Mobile Visual Search Based on Histogram Matching and Zone Weight Learning

Chuang Zhu, Li Tao, Fan Yang, Tao Lu, Huizhu Jia and Xiaodong Xie
Peking University, No. 5 Yiheyuan Road, Beijing, China

{czhu, hzjia}@pku.edu.cn

Abstract. In this paper, we propose a novel image retrieval algorithm for mobile visual search. At first, a short visual codebook is generated based on the descriptor database to represent the statistical information of the dataset. Then, an accurate local descriptor similarity score is computed by merging the $tf-idf$ weighted histogram matching and the weighting strategy in compact descriptors for visual search (CDVS). At last, both the global descriptor matching score and the local descriptor similarity score are summed up to rerank the retrieval results according to the learned zone weights. The results show that the proposed approach outperforms the state-of-the-art image retrieval method in CDVS.

1. Introduction
Content-based image retrieval (CBIR) is always a hot topic to study these years [1]. With the steadily growing amounts of mobile devices, a new type of CBIR technique, mobile visual search [2], is attracting keen attention of researchers. In mobile visual search, there are several significant challenges: limited wireless network bandwidth, small mobile battery capacity, little memory space to store features and the requirement of feature interoperability [3]. To address these challenges, the ISO/IEC moving pictures experts group (MPEG) drafts the compact descriptors for visual search (CDVS) [4].

A typical CDVS application framework is shown in Figure 1. The local features are extracted and compressed to produce compact visual descriptors on the mobile devices and the retrieval is performed on the remote server using the received descriptors. In this paper, we focus on the image retrieval algorithm design.

![Figure 1. Mobile visual search architecture.](image-url)

Generally, for an image retrieval system, the user supplies a query image or a query object by selecting a region of the query image, and then the feature detection module extracts local features...
such as SIFT [5] and SURF [6]; the retrieval system returns a ranked list of images that contain the same object based on the Bag-of-Words (BoW) [7] signature or global descriptors such as fisher vectors (FV) [8] and vector of locally aggregated descriptors (VLAD) [9]. In this paper, the above word “feature” refers to the original uncompressed key point, and the word “descriptor” means the encoded local feature or the aggregated global feature. However, the BoW signature and global descriptors are generated based on the orderless local features, which leads to the disregarding of information about the spatial layout of the features. To further improve image retrieval performance, Philbin et al. [10] propose to add an efficient spatial verification to rerank the results returned from the BoW model. Similarly, in work [11] the authors first retrieve videos using the weighted frequency vector and then rerank them based on the spatial consistency measure. Although the above BoW based approaches can yield decent performance, the fact that mobile visual search generally requires low memory makes these approaches unsuitable because of their very large visual word tables [3]. Thus, the CDVS standard adopts the Scalable Compressed Fisher Vector (SCFV) [12] to conduct image retrieval, and then reranks the returned results by using geometric consistency check (GCC) which includes a ratio test and a fast geometric model estimation [3]. However, the statistical information of image database, such as “inverse document frequency” (idf), is ignored and the global similarity score is discarded in reranking stage.

In this paper, we propose a novel visual search algorithm. First, a short visual codebook is generated based on the database descriptors to represent the statistical information of the dataset. Then, an accurate local descriptor similarity score (LDSS) is computed by merging the “term frequency-inverse document frequency” (tf-idf) weighted histogram matching and the ratio based weighting strategy in CDVS. At last, both the global descriptor matching score (GMS) and the LDSS are summed up to rerank the retrieval results according to the learned zone weights.

The remainder of the paper is structured as follows. In Section 2, we overview the mobile visual search techniques. In Section 3, we present details of our proposed image retrieval algorithm. In Section 4, we discuss the performance comparisons and then in Section 5 we conclude our paper.

2. Mobile visual search overview
The CDVS standard defines the feature extraction process and 6 different query descriptor lengths (0.5KB, 1KB, 2KB, 4KB, 8KB and 16KB) to support different scenarios [3]. The standardized CDVS bitstream makes the interoperability of the descriptors from different devices possible, and CDVS standard provides a matching mechanism for descriptors with different coding rates. For a typical mobile visual search application, there are two stages to perform: CDVS bitstream extraction and image retrieval using CDVS bitstream.

2.1. CDVS Bitstream Extraction
Figure 2 outlines the work flow of the CDVS bitstream extraction, which includes seven building blocks: interest point detection, local feature selection, local feature description, local feature compression, local feature aggregation, local feature location compression and CDVS encoding.

![Figure 2. CDVS Bitstream extraction process.](image-url)
For a query image, the CDVS standard first adopts a low-degree polynomial (ALP) detector [3] to find the interest points by approximating the result of the Laplacian of Gaussian (LoG) filter. Secondly, a subset of local features is selected to meet the bandwidth limitation according to a statistically learned relevance measure, which indicates the priori probability of a feature from query image matching a feature of database image correctly. After local feature selection, each picked local feature is described as original 128-dimensional (1024 bits) SIFT vector [5]. Then, on one hand, CDVS standard adopts a SCFV model to aggregate the local features to build a global descriptor for the query image; on the other hand, CDVS adopts a transform coding scheme followed by a scalar quantization and entropy coding to compress the selected local SIFT features. Besides, the local feature location compression is performed to record the \( x \) and \( y \) location information, which will be used in the GCC step of image retrieval. At last, in CDVS encoding module, the global descriptor, the compressed local features and the coded locations are merged to produce the CDVS bitstream.

![CDVS Bitstream Diagram](image)

**Figure 3.** Image retrieval based on CDVS bitstream.

### 2.2. Image Retrieval Using CDVS Bitstream

The CDVS also specifies the image retrieval framework. With the same extraction flow above, a set of global and local descriptors of the database images are extracted, and then they are indexed to build a global database and a local database, respectively. With the received query image CDVS bitstream, image retrieval will be performed according to the framework in Figure 3.

First, the global and local descriptors are separated out from the query bitstream by the CDVS decoding module. Second, the query global descriptor is compared with each global descriptor in the global database, and based on the GMS a shortlist with the top ranked \( N \) images, such as 500, is returned. In the GMS comparison, the Hamming distance-based similarity score \( S_{GMS} \) is computed according to

\[
S_{GMS}(X,Y) = \frac{\sum_{i=0}^{511} b^X_i b^Y_i w_{Ha(u^X_i,u^Y_i)} (32 - 2 Ha(u^X_i,u^Y_i))}{\sqrt{\sum_{i=0}^{511} b^X_i} \sqrt{\sum_{i=0}^{511} b^Y_i}}
\]

where \( X \) and \( Y \) are the SCFVs of two images. In CDVS, SCFV is built based on a selected subset of Gaussian components (512 in total) from the Gaussian Mixture Model (GMM). In Equation (1), \( b^X_i = 1 \) if \( i \)th Gaussian is selected, if not then \( b^X_i = 0 \). \( Ha(u^X_i,u^Y_i) \) and \( w_{Ha(u^X_i,u^Y_i)} \) are the Hamming distance and the correlation weight of \( i \)th Gaussian between \( X \) and \( Y \), respectively. Third, an exhaustive pairwise comparison is performed to find all matched local descriptor pairs between the query image and each candidate image in the shortlist. Then, the ratio test and GCC are conducted to remove the wrong matched pairs, which are also called outliers. The left \( P \) inlier good match pairs are used to generate the matching score \( L \) between two images. At last, the reranked final image retrieval results are generated according to \( L \).
\[ L = \sum_{i=1}^{p} \cos\left(\frac{\pi \cdot \min_i}{s_{\text{min}_i}}\right) \]  \hspace{1cm} (2)

In Equation (2), \( \cos \) is the cosine transform function, and \( \min_i/s_{\text{min}_i} \) is the ratio between the distance of the closest neighbor and that of the second-closest neighbor in computing the \( i_{th} \) inlier matching pair [5].

3. Proposed image retrieval
In this section, we first present our proposed image retrieval architecture, and then show our designed LDSS computing scheme. Finally, we illustrate our zone weight learning (ZWL) based image reranking algorithm, which takes both GMS and LDSS into consideration.

3.1. Image Retrieval Architecture
The main image retrieval flow is the same with the typical CDVS application illustrated in Figure 3: first generate shortlist and then perform image reranking. As shown in Figure 4, the main differences between our proposed image retrieval architecture and the typical CDVS image retrieval system come from the following three aspects.

First: Local database statistical information is integrated into the image retrieval system. The local database is a collection of local descriptors, and we cluster them to create a \( l \)-size, such as 300, codebook \( C=(c_1,c_2,...,c_l) \) by using \( k \)-means algorithm. Based on the codebook, we can quantize and represent the local descriptors of database images as visual words and then compute the \( \text{idf} \) for each visual word. The \( \text{idf} \) is calculated according to Equation (3).

\[ \text{idf}_{c_i} = \log \frac{N}{N_i} \]  \hspace{1cm} (3)

where \( c_i \) (i from 1 to \( l \)) is the \( i_{th} \) visual word of the codebook, \( N \) is the number of database images and \( N_i \) is the number of images containing visual word \( c_i \). Then the \( \text{idf} \) weighting table is

\[ W_{\text{idf}} = (\text{idf}_{c_1}, \text{idf}_{c_2},...,\text{idf}_{c_l}) \]  \hspace{1cm} (4)

Both the codebook \( C \) and the generated weighting table \( W_{\text{idf}} \) will be used in the image reranking stage.

Second: More information is passed to the image reranking module, including the GMS, the pre-calculated \( \text{idf} \) weighting table and the local descriptor codebook. It is noteworthy that the GMS corresponding to each image in the shortlist is generated in the global matching stage.

Third: The adopted image reranking algorithm is different. Besides GCC, we add \( \text{tf-idf} \) weighted histogram matching to compute the LDSS. Moreover, we utilize both the GMS and the LDSS to rerank the shortlist based on the ZWL.

3.2. LDSS Computing Based on \( \text{tf-idf} \) Weighted Histogram Matching
LDSS represents the similarity between the query image local descriptor (QLD) and the reference image local descriptor (RLD) in the database. We use the following Equation (5) to compute LDSS.
\[ S_{LDSS} = bK + (1-b)L \]  

(5)

where \( K \) is the \textit{tf-idf} weighted histogram matching score (HMS), \( L \) is the reranking criteria used in original CDVS reference model as shown in Equation (2), \( b \) is a constant to balance \( K \) and \( L \). The key of computing LDSS is to find \( K \).

The LDSS computing flow is shown in Figure 5. We perform local descriptor matching for the QLD and a RLD first, and then conduct GCC to remove the wrongly matched pairs. The left inlier pairs are used to compute \( K \). Finally, we compute LDSS based on \( K \) and \( L \).

Let QLD and RLD are

\[
\begin{align*}
QLD & = (q_1, q_2, ..., q_m) \\
RLD & = (r_1, r_2, ..., r_n)
\end{align*}
\]

(6)

where \( q_i \) (\( i \) from 1 to \( m \)) and \( r_j \) (\( j \) from 1 to \( n \)) are two sets of local descriptors in CDVS standard corresponding to the query image and a database reference image, respectively. After GCC, we get \( h \) number of good matching pairs (the inliers) \( M \), as shown in Equation (7).

\[
M(Q_{LD}, R_{LD}) = \{(q_{i_1}, r_{j_1}), (q_{i_2}, r_{j_2}), ..., (q_{i_h}, r_{j_h})\}
\]

(7)

where \( q_{i_0} \) and \( r_{j_0} \) (\( o \) from 1 to \( h \)) are from \( Q_{QLD} \) and \( R_{RLD} \), respectively. Then, \( q_{i_0} \) and \( r_{j_0} \) are quantized to create two weighting histograms \( H(Q) \) and \( H(R) \), according to the codebook and \textit{idf} weighting table. \( H(Q) \) and \( H(R) \) are histograms with \( l \) bins, corresponding to the \( l \) visual words of the codebook. Based on Equation (4), we have

\[
\begin{align*}
H(Q) & = ((idf_{i_1})N_{q_1}, (idf_{i_2})N_{q_2}, ..., (idf_{i_l})N_{q_l}) \\
H(R) & = ((idf_{j_1})N_{r_1}, (idf_{j_2})N_{r_2}, ..., (idf_{j_l})N_{r_l})
\end{align*}
\]

(8)

where \( N_{q_i} \) and \( N_{r_i} \) (\( i \) from 1 to \( l \)) denote the counts of query and reference inlier points, contained in Equation (7), which fall into the \( i \)th bins of \( H(Q) \) and \( H(R) \), respectively.

We know that histogram intersection effectively counts the number of points in two sets that fall into the same bin, and we use this criteria to evaluate the matching degree of \( H(Q) \) and \( H(R) \). Then, we get the \textit{tf-idf} weighted histogram matching score \( K \)

\[
K(H(Q), H(R)) = \sum_{i=1}^{l} \text{min}(H(Q_i), H(R_i))
\]

(9)

where \textit{min} is the minimum function.

Combining Equation (2) and (9), we rewrite Equation (5) as
In this paper, we set $b=0.5$ throughout experiments to achieve good balance between histogram matching score $K$ and CDVS reranking criteria $L$.

3.3. Image Reranking Based on Zone Weight Learning

Generally, image retrieval first sorts the database images based on BoW [7] or global descriptor [12] and yields a shortlist with top ranked images. Then image reranking algorithm will be followed to refine the results based on just local descriptor matching. In this paper, we propose to perform image reranking by using both LDSS and GMS.

Our motivation is that if two images containing the same targets, they should look similar both from the whole and the part at the same time. In statistical text retrieval system [13], a document can be split into different zones, such as title, abstract and body. In each zone, the document is viewed as a sequence of terms. We usually conduct text retrieval by combining the information of different zones. For example, we often retrieve a document like this: find documents with “computer” in the title and “adam” in the author list and the phrase “system architecture” in the body. The matching scores of different zones are weighted and added together to sort the documents. Similarly, in this paper we treat the global information of an image as global zone (similar to the title or abstract zones in text retrieval) and local information as local zone (similar to the body zone in text retrieval). We take both global zone and local zone matching scores into consideration for image reranking.

We propose to use the following equation in image reranking.

$$S = \lambda S_{GMS} + (1-\lambda) S_{LDSS} \quad (11)$$

where $S_{GMS}$ is the GMS presented in Equation (1) and $S_{LDSS}$ is LDSS illustrated in Equation (10). In Equation (11), $\lambda$ and $(1-\lambda)$ are the weights for global zone and local zone, respectively. In the following, we will describe the proposed zone weight learning method to find $\lambda$.

Firstly, the image dataset is divided into a training set and a testing set, and the training set is used to learn zone weights. Then, for the training set, we match each image with the others to generate a series of image pairs, in which the image pairs containing the same targets are labelled as $\alpha$ and the others are labelled as $\beta$. At last, we select $P$ image pairs with label $\alpha$ and $N$ image pairs with label $\beta$ to train $\lambda$ as follows. We first normalize $S_{GMS}$ and $S_{LDSS}$ of the pairs to a range of $[0, 1]$ by using min-max normalization, as shown in Equation (12).

$$y = (x - \text{MIN}) / (\text{MAX} - \text{MIN}) \quad (12)$$

where $x$ is the input value and $y$ is the normalized output. $\text{MIN}$ and $\text{MAX}$ are the minimum and maximum of the input values. Then, let $P_{\alpha}$ and $P_{\beta}$ as

$$P_{\alpha} = \{(S'_{GMS, 1}, S'_{LDSS, 1}), ..., (S'_{GMS, i}, S'_{LDSS, i})\}$$

$$P_{\beta} = \{(S'_{GMS, 1}, S'_{LDSS, 1}), ..., (S'_{GMS, i}, S'_{LDSS, i})\}$$

where $S'_{GMS}$ and $S'_{LDSS}$ are the $i$th normalized GMS and LDSS of the image pairs. For explanation convenience, we randomly select 300 pairs with label $\alpha$ and 300 pairs with label $\beta$, and depict the corresponding reranking scores $S$ with different $\lambda$ values, as shown in Figure 6.

We can see that different $\lambda$ values generate different distributions of $S$. Then, the question becomes how to choose a proper $\lambda$ which generates an $S$ distribution that can be easily classified by a fixed threshold value. We propose to use Equation (14) to train $\lambda$. 

$$S = bK + (1-b)L$$

$$= b\sum_{i=1}^{n} \min(H(Q_i), H(R_i)) + (1-b)\sum_{i=1}^{n} \cos(\frac{\pi}{2} \min_{s_{\min}}) \quad (10)$$

$$\min_{s_{\min}}$$

$$\sum_{i=1}^{n} \min(H(Q_i), H(R_i)) + (1-b)\sum_{i=1}^{n} \cos(\frac{\pi}{2} \min)$$
Figure 6. The distribution of reranking score with different $\lambda$.

$$\hat{\lambda} = \arg\min_{\lambda} \frac{C_{\text{inner}}}{D_{\text{inter}}} \quad (14)$$

where $C_{\text{inner}}$ represents the sum of concentration degree in each class, $D_{\text{inter}}$ denotes the separation degree between classes $\alpha$ and $\beta$. We will formulate Equation (14) in detail in the following. Corresponding to $Pa_{\alpha}$ and $Pa_{\beta}$, we have

$$\begin{align*}
\{ S'_{\alpha}(\hat{\lambda}) &= \{ S'_{\alpha}(\lambda), ..., S'_{\alpha}(\lambda), ..., S'_{\alpha}(\lambda) \} \\
\{ S'_{\beta}(\hat{\lambda}) &= \{ S'_{\beta}(\lambda), ..., S'_{\beta}(\lambda), ..., S'_{\beta}(\lambda) \} \\
\end{align*} \quad (15)$$

with

$$\begin{align*}
\{ S'_{\alpha}(\lambda) &= \lambda S'_{\text{GMS},\alpha} + (1-\lambda) S'_{\text{LDS},\alpha} \\
\{ S'_{\beta}(\lambda) &= \lambda S'_{\text{GMS},\beta} + (1-\lambda) S'_{\text{LDS},\beta} \\
\end{align*} \quad (16)$$

where $i$ from 1 to $P$ and $j$ from 1 to $N$. Let $m_{\alpha}(\lambda)$ and $m_{\beta}(\lambda)$ are the expectations, and $\text{var}_{\alpha}(\lambda)$ and $\text{var}_{\beta}(\lambda)$ are the variances of $S'_{\alpha}(\lambda)$ and $S'_{\beta}(\lambda)$, respectively. Then we can get the expectation of the whole, including both $S'_{\alpha}(\lambda)$ and $S'_{\beta}(\lambda)$.

$$m'(\lambda) = R_{\alpha} m_{\alpha}(\lambda) + R_{\beta} m_{\beta}(\lambda) \quad (17)$$

where $R_{\alpha}=P/(P+N)$ and $R_{\beta}=N/(P+N)$. Finally, we yield $C_{\text{inner}}$ and $D_{\text{inter}}$ as

$$\begin{align*}
\{ C_{\text{inner}} &= R_{\alpha} (\text{var}_{\alpha}(\lambda)) + R_{\beta} (\text{var}_{\beta}(\lambda)) \\
D_{\text{inter}} &= R_{\alpha} (m_{\alpha}(\lambda) - m'(\lambda))^2 + R_{\beta} (m_{\beta}(\lambda) - m'(\lambda))^2 \\
\end{align*} \quad (18)$$

We substitute Equation (18) into Equation (14) producing

$$\hat{\lambda} = \arg\min_{\lambda} \frac{R_{\alpha} (\text{var}_{\alpha}(\lambda)) + R_{\beta} (\text{var}_{\beta}(\lambda))}{R_{\alpha} (m_{\alpha}(\lambda) - m'(\lambda))^2 + R_{\beta} (m_{\beta}(\lambda) - m'(\lambda))^2} \quad (19)$$

Our proposed zone weight learning method will train the best $\lambda$ by minimizing Equation (19). One simple solution for $\lambda$ is directly checking all the results according to Equation (19) by increasing $\lambda$ with a small step, such as 0.01, from 0 to 1. Once we obtain the best $\lambda$, we will rerank the shortlist according to

$$S'(\hat{\lambda}) = \hat{\lambda} S'_{\text{GMS}} + (1-\hat{\lambda}) S'_{\text{LDS}} \quad (20)$$

where $S'_{\text{GMS}}$ and $S'_{\text{LDS}}$ are the normalized values, and $\hat{\lambda}$ is the best $\lambda$.

4. Experiments
Dataset. We evaluate our proposed method on two benchmark datasets: INRIA Holidays [14] and University of Kentucky Benchmark (UKBench) [10]. INRIA Holidays dataset includes 1491 images, and 500 ground truth queries can be used for testing. UKBench dataset contains 2550 tagged ground truth groups, and each group contains 4 pictures of the same object with different views.

Performance evaluation criteria. The mean average precision (mAP) is used to evaluate search performance. The mAP is shown in Equation (21).

\[
\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{r=1}^{M_{i,\text{relevant}}^{r}} P(r)}{M_{i,\text{relevant}}^{r}}
\]

where \(i\) is the query image index and \(N\) is the number of total queries. \(M_{i,\text{relevant}}^{r}\) is the number of relevant images corresponding to the \(i\)th query, \(r\) is the relevant image index and \(P(r)\) is the precision at the cut-off rank of \(r\) relevant image.

Zone weight learning. For each selected dataset, we divide it into two parts. 100 images and 300 images are used in zone weight \(\lambda\) training for INRIA Holidays and UKBench, respectively, and the rest part is used for the visual search performance evaluation. In CDVS, there are six specified bitrate modes (from 0.5KB to 16KB), thus we learn six different weights for them. Figure 7 depicts the learned \(\lambda\) for the two datasets. With the increase of bitrate, more local features are encoded into the CDVS bitstream and thus the reliability of \(S_{LDSS}'\) becomes strong, which results in the decrease of \(\lambda\).

Comparisons. For mobile visual search, CDVS achieves the state-of-the-art performance. The image reranking method based on spatial verification in [10] and the image retrieval algorithm in CDVS are tested to make comparisons with our work. All the methods are integrated into CDVS reference software test model framework 11 (TM 11.0) [15]. Test results are tabulated in Table 1. The results show that our proposed method achieves the best searching performance among all 3 methods: it outperforms both the method in work [10] and the state-of-the-art method in CDVS reference model TM 11.0 among all bitrate modes. Our method can achieve maximum up to 3% gain in some specific bitrate modes, such as 1KB mode for UKBench, compared to the other algorithms.
Table 1. Performance comparisons.

| Dataset   | Bitrate (KB) | mAP  | work [10] | CDVS | Proposed |
|-----------|--------------|------|-----------|------|----------|
| INRIA Holidays | 0.5          | 59.39 | 59.42     | 59.77 |
|           | 1            | 59.38 | 60.93     | 61.29 |
|           | 2            | 64.52 | 66.60     | 67.18 |
|           | 4            | 66.18 | 68.62     | 69.03 |
|           | 8            | 69.20 | 70.43     | 70.55 |
|           | 16           | 68.95 | 70.19     | 70.31 |
| UKBench   | 0.5          | 76.13 | 76.24     | 76.74 |
|           | 1            | 76.11 | 77.45     | 79.36 |
|           | 2            | 80.76 | 82.12     | 82.74 |
|           | 4            | 82.47 | 83.56     | 84.73 |
|           | 8            | 84.28 | 84.50     | 84.99 |
|           | 16           | 84.28 | 84.48     | 84.97 |

5. Conclusion
In this paper, we propose an image retrieval algorithm for mobile visual search. First, a short visual codebook is generated based on the database descriptors and then an accurate LDSS is computed by merging the tf-idf weighted histogram matching and the weighting strategy in CDVS. At last, both the GMS and the LDSS are summed up to rerank the retrieval results according to the learned zone weights. The results show that the proposed approach outperforms the state-of-the-art performance of CDVS. Because most part of our proposed method can be pre-computed offline, the online image retrieval computing complexity is just increased slightly compared with CDVS. In the future, we will focus on low complexity image retrieval algorithm design.

6. References
[1] A referenceThis reference has two entries but the second one is not numbered (it uses the ‘Reference (no number)’ style. Singhai N, Shandilya S K. 2010. A survey on: content based image retrieval systems. International Journal of Computer Applications. 4(2), (2010), 22-26.
[2] Girod B, Chandrasekhar V, Chen D M, et al. 2011. Mobile visual search. IEEE signal processing magazine. 28(4), (2011), 61-76.
[3] Duan L Y, Chandrasekhar V, Chen J, et al. 2016. Overview of the MPEG-CDVS Standard. IEEE Transactions on Image Processing. 25(1), (2016), 179-194.
[4] S. Paschalakis et al. 2015. Information Technology-Multimedia Content Description Interface-Part 13: Compact Descriptors for Visual Search.(2015), ISO/IEC 15938-13.
[5] Lowe D G. 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision. (2004), 60(2), 91-110.
[6] Bay H, Ess A, Tuytelaars T, et al. 2008. Speeded-up robust features (SURF). Computer vision and image understanding. (2008), 110(3), 346-359.
[7] Zhang Y, Jin R, Zhou Z H. 2010. Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics.(2010), 1, 43-52.
[8] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. 2010. Large-scale image retrieval with compressed Fisher vectors. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), (2010), pp. 3384-3391. San Francisco.
[9] H. Jégou, M. Douze, C. Schmid, and P. Perez. 2010. Aggregating local descriptors into a compact image representation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), (2010), pp. 3304-3311. San Francisco.
[10] Philbin J, Chum O, Isard M, et al. Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition,
CVPR’07,(2007), pp: 1-8.

[11] Sivic J, Zisserman A. 2009. Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*. (2009), 31(4), 591-606.

[12] J. Lin, L.-Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao. 2014. Rate- adaptive compact Fisher codes for mobile visual search. *IEEE Signal Process. Lett.* (2014), 21(2), 195-198.

[13] Manning C D, Raghavan P, Schütze H. 2008. Introduction to information retrieval. Cambridge university press, (2008), Cambridge.

[14] Herve Jegou, Matthijs Douze and Cordelia Schmid. 2008. Hamming Embedding and Weak geometry consistency for large scale image search. In: *proceedings of the 10th European conference on Computer vision*. (2008).

[15] Test Model 11: Evaluation framework for Compact Descriptor for Visual Search, (2014), document ISO/IECJTC1/SC29/WG11/N14680.

Acknowledgments

This work was partially supported by grant from the China Postdoctoral Science Foundation under contract No. 2016M590020, the National Science Foundation of China (61421062 and 61602011), the National High Technology Research and Development Program of China (863 Program) under contract No.2015AA015903, the Major National Scientific Instrument and Equipment Development Project of China under contract No. 2013YQ030967, and the National Key Research and Development Program of China under contract No. 2016YFB0401904 and 2016YFB0402001.