IMAGE RETRIEVAL

Presenters: Bao Truong, Thuyen Phan, Dang Nguyen, Khoi Pham
Instructor: Tiep Nguyen

May 9, 2015

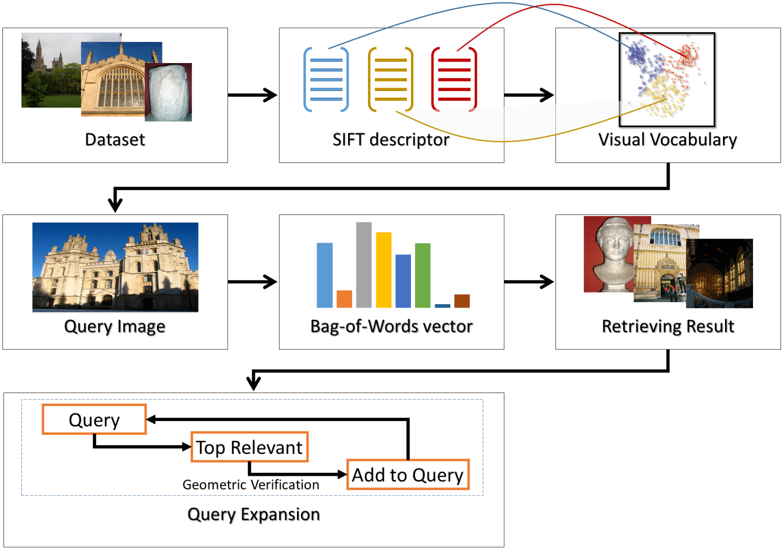
VNUHCM - University of Science - Faculty of Information Technology

OUTLINE

1. Overview
2. Index construction
    - Feature extraction
    - Codebook building
    - Quantization
3. Query
    - TF-IDF weighting
    - Query expansion (with geometric verification)
4. Experimental results
5. Conclusion

OVERVIEW

OVERVIEW


Dataset
SIFT descriptor
Visual Vocabulary
Query Image
Bag-of-Words vector
Retrieving Result
Query
Top Relevant
Add to Query
Geometric Verification
Query Expansion

INDEX CONSTRUCTION

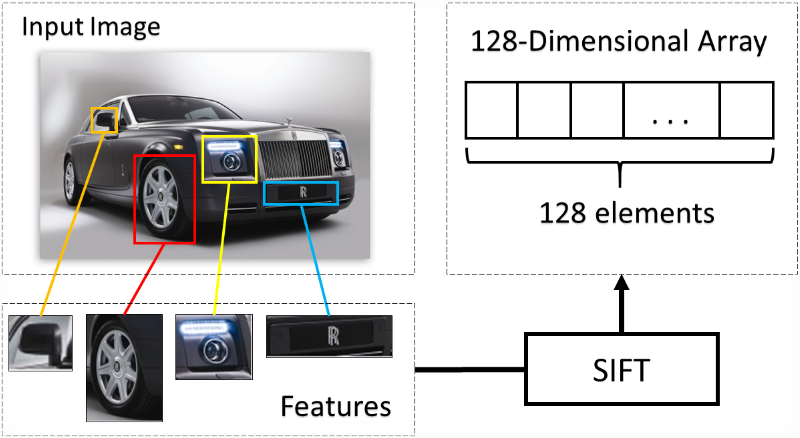
Apply ”compute descriptors” tool from University of Surrey[1] with the following parameters:

-hesaff -sift -noangle

hesaff: Scale and Affine invariant interest point detector
sift: use Scale Invariant Feature Transform (SIFT) descriptor
noangle: no angle estimation

[1] http://kahlan.eps.surrey.ac.uk/featurespace/web/desc (last access: May 2, 2015) .

FEATURE EXTRACTION



CODEBOOK BUILDING

Use **approximate k-Means** to cluster all features to **1M** clusters

- Use Fast Library for Approximate Nearest Neighbors (FLANN)[2]
- Philbin et al.[3] shows that **1M** dictionary size produce best performance on Oxford Building dataset[4]
- Run with **50** iterations

Each image is presented by a **1M-dimensional** vector

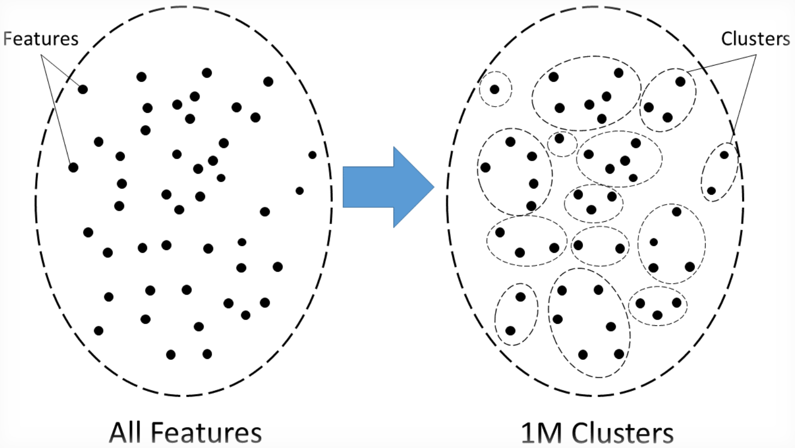
---

[2] http://www.cs.ubc.ca/research/flann (last access: May 2, 2015).

[3] J. Philbin, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases", in Proc. CVPR, 2008.

[4] http://www.robots.ox.ac.uk/vgg/data/oxbuildings (last access: May 2, 2015).

CODEBOOK BUILDING



QUANTIZATION (SOFT ASSIGNMENT)

Soft assignment: Each **128-dimensional** feature vector is reduced to **3-dimensional** by looking for its **3 nearest visual words**
Each of the **nearest visual words** is assigned with weight[5]:

$$\text{weight} = \exp(-\frac{d^2}{2\delta^2})$$

d = distance from feature vector to cluster centroid

$\delta^2 = 6250$

All weights are added to their corresponding visual word in the **1M-dimensional** representation of the image

[5] J. Philbin, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases", in Proc. CVPR, 2008.

QUANTIZATION (NATURAL SOFT ASSIGNMENT)

Use **natural soft assignment** to increase accuracy. 2 steps:

1. **Detection of repetitive structures**:
    - (x_i, s_i, d_i) is feature at location x_i, scale s_i, descriptor d_i
    - 2 features are connected if
        1. L2 distance $|x_i - x_j| < c(s_i + s_j)$
        2. ratio σ of 2 features is in $0.5 < \sigma < 1.5$
        3. 2 features share at least one common visual word

Find connected components of features. The detected components are called repttiles.

2. Weight calculation for each visual word in an image:

w_{id}: weight of visual word i in image d
k_f: number of nearest visual words we consider for feature f
V_f : set of indices of the k_f nearest visual words to feature f

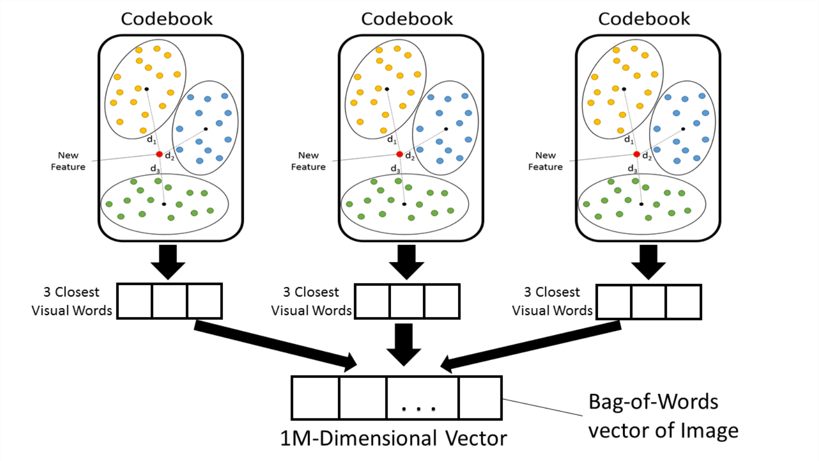
$$w_{id} = \sum_{f \in F_d} \sum_{k=1}^{k_f} 1[V_f(k) = i] \frac{1}{2^{k-1}}$$

where the indicator function $1[V_f(k) = i]$ equals to 1 if visual word i is at position k of V_f

$$k_f = \left\lceil k_{max} \frac{\log \frac{n_d+1}{m_f}}{\max_{f \in F_d} \log \frac{n_d+1}{m_f}} \right\rceil$$

where m_f is the number of features in the repttile of f

QUANTIZATION



QUERY

Similar with text retrieval

- **raw term frequency:** raw $\mathrm{tf}_{i,j}$ = weight of visual word i in image j
- **document frequency:** df_i = # of images that visual word i appears
- **raw inverse document frequency:** raw $\mathrm{idf}_i = |D|/\mathrm{df}_i$

Observation

- The **more time** a visual word occurs, the **less important** it is
- A visual word is **more discriminate** if it occurs in **fewer** images

Hence, it is necessary to **normalize** the **values of TF-IDF**

$$\mathrm{tf}_{i,j} = \frac{\text{raw tf}_{i,j}}{\sum_k \text{raw tf}_{k,j}} \text{ (for all visual words k in image j)}$$

$$\mathrm{idf}_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

Weight of visual word i in image j is therefore: $\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$

The tf-idf weight is used to **compute similarity** between an image d_i and a query q

$$s_{d_i,q} = \vec{\mathrm{tfidf}}_i \cdot \vec{\mathrm{tfidf}}_q = \sum_{j=1}^{|T|} \mathrm{tfidf}_{i,j} \times \mathrm{tfidf}_{q,j}$$

By **sorting** list of images based on their **similarity score** with a query, we achieve the **raw ranked list** which is used for the **Query Expansion** step

Apply **geometric verification** between the query image **Q** and each top-ranked image **A**:

1. (x, y) is a **matched pair** of features if $x \in Q$, $y \in A$, x and y are assigned to the same visual word
2. Randomly choose 4 pairs of features to build the **homography matrix**. A matched pair (x, y) is called **inliner** if apply the computed homography matrix on feature x produces feature y. Repeat 100 times to find the matrix that produces the **largest** number of inliners. These inliers are the **verified** visual words
3. **TF-IDF weight** of the **verified** visual words are added to query

Run this process for all top-ranked images. The added TF-IDF weight are averaged before running the query again

EXPERIMENTAL RESULTS

Table: mAP comparisons between different methods on Oxford 5K dataset

Method	mAP
BoW + soft assignment	0.676
Bow + nat. soft assignment	0.69814
Bow + nat. soft + spatial rerank	0.755482
Bow + nat. soft + query expansion	0.787189

CONCLUSION

Our proposed system **(Bow + nat. soft + query expansion)** achieves **mAP = 0.787** on the Oxford 5K dataset

Future works: port code from MATLAB to C++ to run the system on the **Oxford 100K** dataset

Questions?