PAPER

Highly Efficient Mobile Visual Search Algorithm

Chuang ZHU[†a)], **Xiao Feng HUANG**[††], ***Nonmembers***, **Guo Qing XIANG**[†††], ***Student Member***, **Hui Hui DONG**[†], ***and*** **Jia Wen SONG**[†††], ***Nonmembers***

SUMMARY In this paper, we propose a highly efficient mobile visual search algorithm. For descriptor extraction process, we propose a low complexity feature detection which utilizes the detected local key points of the coarse octaves to guide the scale space construction and feature detection in the fine octave. The Gaussian and Laplacian operations are skipped for the unimportant area, and thus the computing time is saved. Besides, feature selection is placed before orientation computing to further reduce the complexity of feature detection by pre-discarding some unimportant local points. For the image retrieval process, we design a high-performance reranking method, which merges both the global descriptor matching score and the local descriptor similarity score (LDSS). In the calculating of LDSS, the *tf-idf* weighted histogram matching is performed to integrate the statistical information of the database. The results show that the proposed highly efficient approach achieves comparable performance with the state-of-the-art for mobile visual search, while the descriptor extraction complexity is largely reduced.

key words: *mobile visual search, descriptor extraction, feature selection, reranking*

1. Introduction

Content-based image retrieval (CBIR) is always a hot topic to study these years [1]. Generally, the research on CBIR can be classified into two categories: hand-crafted feature based method [2] and deep learning based method [3]–[6]. The first category of CBIR designs specific local features, such as SIFT [7], and performs precise feature matching, geometric consistency check and reranking. The deep learning based algorithms conduct retrieval by applying the output of fully connected layers [3] or the max/sum polling of feature maps of CNN [4], [5] to represent the image.

Background. With the steadily growing amounts of mobile devices, a new type of CBIR technique, mobile visual search [8], 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 [9]. To partly address these challenges, the ISO/IEC moving pictures experts group (MPEG) drafts the compact descriptors for visual search (CDVS) [10]. To future support the large-scale video retrieval applications, the MPEG is carrying out the draft of Compact Descriptors for Video Analysis (CDVA) [13]. The recent work [14] shows that the combination of hand-crafted feature or descriptor, such as CDVS, and deep features can improve image retrieval performance dramatically. Conduct research on highly efficient hand-crafted feature/descriptor based visual search is still important and this paper focuses on the CDVS based mobile visual search.

A typical CDVS application framework is shown in Fig. 1, which is composed of mobile-end descriptor generation and server-end image retrieval. 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. 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.

Problem. In the mobile-end, feature detection module of CDVS adopts a low-degree polynomial (ALP) detector [10] to find the interest points by approximating the result of the Laplacian of Gaussian (LoG) filter. For mobile devices, the complexity of feature detection is still too high, in which the pyramid scale-space construction is the most complicated part. Koutaki et al. proposed a low-complexity scheme which uses 4-basis images to accurately reconstruct the Gaussian or sLoG within the scale range $s \in [1.0, 5.0]$ [11]. The authors further improved the filtering to XY-separable form by Gaussian lobes approximation [12]. Chen et al. [15] proposed a block-based Frequency Domain Laplacian of Gaussian (BFLoG) detector, which has been adopted by the MPEG CDVS standard, to alleviate the computational burden. However, the relationships between different octaves are not utilized to decrease the computing complexity in CDVS. It is meaningful to fur-

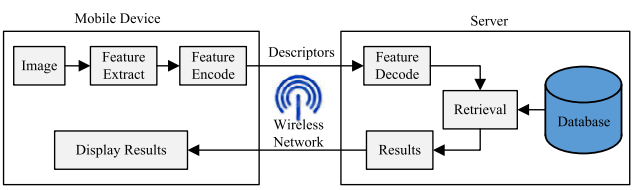

Fig. 1 Mobile device extracts features of the query image and transmits the compressed features (local descriptor and global descriptor) via wireless network. The remote server performs image retrieval algorithm and transmits the results back to the mobile device.

Manuscript received February 27, 2018.
Manuscript revised July 31, 2018.
Manuscript publicized September 14, 2018.

[†]The authors are with Beijing University of Posts and Telecommunications, China.
[††]The author is with the NVIDIA Corporation, China.
[†††]The authors are with the Peking University, China.
a) E-mail: czhu@bupt.edu.cn

DOI: 10.1587/transinf.2018EDP7075

ther reduce the scale-space construction time. In the server-end, the CDVS standard adopts the Scalable Compressed Fisher Vector (SCFV) [16] 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 [9]. However, the statistical information of the image database, such as inverse document frequency (*idf*), is ignored and the global similarity score is discarded in the reranking stage. To summarize, there are two problems to be addressed in the existing CDVS standard:

- How to design a highly efficient feature detection scheme remains a problem to be addressed by considering the relationships between different octaves in the scale-space pyramids. Here, the highly efficient scheme means the low-complexity of feature detection algorithm which achieves comparable retrieval performance when compared with the original CDVS.
- We believe that the ignorance of database statistical information and the global similarity in the image reranking stage will hurt the visual search performance. Research on how to fuse such information into the image reranking algorithm is still an open question.

Approach. For feature detection, we propose a highly efficient algorithm which utilizes the small blocks (say, 32×32) containing important detected local feature points of the higher-level scale space as the initial key area. Then, we use dilation operations on the initial key area to generate an expanded important area (EIA). The EIA is used to guide the low-level scale space construction and feature detection. The Gaussian and Laplacian operations are skipped for the area outside EIA to save the computing time of feature detection. Besides, feature selection is placed before orientation computing to further reduce the complexity by pre-discarding the orientation calculating process of some unimportant local points.

For image reranking, first a short visual codebook is generated to represent the statistical information of the database. 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. Our previous conference article [18] has reported some related contents about reranking and they are included as part of this work.

Outline. The remainder of the paper is structured as follows. In Sect. 2, we review the mobile visual search techniques: compact descriptor extraction and image retrieval. In Sect. 3, we present the details of our proposed highly efficient mobile visual search algorithm. In Sect. 4, we discuss the performance comparisons and then in Sect. 5 we conclude our paper.

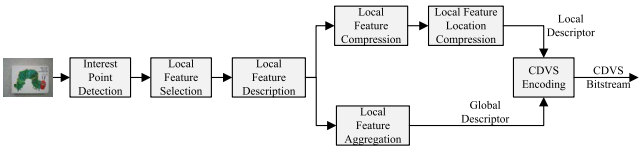

Fig. 2 Compact descriptor extraction process in CDVS standard. Compressed local descriptors and global descriptors are extracted from the query image and combined to form the CDVS bitstream.

2. CDVS Overview

For a typical mobile visual search application based on CDVS, there are two stages to perform: Compact descriptor extraction and image retrieval using CDVS descriptor.

2.1 Compact Descriptor Extraction

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 [7]; feature encoding module, which has gained a lot of interest in recent years [16], [17], [19], will compress the original local features to produce compact descriptors.

Figure 2 outlines the workflow 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.

For a query image, the CDVS standard first adopts ALP detector [10] to find the interest points by approximating the result of the 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 [7]. 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 [18]. 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, the global descriptor, the compressed local features, and the coded locations are merged to produce the CDVS bitstream in encoding module. In the above blocks, the local feature detection (including scale-space construction) is the most time-consuming part and in Sect. 3 we will design a low complexity feature detection scheme.

To conduct feature detection, we first need to build scale-space pyramid of the input image $\mathrm{I}(x, y)$, as shown in Fig. 3. The GS scale-space is constructed according to (1).

$$I_{GS}(x, y, \sigma) = GS(x, y, \sigma) * I(x, y) \tag{1}$$

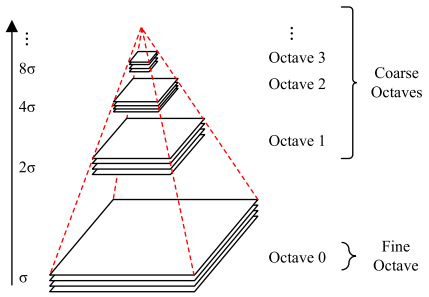

Fig. 3 Scale-space pyramid.

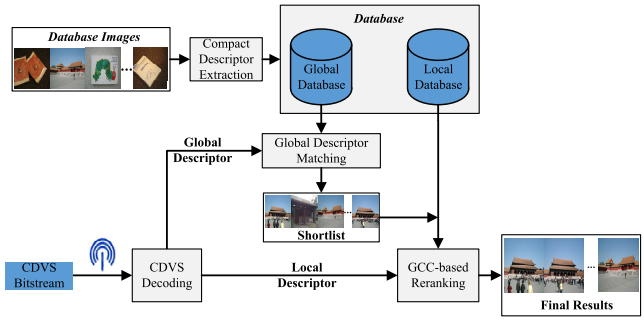

Fig. 4 Image retrieval framework based on CDVS bitstream.

where $GS(x, y, \sigma)$ is a series of Gaussian functions of different scale factors, and * denotes convolution operation. The local extrema of LoG with scale normalization will produce the most stable interest points, and LoG is generated by (2).

$$LoG(x, y, \sigma) = \sigma^2 \nabla^2 I_{GS}(x, y, \sigma) * I(x, y) \tag{2}$$

where ∇^2 is the Laplacian operator. Generally, as depicted in Fig. 3, the continuous scale-space is represented as discrete octaves, and each octave contains S (say 4) smoothed images with scale factor $\sigma_l = 2^{l/S}\sigma_0$, $l = 1, \ldots, S$. The input of the higher octave is the downsampled image with scale $2\sigma_0$ in the previous octave. In this paper, we treat octave 0 as the fine octave and the other as coarse octaves.

2.2 Image Retrieval Based on Compact Visual Descriptor

Generally, the retrieval system returns a ranked list of images that contain the same object based on the Bag-of-Words (BoW) [20] signature or global descriptors such as fisher vectors (FV) [21] and vector of locally aggregated descriptors (VLAD) [22]. However, the BoW signature and global descriptors are generated based on the orderless local features, which lead to the disregarding of information about the spatial layout of the features. To further improve image retrieval performance, Philbin et al. [23] propose to add an efficient spatial verification to rerank the results returned from the BoW model. Similarly, in work [25] 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 methods, which contain very large visual word tables [9], can yield decent performance, the fact that mobile visual search generally requires low memory makes these approaches unsuitable. For mobile search, the CDVS adopts the retrieval framework in Fig. 4.

First, the global and local descriptors are separated out from the query bitstream by the CDVS decoding module. Second, the global descriptor of the query 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 SGMS is computed according to

$$S_{GMS}(X, Y) = \frac{\sum_{i=0}^{511} b_i^X b_i^Y w_{Ha(u_i^X, u_i^Y)}(32 - 2Ha(u_i^X, u_i^Y))}{32\sqrt{\sum_{i=0}^{511} b_i^X}\sqrt{\sum_{i=0}^{511} b_i^Y}} \tag{3}$$

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 Eq. (3), $b_i^X = 1$ if i_{th} Gaussian is selected, if not then $b_i^X = 0$. $Ha(u_i^X, u_i^Y)$ and $w_{Ha(u_i^X, u_i^Y)}$ 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 results are generated according to L.

$$L = \sum_{i=1}^{P} cos(\frac{\pi}{2}\frac{min_i}{smin_i}) \tag{4}$$

In Eq. (4), cos is the cosine transform function, and $min_i/smin_i$ 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 [7]. For image retrieval, this paper focuses on the image reranking part in Fig. 4.

3. Proposed Highly Efficient Mobile Visual Search Algorithm

In this section, we will design a highly efficient mobile visual search algorithm. For the compact descriptor extraction part, we propose a low complexity feature detection scheme which uses the key areas as a guide to save the computing time. For the image retrieval part, the image reranking algorithm of CDVS is redesigned to achieve higher performance by merging more information, while the computing load of reranking just slightly increases.

3.1 Low Complexity Feature Detection Algorithm

The architecture of our feature detection algorithm is sum-

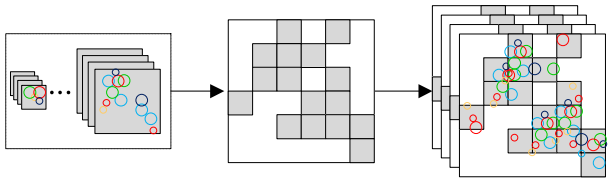

Fig. 5 Low complexity feature detection flow: (a) coarse scale-space construction and feature detection (b) important area mask generation based on key points in the coarse scales (c) fine scale-space construction and feature detection for the important area.

marized in Fig. 5. First, the proposed algorithm will construct the scale-space for the coarse octaves, and then detect the key points in the coarse pyramid. Second, we will conduct the important area generation based on the detected key points in the coarse octaves. Finally, we will perform the fine level scale-space construction and then conduct feature detection according to the generated important area. In the final stage, the scale-space construction and feature detection of the unimportant area in the fine octave are skipped and thus the computing time is saved.

3.1.1 Coarse Scale-Space Construction and Feature Detection

Different from the traditional GS scale-space construction, the coarse octaves, as shown in Fig. 5, of the pyramid are built first. It should be pointed out that the input for the coarse octave 1 is the down-sampled GS image with scale $2\sigma_0$ in octave 0. Thus, the corresponding GS image in octave 0 must be generated prior to the coarse octave building. After coarse GS pyramid construction, the coarse LoG pyramid will be generated and then the feature detection can be performed to produce the most stable image features, as described in the following.

In the generated coarse LoG pyramid, ALP detector is adopted to find the local extremas, which generally are viewed as the interest points. In the ALP detection process, for each detected interest point, a relevance measure r is calculated according to (5) at the same time. The relevance measure r indicates the priori probability of an interest point from query image matching a point of database image correctly [9] and thus a bigger r means a more important key point.

$$r(\sigma, p, d, \rho, p_{\sigma\sigma}) = f_1(\sigma)\cdot f_2(p)\cdot f_3(d)\cdot f_4(\rho)\cdot f_5(p_{\sigma\sigma}) \quad (5)$$

where "$\cdot$" is dot product and $f_1 \sim f_5$ are normative tables statistically learned conditional distributions according to the following characteristics of each key points: the scale σ, the peak response value p of the LoG, the distance d (from the interest point to the image center), the ratio ρ of the squared trace to the determinant of the Hessian, and the second derivative $p_{\sigma\sigma}$ of the scale-space function with respect to scale.

Both the spatial positions and the relevance measure r of each detected point will be taken into account in the following important area generation stage.

3.1.2 Important Area Mask Generation

First, we split the original image with size $W \times H$ into cells of $w \times w$ (say, 32×32) and use a state mask matrix M to represent the important state of the cells. If element $m(i, j)$ of M (i from 0 to $\lceil W/w \rceil - 1$, j from 0 to $\lceil H/w \rceil - 1$) is equal to 1, the corresponding cell will be defined as an important area and the fine scale space construction and feature detection process will be performed for $cell(i, j)$ in Sect. 3.1.3. However, the related computing process will be skipped for the cells with $m(i, j) = 0$.

Second, the matrix M will be initially set according to the positions and relevance measures of detected key points. In detail, all the generated key point positions of the coarse octaves are up-scaled to the original image size accordingly, and then each key point will fall into one specific cell. The total relevance measure for each cell will be generated by (6), and the matrix M will be set according to (7).

$$I(i, j) = \sum r(k) \quad k \in cell(i, j) \quad (6)$$

$$M(i, j) = \begin{cases} 1 & I(i, j) \geq T \\ 0 & I(i, j) < T \end{cases} \quad (7)$$

where T is a threshold. Obviously, if one cell contains more key points with bigger relevance values, it will tend to be judged as an important area. Besides, threshold T can be used to control the proportion of important and unimportant cells: a smaller T will lead to more cells are classified as important areas, and visa versa. Although bigger T can bring more time saving, the image retrieval performance is prone to be ruined in this situation.

Third, the number of elements with value 1 in matrix M is increased by using morphological dilation operation. In this paper, we believe that the areas in the fine octave around the centers of the detected key points are also apt to contain key information. Thus, we expand the important areas through applying the dilatation operator recursively. With each iteration, important areas will grow a cell width to the surrounding areas. In our work, we perform dilatation operation only once.

3.1.3 Fine Scale-Space Construction and Feature Detection

In fact, the construction process of the fine scale-space (octave 0) is the most time-consuming part of the original pyramid. In order to decrease the computing load, the fine scale-space images are also split into cells of size $w \times w$, and the construction of cells with the element $m(i, j) = 0$ will be skipped. It should be pointed out that for the GS pictures, all the skipped cells will be filled with pixels in the corresponding original image. This is because in the feature description stage of CDVS, the key points in the border of the

important areas may need the GS pixels in the unimportant areas and the filled original pixels are the rough estimations of the GS pixels. After the construction of GS pyramid for the important areas, the GS pictures are Laplace filtered in advance to build the LoG pyramid. In the LoG pyramid for the important areas, ALP detector is used to detect the key points.

For the original CDVS standard, in the feature detection process, one or more orientations (up to 4 in CDVS) are assigned to each detected key point. Then feature selection is performed after feature detection to remove some unimportant interest points. However, the orientation computing based on local image gradient directions is a time-consuming task due to the large amount of detected key points. To further reduce the complexity of feature detection, we proposed to conduct feature selection before the orientation computing. The orientation computing is just performed for a small amount of important key points.

3.2 Image Reranking Algorithm

Generally, image retrieval first sorts the database based on BoW [20] or global descriptor [16] 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 proposed image reranking architecture is shown in Fig. 6.

Local database statistical information is integrated into our proposed image reranking architecture. The local database is a collection of local descriptors, and we cluster them to create a l size, such as 300, codebook $C = (c_1, c_2, \ldots, c_l)$ by using k-means algorithm. Based on the codebook, we quantize and represent the local descriptors of database images as visual words and then compute the idf for each visual word. The idf is calculated according to (8).

$$idf_{c_i} = log\frac{N}{N_i} \tag{8}$$

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 idf weighting table is

$$W_{idf} = (idf_{c_1}, idf_{c_2}, \ldots, idf_{c_l}) \tag{9}$$

Both the codebook C and the generated weighting table W_{idf} will be used in LDSS computing.

We propose to use the following equation in image reranking.

$$S = \lambda S_{GMS} + (1-\lambda) S_{LDSS} \tag{10}$$

where S_{GMS} is the GMS presented in (3) and S_{LDSS} is LDSS which will be discussed later. In Eq. (10), λ and $(1-\lambda)$ are the weights for *global* zone and *local* zone, respectively. In the following, we will describe the calculation of S_{LDSS} and weight λ.

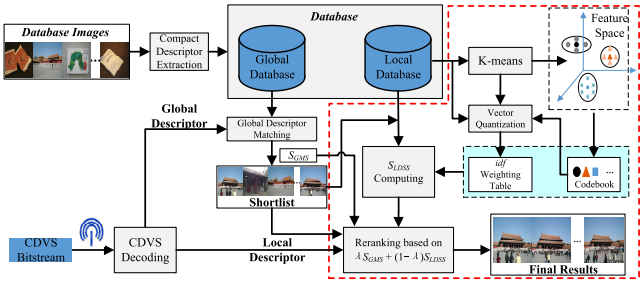

Fig. 6 Image retrieval architecture. The area inside the red dotted rectangle is our proposed image reranking.

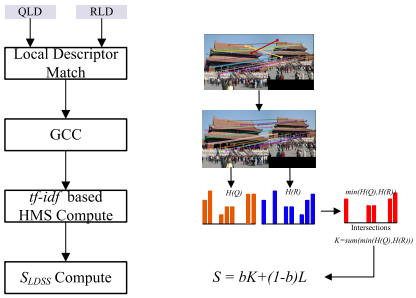

Fig. 7 Histogram matching based LDSS computing flow.

3.2.1 S_{LDSS} Computing Based on *tf-idf* Weighted Histogram Matching

S_{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 (11) to compute S_{LDSS}.

$$S_{LDSS} = bK + (1-b)L \tag{11}$$

where K is the *tf-idf* weighted histogram matching score (HMS), L is the reranking criteria used in original CDVS reference model as shown in (4), and 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 Fig. 7. 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

$$\begin{cases} Q_{QLD} = (q_1, q_2, \ldots, q_m) \\ R_{RLD} = (r_1, r_2, \ldots, r_n) \end{cases} \tag{12}$$

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 Eq. (13).

$$M(Q_{QLD}, R_{RLD}) = \{(q_{i1}, r_{j1}), (q_{i2}, r_{j2}), \ldots, (q_{ih}, r_{jh})\} \tag{13}$$

where q_{io} and q_{jo} (o from 1 to h) are from Q_{QLD} and R_{RLD}, respectively. Then, q_{io} and q_{jo} are quantized to create two weighting histograms $H(Q)$ and $H(R)$, according to the codebook and idf weighting table. $H(Q)$ and $H(R)$ are histograms with l bins, corresponding to the l visual words of the codebook. Based on Eq. (9), we have

$$\begin{cases} H(Q) = ((idf_{c_1})N_{q1}, (idf_{c_2})N_{q2}, \ldots, (idf_{c_l})N_{ql}) \\ H(R) = ((idf_{c_1})N_{r1}, (idf_{c_2})N_{r2}, \ldots, (idf_{c_l})N_{rl}) \end{cases} \tag{14}$$

where N_{qi} and N_{ri} (i from 1 to l) denote the counts of query and reference inlier points, contained in Eq. (13), 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 *tf-idf* weighted histogram matching score K

$$K(H(Q), H(R)) = \sum_{i=1}^{l} min(H(Q_i), H(R_i)) \tag{15}$$

where min is the minimum function.

Combining Eqs. (4) and (15), we rewrite (11) as

$$\begin{aligned} S_{LDSS} &= bK + (1-b)L \\ &= b\sum_{i=1}^{l} min(H(Q_i), H(R_i)) \\ &\quad + (1-b)\sum_{i=1}^{P} cos(2\pi\frac{min_i}{smin_i}) \end{aligned} \tag{16}$$

In this paper, we set $b = 0.5$ throughout experiments to achieve a good balance between histogram matching score K and CDVS reranking criteria L.

3.2.2 Learning of Zone Weight λ

Zone weight λ is a balancing factor to decide the reranking contributions of S_{LDSS} and S_{GMS}, respectively.

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 labeled as α and the others are labeled as β. At last, we select P image pairs with label α and N image pairs with label β to train λ 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 Eq. (17).

$$y = (x - MIN)/(MAX - MIN) \tag{17}$$

where x is the input value and y is the normalized output. MIN and MAX are the minimum and maximum of the input values. Then, let Pa_α and Pa_β as

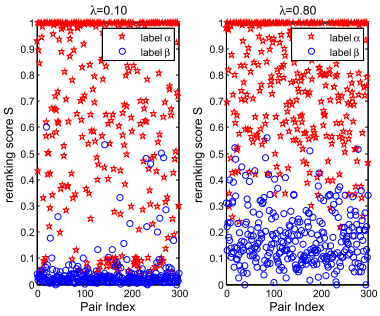

Fig. 8 The distribution of reranking score with different λ.

$$\begin{cases} Pa_\alpha = \{(S'_{GMS_{\alpha 1}}, S'_{LDSS_{\alpha 1}}), \ldots, (S'_{GMS_{\alpha i}}, S'_{LDSS_{\alpha i}}), \\ \qquad \ldots, (S'_{GMS_{\alpha P}}, S'_{LDSS_{\alpha P}})\} \\ Pa_\beta = \{(S'_{GMS_{\beta 1}}, S'_{LDSS_{\beta 1}}), \ldots, (S'_{GMS_{\beta j}}, S'_{LDSS_{\beta j}}), \\ \qquad \ldots, (S'_{GMS_{\beta N}}, S'_{LDSS_{\beta N}})\} \end{cases} \tag{18}$$

where S'_{GMS_i} and S'_{LDSS_i} are the i_{th} normalized GMS and LDSS of the image pairs. For explanation convenience, we randomly select 300 pairs with label α and 300 pairs with label β, and depict the corresponding reranking scores S with different λ values, as shown in Fig. 8.

We can see that different λ values generate different distributions of S. Then, the question becomes how to choose a proper λ which generates a S distribution that can be easily classified by a fixed threshold value. We propose to use Eq. (19) to train λ.

$$\hat{\lambda} = \underset{\lambda}{argmin} \frac{C_{inner}}{D_{inter}} \tag{19}$$

where C_{inner} represents the sum of concentration degree in each class, D_{inter} denotes the separation degree between class α and class β. We will formulate Eq. (18) in detail in the following. Corresponding to Pa_α and Pa_β, we have

$$\begin{cases} S'_\alpha(\lambda) = \{S'_{\alpha 1}(\lambda), \ldots, S'_{\alpha i}(\lambda), \ldots, S'_{\alpha P}(\lambda)\} \\ S'_\beta(\lambda) = \{S'_{\beta 1}(\lambda), \ldots, S'_{\beta j}(\lambda), \ldots, S'_{\beta N(\lambda)}\} \end{cases} \tag{20}$$

with

$$\begin{cases} S'_{\alpha_i}(\lambda) = \lambda S'_{GMS_{\alpha i}} + (1-\lambda)S'_{LDSS_{\alpha i}} \\ S'_{\beta_j}(\lambda) = \lambda S'_{GMS_{\beta j}} + (1-\lambda)S'_{LDSS_{\beta j}} \end{cases} \tag{21}$$

where i starts from 1 to P and j starts from 1 to N. Let $m_\alpha(\lambda)$ and $m_\beta(\lambda)$ are the expectations, and $var_\alpha(\lambda)$ and $var_\beta(\lambda)$ 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 $S'_\beta(\lambda)$.

$$m'(\lambda) = R_\alpha m_\alpha(\lambda) + R_\beta m_\beta(\lambda) \tag{22}$$

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

$$\begin{cases} C_{inner} = R_\alpha(var_\alpha(\lambda)) + R_\beta(var_\beta(\lambda)) \\ D_{inter} = R_\alpha(m_\alpha(\lambda) - m'(\lambda))^2 + R_\beta(m_\beta(\lambda) - m'(\lambda))^2 \end{cases} \tag{23}$$

Table 2 Performance comparison between our proposed low complexity descriptor extraction and the original CDVS method.

Dataset	UK		INRIA		Graphics		Paintings		Buildings	
Bitrate	Δ mAP	Δ T	Δ mAP	Δ T	Δ mAP	Δ T	Δ mAP	Δ T	Δ mAP	Δ T
0.5	−1.1	−12.2%	−1.78	−3.3%	−1.06	−11.7%	−0.58	−14.0%	−2.06	−32.3%
1	−0.9	−17.0%	−1.59	−5.3%	−0.82	−7.9%	−1.28	−15.1%	−1.47	−23.3%
2	−0.96	−17.7%	−2.39	−7.7%	−1.36	−7.1%	−0.75	−16.0%	−0.77	−16.0%
4	−1.17	−18.6%	−2.5	−6.1%	0.01	−8.4%	−0.02	−11.8%	−0.34	−16.4%
8	−0.79	−21.9%	−2.22	−6.1%	0.03	−10.0%	−1.11	−5.8%	0.52	−21.9%
16	−0.76	−23.8%	−2.33	−11.2%	0.07	−11.5%	−1.17	−8.0%	0.37	−31.3%
Average	**−0.95**	**−18.5%**	**−2.135**	**−6.6%**	**−0.52**	**−9.4%**	**−0.82**	**−11.8%**	**−0.63**	**−23.5%**

Table 1 Typical values of used parameters in this paper.

Parameter	T	b	P	N
Value	0.002072	0.5	300	300

We substitute Eq. (23) into Eq. (19) producing

$$\hat{\lambda} = \underset{\lambda}{argmin} \frac{R_\alpha(var_\alpha(\lambda)) + R_\beta(var_\beta(\lambda))}{R_\alpha(m_\alpha(\lambda) - m'(\lambda))^2 + R_\beta(m_\beta(\lambda) - m'(\lambda))^2} \tag{24}$$

Our proposed zone weight learning method will train the best λ by minimizing (24). One simple solution for λ is directly checking all the results of (24) by increasing λ with a small step, such as 0.01, from 0 to 1.

Once we obtain the best λ, we will rerank the shortlist according to

$$S'(\hat{\lambda}) = \hat{\lambda} S'_{GMS} + (1 - \hat{\lambda}) S'_{LDSC} \tag{25}$$

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

The typical values (used in this paper) of the related parameters in this section are summarized in Table 1. The adopted parameter λ will be presented in Sect. 4.

4. Experiments

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. The standardized CDVS bitstream makes the interoperability of the descriptors from different devices possible. We make comparisons for all the available modes. The experiments are conducted based on CDVS reference software test model framework 11 (TM 11.0) [26].

Dataset. We evaluate our proposed method on benchmark datasets University of Kentucky Benchmark (UK) [23] and INRIA Holidays (INRIA) [24]. UK dataset contains 2550 tagged ground truth groups, and each group contains 4 pictures of the same object with different views. INRIA dataset includes 1491 images, and 500 ground truth queries can be used for testing. Besides, 2500 graphic, 455 painting, and 466 building images are selected from the MPEG-7 CDVS dataset to further evaluate our algorithm.

Performance evaluation criteria. The mean average precision (mAP) and computing complexity are used to evaluate our proposed highly efficient visual search algorithm.

The mAP is used to evaluate the image retrieval performance. The mAP is shown in Eq. (26).

$$mAP = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{r=1}^{M^i_{relevant}} P(r)}{M^i_{relevant}} \tag{26}$$

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

The time-saving is adopted to evaluate the complexity reduction by using our feature detection. Large time-saving means high degree efficientness of feature detection algorithm, and vice versa. To realize this, we calculate ΔT which is defined as

$$\Delta T = \frac{T_{pro\_DE} - T_{org\_DE}}{T_{org\_DE}} \times 100\% \tag{27}$$

where T_{pro_FD} and T_{org_FD} represent the compact descriptor extraction time with the proposed algorithm and with the original CDVS implementation, respectively. When comparing ΔT, we indicate the total descriptor extraction time including the processes other than feature detection, such as feature compression.

4.1 Low Complexity Descriptor Extraction

We compare our proposed low complexity method with the descriptor extraction in original CDVS on a PC with a quad-core 2.4GHz CPU and 8GB memory. As shown in Table 2, our proposed method works very well on datasets UK and Buildings, and the average time-saving ratio can reach up to 18.5% and 23.5% respectively when compared to the original CDVS. In other words, the proposed method can largely reduce the complexity of descriptor extraction on the above two datasets, but only introduces average 0.95% and 0.63% mAP loss for the image retrieval performance. For INRIA dataset, our method saves only 6.6% computing time but introduces 2.135% mAP loss. The reason is that there are no obviously unimportant areas for INRIA, and all regions have almost equally important attributes. Our method achieves the middle improvements for Graphics and Paintings.

4.2 Image Reranking Comparisons

Zone weight learning. In this paper, we use UK dataset to generate zone weights. For UK dataset, we divide it into two parts. We choose 300 images for zone weight λ training, and the rest dataset is used for the visual search performance evaluation. Table 3 depicts the learned λ. 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 λ. Figure 9 shows the image retrieval performance improvements on different datasets with learned zone weight λ. Although the zone weights are trained based on UK dataset, different retrieval performance gains are introduced on the other datasets. In the following evaluation, we use the same zone weights as shown in Table 3 for all the selected datasets.

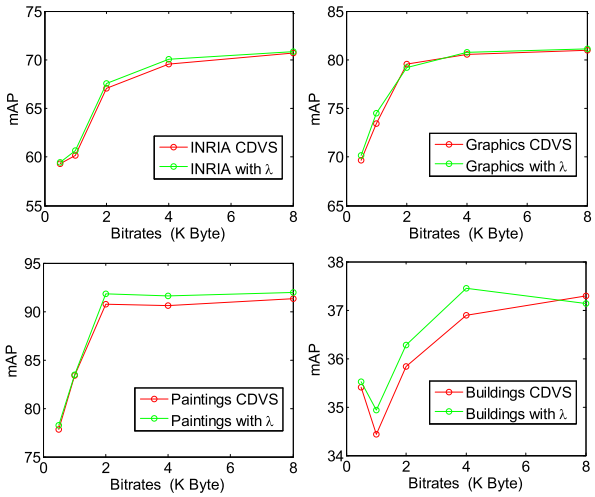

Fig. 9 The effectiveness of λ for different datasets.

Table 3 Learned λ in different bitrate modes.

Bitrate (KB)	0.5	1	2	4	8	16
λ	0.66	0.53	0.44	0.44	0.36	0.35

Reranking Comparisons. The image reranking method based on spatial verification in [23] and the image retrieval algorithm in CDVS are tested to make comparisons with our reranking algorithm. All the methods are integrated into TM 11.0. Test results are tabulated in Table 4. The results show that our proposed reranking achieves the best searching performance among all 3 methods.

4.3 Results of Proposed Highly Efficient Visual Search

For mobile visual search, CDVS achieves the state-of-the-art performance. To evaluate our proposed highly efficient visual search, in this part, we integrate both the low complexity feature detection and the proposed reranking method into TM11 together. Both our method and the original CDVS are tested.

The image retrieval performance is summarized in Table 5. Our proposed highly efficient visual search algorithm achieves comparable retrieval performances in average for all bitrates, which are just less than 0.3% mAP loss than the state-of-the-art for dataset UK, Graphics, Paintings, and Buildings. For descriptor extraction, as discussed in Sect. 4.1, our method can save between 9.4% and 23.5% computing time for these four datasets, which can largely alleviate the complexity problem for the mobile devices. However, for INRIA dataset, the retrieval performance loss still reaches 1.51% even when enabling our proposed reranking scheme. For the server-end image retrieval process, most part of our proposed reranking method can be pre-computed offline (stored as tables), the online image retrieval computing complexity is just increased slightly compared with CDVS.

Table 4 Image reranking performance comparisons.

Dataset	UK			INRIA			Graphics			Paintings			Buildings		
Bitrate	mAP														
	[23]	CDVS	Proposed	[23]	CDVS	Proposed	[23]	CDVS	Proposed	[23]	CDVS	Proposed	[23]	CDVS	Proposed
0.5	76.17	76.27	**76.8**	59.07	59.28	**59.44**	69.28	69.67	**70.13**	77.52	77.79	**78.25**	35.23	35.41	**35.52**
1	76.18	77.44	**79.39**	58.58	60.15	**60.65**	71.81	73.4	**74.51**	83.04	83.38	**83.47**	34.08	34.43	**34.93**
2	80.76	82.25	**83.03**	64.83	67.09	**67.54**	78.68	79.57	**79.2**	90.07	90.74	**91.8**	34.97	35.83	**36.28**
4	82.59	83.86	**84.99**	67.18	69.57	**70.05**	79.38	80.53	**80.8**	90.23	90.61	**91.63**	35.67	36.9	**37.45**
8	84.23	84.62	**85.03**	69.95	70.71	**70.82**	80.23	81.01	**81.15**	91.31	91.31	**91.95**	36.65	37.3	**37.14**
16	84.2	84.64	**85.05**	69.81	70.64	**70.65**	80.21	80.99	**81.11**	91.4	91.4	**91.42**	36.99	37.65	**37.48**
Average	80.69	81.51	**82.4**	64.90	66.24	**66.53**	76.60	77.53	**77.82**	87.26	87.54	**88.09**	35.60	36.25	**36.47**

Table 5 Image retrieval performance of proposed visual search.

Dataset	UK			INRIA			Graphics			Paintings			Buildings		
Bitrate	mAP		Δ mAP	mAP		Δ mAP	mAP		Δ mAP	mAP		Δ mAP	mAP		Δ mAP
	CDVS	Proposed		CDVS	Proposed		CDVS	Proposed		CDVS	Proposed		CDVS	Proposed	
0.5	76.27	75.71	**−0.56**	59.28	58.2	**−1.08**	69.67	69.25	**−0.42**	77.79	77.79	**0**	35.41	33.452	**−1.96**
1	77.44	78.34	**0.9**	60.15	59.73	**−0.42**	73.4	73.76	**0.36**	83.38	82.63	**−0.75**	34.43	33.39	**−1.04**
2	82.25	81.85	**−0.4**	67.09	65.47	**−1.62**	79.57	78.05	**−1.52**	90.74	91.05	**0.31**	35.83	35.92	**0.09**
4	83.86	83.78	**−0.08**	69.57	67.49	**−2.08**	80.53	80.51	**−0.02**	90.61	91.61	**1**	36.9	36.92	**0.02**
8	84.62	84.23	**−0.39**	70.71	68.84	**−1.87**	81.01	81.01	**0**	91.31	90.84	**−0.47**	37.3	37.97	**0.67**
16	84.64	84.29	**−0.35**	70.64	68.67	**−1.97**	80.99	81.16	**0.17**	91.4	90.23	**−1.17**	37.65	38.12	**0.47**
Average	81.51	81.37	**−0.15**	66.24	64.73	**−1.51**	77.53	77.29	**−0.24**	87.54	87.36	**−0.18**	36.25	35.96	**−0.29**

5. Conclusion

In this paper, we propose a highly efficient compact mobile visual search algorithm. For descriptor extraction process, we propose a low complexity feature detection algorithm which utilizes the detected local key points of the coarse octaves to guide the scale space construction and feature detection in the fine octave. Besides, feature selection is placed before orientation computing to further reduce the complexity of feature detection. For the image retrieval process, we design a high-performance reranking method, which merges both the GMS and the LDSS. The test results show that the proposed highly efficient approach achieves comparable performance with the state-of-the-art for mobile visual search, while the descriptor extraction complexity is reduced largely in average for most datasets. In the future, we target at highly efficient algorithm with adaptive cell partition size. Besides, we will focus on low complexity scheme for the hybrid feature (including both the handcrafted feature and deep feature) based mobile visual search.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (61602011, 61671078, 61701031), Director Funds of Beijing Key Laboratory of Network System Architecture and Convergence (2017BKL-NSAC-ZJ-06), and 111 Project of China (B08004, B17007). This work is conducted on the platform of Center for Data Science of Beijing University of Posts and Telecommunications.

References

[1] N. Singhai and S.K. Shandilya, "A Survey On: Content Based Image Retrieval Systems, International Journal of Computer Applications, vol.4, no.2, pp.22–26, 2010.
[2] G.-H. Liu and J.-Y. Yang, "Content-based image retrieval using color difference histogram," Pattern recognition, vol.46, no.1, pp.188–198, 2013.
[3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in European Conference on Computer Vision, Springer, vol.8689, pp.584–599, 2014.
[4] H. Azizpour, A.S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.36–45, 2015.
[5] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in Proceedings of the IEEE International Conference on Computer Vision, pp.1269–1277, 2015.
[6] G. Tolias, R. Sicre, and H. Jegou, "Particular object retrieval with integral max-pooling of cnn activations," arXiv preprint arXiv:1511.05879, 2015.
[7] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol.60, no.2, pp.91–110, 2004.
[8] B. Girod, V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham, "Mobile visual search," IEEE Signal Process. Mag., vol.28, no.4, pp.61–76, 2011.
[9] L.-Y. Duan, V. Chandrasekhar, J. Chen, J. Lin, Z. Wang, T. Huang, B. Girod, and W. Gao, "Overview of the MPEG-CDVS Standard," IEEE Trans. Image Process., vol.25, no.1, pp.179–194, 2016.
[10] S. Paschalakis et al., Information Technology Multimedia Content Description Interface Part 13: Compact Descriptors for Visual Search, ISO/IEC 15938-13, 2015.
[11] G. Koutaki and K. Uchimura, "Scale-space processing using polynomial representations," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2744–2751, 2014.
[12] G. Koutaki and K. Uchimura, "XY-Separable Scale-Space Filtering by Polynomial Representations and Its Applications," IEICE Trans. Inf. & Syst., vol.E100-D, no.4, pp.645–654, 2017.
[13] L.-Y. Duan, V. Chandrasekhar, S. Wang, et al., "Compact descriptors for video analysis: The emerging MPEG standard," arXiv preprint arXiv:1704.08141, 2017.
[14] Y. Lou, Y. Bai, J. Lin, S. Wang, J. Chen, V. Chandrasekhar, L.-Y. Duan, T. Huang, A.C. Kot, and W. Gao, "Compact deep invariant descriptors for video retrieval," Data Compression Conference (DCC), IEEE, pp.420–429, 2017.
[15] J. Chen, L.-Y. Duan, F. Gao, J. Cai, A.C. Kot, and T. Huang, "A low complexity interest point detector," IEEE Signal Process. Lett., vol.22, no.2, pp.172–176, 2015.
[16] J. Lin, L.-Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao, "Rate-adaptive compact Fisher codes for mobile visual search," IEEE Signal Process. Lett., vol.21, no.2, pp.195–198, 2014.
[17] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.1, pp.117–128, Jan. 2011.
[18] C. Zhu, L. Tao, F. Yang, T. Lu, H. Jia, and X. Xie, "Mobile Visual Search Based on Histogram Matching and Zone Weight Learning[C]," Journal of Physics: Conference Series, IOP Publishing, vol.933, 1, 012001, 2018.
[19] S. Paschalakis et al., CDVS CE2: Local Descriptor Compression, document ISO/IEC JTC1/SC29/WG11/M28179, Jan. 2013.
[20] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," International Journal of Machine Learning and Cybernetics, vol.1, no.1-4, pp.43–52, 2010.
[21] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), San Francisco, pp.3384–3391, 2010.
[22] H. Jgou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), San Francisco, pp.3304–3311, 2010.
[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07, pp.1–8, 2007.
[24] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," European conference on computer vision, Springer, Berlin, Heidelberg, vol.5302, pp.304–317, 2008.
[25] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," IEEE Trans. Pattern Anal. Mach. Intell., vol.31, no.4, pp.591–606, 2009.
[26] Test Model 11: Evaluation framework for Compact Descriptor for Visual Search, document ISO/IECJTC1/SC29/WG11/N14680, 2014.

Chuang Zhu received the B.S. degree in computer science from The North University of China, Taiyuan, China, in 2008 and the Ph.D. degree in the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, in 2015. He is currently a Lecturer with the School of Information and Communication Engineering, BUPT. His research focuses on mobile visual search, multimedia processing, and big data analytics.

Xiaofeng Huang received the B.S. degree in Microelectronics from The Nanjing University of Posts and Telecommunications, Nanjing, China, in 2010 and the Ph.D. degree in the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, in 2016. He is currently with the NVIDIA Corporation. His research focuses on multimedia processing, video coding and VLSI chip design.

Guoqing Xiang received the B.S. degree in Microelectronics from HeFei University of Technology, Hefei, China, in 2014. He is currently pursuing the Ph.D. degree from the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research focuses on video coding, image processing and VLSI chip design.

Huihui Dong is currently pursuing the B.S. degree in communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include image processing, pattern recognition and big data analysis.

Jiawen Song received the B.S. degree in computer science from The Zhongnan University of Economics and Law. He is currently pursuing the Master degree from the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research focuses on computer vision, multimedia retrieval and video coding optimization.