H-026

Compact Binary Local Descriptor for Object Identification

Kota Iwamoto *Ryota Mase* *Toshiyuki Nomura*
Information and Media Research Labs., NEC Corporation

1. Introduction

Robust local descriptors, such as SIFT [1], GLOH [2], and SURF [3] descriptors, have been extensively used in computer vision applications due to their capability to identify objects or scenes in images with robustness against camera viewpoints and lighting conditions. Together with the recent proliferation of camera equipped mobile devices, these local descriptors have enabled mobile visual search services [4] of objects captured by a camera, such as search services of commercial products or landmarks for providing related information to the users. In these services, the captured images or the local descriptors extracted on the terminal are transmitted over a network to the server, and the search is carried out on the server side with a large database. Size of JPEG image or SIFT descriptors extracted from an image can be tens if not hundreds of kilobytes, which makes transmission over a network slow, becoming the bottleneck for causing response latency of the search service.

Recently, small size descriptors [5] have been developed for fast transmission of descriptors over a network. MPEG (Moving Picture Expert Group) has released a call for proposals [6] for standardizing such compact descriptors for mobile visual search applications. For a practical search service, it is essential for the descriptor size to be small, but at the same time object identification accuracy must not be lost.

In this paper, we propose a new compact binary local descriptor for object identification. We show that the proposed descriptor achieves an object identification accuracy that is comparable with SIFT descriptor, but with a much smaller size.

2. Proposed Compact Binary Local Descriptor

We propose a new scalable binary descriptor for compact representation of features around keypoints detected in an image. The proposed descriptor is extracted by binarizing histogram of quantized gradient orientations of a local patch centered at a keypoint, and then progressively selecting subset of bits for size reduction. The bit-length of the binary descriptor is progressively scalable, ranging anywhere from 32 to 150 bits. Descriptors of different bit-lengths can interoperate (can match) with each other, providing scalability and flexibility in the representation for wide range of applications.

Keypoint detection is carried out using a conventional scale-invariant keypoint detector. We use the DoG (Difference-of-Gaussian) detector used in SIFT [1], but it will also work with any other detector, such as the fast-Hessian detector used in SURF [2].

The extraction of the proposed binary descriptor from a detected keypoint is illustrated in Figure 1. It is carried out in the following steps; histogram creation and expansion, histogram binarization, and progressive bit selection.

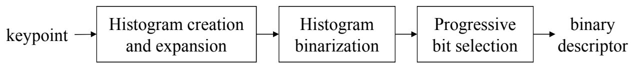

Figure 1: Extraction of the compact binary descriptor.

2.1 *Gradient histogram creation and expansion*

The binary descriptor is extracted from a local patch centered around the detected keypoint, where the patch is sized and rotated to the scale and the dominant orientation of the keypoint. The local patch is partitioned into 5×5=25 blocks, and for each block a histogram of gradient orientations quantized to 6 bins is created. This results in a histogram of 25×6=150 dimensions.

We then construct a hierarchical histogram of a larger block area by expanding the original 150 dimensional histogram, illustrated in Figure 2. Here we refer the original 5×5=25 blocks as layer 1. The histograms of each 2×2=4 blocks of the layer 1 are aggregated to form a new histogram of 4×4=16 blocks with 16×6=96 dimensions, here referred as layer 2. Furthermore, the histograms of each 2×2=4 blocks of layer 2 are aggregated to form a new histogram of 3×3=9 blocks with 9×6=54 dimensions, here referred as layer 3. This results in histogram of 150+96+54=300 dimensional feature vector. When constructing the hierarchical histogram of higher layers, we use the only the histogram elements (bins) that are not selected in the selection process described in the following section 2.3.

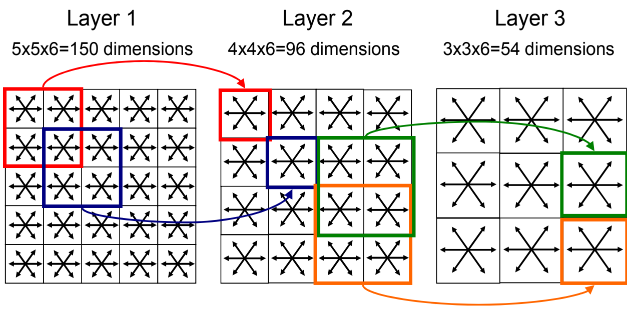

Figure 2: Creation and expansion of gradient histogram.

2.2 *Histogram binarization*

The 300 dimensional histogram is binarized by the following procedure. Each histogram element h_i is converted to a bit b_i based on the following equation, where i={0,1,2,3,4,5} denotes the 6 histogram elements corresponding to the quantized orientation in each block.

$$b_i = \begin{cases} 1 & (if\, h_i > th) \\ 0 & (if\, h_i \le th) \end{cases}, \qquad (1)$$

where the threshold th is determined as,

$$th = \alpha \sum_{i=0}^{5} h_i \, . \qquad (2)$$

Here α is a preset coefficient, which is set at different values for each layer. This binarization results in 150 bits for layer 1, 96 bits for layer 2, and 54 bits for layer 3.

2.3 *Progressive bit selection*

The bits are progressively selected to achieve progressively scalable binary descriptor, for size reduction of the descriptor. First, bits from layer 1 are selected up to 75 bits (out of 150), then bits from layer 2 are selected up to 48 bits (out of 96), then finally bits from layer 3 are selected up to 27 bits (out of 54), totaling 150 bits at the full size. Figure 3 illustrates the progressive bit selection of layer 1.

The selection order of bits is determined so that the correlation between the bits becomes small, by selecting the bits corresponding to different orientation for the adjacent blocks. 75 bits selected from layer 1 show that no bits corresponding to same orientation are selected for the adjacent blocks. The blocks selecting the even number orientations i={0,2,4} and the blocks selecting odd number orientations i={1,3,5} are located alternately. Also, the bits are progressively selected from all the blocks first, before selecting second bits from a block. Bit selections for layers 2 and 3 follow the same pattern. This pattern of bit selection achieves high accuracy even under reduced amount of bit-length. Any number of bits can be selected in this order, but practically larger than 32bits. This allows flexible adjustment for achieving the required feature size for an image.

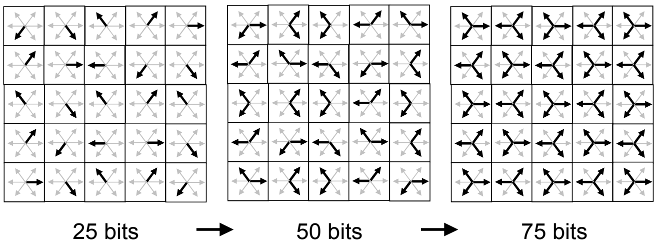

Figure 3: Progressive bit selection of layer 1.

3. Matching of Compact Binary Descriptors

Matching of compact binary descriptors between two images is carried out using a commonly used method of object matching. First, keypoint correspondences between the images are determined, by using a Hamming distance or a Jaccard score as a distance metric between the binary descriptors. Then, geometric verification using RANSAC is carried out to find a plausible geometric transformation (affine, homography, fundamental matrix) between the corresponding keypoints. If sufficient number of inliers is found, the two images are declared to be matching images depicting a same object.

4. Evaluation

Evaluation is carried out to test the performance of object identification of SIFT descriptor [1] using 1,024bits (=128bytes) and the proposed compact binary local descriptor using 96bits (=12bytes). We compare the performance the descriptors by pair-wise matching of images with objects captured from various camera viewpoints. The task is to determine whether the two images depict the same object or not. Table 1 summarizes the 3 dataset used for evaluation. Dataset (a) is composed of planar objects (CDs/books/ documents), (c) is composed of 3D objects and (b) is composed of both. The matching pairs are images which depict the same objects, and the non-matching pairs are images which depict different objects. The non-matching pairs are used to determine a threshold for achieving a false alarm rate of no more than 0.3%. Using that threshold, matching pairs are used to evaluate of the detection rate (true positive rate).

For evaluation, we use the same DoG-based keypoint detector used in [1] for both the SIFT descriptor and the proposed compact binary descriptor. For each image, we select up to 500 keypoints in the order of larger scales of DoG scale-space representation. This results in a total image feature size of 64Kbytes for SIFT descriptor, and 6Kbytes for the compact binary descriptor. For distance metrics, we used the L2 distance for SIFT, and the Jaccard score for the compact binary descriptor. We used the fundamental matrix in the geometric verification

Table 2 shows the results of the detection rate at a false alarm rate of 0.3%. They show that the proposed compact binary descriptor achieves a detection rate that is comparable with SIFT, but at a size smaller than 1/10 of SIFT.

Table 1: Evaluation dataset.

dataset	number of pairs
(a) graphics&texts	3000(matching), 30000(non-matching)
(b) common objects	2550(matching), 25500(non-matching)
(c) buildings	4005(matching), 48675(non-matching)

Table 2: Detection rate (at false alarm rate of 0.3%).

dataset	SIFT (128 bytes/keypoint)	Proposed (12bytes/keypoint)
(a) graphics&texts	96.9%	96.0%
(b) common objects	88.5%	85.5%
(c) buildings	82.7%	79.5%

5. Conclusion

We have proposed a compact binary local descriptor for object identification. The proposed binary descriptor is extracted by binarizing histogram of quantized gradient orientations of a local patch centered at a keypoint. Subset of bits is progressively selected from the binarized histogram forming a descriptor with a size of 32-150 bits. Experimental results using images with planar and 3D objects show that the proposed descriptor achieves object identification accuracy that is comparable with SIFT descriptor, but at a descriptor size smaller than 1/10 of SIFT.

References

[1] D. Lowe, “Distinctive image features from scale-invariant keypoints”, IJCV, 60, 2 (2004), 2004.

[2] K. Mikolajczyk *et. al.*, “A Performance Evaluation of Local Descriptors”, IEEE Trans. on PAMI, Vol.10, no.27, 2005.

[3] H. Bay *et. al.*, “SURF: Speeded Up Robust Features”, CVIU, Vol. 110, No. 3, 2008.

[4] B. Girod *et. al.*, “Mobile Visual Search”, Proc. Special Issue: Mobile Media Search, IEEE Signal Processing Magazine, 2011.

[5] V. Chandrasekhar, *et. al.*, “Compressed Histogram of Gradients: A Low Bitrate Descriptor”, IJCV, 2011.

[6] MPEG, "Call for Proposals for Compact Descriptors for Visual Search", ISO/IEC JTC1/SC29/WG11 N12201, 2011/07.