PAPER

Image Retrieval Framework Based on Dual Representation Descriptor

Yuichi YOSHIDA[†a)], ***Member and*** **Tsuyoshi TOYOFUKU**[†], ***Nonmember***

SUMMARY Descriptor aggregation techniques such as the Fisher vector and vector of locally aggregated descriptors (VLAD) are used in most image retrieval frameworks. It takes some time to extract local descriptors, and the *geometric verification* requires storage if a real-valued descriptor such as SIFT is used. Moreover, if we apply binary descriptors to such a framework, the performance of image retrieval is not better than if we use a real-valued descriptor. Our approach tackles these issues by using a dual representation descriptor that has advantages of being both a real-valued and a binary descriptor. The real value of the dual representation descriptor is aggregated into a VLAD in order to achieve high accuracy in the image retrieval, and the binary one is used to find correspondences in the *geometric verification* stage in order to reduce the amount of storage needed. We implemented a dual representation descriptor extracted in semi-real time by using the CARD descriptor. We evaluated the accuracy of our image retrieval framework including the *geometric verification* on three datasets (holidays, ukbench and Stanford mobile visual search). The results indicate that our framework is as accurate as the framework that uses SIFT. In addition, the experiments show that the image retrieval speed and storage requirements of our framework are as efficient as those of a framework that uses ORB.

key words: *image retrieval, local descriptor*

1. Introduction

Descriptor aggregation techniques such as the Fisher vector and vector of locally aggregated descriptors (VLAD) are used for instance searches or object-based image retrieval because of their robustness and runtime efficiency [1]. The frameworks are composed of four stages, i.e., *local descriptor extraction*, *descriptor aggregation*, *search*, and *geometric verification*.

Image retrieval frameworks often use SIFT as a local descriptor because of its robustness [2]. SIFT descriptors, however, require a lot of time to extract local descriptors from an image. *Geometric verification* filters out *search errors* by using spatial invariant features. *Search errors* arise when the query image does not always include any objects in the database and incorrect results are returned by nearest neighbor search in the *Search* stage. The *geometric verification* stage needs to store descriptors of all images in the database.

A straightforward approach to tackling the issues of speed and storage size of image retrieval is to use a binary descriptor. The binary descriptors such as ORB and BRISK that have bits at each keypoint [3]–[5]. Aggregation vectors with binary descriptors are not as robust as aggregation vectors with real-valued descriptors such as SIFT [6]. Thus, this approach sacrifices accuracy of image retrieval.

We propose a framework based on a dual representation descriptor that extracts real-valued and binary descriptors simultaneously and quickly. CARD can extract dual representations in semi-real time by probing the real-valued descriptors before binarizing them. The existing frameworks do not use two descriptors that are extracted simultaneously for different purposes. Our framework is as robust as typical frameworks that use SIFT [1], [7], [8], and it can reduce both the time needed to extract local descriptors and storage size as well as an approach that uses binary descriptors [6].

Our contributions are as follows. First, we propose a fast and compact image retrieval framework based on a dual representation descriptor. Second, we implemented the framework with the CARD descriptor [9]. Third, we conducted evaluations on three datasets showing that our framework accelerates image retrieval and reduces the amount of storage needed for the *geometric verification* without sacrificing accuracy of image retrieval. In particular, in terms of the receiver operating characteristic (ROC) curves of image retrieval, our framework outperforms the existing methods on databases that include a lot of planar objects.

2. Related Work

2.1 Local Descriptor Extraction and Descriptor Aggregation

Local descriptor extraction is the most time-consuming step in image retrieval frameworks based on the descriptor aggregation technique because extracting real-valued descriptors like SIFT entails high computational costs. In particular, one has to compute rotations of many image patches in order to keep the orientation invariance of the descriptors. SURF descriptors can be extracted much faster by making an approximation of SIFT descriptors [10], but even these cannot be extracted as fast as binary descriptors. The binary descriptors such as CARD, ORB, and BRISK obtain binary values by using simple binary tests between pixels in a smoothed image patch instead of computing gradients from the patch [3]–[5], [9]. The extracted descriptors are passed to the next *descriptor aggregation* stage.

Descriptor aggregation aggregates local descriptors

Manuscript received February 8, 2017.
Manuscript revised May 31, 2017.
Manuscript publicized July 6, 2017.

[†]The authors are with DENSO IT Laboratory, Inc., Tokyo, 150–0002 Japan.
a) E-mail: yyoshida@d-itlab.co.jp

DOI: 10.1587/transinf.2017EDP7050

into a single vector as a signature of an image. The vector is an aggregation vector. The accuracy of an image retrieval framework depends on the type of local descriptor and the method of aggregating them. Most frameworks use the Fisher vector or vector of locally aggregated descriptors (VLAD). The Fisher vector is a gradient vector of an input image's likelihood with respect to the parameters of a distribution, and it is scaled with the Fisher information matrix [11]. It is often computed under the assumption that SIFT descriptors are generated from a Gaussian mixture model. VLAD simplifies the Fisher vector in order to reduce the computational cost [1]. It ignores terms that include the covariance and weights from the Fisher vector.

Uchida *et al*. proposed an image retrieval method that aggregates ORB descriptors [6]. Their approach computes Fisher vectors by aggregating ORB descriptors under the assumption that they are generated from a Bernoulli mixture model [12]. Fisher vectors composed of ORB descriptors were compared with bag of binary words [13] on the Stanford mobile visual search dataset (SMVSD) [14]. Although they were found to be better than the competitor, they did not compare their approach with the typical VLAD.

2.2 Search and Geometric Verification

The *search* stage finds candidate images in the database that is the most similar to the query image. We can apply a nearest neighbor search to this stage. Generally speaking, *search* returns a ranked list composed of the top N most similar images to the query, whereas the next stage, *geometric verification*, removes *search errors* from the ranked list by using spatially invariant features [15].

The *search* and *geometric verification* stages have been studied from the viewpoints of efficiency and precision. The *search* stage can use nearest neighbor searches. In particular, PQTable is 10^2 to 10^5 times faster than other existing methods, and it calculates the nearest neighbor of 64-bit binary vectors among a list containing 10^9 vectors in 10 milliseconds [16]. Xinchao *et al*. proposed an efficient method that enables one to check the geometric relation of a pair of images in 1 millisecond for the purpose of making a *geometric verification* [17].

2.3 Issues

Image retrieval involves three important measurements, i.e., accuracy, speed of image retrieval, and amount of storage for descriptors. There haven't been any attempts yet aimed at resolving these issues simultaneously.

1. *Local descriptor extraction* is important for accelerating most image retrieval frameworks. We can execute the *search* and *geometric verification* 10 to 10^2 times faster than SIFT extraction during the *local descriptor extraction* step if we use adequate methods for those components.
2. The *geometric verification* needs to store all positions in the image and descriptors of keypoints in order to find correspondences between them. Because the amount of storage linearly increases with the number of database images, the descriptor must be compact.
3. These two issues can be resolved if we use ORB as the local descriptor to be aggregated. The frameworks that have been developed so far that use an aggregation vector with binary descriptors have not shown sufficient retrieval performance compared with real-valued descriptors like SIFT.

3. Image Retrieval Framework Based on Dual Representation Descriptor

3.1 Dual Representation Descriptor

We propose an image retrieval framework based on a dual representation descriptor that obtains real-valued and binary descriptors at each keypoint. The real-valued descriptors are aggregated into a single vector in order to represent an image, and they are based on a histogram of gradients. Such an aggregation vector is as robust as one with SIFT descriptors. Our approach applies binary descriptors to the *geometric verification*. The *geometric verification* needs the positions and descriptors of all images of the keypoints in order to compute correspondences. We can reduce the amount of storage needed because binary descriptors are more compact than real-valued ones.

3.2 Our Image Retrieval Framework

It is difficult to implement the dual representation descriptor by using straightforward methods or by combining existing approaches. We cannot obtain two types of descriptor at the same time by using a typical binary descriptor approach, which directly outputs binary values by making simple binary tests between pixels. Moreover, while we could convert the SIFT descriptor into a binary descriptor by using locality sensitive hashing (LSH) [18], this approach would clearly take more time than the original SIFT approach takes.

We implemented a dual representation descriptor incorporating the CARD descriptor [9]. The binary descriptor of a CARD descriptor is indirectly obtained. It is converted from real-valued descriptors after quickly extracting real-valued ones from the image patches. The implementation of the dual representation descriptor is described below.

(1) Local descriptor extraction

The real-valued descriptors of CARD are obtained by using a lookup table (LUT) after obtaining a histogram of gradients in a circular patch, like GLOH [19]; this avoids having to rotate image patches in order to keep orientation invariance. Real-valued descriptors $\mathbf{x}$ is aggregated into a VLAD. Next, the real-valued descriptors are converted into binary descriptors by using a random projection technique. Here, let $\mathbf{b}$ be an binary vector, B be the dimension of the binary vector (In the default parameters of CARD, $B = 128$,

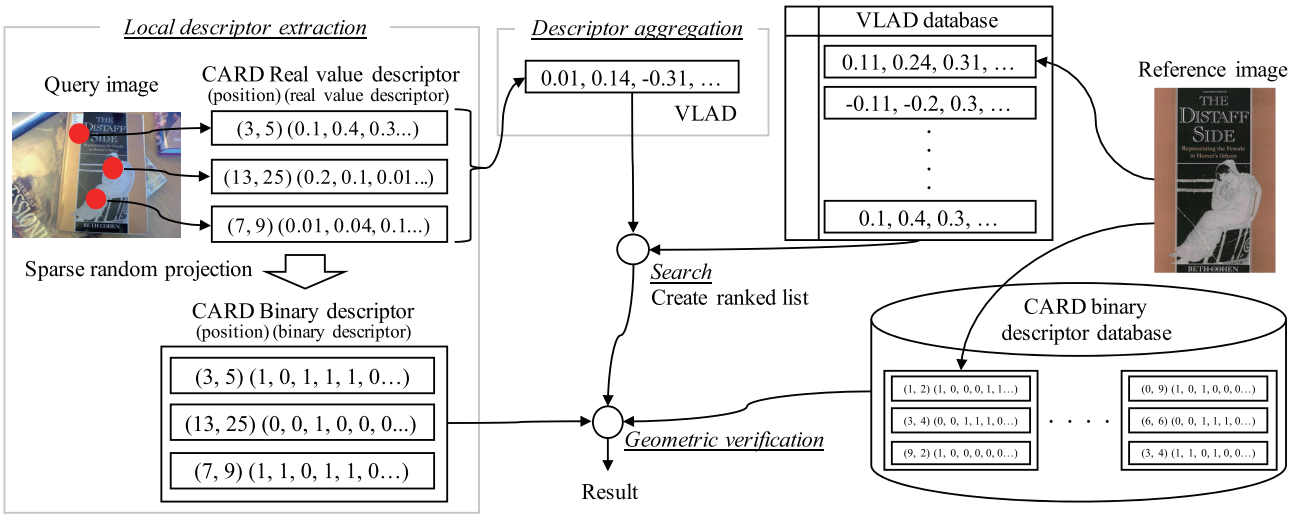

Fig. 1 Overview of proposed framework.

$D = 136$), and $\mathbf{W}$ be a sparse random projection matrix; $\mathbf{x}$ is converted into a binary vector $\mathbf{b}$ as follows:

$$\mathbf{b} = \frac{\mathrm{sgn}(\mathbf{W} \cdot \mathbf{x}) + 1}{2} \tag{1}$$

where $\mathbf{b} \in \{0, 1\}^B$, $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{W} \in \{-1, 0, 1\}^{B \times D}$. Equation (1) can be executed without incurring a high computational cost because $\mathbf{W}$ takes only three entries $\{-1, 0, 1\}$, each with probabilities $\{\frac{1}{2\sqrt{D}}, 1 - \frac{1}{\sqrt{D}}, \frac{1}{2\sqrt{D}}\}$. $\mathbf{W}$ is very sparse because zero elements have a large probability, for example, 91.4% when $D = 136$.

(2) Descriptor aggregation

$\mathbf{x}$ is converted into a VLAD vector. VLAD uses the residual between the original vector $\mathbf{x}_t$ and its nearest centroid obtained by k-means clustering. Here, let $\boldsymbol{\mu}$ be the centroids of k-means clusters, T be the number of keypoints in an image, $\mathrm{NN}(\mathbf{x}_t)$ be $\mathbf{x}_t$'s nearest centroid, and K be the number of centroids; the dimension of VLAD vectors $\mathbf{v}$ is $D \cdot K$, and a block of VLAD vectors is:

$$\mathbf{v}_i = \sum_{\mathbf{x}_t : \mathrm{NN}(\mathbf{x}_t) = i} \mathbf{x}_t - \boldsymbol{\mu}_i, \tag{2}$$

where $\mathbf{v}_i \in \mathbb{R}^D$, $i = 1, \ldots, K$, $t = 1, \ldots, T$. Both the Fisher vector and VLAD are composed by concatenating components, and each component has a different scale. The concatenated vector should be normalized at each component instead of using general L2 normalization. Intra L2 and power-law normalization methods are often used in order to normalize the concatenated vector [1], [20].

(3) Search and geometric verification

A lot of methods can be used in these two stages. We chose methods on the basis of the evaluation of the image retrieval framework. The VLAD database stores VLAD vectors that were computed from reference images beforehand. The *search* stage creates a ranked list composed of the N nearest VLAD vectors to the query image's VLAD. To reduce the computational cost and size of the database of the *search* stage, aggregation vectors are usually compared with each other by using approximate nearest neighbor searches. One approach is random projection, which is CARD's binarizing technique. In addition, there are various approaches such as product quantization [21] and FLANN [22]. We used the random projection technique in the experimental evaluation. In comparative experiments, the cosine distance between the two aggregation vectors was computed without any approximations in order to eliminate the effect of such algorithms on the results.

The *geometric verification* eliminates *search errors* from the ranked list. The ranked list includes the N nearest neighbors as candidate images. For each candidate image, the positions and descriptors of all keypoints in the image are loaded from the descriptor database. The homography matrix between the query image and the candidate image is estimated by applying RANSAC [23] to their correspondences. Finally, the number of inlier correspondences of which the projection error from the query image to the candidate image was less than a certain threshold is obtained. The ranked list is ordered by the number of inlier correspondences again. The image retrieval result is the top one of the re-ordered ranked list. Furthermore, if the number of inlier correspondences of the top one is less than a threshold, it is rejected regardless of the nearest neighbor of the query. Our framework enables one to adjust the balance between the true-positive and false-positive rates of image retrieval by changing the threshold, which is defined as the *geometric verification threshold*.

4. Experiments

4.1 Overview and Protocol

First, we studied how well our framework, which aggregates CARD real-valued vectors into VLAD and uses

Table 1 Number of positive and negative samples for each database.

Detail	holidays	ukbench	SMVSD
References (database)	500	2,550	492
Positive samples (query)	1,491	10,200	1,968
Negative samples (query)	1,500	4,000	2,000

Table 2 Extraction speed and data size of local descriptors.

Name	Time [ms]	Size [byte]	Type	Implementation
ORB	6	16	Binary	OpenCV
CARD	**10**	**16**	**Binary**	**Original (in C)**
SURF	60	256	Real	OpenCV
SIFT	160	512	Real	OpenCV

binary-valued vectors for geometric verification, works in a practical runtime environment. Next, we studied how much our framework accelerates the image retrieval process and reduces the amount of storage required compared with other approaches. We conducted experiments on three datasets, *holidays* [24], ukbench [25], SMVSD[†] [14]. All images were resized to less than 480 pixels beforehand. The SMVSD images are categorized into references or queries, but the images in the other databases have not been categorized in this way. Holidays and ukbench are composed of many sets of images. The images in each set include the same object. We chose one image from each set as a reference image and let the remaining images be positive-sample query images. Furthermore, we selected from mirflickr25k negative samples that did not include any objects in the three datasets. Table 1 describes the samples. Each of the positive-sample query images should be assigned to an appropriate reference image, whereas all negative-samples should be rejected.

We used the mirflickr25k dataset[††] to learn each parameter for the *descriptor aggregation*. For VLAD and bag of visual words vector (BOVW), the centroids that are needed to calculate them were learned using the k-means algorithm. The parameter sets of the Bernoulli mixture model that is needed to calculate Fisher vectors from the binary vectors were learned with the EM-algorithm. These three parameter sets were learned using one million descriptors extracted from mirflickr25k.

4.2 Evaluation

We verified our method by comparing the proposed method with a naive one that does not execute the *geometric verification* after the *search*. In this evaluation, all VLAD vectors in the database were quantized in order to shorten the time needed to execute the *search* and reduce the size of the database. VLAD vectors were converted into binary vectors with the random projection method. We calculated the mean VLAD vector, which was the center of the VLAD vectors of all images in mirflickr25k, beforehand. The mean VLAD vector was subtracted from all VLAD vectors. The centered VLAD vectors were normalized by using the intra L2 normalization method before converting them into binary vectors [20]. The number of centroids for VLAD, K, was 64. Each of the 64 blocks among the normalized VLAD vectors was converted into 256 bit binary vectors by using random projection.

[†]We used the *book covers*, *cd covers*, *dvd covers*, *museum paintings*, and *video frames* categories.

[††]http://press.liacs.nl/mirflickr/mirdownload.html

We calculated the true positive rate (a correct result is returned when a positive-sample is posted as a query image) and false positive rate (a negative-sample is not rejected when it is posted) while varying thresholds over the three datasets. The naive method outputs the top one of the ranked list as the image retrieval result. As well as the proposed method, the naive one should reject negative-samples in order to evaluate the true positive and false positive rates. The naive method rejects the top one in the list if its *search score* is less than the threshold, which is defined as the *naive threshold*. We drew the receiver operating characteristic (ROC) curves for each dataset while varying the *geometric verification threshold* and *naive threshold* in Fig. 2. We simultaneously calculated top-1 (naive) and top-5 precision (probability that the ranked list includes an appropriate reference image) in the *search* step and the precision of the *geometric verification* step (proposed) (Fig. 3). Figure 2 shows that our method has better precision and reproducibility compared with the naive one. Figure 3 shows that the *geometric verification* of our framework improves the results of *search*.

4.3 Comparative Experiments

We compared our framework using the dual representation descriptor with straightforward frameworks using ORB, SIFT, and SURF. We assumed that the VLAD composed of CARD real-valued vectors is as robust as SIFT and more than robust than the Fisher vectors of ORB. To verify this assumption, we needed to demonstrate that the accuracy of our approach can be compared with the SIFT-based approach and that our method outperforms the approach using the Fisher vector of ORB.

We extracted these descriptors by using the default parameters of each implementation (Table 2). SIFT, SURF, and CARD (real-valued vector) were converted into VLAD, and ORB was converted into a Fisher vector. Moreover, a bag of visual words vector (BOVW) was calculated using SIFT in order to compare our approach with BOVW, which is a popular aggregation method [7]. In particular, ORB was converted into a Fisher vector by using the method described by Uchida *et al*. [6].

We normalized VLAD by using the intra L2 normalization method and the Fisher vector of ORB by using power-law and intra L2 normalizations [20]. To study the effect of varying K, the number of centroids for VLAD and BOVW was set to 8, 16, 32, 64, 128, or 256. The size of the Bernoulli mixture model was set to the same values. To eliminate the effect of approximate nearest neighbor searches from the results, the cosine distance between the

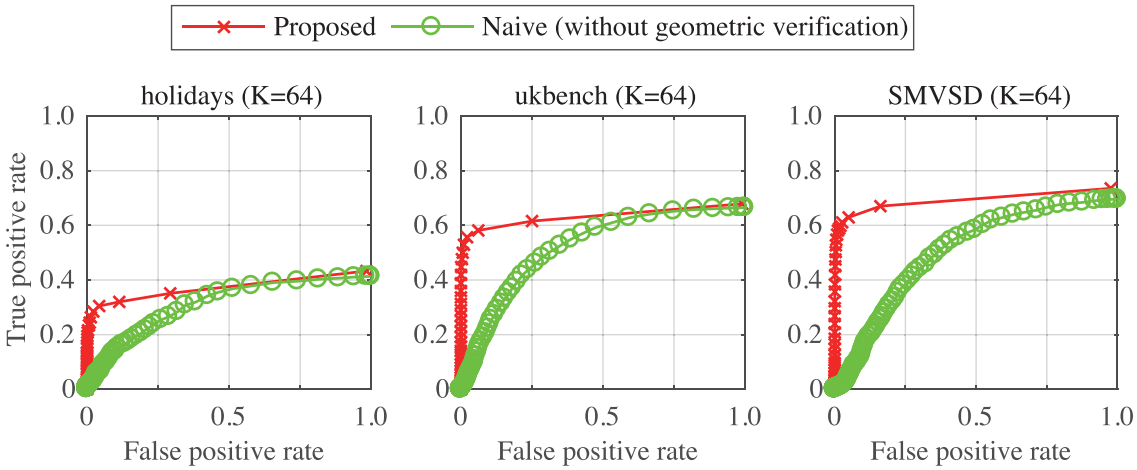

Fig. 2 Receiver operating characteristic (ROC) curves of true-positive and false-positives rates in image retrieval test on the three datasets, comparing the proposed method with the naive one that does not execute *geometric verification* after *search*.

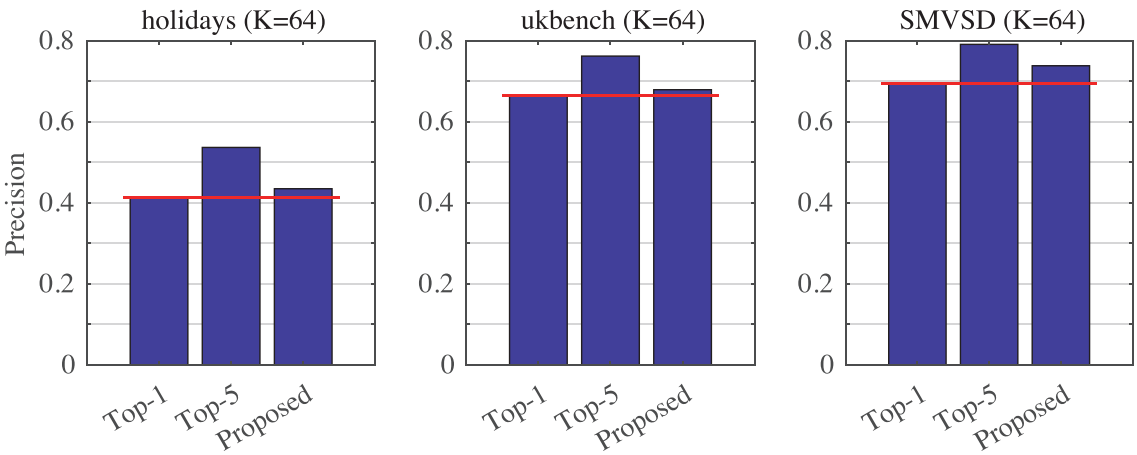

Fig. 3 Top-1 precision (naive), top-5 precision (probability that ranked list includes an appropriate reference image), and proposed (precision of *geometric verification*).

two aggregation vectors was computed without any approximations.

4.3.1 Efficiency

We evaluated the time it took to extract the local descriptors and their data size in an actual runtime environment. Here, the time needed to extract each descriptor a thousand times was measured and the average taken. The scores were measured on a 16-core 3GHz Xeon CPU, mutli-threaded, without any GPUs. Table 2 shows the results. In addition, Fig. 5 compares the total image retrieval times in the case of CARD and the descriptors. We refer to the results of Chen *et al*.'s system for the times of the search and geometric verification [8]. Their system takes 410 milliseconds for image retrieval, and the descriptor extraction takes 230 milliseconds on average. Our *local descriptor extraction* was faster than the other descriptors, excluding ORB. These results show that total image retrieval time in the case of using CARD or ORB were about twice as fast as SIFT.

4.3.2 Aggregation Performance

We evaluated the performance of our aggregation vector by using the mean average precision computed over the three datasets. The mean average precision was obtained using the ranked list ordered by the cosine distance, which was composed of candidate images in the database. Figure 4 shows the mean average precision of each descriptor for the three datasets. Our method gave better results than the others on all datasets; ORB and BOVW did not work as well.

4.3.3 Image Retrieval Performance

This evaluation investigated the overall performance of the image retrieval, including the *geometric verification*. In addition, we measured the actual size of the database for the *geometric verification* in order to confirm that our dual

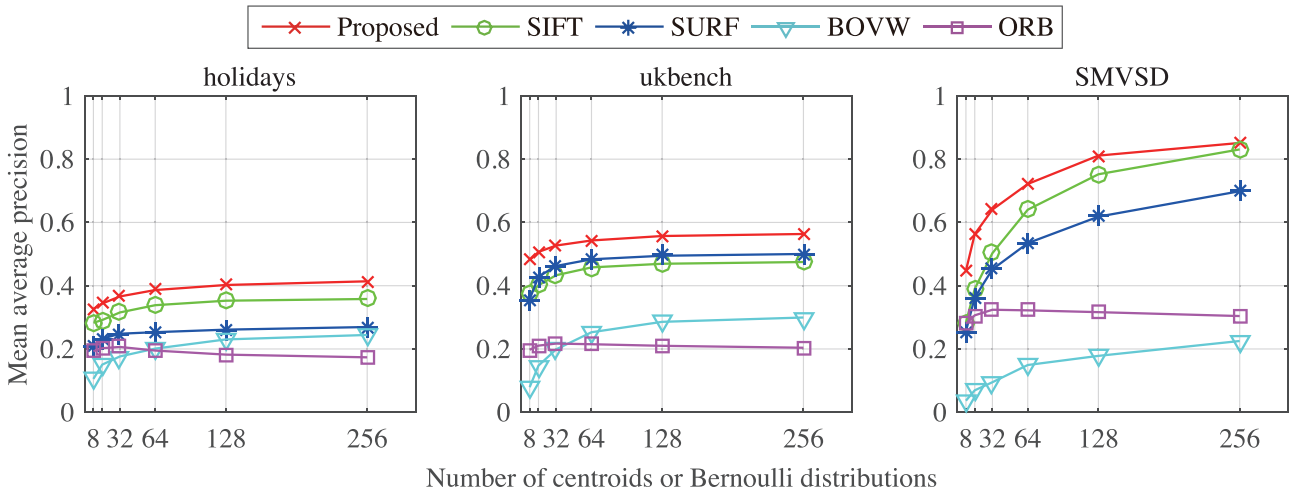

Fig. 4 Mean average precision over the three datasets.

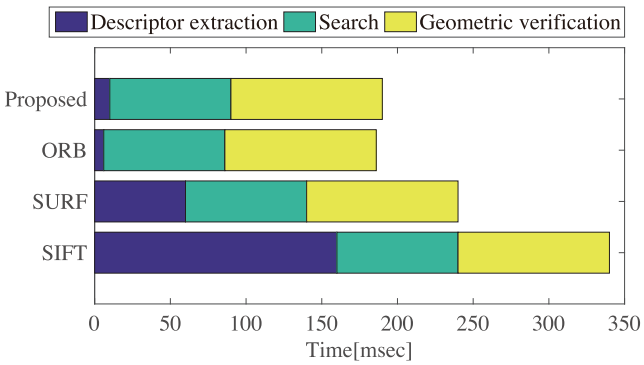

Fig. 5 Total image retrieval time in the case of CARD and descriptors. We referred to the results of Chen *et al.*'s system for the times of the search and geometric verification [8].

representation descriptor actually reduced the size of the database. We calculated the true positive rate and false positive rate while varying the *geometric verification threshold*. We let the size of the ranked list N be five in the *search* and *geometric verification*. We then drew the receiver operating characteristic (ROC) curves for each K and dataset in Fig. 6. Figure 7 shows the top-five precisions of these evaluations. Results for ORB are eliminated from Fig. 6 because its top-five precisions were almost 0% (Fig. 7). Our framework outperformed SIFT and SURF under certain conditions.

Figure 8 shows the actual size of the database for the *geometric verification* in the image retrieval experiments. Our method and ORB made the database more compact than SIFT and SURF did.

4.4 Discussion

The *geometric verification* of our framework improves the results of *search*. Figure 2 shows that our approach can reduce the false positive rate of image retrieval, but Fig. 3 shows that it only slightly improves precision. Thus, the *geometric verification* of our framework improves the reproducibility of the image retrieval rather than its precision.

Our CARD descriptor performed a little better than SIFT and SURF on all datasets as regards the mean average precision of *search* using aggregation vectors. Regarding the accuracy of the image retrieval, on the ukbench and SMVSD, CARD outperformed the other descriptors in the case of small K. The evaluation shows that CARD is better than the other descriptors. However, CARD is theoretically not as robust as SIFT because it approximates the orientation invariance by using a LUT. The results of the evaluation are due to the properties of the keypoint detector. SMVSD has the most planar objects, with ukbench and holidays having the second and third most. The Fisher vector and VLAD become less robust as fewer stable keypoints are extracted from an image. CARD's keypoint detector is a corner detector, so it can extract keypoints from images that include book covers or CD/DVD covers more stably than can SIFT using the BLOB detector. Such images have many clear corners, for example, around the characters indicating the content.

Let us point out why the ROC curves for ORB are not as good as those for BOVW, whereas the mean average precision of ORB is as good as that of BOVW. Our image retrieval performance evaluations required that the *descriptor aggregation* and *search* include a positive sample in the ranked list. Mean average precision will be higher even if none of the correct images are included in the top five. Figure 7 shows that the top-five precisions in the case of ORB are lower than in the other cases. Therefore, we consider that the aggregation vector with ORB is not suitable for the purpose of image retrieval even though ORB's mean average precision is better than that of BOVW.

5. Conclusion

We proposed an image retrieval framework based on the dual representation descriptor that extracts real-valued and binary descriptors simultaneously and quickly. The pro-

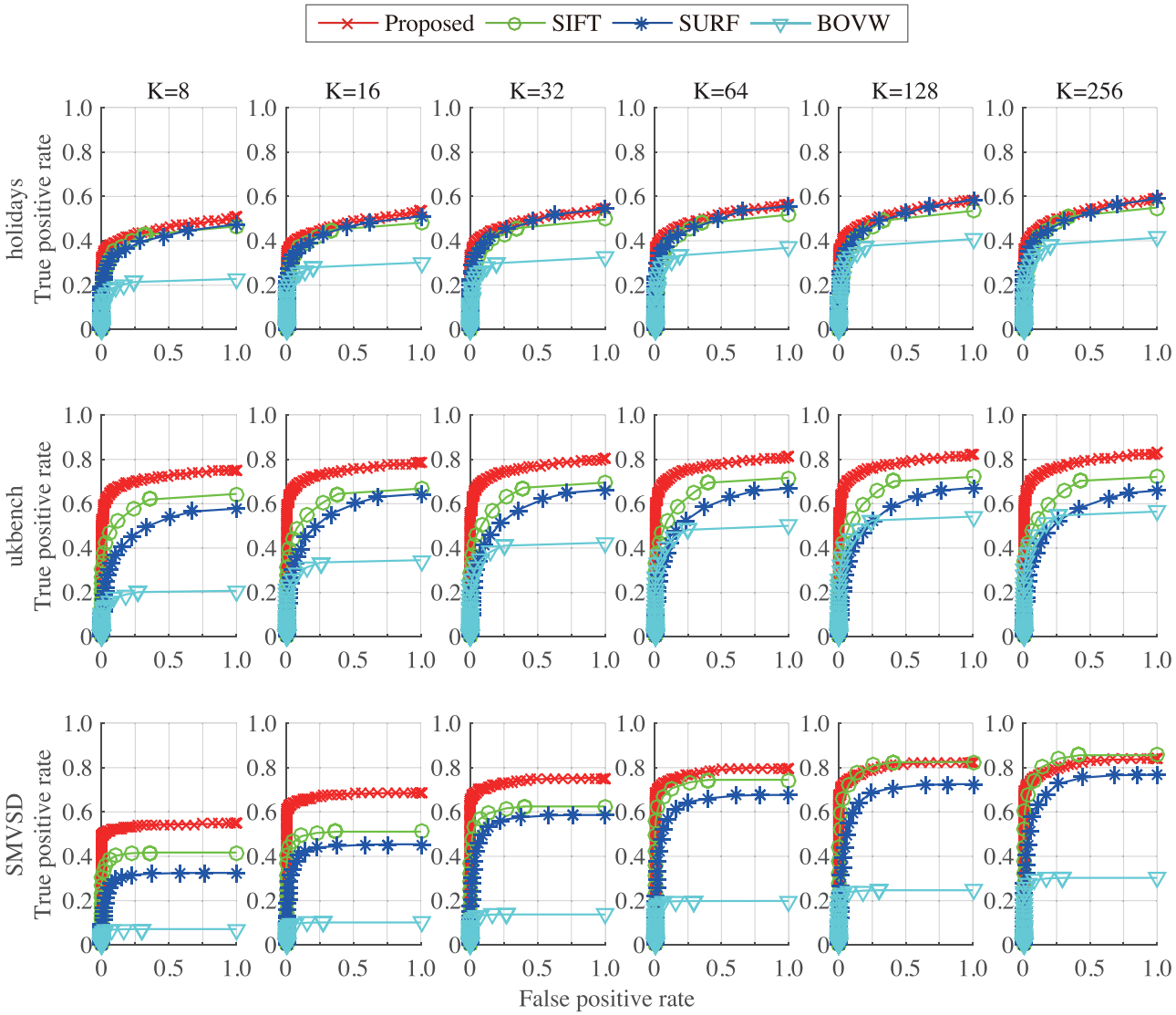

Fig. 6 Receiver operating characteristic (ROC) curves of true positive and false positives in image retrieval test on the three datasets.

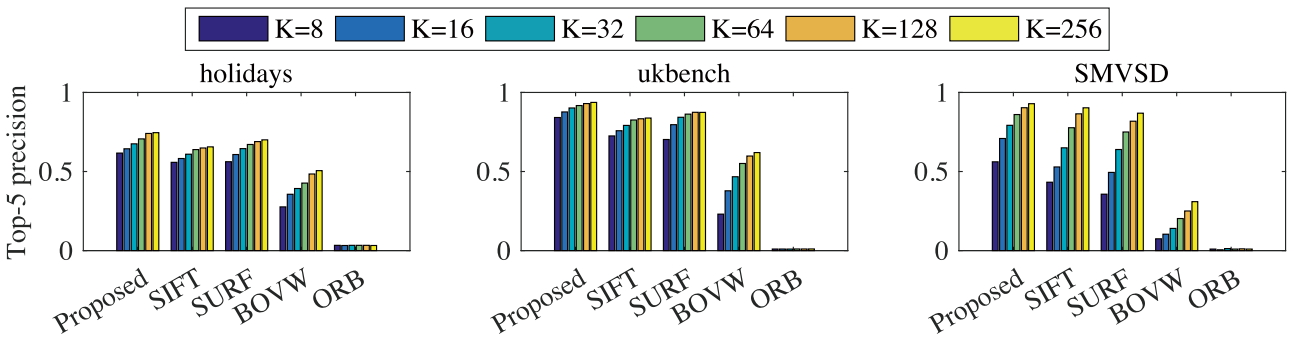

Fig. 7 Top-five precisions of image retrieval on the three datasets.

posed descriptor has the advantages of both. Our framework overcomes the trade-off between the retrieval speed, amount of storage, and accuracy of such frameworks. We implemented it using the CARD descriptor and evaluated it on three datasets. The experimental results show that the image retrieval of our framework is as fast as that of a framework using ORB and that it reduces the storage requirements to about 4% of SIFT. The mean average precision of the ag-

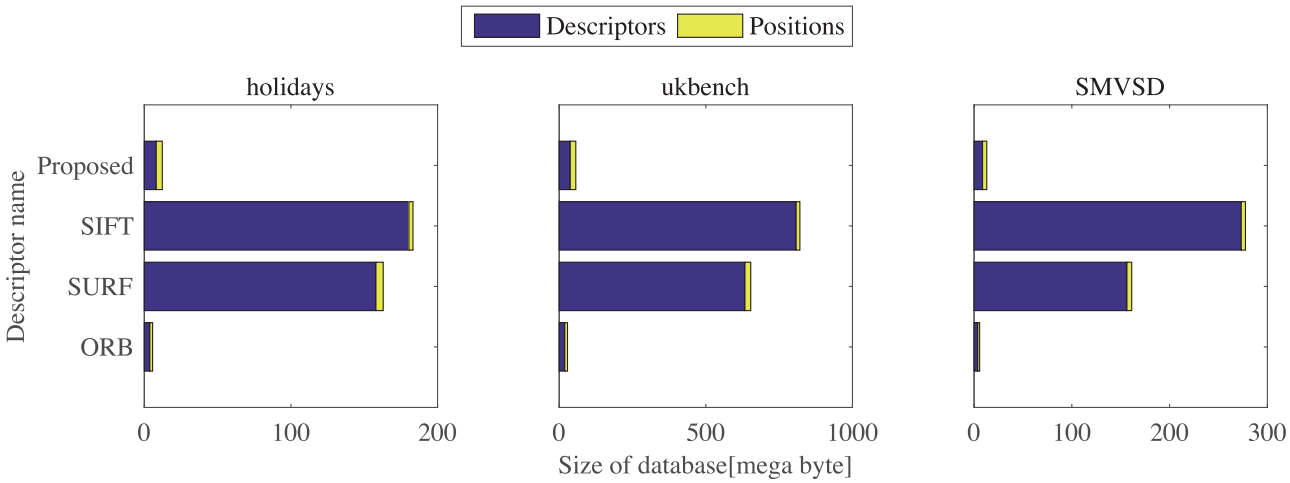

Fig. 8 Size of database for geometric verification in image retrieval on the three datasets. The database was composed of positions in an image and descriptors of all keypoints.

gregation vector and the accuracy of the total image retrieval are as good as using SIFT or SURF.

References

[1] H. Jégou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3304–3311, June 2010.

[2] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vision, vol.60, no.2, pp.91–110, Nov. 2004.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," IEEE International Conference on Computer Vision (ICCV), pp.2564–2571, 2011.

[4] S. Leutenegger, M. Chli, and R.Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," IEEE International Conference on Computer Vision (ICCV), pp.2548–2555, 2011.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," Proc. 11th European Conference on Computer Vision, pp.778–792, 2010.

[6] Y. Uchida and S. Sakazawa, "Image retrieval with Fisher vectors of binary features," 2013 2nd IAPR Asian Conference on Pattern Recognition (ACPR), pp.23–28, Nov. 2013.

[7] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," IEEE International Conference on Computer Vision (ICCV), pp.1470–1477, Oct. 2003.

[8] D.M. Chen and B. Girod, "Memory-efficient image databases for mobile visual search," IEEE MultiMedia, vol.21, no.1, pp.14–23, Jan. 2014.

[9] M. Ambai and Y. Yoshida, "CARD: Compact and real-time descriptors," IEEE International Conference on Computer Vision (ICCV), pp.97–104, Nov. 2011.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," Proc. 9th European Conference on Computer Vision, pp.404–417, 2006.

[11] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1–8, June 2007.

[12] M.Á. Carreira-Perpiñán and S Renals, "Practical identifiability of finite mixtures of multivariate Bernoulli distributions," Neural Comput., vol.12, no.1, pp.141–152, Jan. 2000.

[13] D. Gálvez-López and J.D. Tardós, "Real-time loop detection with bags of binary words," 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.51–58, Sept. 2011.

[14] V.R. Chandrasekhar, D.M. Chen, S.S. Tsai, N.-M. Cheung, H. Chen, G. Takacs, Y. Reznik, R. Vedantham, R. Grzeszczuk, J. Bach, and B. Girod, "The Stanford mobile visual search data set," Proc. Second Annual ACM Conference on Multimedia Systems, pp.117–122, 2011.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[16] Y. Matsui, T. Yamasaki, and K. Aizawa, "PQTable: Fast exact asymmetric distance neighbor search for product quantization using hash tables," IEEE International Conference on Computer Vision (ICCV), pp.1940–1948, Dec. 2015.

[17] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5153–5161, June 2015.

[18] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni, "Locality-sensitive hashing scheme based on P-stable distributions," Proc. Twentieth Annual Symposium on Computational Geometry, pp.253–262, 2004.

[19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.10, pp.1615–1630, Oct. 2005.

[20] R. Arandjelovic and A. Zisserman, "All about VLAD," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1578–1585, June 2013.

[21] 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.

[22] M. Muja and D.G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," IEEE Trans. Pattern Anal. Mach. Intell., vol.36, no.11, pp.2227–2240, 2014.

[23] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol.24, no.6, pp.381–395, June 1981.

[24] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," Proc. 10th European Conference on Computer Vision, pp.304–317, Oct. 2008.

[25] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2161–2168, June 2006.

Yuichi Yoshida received the B.S. and M.S. degrees in system engineering science from Osaka University in 2001 and 2003. From 2003 to 2007, he was a researcher at NTT Corporation. Currently, he is a Researcher in the research and development group at Denso IT Laboratory, Inc. Tokyo, Japan. His research interests include computer vision, machine learning, and mobile computing.

Tsuyoshi Toyofuku currently, works in the research and development group at Denso IT Laboratory, Inc. Tokyo, Japan.