimating homography

RANSAC loop:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \mathbf{H}\, p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

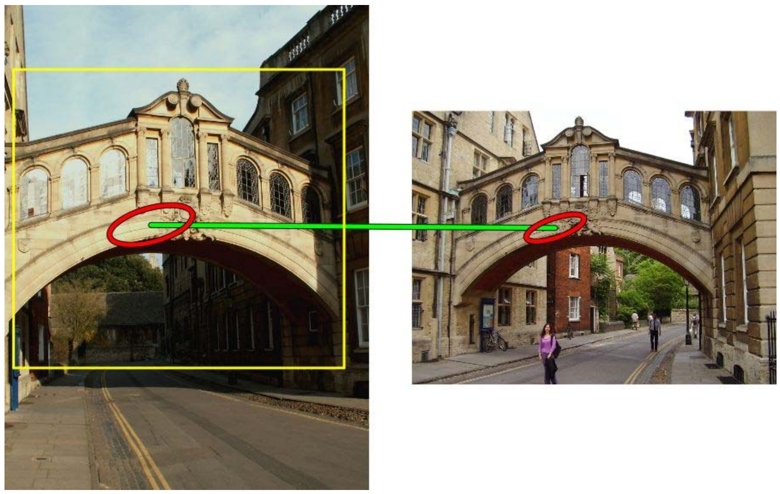
Slide of A. Efros

RANSAC

Slide of A. Efros

Estimating spatial correspondences

1. Test each correspondence



Slide credit: J. Sivic

Estimating spatial correspondences

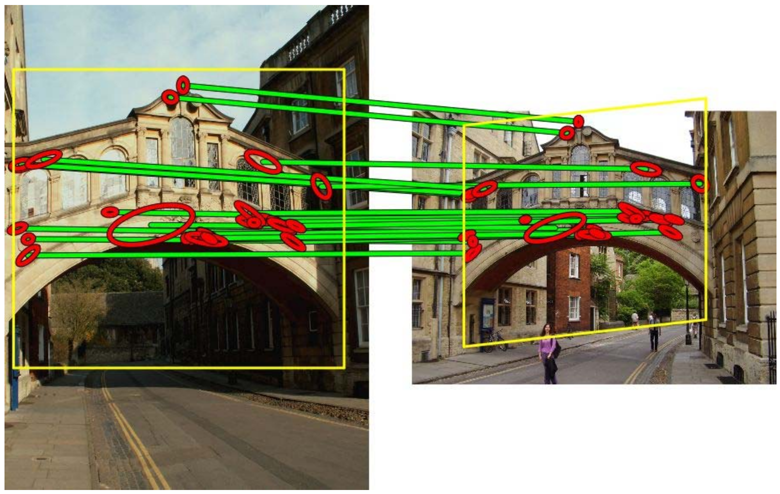
2. Compute a planar affine transformation (6 dof)

Need just one correspondence

Slide credit: J. Sivic

Estimating spatial correspondences

3. Score by number of consistent matches

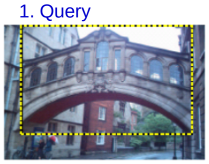


Re-estimate full affine transformation (6 dof)

Slide credit: J. Sivic

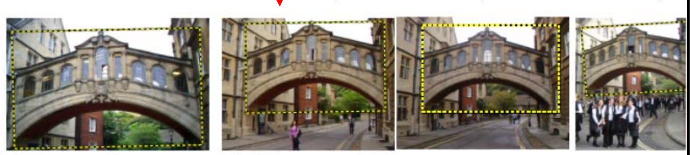
Verification by spatial layout - overview

1. Query



2. Initial retrieval set (bag of words model)

3. Spatial verification (re-rank on # of inliers)



Slide credit: J. Sivic

Oxford buildings dataset

- Automatically crawled from flickr
- Consists of:

Dataset	Resolution	# images	# features	Descriptor size
i	1024 × 768	5,062	16,334,970	1.9 GB
ii	1024 × 768	99,782	277,770,833	33.1 GB
iii	500 × 333	1,040,801	1,186,469,709	141.4 GB
Total		1,145,645	1,480,575,512	176.4 GB

Slide credit: J. Sivic

Oxford buildings dataset

- Landmarks plus queries used for evaluation

- Ground truth obtained for 11 landmarks
- Evaluate performance by mean Average Precision

Slide credit: J. Sivic

Query Expansion in text

In text :

- Reissue top n responses as queries
- Pseudo/blind relevance feedback
- Danger of topic drift

In vision:

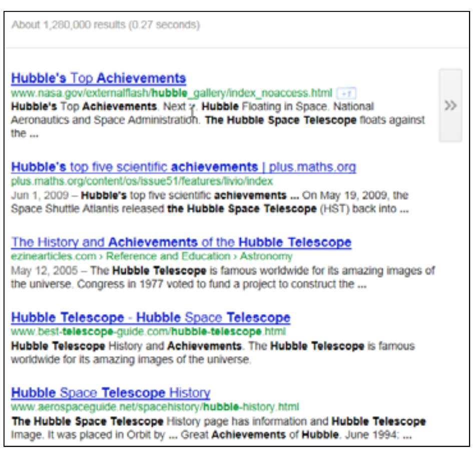
- Reissue spatially verified image regions as queries

Slide credit: J. Sivic

Query expansion in text - example

Original query: Hubble Telescope Achievements

Query expansion: Select top 20 terms from top 20 documents



Added terms:

telescope
hubble
space
nasa
ultraviolet
shuttle
mirror
telescopes
earth
discovery
orbit
flaw
scientists
launch
stars
universe
mirrors
light
optical
species

Example from: Jimmy Lin, University of Maryland

Slide credit: J. Sivic

Automatic query expansion

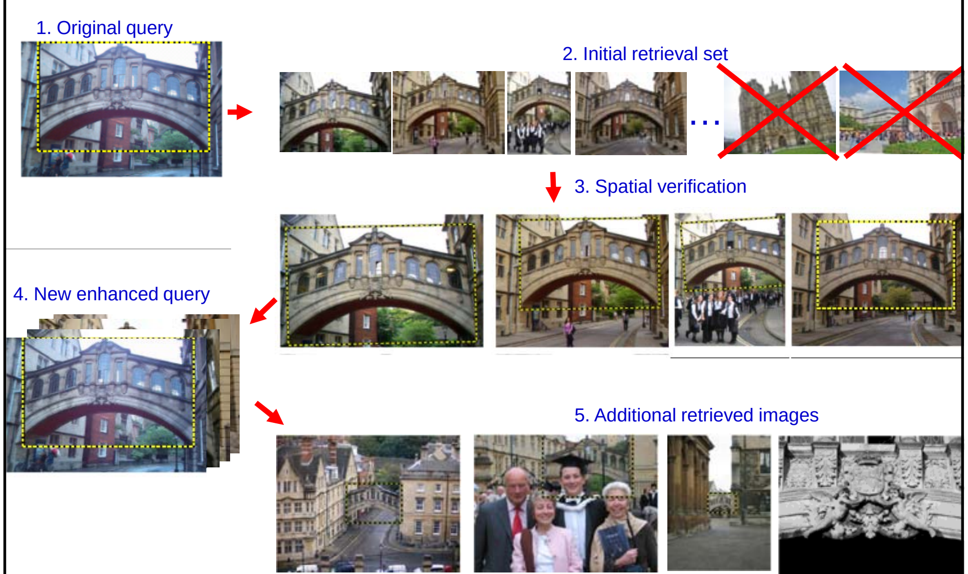
Visual word representations of two images of the same object may differ (due to e.g. detection/quantization noise) resulting in missed returns

Initial returns may be used to add new relevant visual words to the query

Strong spatial model prevents ‘drift’ by discarding false positives

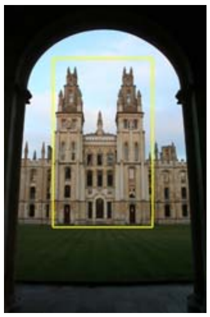
[Chum, Philbin, Sivic, Isard, Zisserman, ICCV’07]

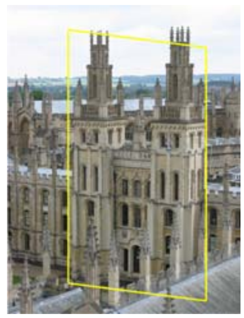
Visual query expansion - overview



Slide credit: J. Sivic

Query Expansion

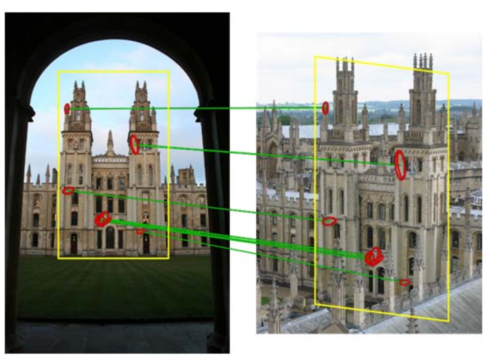

Query Image

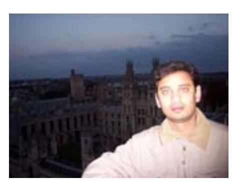

Originally retrieved image


Originally not retrieved

Slide credit: J. Sivic

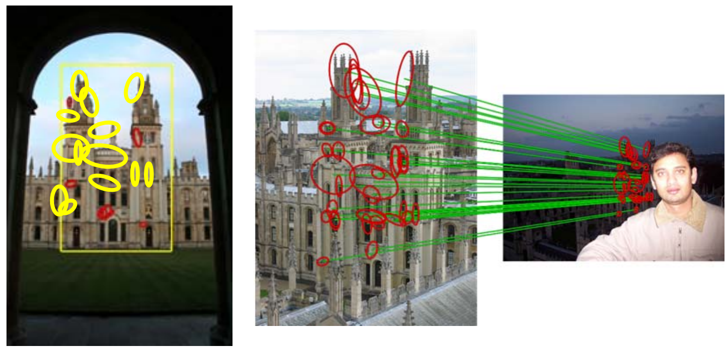
Query Expansion





Slide credit: J. Sivic

Query Expansion

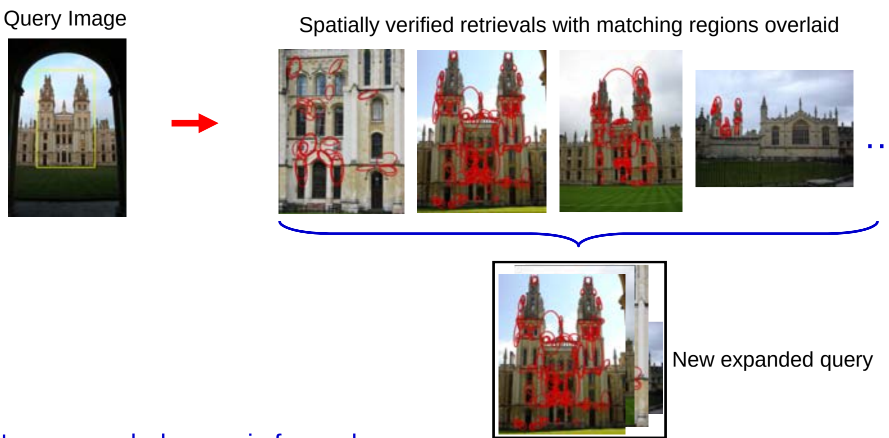


Slide credit: J. Sivic

Query Expansion

Slide credit: J. Sivic

Query Expansion

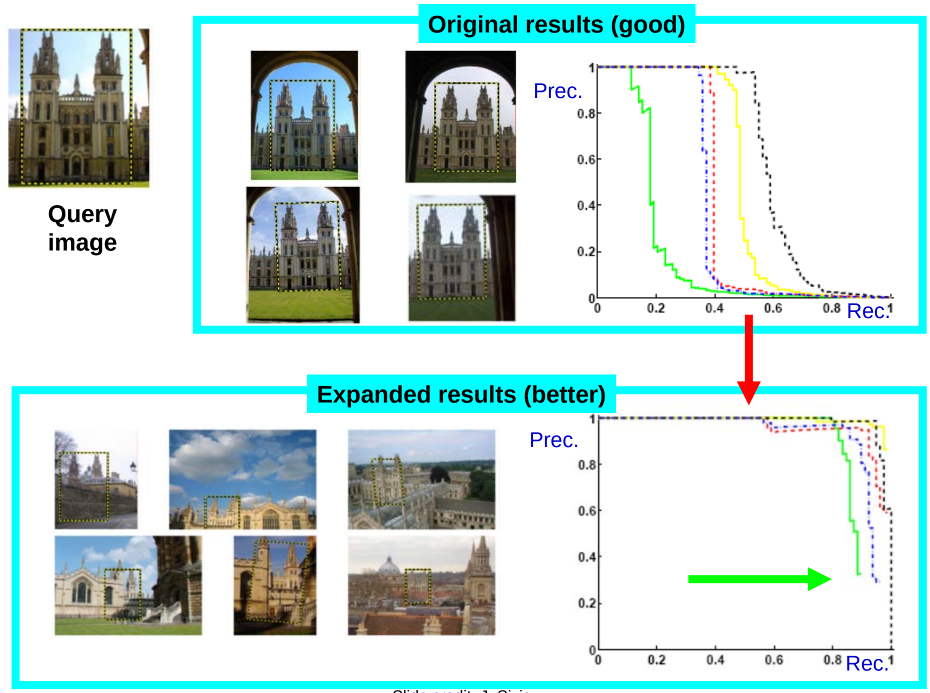


New expanded query is formed as

- the average of visual word vectors of spatially verified returns
- only inliers are considered
- regions are back-projected to the original query image

Query Expansion

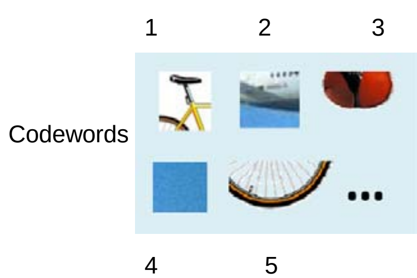
Slide credit: J. Sivic

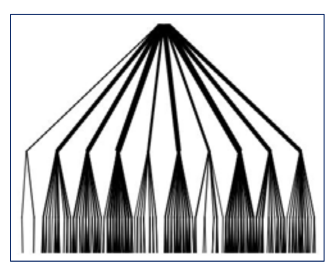


Slide credit: J. Sivic

Issues of Tree-based Indexing

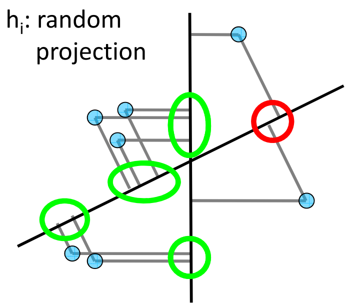
- Robust visual search requires large trees
  > 1 million nodes
  - Tree index is compact – 20 bits per feature
  - But difficult to store large trees at mobile clients
- Another problem:
  In high-dimensional space, back-tracing for NN search becomes a bottleneck
- Codeword index does not preserve proximity





Alternative: Hashing as Compact Code

- Locality Sensitive Hashing (LSH) [Indyk & Motwani 98]



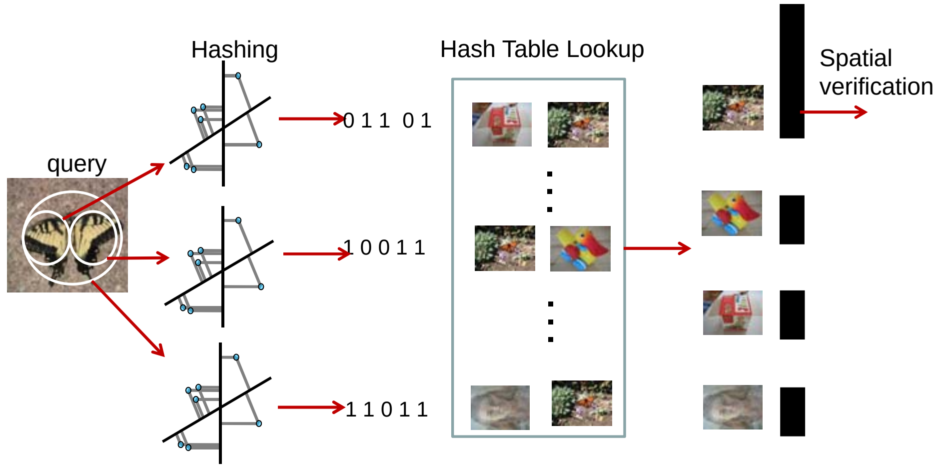
X	x_1	x_2	x_3	x_4	x_5
h_1	0	1	1	0	1
h_2	1	0	1	0	1
…	…	…	…	…	…
h_k	…	…	…	…	…
	010…	100…	111…	001…	110…

- Solve approximate NN problem with probabilistic guarantee

For any two points p, q:
– if |p-q| < r then Pr[h(p)=h(q)] > P_1
– if |p-q| > cr then Pr[h(p)=h(q)] < P_2

Figure: J. Sivic

Hash based image retrieval



64

Beyond LSH: Compact and Balanced

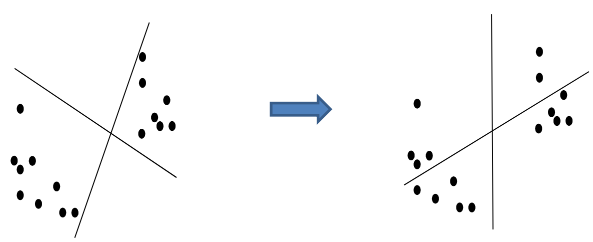
- Pure random projections lead to redundancy and imbalance
- SPICA Hash (He et al CVPR 11): jointly optimizes search accuracy & time

$$D(Y) = \sum_{p,q=1}^{N} W_{pq} \| Y_p - Y_q \|^2 \leq \eta$$

Locality sensitive

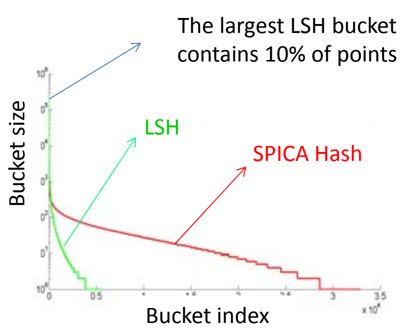
$$\min I(y_1, \ldots, y_k, \ldots, y_M)$$
$$\text{while } E(y) = \sum_{p=1}^{N} Y_p = 0$$

Balanced bucket size

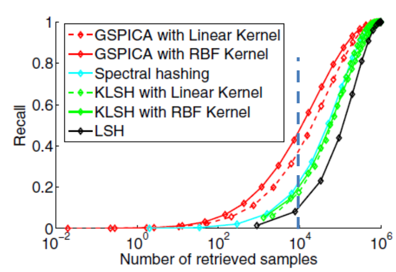


Beyond LSH: Compact and Balanced

- SPICA Hash: jointly optimize search accuracy & time
- Results



Reduce candidate set from 1M to 10K @ 0.5 recall



Beyond Randomness: Semi-Supervised Learned Projection

Wang, Kumar, and Chang, CVPR, ICML 2010

- Given Pair-Wise Relations

$$(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}: \text{ neighbor pair}$$
$$(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \text{ nonneighbor-pair}$$

- Measure empirical fit of hash bits

$$J(\mathbf{H}) = \sum_k \left\{ \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} h_k(\mathbf{x}_i)h_k(\mathbf{x}_j) - \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} h_k(\mathbf{x}_i)h_k(\mathbf{x}_j) \right\}$$

- Are the partitions balanced? Measure hash bit variance

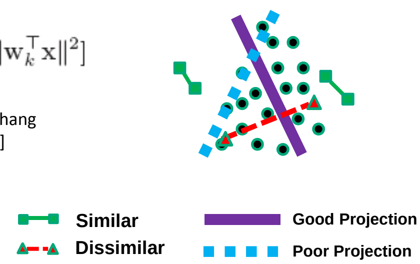
$$\sum_k E[\|h_k(\mathbf{x}) - \mu_k\|^2]$$

- Elegant eigen-decomposition solution

$$J(\mathbf{W}) = \frac{1}{2}\text{tr}\left[\mathbf{W}^\top \mathbf{X}_l \mathbf{S} \mathbf{X}_l^\top \mathbf{W}\right] + \frac{\eta}{2}\sum_k E[\|\mathbf{w}_k^\top \mathbf{x}\|^2]$$

$$S_{ij} = \begin{cases} 1 & : \ (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ -1 & : \ (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ 0 & : \ \textit{otherwise.} \end{cases}$$

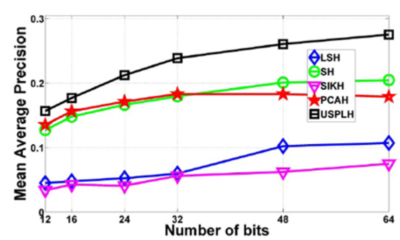
[Wang, Kumar, Chang CVPR10, ICML10]



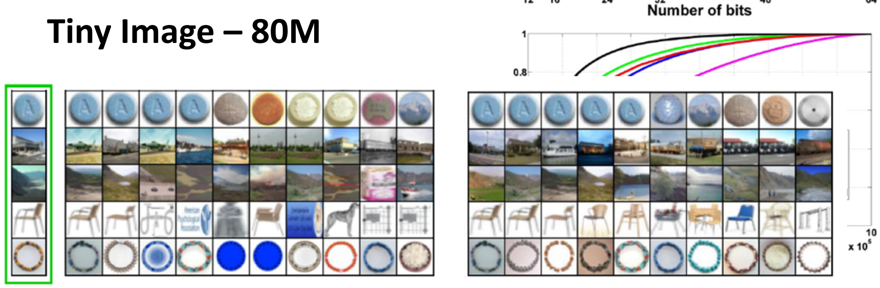
- Incremental learning via AdaBoosting

- **Learned projection hash increases accuracy > 2X**
- **Query time: a few seconds**
- **Compact code - 48 bits vs. 128 bytes per sample)**
- **Challenge: scale to billions**

SIFT-1M

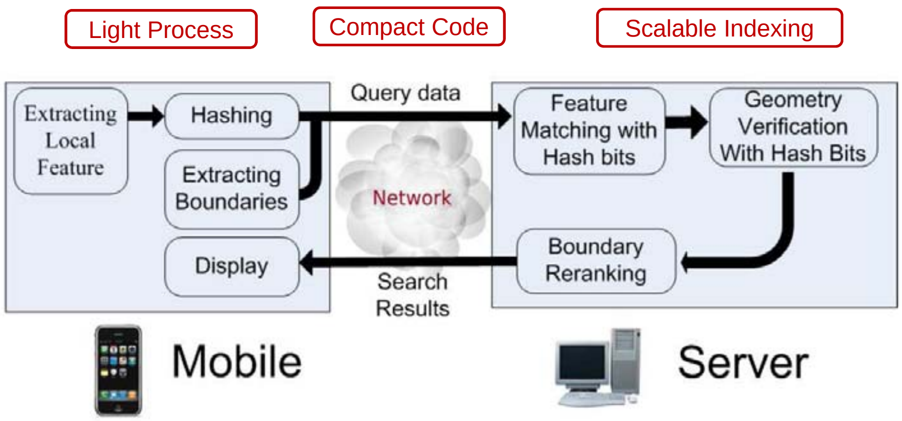


Tiny Image – 80M



Binary Reconstructive Embedding | **Learned Projection Hash**

Columbia Mobile Product Search using Bags of Hash Bits (BoHB)



69

Additional Feature: Boundary

- Use automatic salient object segmentation for every image in DB [Cheng et al, CVPR 2011]
- Boundary features:
  normalized central distance, Fourier magnitude
- Invariance: translation, scaling, rotation

70

Boundary Feature – Central Distance

- Sample N points p(n) with equal distance along boundary
- Compute the distance from every sample point p(n) to the boundary center c
- Normalized the distance vector by the maximal distance

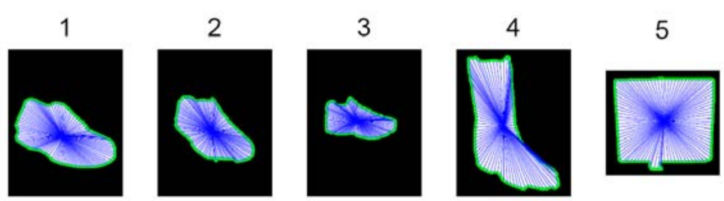
$$\mathbf{D}(n) = \frac{\|\mathbf{p}(n) - \mathbf{c}\|_2}{\max_n \|\mathbf{p}(n) - \mathbf{c}\|_2}$$

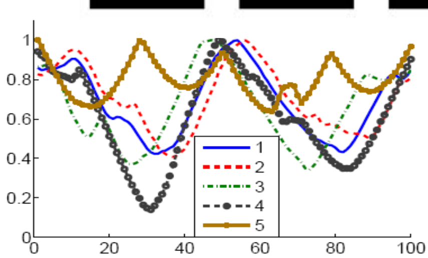
- Apply FFT on the distance vector, and take the magnitude part $\mathbf{F}(n) = |f[\mathbf{D}(n)]|$

- Invariance: translation, scale, rotation

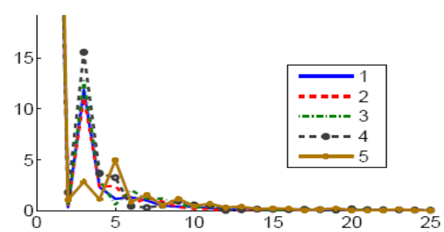
71

Boundary Feature – Central Distance





Distance to Center D(n)



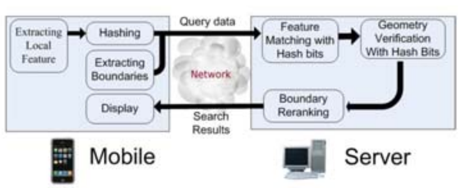
FFT: F(n)

72

Reranking with boundary feature

73

Columbia MPS System:
Bags of Hash Bits and Boundary features



Server:

- **400,000 product images crawled from Amazon, eBay and Zappos**
- **Hundreds of categories; shoes, clothes, electrical devices, groceries, kitchen supplies, movies, etc.**

Speed

- **Feature extraction: ~1s**
- **Transmission: 80 bits/feature, 1KB/im**
- **Serer Search: ~0.4s**
- **Download/display: 1-2s**

video demo (50")

Multi-View Challenge

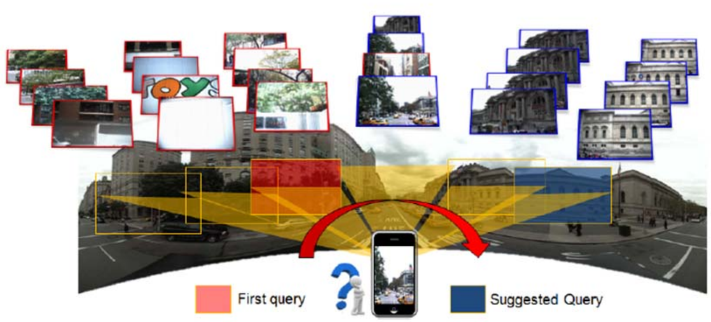
How to guide the user to take a successful mobile query?
- Which view will be the best query?
  - For example, in mobile location search:

  - Or in mobile product search:

75

Solution: Active Query Sensing

- Guide User to a More Successful Search Angle
  - Active Query Sensing [Yu, Ji, Zhang, and Chang, ACMMM '01]

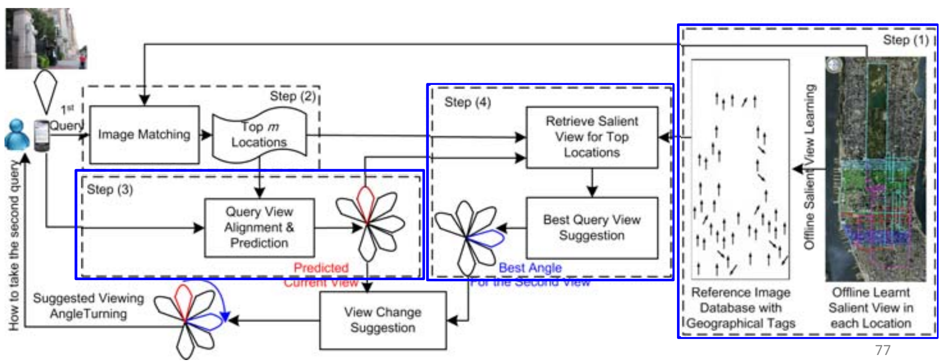


Video demo
Mobile App Demo

Active Query Sensing System

Basic Idea and Workflow

- Offline
  - Salient view learning for each reference location
- Online
  - Viewing angle prediction of the first query
  - Suggest new views by majority voting



Predict the view angle of query

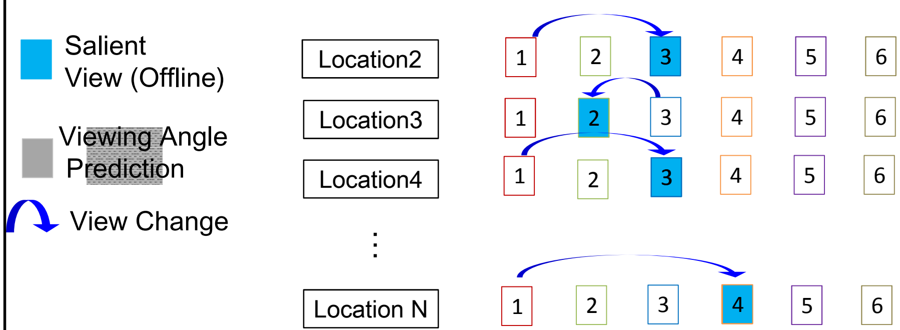
Offline Training: Train view prediction classifiers offline

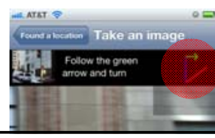
Online Prediction: View alignment based on the image matching

Our solution is to combine them both

Active Query Sensing

- Find the most reliable view angle and suggest angle change



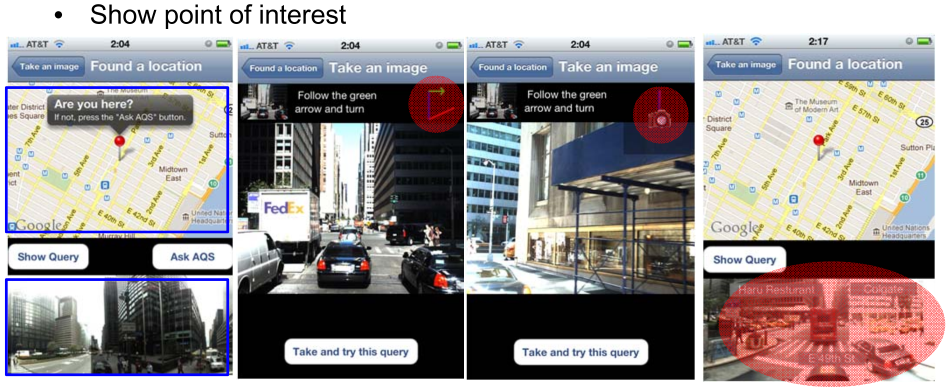


Turn 90 degrees to the right

79

User Interface

- Help user determine whether the first query is correct
  - Panorama
  - Geographical map context
- Guide the user to take the second query
  - Compass, camera icon
- Show point of interest



Mid-Term Project

- Possible Topics (not limited to these):
  - Review and test new features (local features, sketch, depth, etc.)
  - Review and test multi-feature fusion methods
  - Visual search of specific types of data
    - Patent diagrams, fashion, consumer, street view, nature, etc.
  - Visual summaries: panorama, photo poster, slide show
  - New user interfaces (search by sketch, gesture, voice)

Mobile Search Project

- Tools and systems on mobile devices
- Real-time feature extraction and tracking from images/videos, PTAM1, PTAM2
- Salient object detection and indexing (remember where the objects are UCLA project)
- Real-time object detection, text detection, OCR
- Combine GPS and local information

Mid-Term Project

- Focus on reviews in mid-term projects (testing welcome)
  - expand to the final project with new ideas and experiments
- OK to use data in HW#1 and #2
- Two persons in a team, but exceptions considered.
- 3/5: 1-page proposal due (summary, work plan, & references)
- 3/26: mid-term report and narrated slides (15 mins)
- 4/30: final project report due 4/30