# Search by Mobile Image Based on Visual and Spatial Consistency

Xianglong Liu, Yihua Lou, Wei Yu, Bo Lang

State Key Laboratory of Software Development Environment
Beihang University, P.R.China

July 2011

# Outline

# Search by Mobile Image

# Search by Mobile Image

## definition

- given a query image (mobile image) taken by the mobile phone, retrieve its most similar image in a large scale image database
- usually the most similar image is the original image taken photos of and associated with its relevant information.

# Search by Mobile Image

## definition

- given a query image (mobile image) taken by the mobile phone, retrieve its most similar image in a large scale image database
- usually the most similar image is the original image taken photos of and associated with its relevant information.

## application

- an easy and useful way to take photos of objects using mobile phones and then search the related information only by submitting these photos to the visual search engine.
- many applications: Google goggles, other applications like CD search and street search.


**Figure:** Some examples of mobile images


**Figure:** Some examples of mobile images

large image appearance variations caused by background clutter, foreground occlusion, and differences in viewpoint, orientation, scale and light conditions

# Related Works

# Related Works

## Basic approach

Most of image retrieval systems achieve efficiency by

- quantizing local features like Scale-Invariant Feature Transform (SIFT) into visual words
- applying scalable textual indexing and retrieval schemes

# Related Works

## Basic approach

Most of image retrieval systems achieve efficiency by

- quantizing local features like Scale-Invariant Feature Transform (SIFT) into visual words
- applying scalable textual indexing and retrieval schemes

## Issues

Two issues unsolved have critical effect on search by mobile images with large variations:

- the discriminative power of local features is limited due to quantization;
- the spatial relationship between features are not exploited enough.

# Related Works

# Related Works

## Solutions

- for issue (1), techniques like soft assignment and multitree scheme
- for issue (2), the geometric verification as an important post-processing step but computationally expensive

# Related Works

## Solutions

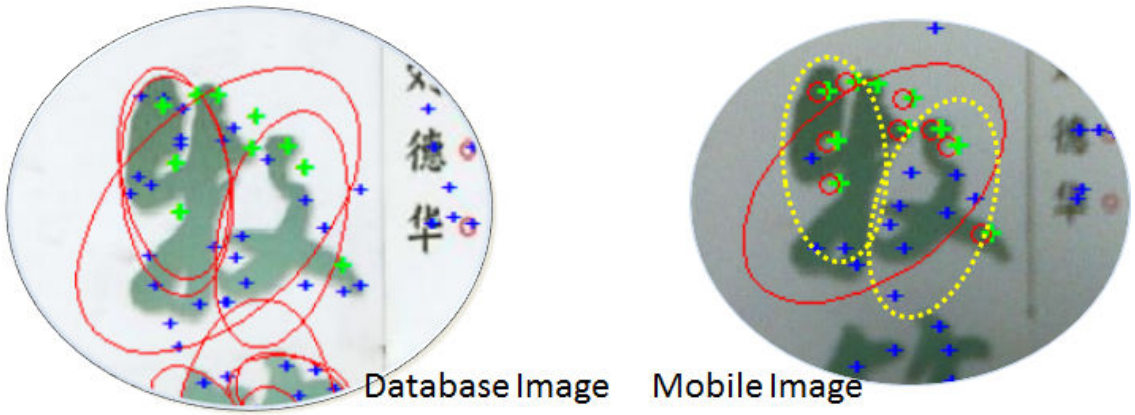
- for issue (1), techniques like soft assignment and multitree scheme
- for issue (2), the geometric verification as an important post-processing step but computationally expensive

## Problems

few work tried to solve both issues at the same time except bundled features

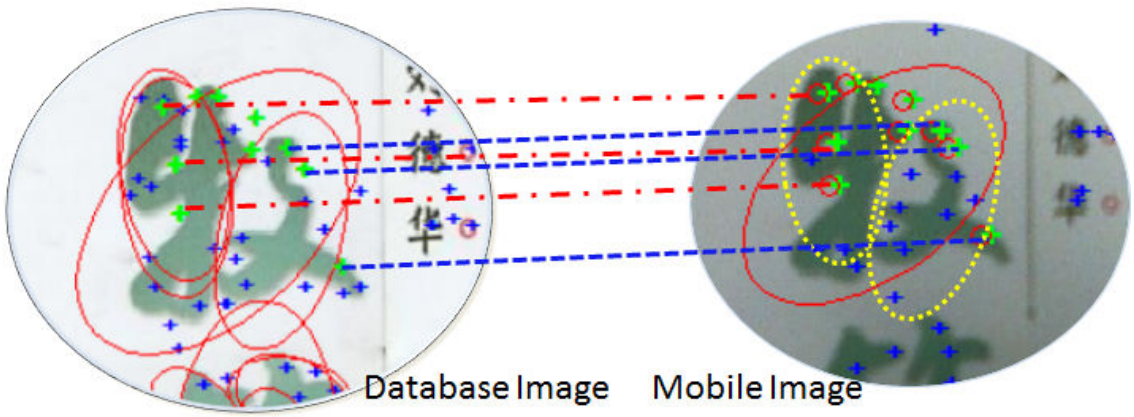
- match between all regions is time consuming and no rotation assumption is not suitable for mobile images
- for mobile images, due to the variations, some MSERs are not repeatable which degrades the match accuracy

# Observations




## Observations

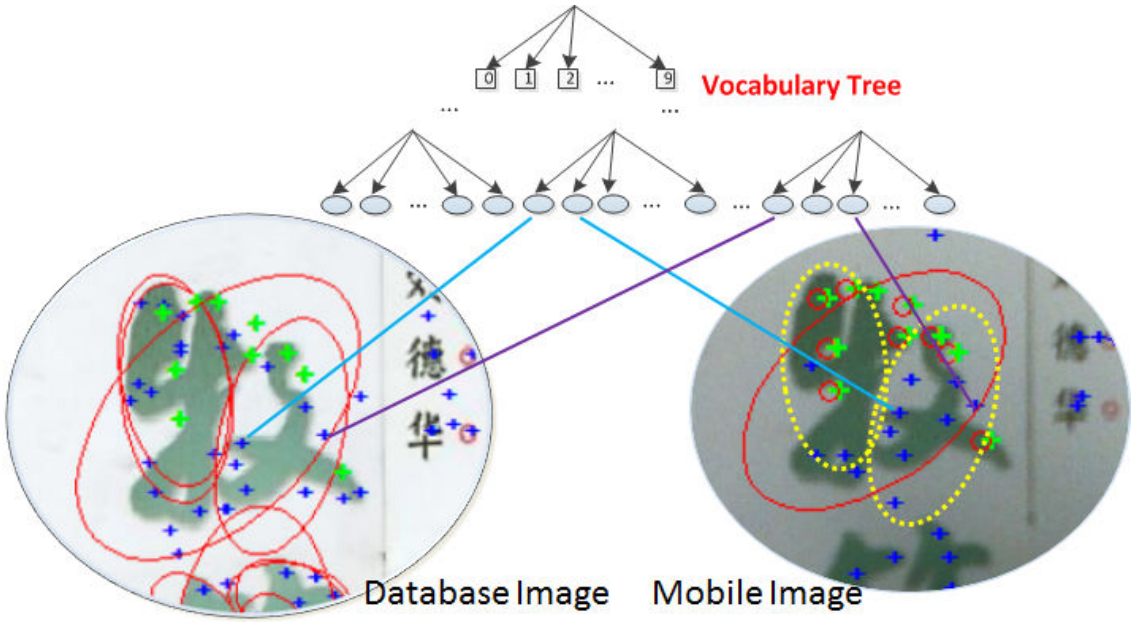
# Observations




## Observations

- usually corresponding regions of query image and original images have more common SIFT features matched

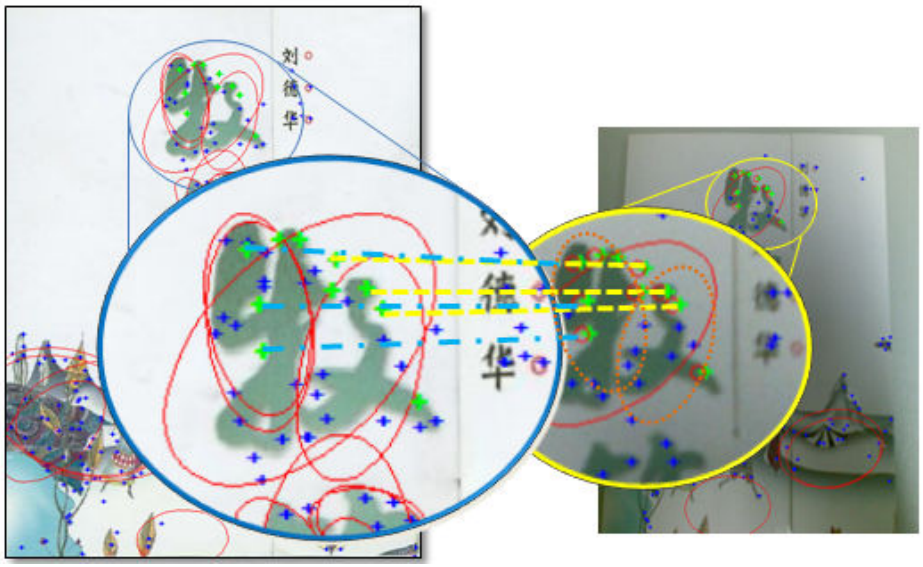
# Observations




## Observations

- usually corresponding regions of query image and original images have more common SIFT features matched
- in the corresponding regions which might be unrepeatable, there exist a number of corresponding features that quantized to neighbor words in vocabulary tree.

# Key Idea: Search Based on Visual and Spatial Consistency



## Key idea

Find the corresponding regions using the exactly matched features, then features in these regions can be grouped and softly matched

- discriminative ability: group local features using the exactly matched points and their geometric relationships
- quantization loss: soft match scheme for the grouped features

Visual and spatial information are coupled consistently.

# Visual Features

# Visual Features

## Point feature

The SIFT feature: invariant to image variations like scale and rotation:

- bag-of-words: quantizes SIFT features
- quantize them using the vocabulary tree built by hierarchical k-means clustering

# Visual Features

## Point feature

The SIFT feature: invariant to image variations like scale and rotation:

- bag-of-words: quantizes SIFT features
- quantize them using the vocabulary tree built by hierarchical k-means clustering

## Grouped feaure

Group features using region features like the Maximally Stable Extremal Region (MSER)

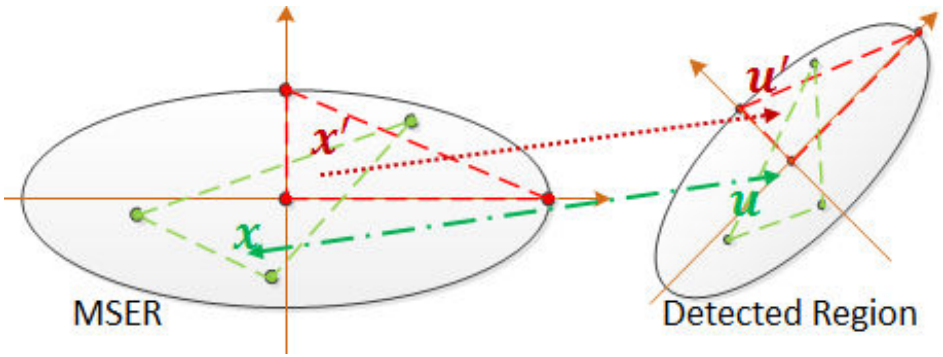
- database images: free of variations, MSERs can be extracted well and indexed.
- mobile query images: large variations, MSER detector usually fails and cannot be applied directly to grouping features.

# Feature group

Grouped features have been proven to be more discriminative than individual local features. The question is how to correctly group features in query image.

# Feature group

Grouped features have been proven to be more discriminative than individual local features. The question is how to correctly group features in query image.




Motivated by the observation: usually corresponding regions have more common SIFT features exactly matched, a straightforward way to detect the corresponding region:

- estimate the affine transformation $A$ from randomly selected 3 pairs of exactly matched points $(\mathbf{x}, \mathbf{u})$.
- detect the corresponding region by affine transformation $\mathbf{A}\mathbf{x}' = \mathbf{u}'$

# Notation

# Notation

- $S = \{p_i\}$: SIFT points reprensented by visual words

# Notation

- $S = \{p_i\}$: SIFT points reprensented by visual words
- $R = \{r_k\}$: Region features like MSER

# Notation

- $S = \{p_i\}$: SIFT points reprensented by visual words
- $R = \{r_k\}$: Region features like MSER
- $G_k = \{p_i | p_i \in r_k, p_i \in S\}$: feature group, $p_i \in r_k$ means that the point feature $p_i$ falls inside the region $r_k$.

# Notation

- $S = \{p_i\}$: SIFT points reprensented by visual words
- $R = \{r_k\}$: Region features like MSER
- $G_k = \{p_i | p_i \in r_k, p_i \in S\}$: feature group, $p_i \in r_k$ means that the point feature $p_i$ falls inside the region $r_k$.
- $G_k \rhd Q$: for each $G_k$, if its corresponding feature group $G'_k$ in query image $Q$ can be detected through the above process:
  - enlarge the ellipse of the MSER by factor 1.5 when computing $p_i \in r_k$
  - discard if it is empty or its ellipse spans more than half the width or height of the image
  - SIFT features that do not belong to any $G_k$ are treated to fall into the same region $r_0$ Here, $G_0$ unsatisfies $G_0 \rhd Q$ because it is not a MSER of image $I$.

# Soft Match

After corresponding regions are detected, features that fall into a same region are grouped, and soft match can be made between points in them to alleviate quantization loss:

# Soft Match

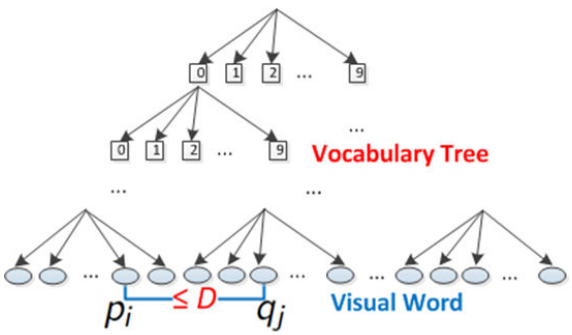
After corresponding regions are detected, features that fall into a same region are grouped, and soft match can be made between points in them to alleviate quantization loss:

- point features are quantized and represented by visual words $p_i$, $q_j$ in our visual vocabulary $W$

# Soft Match

After corresponding regions are detected, features that fall into a same region are grouped, and soft match can be made between points in them to alleviate quantization loss:

- point features are quantized and represented by visual words $p_i$, $q_j$ in our visual vocabulary $W$
- if $|p_i - q_j| \leq D$ which means they are close in $W$, then we call that they are neighbors and denote $\mathcal{N}_{p_i}^{D}(q_j)$:

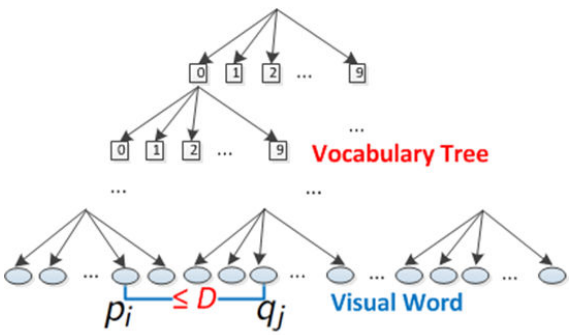



$$\mathcal{N}_{p_i}^{D}(q_j) = \begin{cases} 1, & \text{if } \|p_i - q_j\| \leq D; \\ 0, & \text{if } \|p_i - q_j\| > D \end{cases}$$

# Soft Match

After corresponding regions are detected, features that fall into a same region are grouped, and soft match can be made between points in them to alleviate quantization loss:

- point features are quantized and represented by visual words $p_i$, $q_j$ in our visual vocabulary $W$
- if $|p_i - q_j| \leq D$ which means they are close in $W$, then we call that they are neighbors and denote $\mathcal{N}_{p_i}^{D}(q_j)$:




$$\mathcal{N}_{p_i}^{D}(q_j) = \begin{cases} 1, & \text{if } \|p_i - q_j\| \leq D; \\ 0, & \text{if } \|p_i - q_j\| > D \end{cases}$$

- exponential weighting function gives higher importance to soft match of closer points:

$$\omega_{p_i,q_j}^{D} = \mathcal{N}_{p_i}^{D}(q_j) e^{-\|p_i - q_j\|}$$

# Score

In retrieval, the TF-IDF score is used to measure the similarity:

# Score

In retrieval, the TF-IDF score is used to measure the similarity:

- for visual word $p_i$ in the feature group $G_k$ of database image $I$
  - if $G_k \rhd Q$, then visual words of $G_k$ and corresponding $G'_k$ detected in $Q$ can be both exactly and softly matched;
  - Otherwise, only exact match is operated on words of $G_k$ and $Q$.

# Score

In retrieval, the TF-IDF score is used to measure the similarity:

- for visual word $p_i$ in the feature group $G_k$ of database image $I$
  - if $G_k \rhd Q$, then visual words of $G_k$ and corresponding $G'_k$ detected in $Q$ can be both exactly and softly matched;
  - Otherwise, only exact match is operated on words of $G_k$ and $Q$.
- matching score $M_Q(G_k)$ for feature group $G_k$:

$$M_Q(G_k) = \begin{cases} \lambda_{G_k, G'_k} \sum\limits_{p_i \in G_k, q_j \in G'_k} \omega^D_{p_i, q_j} v_{p_i} v_{q_j}, & \text{if } G_k \rhd Q; \\ \sum\limits_{p_i \in G_k, q_j \in Q} \omega^0_{p_i, q_j} v_{p_i} v_{q_j}, & \text{otherwise} \end{cases}$$

# Score

In retrieval, the TF-IDF score is used to measure the similarity:

- for visual word $p_i$ in the feature group $G_k$ of database image $I$
  - if $G_k \rhd Q$, then visual words of $G_k$ and corresponding $G'_k$ detected in $Q$ can be both exactly and softly matched;
  - Otherwise, only exact match is operated on words of $G_k$ and $Q$.
- matching score $M_Q(G_k)$ for feature group $G_k$:

$$M_Q(G_k) = \begin{cases} \lambda_{G_k,G'_k} \sum\limits_{p_i \in G_k, q_j \in G'_k} \omega^D_{p_i,q_j} v_{p_i} v_{q_j}, & \text{if } G_k \rhd Q; \\ \sum\limits_{p_i \in G_k, q_j \in Q} \omega^0_{p_i,q_j} v_{p_i} v_{q_j}, & \text{otherwise} \end{cases}$$

- more common visual words means higher confidence and higher score:

$$\lambda_{G_k,G'_k} = \ln \sum_{p_i \in G_k, q_j \in G'_k} \mathcal{N}^0_{p_i}(q_j).$$

# Score

# Score

- a database image $I$ is scored $S$ for the query image $Q$:

$$S_Q(I) = \sum_{G_k} M_Q(G_k).$$

# Score

- a database image $I$ is scored $S$ for the query image $Q$:

$$S_Q(I) = \sum_{G_k} M_Q(G_k).$$

- to compute the score efficiently, rewrite it:

$$S_Q(I) = \sum_{p_i \in G_k, q_j \in Q} \mathcal{N}^0_{p_i}(q_j) v_{p_i} v_{q_j} - S^{\mathcal{W}}_Q + S^{\mathcal{N}}_Q,$$

where:

$$S^{\mathcal{W}}_Q = \sum_{G_k \rhd Q} \sum_{\substack{p_i \in G_k \\ q_j \notin G'_k}} \mathcal{N}^0_{p_i}(q_j) v_{p_i} v_{q_j}$$

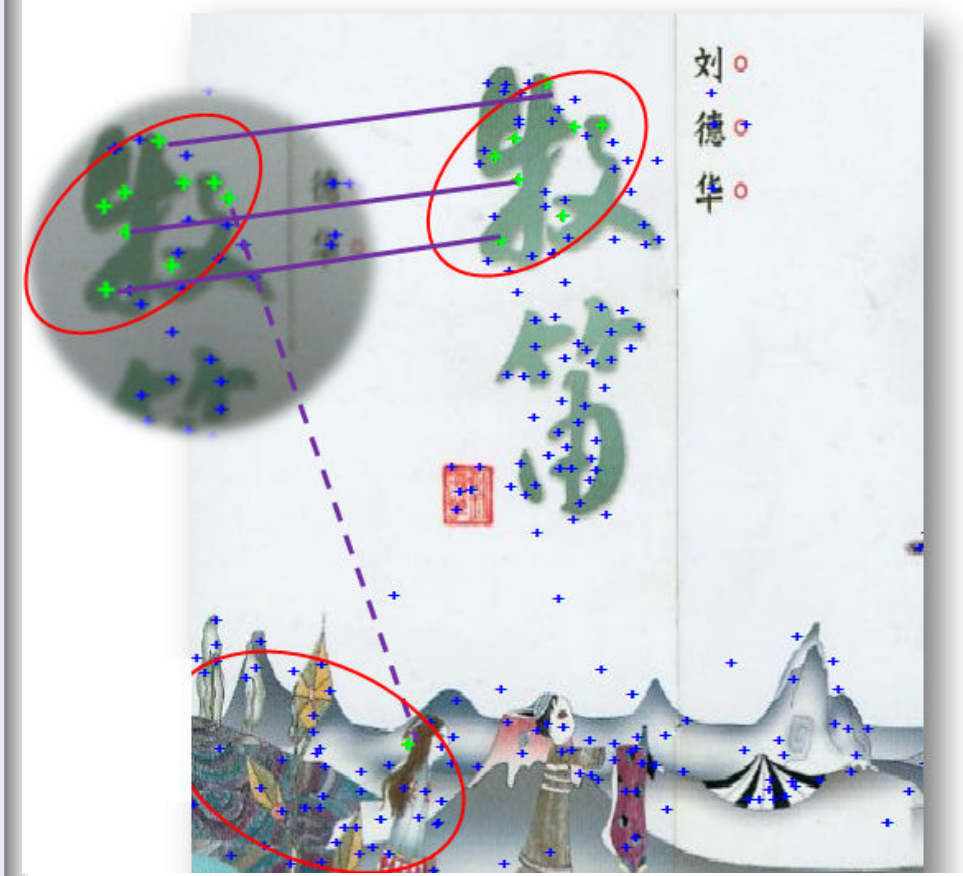
$$S^{\mathcal{N}}_Q = \sum_{G_k \rhd Q} \sum_{\substack{p_i \in G_k \\ q_j \in G'_k}} (\lambda_{G_k, G'_k} - \frac{\mathcal{N}^0_{p_i}(q_j)}{N_{p_i}}) \omega^D_{p_i, q_j} v_{p_i} v_{q_j}.$$

# Score

$$S_Q(I) = \sum_{p_i \in G_k, q_j \in Q} \mathcal{N}^0_{p_i}(q_j) v_{p_i} v_{q_j} - S^{\mathcal{W}}_Q + S^{\mathcal{N}}_Q$$

$S_Q(I)$ contains three parts:

- part 1: standard TF-IDF score

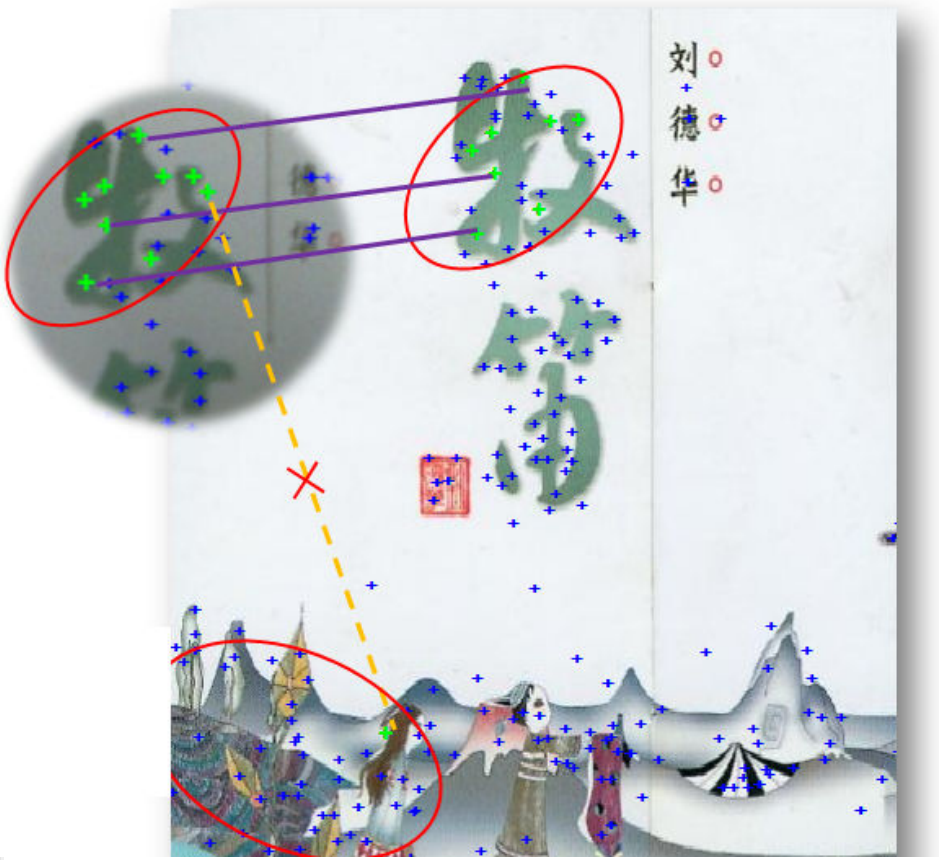


# Score

$$S_Q(I) = \sum_{p_i \in G_k, q_j \in Q} \mathcal{N}^0_{p_i}(q_j) v_{p_i} v_{q_j} - S^{\mathcal{W}}_Q + S^{\mathcal{N}}_Q$$

$S_Q(I)$ contains three parts:

- part 1: standard TF-IDF score
- part 2: score correction removing some wrong matches in $G_k$ ($G_k \triangleright Q$) by feature grouping.

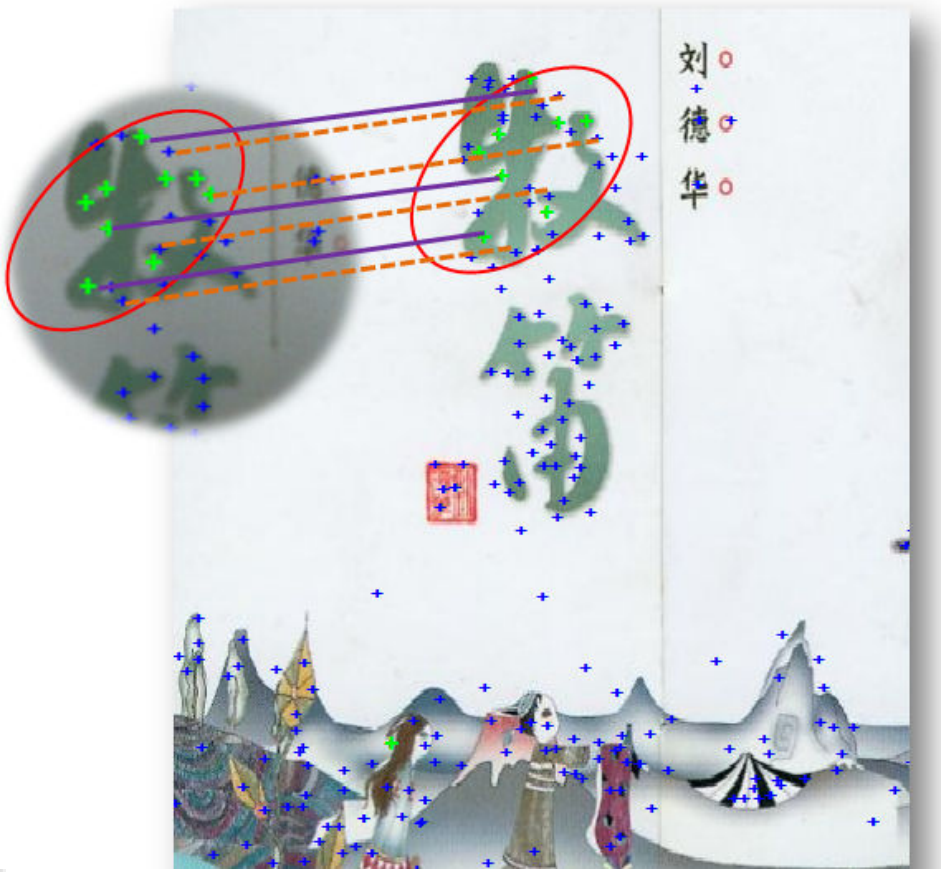


# Score

$$S_Q(I) = \sum_{p_i \in G_k, q_j \in Q} \mathcal{N}^0_{p_i}(q_j) v_{p_i} v_{q_j} - S^{\mathcal{W}}_Q + S^{\mathcal{N}}_Q$$

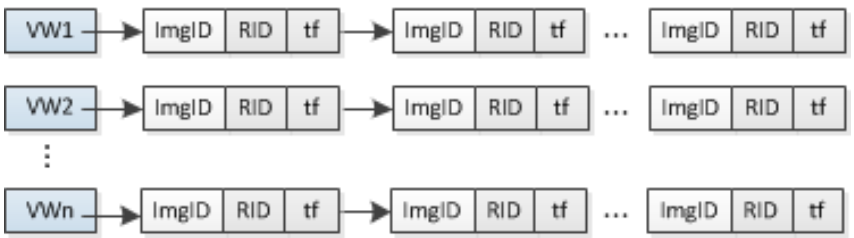
$S_Q(I)$ contains three parts:

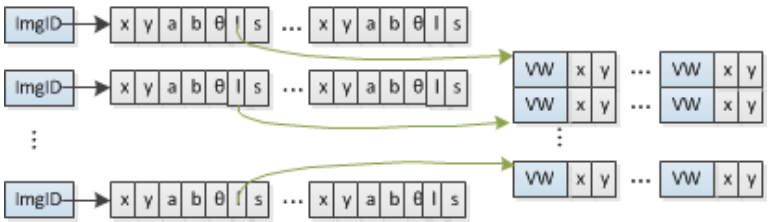
- part 1: standard TF-IDF score
- part 2: score correction removing some wrong matches in $G_k$ ($G_k \triangleright Q$) by feature grouping.
- part 3: additive score from soft match: $\lambda_{G_k, G'_k} > 1$ and $\frac{\mathcal{N}^0_{p_i}(q_j)}{N_{p_i}} \leq 1$, $S^{\mathcal{N}}_Q \geq 0$ which is very helpful for search by mobile images and especially partial ones.



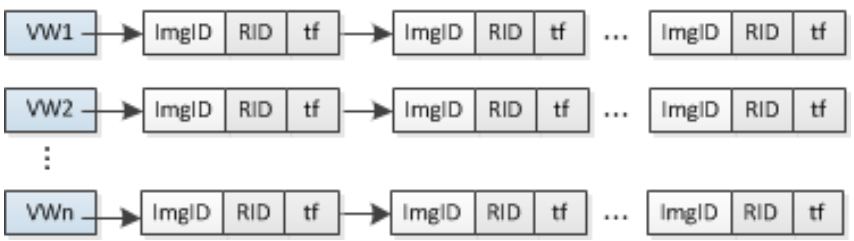
# Index

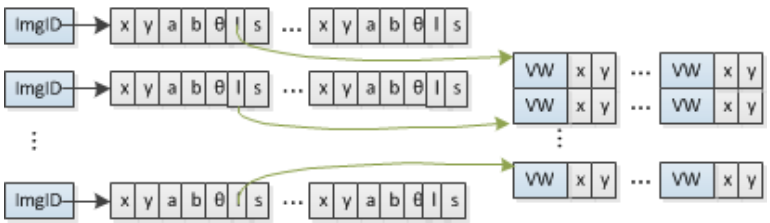
inverted file index for large-scale indexing and retrieval



# Index

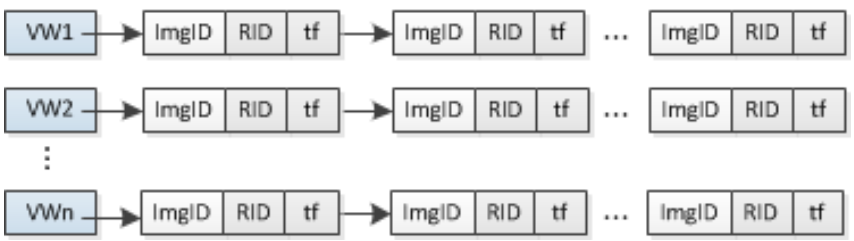


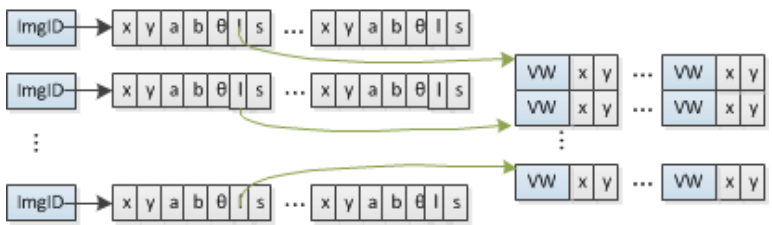




inverted file index for large-scale indexing and retrieval

- each visual word has an list containing images and MSERs where it appears and its positions.

# Index

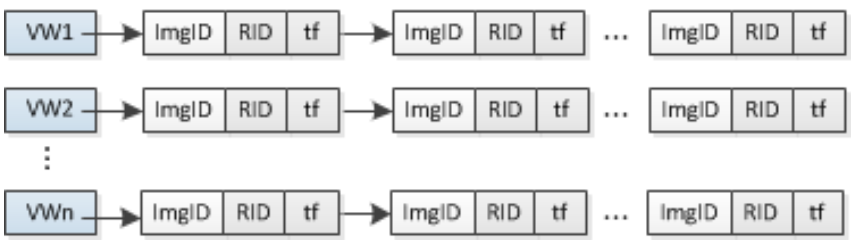


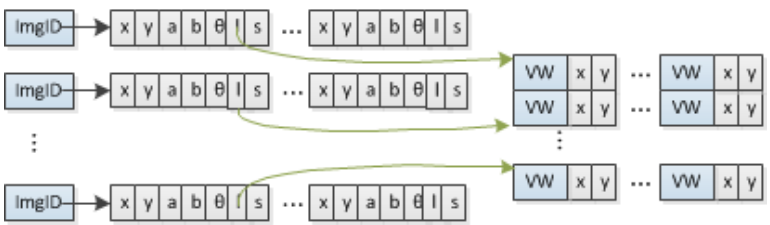




inverted file index for large-scale indexing and retrieval

- each visual word has an list containing images and MSERs where it appears and its positions.
- another table records central point coordinates, lengths of major and minor axis, and angles for MSERs of each database image.

# Index




inverted file index for large-scale indexing and retrieval

- each visual word has an list containing images and MSERs where it appears and its positions.
- another table records central point coordinates, lengths of major and minor axis, and angles for MSERs of each database image.
- retrieval by traversing the inverted file index and scoring the database image through local soft match

# Data

# Data

## a reference database

Recognition Benchmark Images: full data set of UKBench

# Data

## a reference database

Recognition Benchmark Images: full data set of UKBench

## a collected database

# Data

## a reference database

Recognition Benchmark Images: full data set of UKBench

## a collected database

- basic data set: one million images including posters and CD covers of 4000 most popular singers in Google Music

# Data

## a reference database

Recognition Benchmark Images: full data set of UKBench

## a collected database

- basic data set: one million images including posters and CD covers of 4000 most popular singers in Google Music
- manually take photos of sampled CD covers using camera phones (CHT9000 and Nokia 2700c, 2.0 Mega pixels cameras) with background cluttering, foreground blocking, and different light conditions, viewpoints, scale and rotations

# Data

## a reference database

Recognition Benchmark Images: full data set of UKBench

## a collected database

- basic data set: one million images including posters and CD covers of 4000 most popular singers in Google Music
- manually take photos of sampled CD covers using camera phones (CHT9000 and Nokia 2700c, 2.0 Mega pixels cameras) with background cluttering, foreground blocking, and different light conditions, viewpoints, scale and rotations
- 100 representative mobile images selected and labeled as our queries

# Data

## a reference database

Recognition Benchmark Images: full data set of UKBench

## a collected database

- basic data set: one million images including posters and CD covers of 4000 most popular singers in Google Music
- manually take photos of sampled CD covers using camera phones (CHT9000 and Nokia 2700c, 2.0 Mega pixels cameras) with background cluttering, foreground blocking, and different light conditions, viewpoints, scale and rotations
- 100 representative mobile images selected and labeled as our queries
- build three smaller data sets: 5K, 30K, and 100K by sampling the basic data set (http://www.nlsde.buaa.edu.cn/xlliu/icme2011).


**Figure:** Data set examples: database and mobile images


**Figure:** Data set examples: database and mobile images

## Measurement

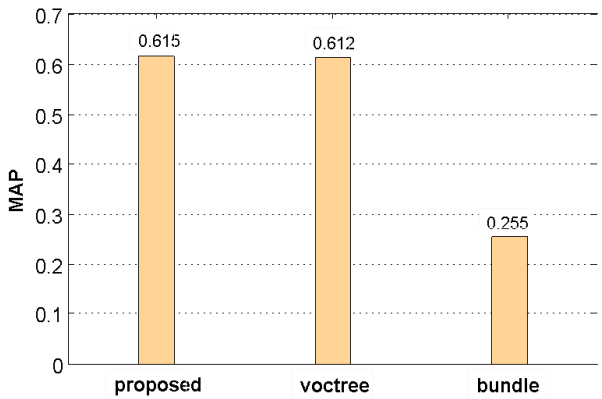
- mean reciprocal rank (MRR) $MRR = \frac{1}{n}\sum_i \frac{1}{rank_i}$: concern whether the original database image is ranked on the top
- compared with Nister et al. 2006 ("voctree") and Wu et al. 2009 ("bundle")
- all methods are running with and without geometric reranking
- use a vocabulary of 1M visual words following "voctree"
- the maximum neighbor distance $D$ is set to 10

# UKBench Experiments

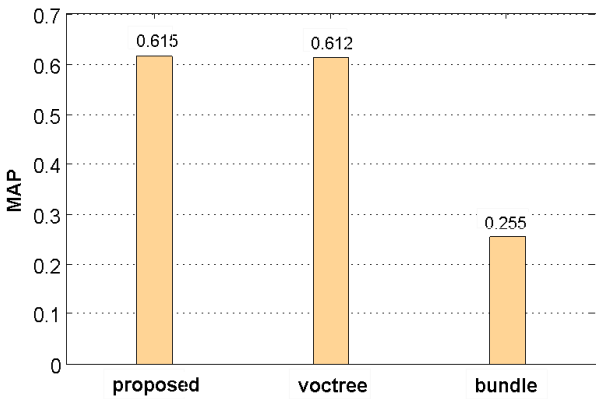
- Experiments
  - images: typical variations of viewpoint and orientation, other slight noises (light, occlusion, background, etc.)
  - query images: sampled from the data set.
  - mean Average Precision (mAP)

# UKBench Experiments

- Experiments
  - images: typical variations of viewpoint and orientation, other slight noises (light, occlusion, background, etc.)
  - query images: sampled from the data set.
  - mean Average Precision (mAP)
- Results
  - close to performance of "voctree" (over 60%)
  - outperform "bundle" significantly

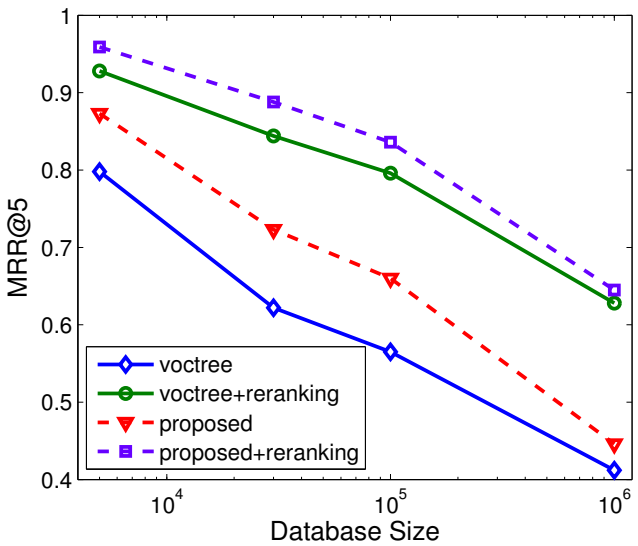


# UKBench Experiments



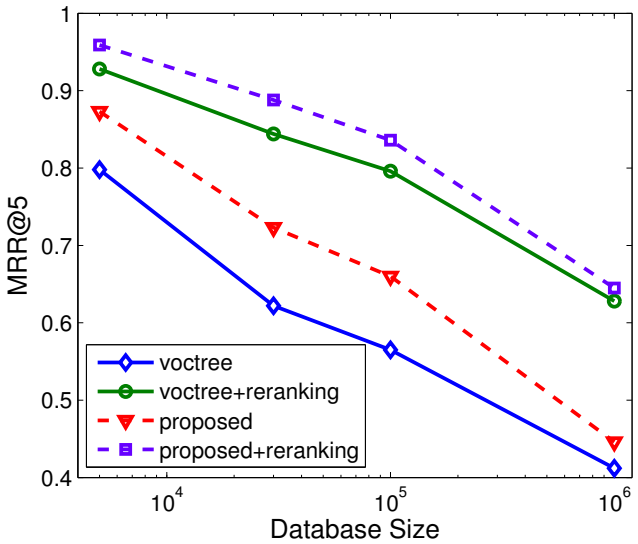

- Experiments
  - images: typical variations of viewpoint and orientation, other slight noises (light, occlusion, background, etc.)
  - query images: sampled from the data set.
  - mean Average Precision (mAP)
- Results
  - close to performance of "voctree" (over 60%)
  - outperform "bundle" significantly
- Discussion (Why "bundle" fails?)
  - some MSERs and SIFT points are not repeatable due to variations;
  - no rotation assumption, while usually the mobile images are rotated
  - SIFT points that fall into no MSER are discarded

# Collected Dataset Experiments




- mobile images photographed in real environments like CD shops
- without re-ranking, MRR of the proposed method is around 10% higher than that of "voctree"; after re-ranking for both methods, it still outperforms "voctree"
- both methods outperform "bundle" significantly (MRRs of 'bundle' are lower than 20%)

# Collected Dataset Experiments




- mobile images photographed in real environments like CD shops
- without re-ranking, MRR of the proposed method is around 10% higher than that of "voctree"; after re-ranking for both methods, it still outperforms "voctree"
- both methods outperform "bundle" significantly (MRRs of 'bundle' are lower than 20%)

**Figure:** Corresponding regions: (a) original image; (b) bundled; (c) proposed

**Table:** MRR performance of different values of $D$

| $D$ | 0 | 1 | 3 | 5 | 10 | 15 |
|---|---|---|---|---|---|---|
| 5K | 0.836 | 0.854 | 0.862 | 0.873 | **0.877** | 0.854 |
| 30K | 0.712 | 0.718 | 0.727 | **0.746** | 0.731 | 0.709 |
| 100K | 0.632 | 0.641 | 0.651 | 0.652 | **0.665** | 0.645 |
| 1M | 0.459 | 0.462 | 0.450 | 0.435 | **0.463** | 0.431 |

## Impact of soft match

The most effective value of D is around 10.

**Table:** MRR performance of different values of $D$

| $D$ | 0 | 1 | 3 | 5 | 10 | 15 |
|---|---|---|---|---|---|---|
| 5K | 0.836 | 0.854 | 0.862 | 0.873 | **0.877** | 0.854 |
| 30K | 0.712 | 0.718 | 0.727 | **0.746** | 0.731 | 0.709 |
| 100K | 0.632 | 0.641 | 0.651 | 0.652 | **0.665** | 0.645 |
| 1M | 0.459 | 0.462 | 0.450 | 0.435 | **0.463** | 0.431 |

## Impact of soft match

The most effective value of D is around 10.

**Table:** Average query time

| | voctree | proposed |
|---|---|---|
| time | 0.87s | 1.34s |

## Runtime

It indicates that the proposed approach takes no much more query time but achieves higher retrieval accuracy than "voctree".

# Thanks to

Wei Su, Hao Su, Anonymous Reviewers, and YOU.




# Web and Mobile Application Demo