IEICE TRANS. INF. & SYST., VOL.E94–D, NO.6 JUNE 2011

PAPER

Hilbert Scan Based Bag-of-Features for Image Retrieval

Pengyi HAO[†a)], *Nonmember and* **Sei-ichiro KAMATA**[†b)], *Member*

SUMMARY Generally, two problems of bag-of-features in image retrieval are still considered unsolved: one is that spatial information about descriptors is not employed well, which affects the accuracy of retrieval; the other is that the trade-off between vocabulary size and good precision, which decides the storage and retrieval performance. In this paper, we propose a novel approach called Hilbert scan based bag-of-features (HS-BoF) for image retrieval. Firstly, Hilbert scan based tree representation (HSBT) is studied, which is built based on the local descriptors while spatial relationships are added into the nodes by a novel grouping rule, resulting of a tree structure for each image. Further, we give two ways of codebook production based on HSBT: multi-layer codebook and multi-size codebook. Owing to the properties of Hilbert scanning and the merits of our grouping method, sub-regions of the tree are not only flexible to the distribution of local patches but also have hierarchical relations. Extensive experiments on caltech-256, 13-scene and 1 million ImageNet images show that HS-BoF obtains higher accuracy with less memory usage.

key words: *image search, feature representation, bag-of-features, Hilbert scanning*

1. Introduction

Searching for similar images in a large unannotated image database is a challenging task not only because of the large memory but also due to the presence of occlusion, background clutter, viewpoint and lighting changes. For example, we may want to search some images containing a certain object (maybe different size or under different viewpoints), or depicting a similar scene as the input image. Many previous approaches have addressed the problem of matching such transformed images, like Refs. [1], [2]. They are in most cases based on local invariant descriptors, and matching descriptors using an efficient indexing structure [3]. Various approximate nearest neighbor search algorithms such as kd-tree [1] are also researched recently. The problem with these approaches is that all individual descriptors need to be compared to and stored.

In this context, bag-of-features (BoF) approach [4] which captures the invariance aspects of local keypoint features has recently attracted numerous research attentions. The basic idea of BoF is to depict each image as an orderless collection of local keypoint features [5]. BoF based search first extracts a set of local descriptors for each image, such as the popular SIFT descriptor [1]. These descriptors are very discriminant and invariant to local transformations. Furthermore, image comparison based on local description is robust to cropping, clutter, change in viewpoint, illumination change, etc. [6]. Then, the distribution of descriptors in descriptor space is quantized into visual words. An image can be described as a histogram of votes for visual words. Fast access to the frequency vectors is obtained by an inverted file [7] system.

The BoF approach, although is simple, it doesn't consider the spatial relationship among descriptors. Different strategies have been proposed to improve BoF. For instance, Ref. [8] used pyramid matching scheme to aggregate statistics of features over fixed subregions. This approach was worked by computing rough geometric correspondence on a global scale, which was an extension of an orderless BoF image representation for recognizing natural scene categories. However, fixed subregions are not strong enough to preserve the coherence and compactness of local features which is very useful for extract the spatial information among these descriptors. Reference [9] exploited spatial relations between features using full segmentation masks, but it did not support a large number of categories. Reference [10] proposed an ordered BoF to encode geometric information of objects within an image by projecting local features to different directions and then selecting the most representative ones, which is a new class of bag-of-features for representing images.

In our research, we focus on orderless bag-of-features approach which can do retrieval quickly yet the search accuracy is not high. For BoF approach, if a little higher accuracy was obtained, much larger memory would be needed. Here, we want to explore spatial relationships among objects in an image automatically, without any label or manual handling, then add these information into bag-of-features to get higher accuracy with less memory usage. Hilbert space filling curve has the property to preserve the locality between objects of multidimensional space in the linear space. If the distance between two points in the two-dimensional image is small, the distance between the same pair of points in the one-dimensional sequence is also small in most cases. So a novel approach called Hilbert scan based bag-of-features (HS-BoF) is proposed in this paper. Firstly, Hilbert scan based tree representation (HSBT) is studied, which is built based on the local descriptors. Spatial relationships are added into nodes by a new grouping rule, resulting of a tree structure for each image. Further, vi-

Manuscript received September 6, 2010.
Manuscript revised January 28, 2011.

[†]The authors are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu-shi, 808–0135 Japan.
a) E-mail: haopy@toki.waseda.jp
b) E-mail: kam@waseda.jp

DOI: 10.1587/transinf.E94.D.1260

sual words with significant spatial information are generated. We have two kinds of codebook production methods, one is multi-layer codebook which forms a hierarchically codebook according to the tree structure of HSBT and assigns descriptors to visual words in a multi-layer way, the other is multi-size codebook which gets the vocabularies using all the nodes in the tree once. Since the regions obtained using Hilbert curve are irregular, adaptable and flexible subregions are shaped according to the coherence of interest points, which benefits the irregular objects in images as much as possible. In addition, the spatial relations between descriptors are also exploited by boosting the features of objects and suppressing background features adaptively, which is not limited to the categories of objects.

The rest of paper is organized in the following way: first, we introduce Hilbert scanning in Sect. 2, then, the proposed Hilbert scan based representation (HSBT) is described in Sect. 3, Hilbert scan based bag-of-features (HSBoF) is presented in Sect. 4, Sect. 5 gives dataset, evaluation metrics, experiments and analysis, finally, conclusions and future work are given in the last section.

2. Hilbert Scan

In 1891, David Hilbert first described a continuous fractal space filling curve named Hilbert curve which passed through each point of the unit square exactly once and never crossing itself. In the application of data analysis, it is used for scanning data in two-dimensional space. This scanning way is called Hilbert scan. It is a one-to-one mapping from two-dimensional space to one-dimensional space and has been demonstrated that if a square corresponds to an interval, its sub-squares correspond to the sub-intervals of that interval. Because it can preserve point neighborhoods as much as possible, Hilbert scan has been widely applied in image processing tasks. Reference [12] gave an interactive method for classify multi-spectral images using a Hilbert curve, which can be performed easily instead of using N-dimensional data directly. Reference [13] proposed a compression approach for color image compression using two-dimensional Hilbert curve.

As a mapping technology, Hilbert curve has the property to preserve the locality between objects of multidimensional space in the linear space. This property was called clustering in some literature, such as Ref. [14]. The clustering properties makes it useful in computer science. If we take a curve like the one shown in Fig. 1 and straighten it out, points that are close together in the two-dimensional space will also tend to be close together in the linear sequence. Based on the clustering merit, Ref. [15] introduced a shape representation method using the combination of the Hilbert space filling curve and Wavelet analysis.

Currently, there existed several algorithms for the two-dimensional Hilbert scan like Refs. [11], [16]. However, these algorithms had more or less restrictions on their applications, such as complex, requiring square-sized image. In order to improve the Hilbert scan for general application,

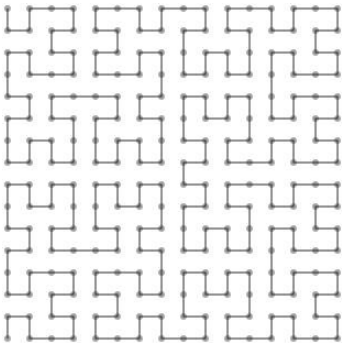


Fig. 1 A 16×16 Hilbert curve in the 2D space.

a Pseudo-Hilbert scan algorithm for arbitrarily-sized arrays was given in Ref. [17]. It is a non-recursive algorithm based on two look-up tables and the size of a scanned rectangle is arbitrary. The more important merit is that it holds the point neighborhoods as well. Therefore, it is useful for our task.

3. Hilbert Scan Based Tree Representation (HSBT)

3.1 Preparation

Let $m_1 \times m_2$ be the resolution of image I, m_1 can be not same with m_2. After extracting keypoints using the DoG detector and described by the SIFT descriptor, image I is mapped from 2D space to 1D sequence by Pseudo-Hilbert scan algorithm for arbitrarily-sized arrays. To simplify the notations, we use $\Theta = \{S, L, F, R\}$ to represent the 1D sequence, where S is the set of keypoints obtained from I, L is the set of keypoints' locations in image I, F is the set of keypoints' features, R is the set of segments produced by partitioning Θ. Here, we call each segment a region. Hilbert scan based tree will be built for image I with several groupings based on keypoints' distribution and features. Here, one grouping means incorporating or merging all the rest regions in one set into main regions. The explanations about rest region and main region will be given in the next section.

The structure of each region in each grouping is consisted by two components: mark and data. For the region R_j in the i-th grouping, it is marked by R^i_j. It has three data: the number of keypoints in this region (n^i_j), the center of gravity of keypoints in this region (g^i_j), and the clustering center of keypoints in this region (b^i_j). g^i_j is calculated based on the location set L^i_j, b^i_j is calculated using fuzzy C-means clustering on the feature set F^i_j.

3.2 Grouping Based on Hilbert Scanning

The set of regions in the i-th grouping is R^i, $R^i = \{R^i_j | j = 1, 2, \cdots, \nu(i)\}$, where $\nu(i)$ is the regions' number of i-th grouping. Let the data set of regions in the i-th grouping are N^i, G^i and B^i. $N^i = \{n^i_j | j = 1, 2, \cdots, \nu(i)\}$, $G^i = \{g^i_j | j = 1, 2, \cdots, \nu(i)\}$, $B^i = \{b^i_j | j = 1, 2, \cdots, \nu(i)\}$. The steps are listed as follows:

1. Initialization. Let $i = 1$, initialize R^i, N^i, G^i, B^i;

2. Select main regions. Select main regions from R^i. It means R^i is divided into R^i_{main} and R^i_{rest}, $\nu(i) = \nu^i_{main} + \nu^i_{rest}$, where ν^i_{main}, ν^i_{rest} are the number of regions in R^i_{main} and R^i_{rest} respectively;

3. Merging regions. If $\nu^i_{rest} \neq 0$, merge the regions in R^i_{rest} to R^i_{main}; $i = i + 1$; Calculate the N^i, G^i and B^i of R^i; go to 2;

4: If $\nu^i_{rest} = 0$, stop the grouping.

The process that all the regions in R^i_{rest} are merged into R^i_{main} which is first selected from R^i is called i-th grouping.

Initialization: In the step of initialization, an interval δ is needed to partition Θ into $\nu(1)$ regions at first, which means image I is divided into $\nu(1)$ irregular parts. Here, $\nu(1) = (m_1 \times m_2)/\delta$. Figure 6 (a), (e), (f) give some examples. Let $R^1 = \{R^1_1, R^1_2, \cdots, R^1_{\nu(1)}\}$, if the region R^1_j ($j = 1, 2, \cdots, \nu(1)$) has no keypoint, it will be filtered. After filtering the blank regions, R^1 is taken as the initial region set. Correspondingly, N^1, G^1, B^1 can be got based on R^1.

Main region selection: In each grouping, main regions are needed to be selected at first. The selecting process in the i-th grouping is described as follows:

Sptep1: Sort the regions in the set R^i. After sorting, $R^i = \{R^i_j | j = 1, 2, \cdots, \nu(i)\}$ is changed as $\{R^i_{\varphi(j)} | \varphi(j) \in [1, 2, \cdots, \nu(i)]\}$; For the sorted set, $n^i_{\varphi(1)} \geq n^i_{\varphi(2)} \geq \cdots \geq n^i_{\varphi(\nu(i))}$;

Step2: If $\sum_{j=1}^{\varphi(s)} n^i_{\varphi(j)} > Th \times M$ and $\sum_{j=1}^{\varphi(s-1)} n^i_{\varphi(j)} < Th \times M$, Where, M is the total number of keypoints in the image, Th is a threshold, $0 < Th < 1$; then $R^i_{main} = \{R^i_{\varphi(1)}, R^i_{\varphi(2)}, \cdots, R^i_{\varphi(s)}\}$, $1 \leq \varphi(s) \leq \nu(i)$;

Step3: $\nu(i + 1) = \varphi(s)$; $R^i_{rest} = R^i - R^i_{main}$.

From the selecting process, it can be seen that the number of regions in the next grouping equals to the number of main regions in the current grouping, and the number of main regions decreases with the increase of groupings.

Merge rest regions: After selecting main regions in the i-th grouping, it is needed to judge whether there are remaining regions or not. If there are regions in R^i_{rest}, they will be merged into main regions.

Assume that we are given three adjacent regions in the i-th grouping on Θ: R^i_x, R^i_y, R^i_z, Where $R^i_y \in R^i_{rest}$, $R^i_x \in R^i_{main}$, $R^i_z \in R^i_{main}$, $x < y < z$. Now a question is coming that R^i_y will be merged into R^i_x or R^i_z. In order to solve this problem, a merging rule is given in Eq. (1),

$$\frac{n^i_x}{|g^i_x - g^i_y|} > \frac{n^i_z}{|g^i_z - g^i_y|}. \tag{1}$$

It means that if the region R^i_x has larger gravitation, R^i_y will be merged into R^i_x, otherwise, it will be merged into R^i_z. The merging process is marked as $R^i_y \rightarrow R^i_x$ or $R^i_y \rightarrow R^i_z$. Note that if R^i_x and R^i_z do not exist, R^i_y will be filtered.

Here, an example about how to do grouping is given in Fig. 2. It includes four groupings. The main regions in the first three groupings are $R^1_{main} = \{R^1_1, R^1_3, R^1_6, R^1_8, R^1_{10}\}$, $R^2_{main} = \{R^2_3, R^2_8, R^2_{10}\}$ and $R^3_{main} = \{R^3_8\}$ respectively. Based on the merging rule, the other regions in each grouping will be merged into the main regions. For instance, in the second grouping, $R^2 = \{R^2_1, R^2_3, R^2_6, R^2_8, R^2_{10}\}$, R^2_3, R^2_8, R^2_{10} are main regions. Based on the merging rule, R^2_1 will be merged into R^2_3, R^2_6 will be merged into R^2_8. So, in the third grouping, there are three regions in the region set: $R^3 = \{R^3_3, R^3_8, R^3_{10}\}$.

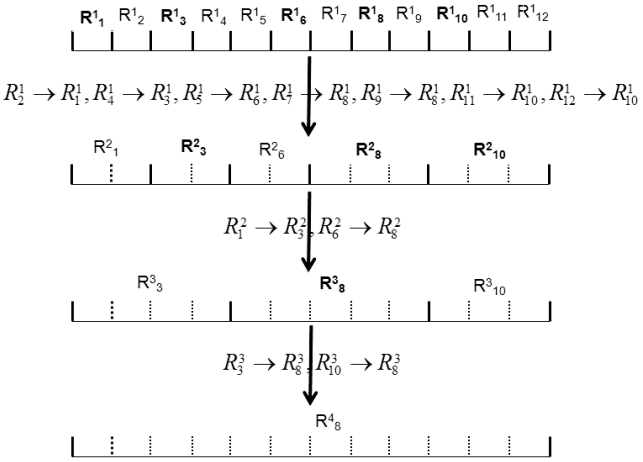


Fig. 2 An example of grouping.

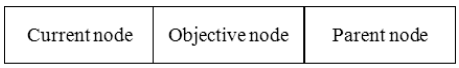


Fig. 3 Node structure.

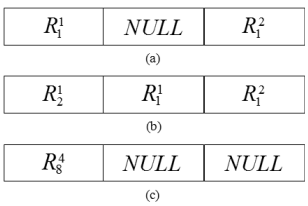


Fig. 4 Examples of node structure. (a) R^1_1, (b) R^1_2, (c) R^4_8.

3.3 Building a Tree for an Image

In fact, the process of grouping introduced above can be expressed by a tree. The node of tree is defined as Fig. 3. It consists of three parts: current node, objective node and parent node. Here, current node is one region in the current grouping, objective node is the region that current one will be merged into, parent node is one region in the next grouping. If the region is a main region and has no objective node, its objective node is "NULL". Specially, the regions in the last grouping have no objective node and parent node. For the regions R^1_1, R^1_2 and R^4_8 shown in Fig. 2, R^1_1 is a main region in the first grouping, R^1_2 is not a main region and will be merged into R^1_1 ($R^1_2 \rightarrow R^1_1$), R^4_8 is the region in the last grouping. The node structures of these regions are illustrated in Fig. 4 (a), (b), (c) respectively.

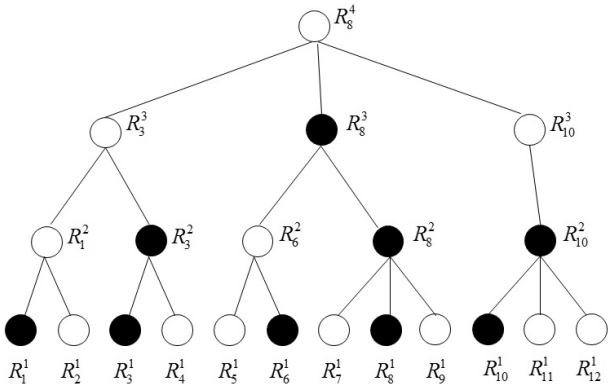


Fig. 5 An example of reverse tree built based on grouping.

Based on the definition of node, we can express the grouping process by a tree, where lower levels have more local information and higher levels contain more global information. Figure 5 gives an example which is the tree of the groupings in Fig. 2, where the black nodes are main regions in each grouping.

4. Hilbert Scan Based Bag-of-Features (HS-BoF)

Local descriptors extracted by SIFT are represented by HSBT, then they are quantized into visual words. The details of codebook generation, frequency vectors production and distance calculation will be given in this section.

4.1 Image Representation by HSBT

In Sect. 3, the details about the construction of HSBT have been presented. Here, we give some examples (see Fig. 6) and briefly explain the role it will play in bag-of-features based search.

Here, in order to show the regions produced by proposed HSBT clearly, they are painted by 20 colors (actually the colors have no relations with our experiments). Figure 6 (a) is the state before grouping. They are produced by $\delta = 500$. Figure 6 (e), (f) show the amplified results of the first and third images in Fig. 6 (a). It can be observed that the original images are divided into lots of regions which are irregular with each other. In fact, the contents in an image such as bird, building, people, etc., also have irregular shapes. Figure 6 (b) and Fig. 6 (c) show the first and second grouping results of HSBT of left images. Figure 6 (d) is the final grouping results of the given images. It can be seen clearly that the main objects are preserved and other less important things are wiped out. This is very beneficial to our work. Reference [9] used full segmentation masks to boost the weights of objects and suppress the weights of background features. Our approach boosts the features of objects adaptively, which is not limited to the categories of objects. In addition, since Hilbert scan preserves the correlation of interest points in an image, the one-dimensional sequence is divided into several parts according to the coherence of points. So the regions which have correlation

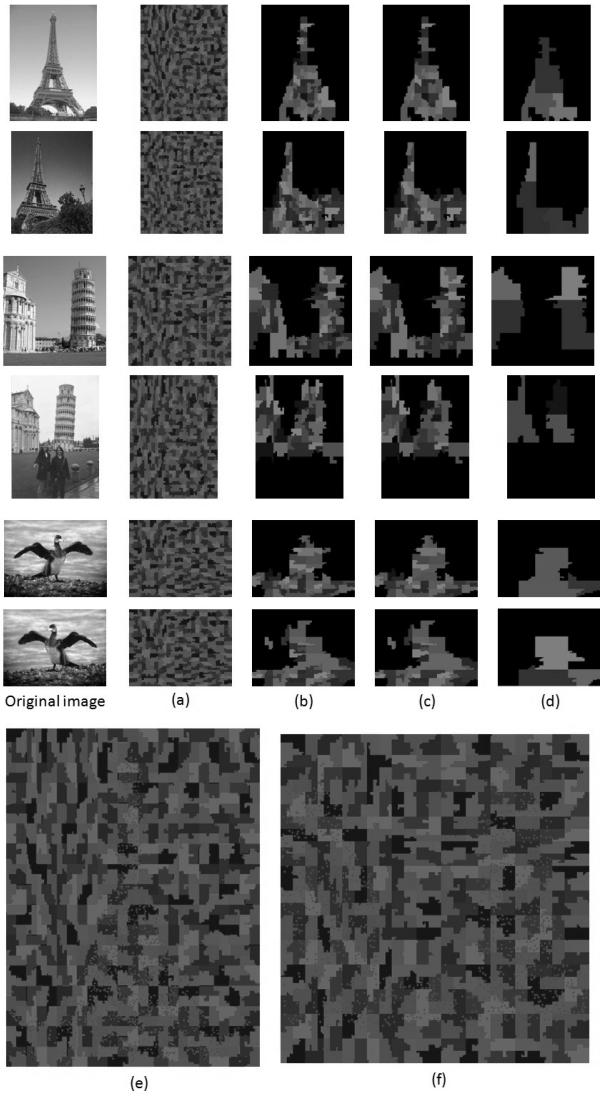


Fig. 6 Examples of grouping in HSBT.

with each other in image are connected. In Ref. [8], the spatial information is explored by dividing image into four subregions with same size and repeating this process several times. Here, the subregions are formed by their own characters and the upper subregions (upper nodes in the tree) are shaped by the lower subregions (lower nodes in the tree). They contain the relationships between layers.

4.2 Codebook Generation

Assuming there are X images in the training set, after HSBT representation, each image has a feature set $B_a = \{B_a^1, B_a^2, \ldots, B_a^i, \ldots, B_a^{h_a}\}, a = 1, 2, \ldots, X$, where h_a is the tree' height of image a. In our experiments, each region in the grouping has one clustering center of fuzzy C-means. Then K-means algorithm is used to produce a codebook. Clearly, the input of K-means is the feature set $F' = \{B_a | a = 1, 2, \ldots, X\}$ instead of F (keypoints' features). The size of codebook is determined by the number of K. A small K

may lack the discriminative power since two interest points may be assigned into the same cluster even if they are not similar to each other. A large K, on the other hand, is less generalizable, less forgiving to noises, and incurs extra processing overhead. In order to get the trade-off between discrimination and generalization, two methods for generating codebook will be given at here.

One way of codebook production is using the tree structure of HSBT representation. One layer of images $B^i = \{B^i_1, B^i_2, \ldots, B^i_X\}$ will be quantized to one set of visual words c_i, $i = 1, 2, \ldots, n$, where n is the max height of Hilbert scan based trees in the training set, $n = \max\{h_1, h_2, \ldots, h_X\}$. This codebook is produced from layer 1 to layer n, so, we call it multi-layer codebook (ML-codebook). Because the number of regions is decreasing with the growth of layers, the vocabulary sizes of lower layers are larger than upper layers. Assuming the numbers of clusters from layer 1 to layer n are $k_1, k_2, \ldots, k_n$, $k_1 \geq k_2 >, \ldots, \geq k_n$. We can get the multi-layer codebook: $\{c_1, c_2, \ldots, c_n\}$, where $c_i = \{c_{i,1}, c_{i,2}, \ldots, c_{i,k_i}\}, i = 1, 2, \ldots, n$. The vocabulary size of ML-codebook is $k_L = k_1 + k_2 +, \ldots, +k_n$. In Sect. 5, the choice of parameter n and the vocabulary size in each layer will be given in details. Figure 7 shows this process.

The other way is using all the features in F' once to produce a codebook. We call it multi-size codebook (MS-codebook). Figure 8 shows this process. In order to distinguish from ML-codebook, we use c' to label the clustering center of K-means. Assuming that there are k_S clusters, we can get the multi-size codebook as follows: $\{c'_1, c'_2, \ldots, c'_{k_S}\}$.

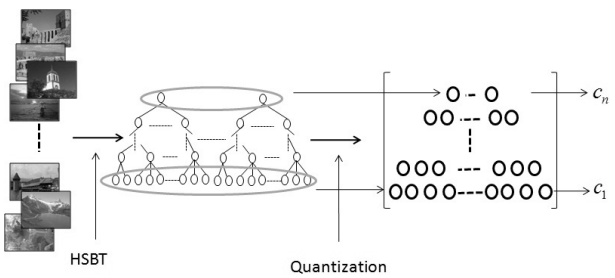


Fig. 7 The formation of multi-layer visual words.

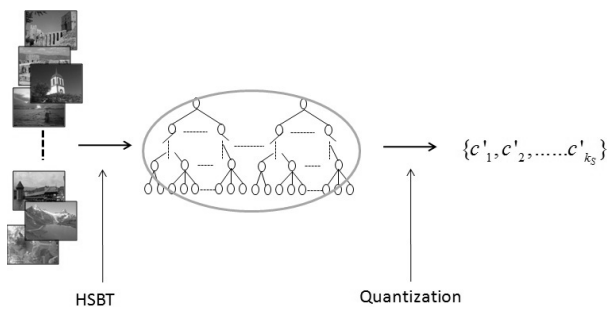


Fig. 8 The formation of multi-size visual words.

4.3 Assigning the Descriptors to Visual Words

Because ML-codebook is more complex than MS-codebook, here the details of assigning the descriptors to visual words according to ML-codebook are listed. Assuming that the descriptor set of image x after HSBT representation is B_x. Each descriptor b^j_i in B_x is assigned to the nearest visual word. Figure 9 shows this process clearly.

Step1: For an image, all the frequencies $f_{i,j}$ of assigning its descriptors to visual words are initialized to 0, where $j = 1, 2, \ldots, k_1$ for $i = 1$; $j = 1, 2, \ldots, k_2$ for $i = 2$; $\ldots$; $j = 1, 2, \ldots, k_n$ for $i = n$.

Step2: For each descriptor b^j_i and each centroid y^h_i of the i-th layer of the codebook, increase the frequency by $f_{i,j} = f_{i,j} + f_q(b^j_i, y^h_i)$, where $f_q(.,.)$ is a matching function that reflects the similarity between descriptors b^j_i and y^h_i. Here, $f_q(.,.)$ is defined as:

$$f_q(b^j_i, y^h_i) = \begin{cases} 1, \text{if } b^j_i \text{ is a k-NN of } y^h_i \\ 0, \qquad\qquad otherwise \end{cases}. \tag{2}$$

Step3: The histogram of visual words occurrences is normalized with the L_1 norm, generating frequency vectors of n layers: $\{f_1, f_2, \ldots, f_n\} = \{(f_{1,1}, f_{1,2}, \ldots, f_{1,k_1}), (f_{2,1}, f_{2,2}, \ldots, f_{2,k_2}), \ldots, (f_{n,1}, f_{n,2}, \ldots, f_{n,k_n})\}$.

Similarly, for MS-codebook, we can get $\{f'_1, f'_2, \ldots, f'_{k_S}\}$.

4.4 Weighting Frequency Vectors

Like the text retrieval, each term (image) in the database is represented by a vector of word frequencies. However, it is usual to apply a weighting to the components of this vector, rather than use the frequency vector directly for indexing. A fundamental difference with text retrieval is that text words are sampled naturally according to language context but visual words are the outcomes of data clustering. The former carries semantic sense, while the latter infers statistical information. Here, the components of the frequency vector are weighted using the strategy given in Ref. [3]. Denoting by η the number of images in the database and by η_j the number of images containing the j-th visual word, the weighting

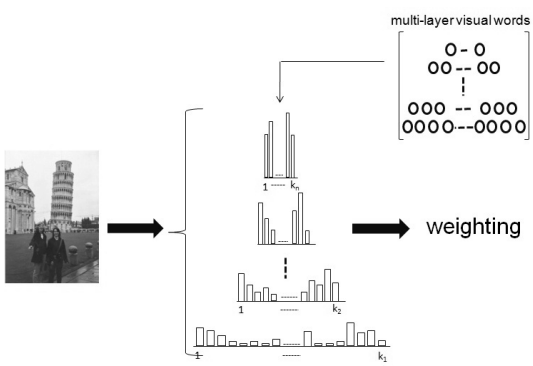


Fig. 9 An example of frequency vector generation using ML-codebook.

component $w_{i,j}$ associated with the i-th layer of the image is given by

$$w_{i,j} = f_{i,j} log \frac{\eta}{\eta_j} \tag{3}$$

The visual word frequency vector got by ML-codebook is: $\{(w_{1,1}, w_{1,2}, \ldots, w_{1,k_1}), (w_{2,1}, w_{2,2}, \ldots, w_{2,k_2}), \ldots, (w_{n,1}, w_{n,2}, \ldots, w_{n,k_n})\}$; the visual word frequency vector got by MS-codebook is: $\{w'_1, w'_2, \ldots, w'_{k_S}\}$. These vectors are a compact representation of the image.

4.5 Distance Calculation

We use two kinds of methods to calculate distances. One is the cosine of angle used in Ref. [4], the other is CHI-Square. For the vectors generated based on MS-codebook, the distance calculation is simple. Here, we just talk about the distance calculation between vectors got by ML-codebook. Given the visual word vector $w_q = \{q_1, q_2, \ldots, q_{n_1}\}$ of a query, $q_i = \{q_{i,1}, q_{i,2}, \ldots, q_{i,k_i}\}, i = 1, \ldots, n_1$, similar images in the database are represented by vectors $w_v = \{v_1, v_2, \ldots, v_{n_2}\}$, $v_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,k_i}\}, i = 1, \ldots, n_2$. Here, n_1 and n_2 are the max numbers of layers for w_q, w_v respectively. The distances calculated by cosine of angle and CHI-Square are defined as follows:

$$d_{cosine}(w_q, w_v) = \sum_{i=1}^{\min(n_1,n_2)} \frac{q_i \bullet v_i}{|q_i||v_i|}, \tag{4}$$

$$d_{CHI}(w_q, w_v) = \sum_{i=1}^{\min(n_1,n_2)} \sum_{j=1}^{k_i} \frac{(q_{i,j} - v_{i,j})^2}{q_{i,j} + v_{i,j}}. \tag{5}$$

5. Experiments

Datasets We evaluate our approach on an object dataset (Caltech-256), a scene dataset (13-scene categories) and also a large scale dataset (1066940 images from ImageNet, we call it imageNet1M).

Caltech-256 was created by the California institute of technology in 2007. Each object category contains between 80 and 827 images. It represents a diverse set of lighting conditions, poses, backgrounds, image sizes and camera systematics. The categories represent a wide varity of natural and artificial objects in various settings. 13-scenes dataset is one of the most common scene database used in the literature so far. Each category contains 200 to 400 images and ranges from natural scenes to man-made environments. ImageNet1M is provided for test the performance of the proposed approach on large scale collections. The images were crawled from about 1000 popular synsets in ImageNet.

Evaluation criteria In order to evaluate the proposed method, we have used two standard evaluation measures, namely the mean average precision (mAP) and the recall at particular ranks (Recall@R).

mAP: For each query image we obtain average precision computed as the area under the precision-recall curve. Precision is the number of retrieved positive images relative to the total number of images retrieved. Recall is the number of retrieved positive images relative to the total number of positives in the corpus. The mAP is then the mean for a set of queries. Recall@R: Measuring the recall at a particular rank R, i.e., the ratio of relevant images ranked in top R positions.

5.1 Object Dataset

For obtaining codebook, 12800 examples were randomly chosen from all categories (50 images from 256 categories). Then we randomly got 5 images from the rest of each category as queries. Figure 10 illustrates the results with different numbers of layers in ML-codebook. Here, in the formation of HSBT, $\delta = 500$ for dividing the 1D sequences to segments, $Th = 0.8$ for selecting main regions. Clearly, more layers bring better precision. This is because that there are much global information with the growth of the layers. However, there is only a slight increase from fifteen layers to twenty layers. This is not surprising since that fewer frequent visual words are generated with the increase of layers. In this case, the information added into the codebook is not very rich. On the other hand, the complexity of computation and the memory usage increase with the growth of layers in codebook. So, $n = 10$ will be used for the next experiments on caltech-256 dataset.

Table 1 compares the proposed HS-BoF with other methods under different vocabulary sizes (10k, 20k, 50k, 100k) in terms of mAP, the two different distance calculation methods are also compared. In order to be fair, same K-means method was also used to do quantization for BoF

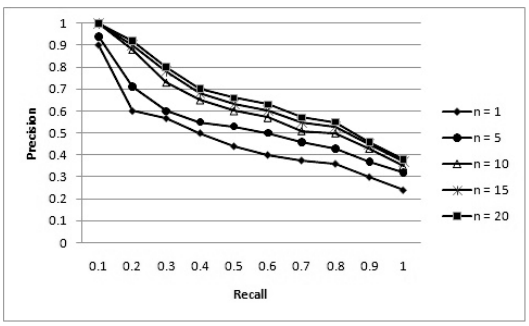


Fig. 10 Results on different layers of codebook.

Table 1 Comparison of mAP on caltech-256 dataset.

Word size	Distance	BoF	SPM	HS-BoF (MS)	HS-BoF (ML)
10k	Cosine	0.400	0.373	0.385	0.436
	CHI-Square	0.438	0.391	0.491	0.605
20k	Cosine	0.490	0.411	0.492	0.523
	CHI-Square	0.541	0.437	0.588	0.667
50k	Cosine	0.515	0.446	0.514	0.531
	CHI-Square	0.573	0.472	0.595	0.669
100k	Cosine	0.527	0.469	0.498	0.507
	CHI-Square	0.604	0.499	0.570	0.622

and spatial pyramid matching (SPM) in our implementation. In this table, four methods are compared, which are traditional bag-of-features (BoF), one BoF model with spatial information (SPM), the proposed Hilbert scan based bag-of-features (HS-BoF) with ML-codebook and MS-codebook respectively. Here, $n = 10$ for ML-codebook, two levels in SPM. Note that, for ML-codebook, the vocabulary size is the sum of visual words of all the layers (the vocabularies of the first layer are 5k, 10k, 20k, 50k respectively in our experiments). First, one can observe that CHI-Square achieves better results than the Cosine of vectors. Note that most of researches rank the images at the retrieval stage by their normalized scalar product (cosine of angle) between the query vector and all image vectors in the database. Second, SPM has a worse performance than BoF on this dataset, maybe it is because that SPM was particularly designed for natural scene categorization, the horizontal and vertical divisions in spatial space are improper for object images. Third, HS-BoF results in significant improvements, which shows the effectiveness of the spatial information in HS-BoF. Finally, in spite of the better performances, mAPs of HS-BoF decrease slightly using 100k vocabularies. The reason is that, for codebook generation, fuzzy c-means' clustering features of regions are used instead of the original keypoints' features. When using 100k vocabularies, the number of visual words is almost equal to the number of descriptors, in this case, it becomes very similar to an approach which matches individual descriptors, therefore, quantified features are less discriminative, which affects the precision of capturing correct spatial information.

Figure 11 shows the rate of relevant images found in the top R images. Here, CHI-Square was used. The max layer was teen for ML-codebook. The vocabulary size of the first layer was $5k$. In Fig. 11, it can been seen that HS-BoF with ML-codebook gets most positive images at the same number of top images comparing with HS-BoF (MS-codebook) and BoF. It increases the recall by 14% in comparison with standard BoF when the number of top images is 50.

5.2 Scene Dataset

650 examples were randomly chosen from all categories (50 images from each category) to get codebook. 260 queries from the rest images (20 images from each category) were used to test. For this dataset, we also compared the performance of HS-BoF with ML-codebook under different layers at first. We found that 15 layers was the best choice, so in the next experiments $n = 15$ for ML-codebook. CHI-square was also used. Figure 12 gives the results of several methods under different vocabulary sizes (500, 1k, 2k, 5k, 10k, 20k). One can observe that HS-BoF with ML-codebook outperforms others. HS-BoF with MS-codebook is comparable with SPM using small vocabularies such as 500, 1k and 2k, a little better than SPM using 5k and 10k. When using a large vocabulary such as 20k, SPM performs a little better than HS-BoF with MS-codebook, but still lower than HS-BoF with ML-codebook. For the slightly decreased performance of HS-BoF using 20k vocabularies, the reason is same with object dataset. Secondly, SPM is better than traditional BoF in this dataset. Thirdly, the best mAP for scene dataset is 0.548 by proposed HS-BoF (ML-codebook), which is lower than caltech-256 dataset (0.669). The reason is that there are not distinct things in most natural scene images, in other words, natural scene images are smoother than object images, after HSBT processing, some parts will be filtered, which affects the precision.

Table 2 illustrates the mAP of each category of 13-scene dataset using HS-BoF (ML-codebook) with 10k vocabularies. 'forest' obtains the best result. These categories such as 'street', 'insidecity' have acceptable results. However, the categories such as 'bedroom', 'highway' can not achieve satisfying mAPs by our approach. The reason is that fewer keypoints are detected on road and bed while noisy objects like flowers, lights, tables in this kind of images have many keypoints, thus the road and bed are filtered finally in

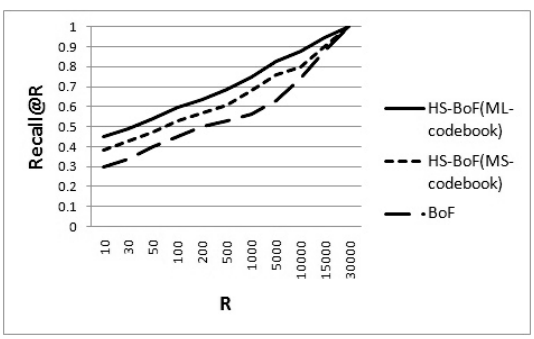


Fig. 11 Rate of relevant images found in the top R images.

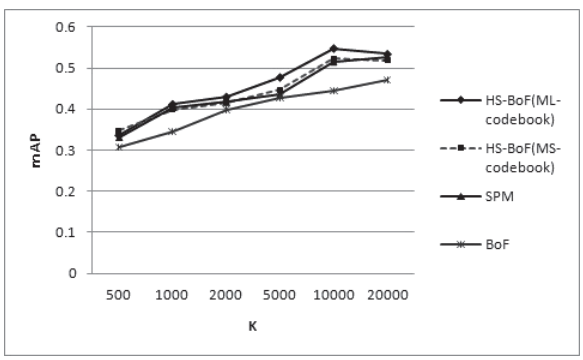


Fig. 12 Comparison of mAP on scene dataset.

Table 2 The mAP of each categoty of 13-scene dataset.

Category	mAP	Category	mAP
forest	0.758	open country	0.512
inside city	0.725	coast	0.496
street	0.716	PARoffice	0.439
tall building	0.703	living room	0.380
CALsuburb	0.629	highway	0.325
kitchen	0.611	bedroom	0.276
mountain	0.554		

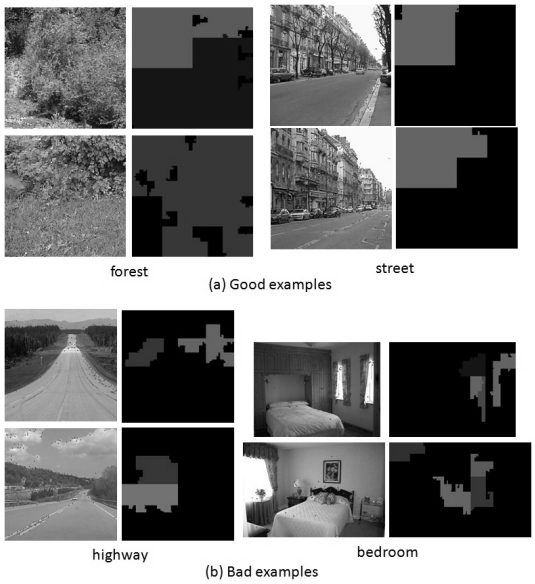


Fig. 13 Some good examples and bad examples of 13-scene dataset.

our approach. Further, those noisy objects may have large differences in same category. Strictly speaking, these images which have many discrepant noisy objects are not similar with each other, however, they have same meaning. If some semantic information is added into the retrieval system, this kind of images would be probably searched. Some visualized examples are shown in Fig. 13. In each column, the left images shows the extracted keypoints and the right ones are the results of 15th's grouping.

5.3 ImageNet1M

50000 images from all categories were used to get codebook, 800 images from the rest were tested as queries. We set the threshold δ to 500, Th to 0.8 again, and $n = 10$ for ML-codebook. CHI-square was used. The best mAP is 0.575 for HS-BoF with ML-codebook, 0.519 for MS-codebook, using 50k, 100k, 200k vocabularies. To achieve comparable accuracy, HS-BoF with ML-codebook requires 784 bytes per image, to be compared, with 17952 bytes for BOF. With a 8 GB memory computer, HS-BoF takes 5.37 seconds for feature extraction, HSBT processing and searching, while the average query time is 4.74 seconds using MS-codebook.

6. Conclusions and Future Work

In this study, Hilbert scan based bag-of-features (HS-BoF) is proposed for image search. Hilbert scan based tree representation (HSBT) is built based on local descriptors with spatial relationships. The features of objects are boosted and background features are suppressed adaptively by the proposed grouping strategy, which benefits our search. Multi-layer codebook and multi-size codebook generation are given based on HSBT. Experiments show that HS-BoF obtains higher accuracy than BoF with smaller memory usage. In addition, HS-BoF with multi-layer codebook performs better than multi-size codebook in terms of retrieval accuracy, while multi-size codebook spends fewer query time. Moreover, we investigate CHI-Square in our research and extensive experiments demonstrate that it improves our search quality.

From the visualized HSBT results, an interesting thing is found that the representation of that images came from the same category is similar while different categories is distinct. This point also can be seen from Fig. 6 and Fig. 13, even though where the colors painted are not same (these colors are irrelevant with our experiments). So maybe it can be improved to label objects in images automatically, then object recognition and image search can be done based on these labels. In addition, how to maintain the high accuracy while using low dimensional vectors to represent images is our future work.

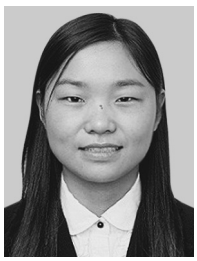
Acknowledgments

We would like to thank all the people providing their data for test and all the observers for giving their contributions to this study.

References

[1] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol.60, no.2, pp.91–110, 2004.
[2] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," Int. J. Comput. Vis., vol.60, no.1, pp.63–86, 2004.
[3] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissmilarity measure," IEEE Trans. Pattern Anal. Mach. Intell., vol.32, no.1, pp.2–11, 2010.
[4] 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, 2003.
[5] J. Yu-Gang, N. Chong-Wah, and Y. Jun, "Towards optimal bag-of-features for object categorization and semantic video retrieval," ACM International Conference on Image and Video Retrieval (CIVR), July 2007.
[6] M. Douze, H. Jegou, and H. Sandhawalia, "Evaluation of GIST descriptors for web-scale image search," ACM International Conference on Image and Video Retrieval (CIVR), July 2009.
[7] J. Zobel, A. Moffat, and K. Ramamohanarao, "Inverted files versus signature files for text indexing," ACM Trans. Database Syst., vol.23, no.4, pp.453–490, 1998.
[8] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.
[9] M. Marszalek and C. Schmid, "Spatial weighting for bag-of-features," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.
[10] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Spatial-bag-of-features," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.
[11] S. Kamata, R.O. Eason, and Y. Bandou, "A new algorithm for N-dimensional Hilbert scanning," IEEE Trans. Image Process., vol.8, no.7, pp.964–973, July 1999.
[12] S. Kamata, M. Niimi, and E. Kawaguchi, "Interactive analysis of multi-spectral image using a Hilbert curve," International Association for Pattern Recognition (IAPR), pp.93–97, 1994.

[13] S. Kamata and Y. Hayashi, "Region-based scanning for image compression," IEEE International Conference on Image Processing (ICIP), pp.895–898, 2000.
[14] H.V. Jagadish, C. Faloutsos, and H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve," IEEE Trans. Knowl. Data Eng., vol.13, no.1, pp.124–141, 2001.
[15] Y. Ebrahim, M. Ahmed, S.C. Chan, and W. Abdelsalam, "An efficient shape preprsentation and description technique," IEEE International Conference on Image Processing (ICIP), pp.441–444, 2007.
[16] T. Agui, T. Nagae, and M. Nakajima, "Generalized Peano scans for arbitrary-sized arrays," IEICE Trans. Inf. & Syst., vol.E74, no.5, pp.1337–1342, May 1991.
[17] J. Zhang, S. Kamata, and Y. Ueshige, "A pseudo-Hilbert scan for arbitrarily-sized arrays," IEICE Trans. Fundamentals, vol.E90-A, no.3, pp.682–690, March 2007.



Pengyi Hao is a Ph.D. student of the Graduate School of Information, Production and Systems, Waseda University, Japan. She received her first M.E. degree in computer application and technology from Shanghai University, China, in March 2010, and the second M.E. degree in computer science from Waseda University, Japan, in July 2010. Her current research interests are multimedia retrieval and pattern recognition.

Sei-ichiro Kamata received the M.S. degree in computer science from Kyushu University, Japan, in 1985, and the doctor of engineering degree from the department of Computer Science, Kyushu Institute of Technology, Japan, in 1995. From 1985 to 1988, he was in NEC, Ltd., Kawasaki, Japan. In 1988, he joined the faculty at Kyushu Institute of Technology. From 1996 to 2001, he was an associate professor in the department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University. Since 2003, he has been a professor in Graduate School of Information, Production and Systems, Waseda University. In 1990 and 1994, he was a visiting researcher at the University of Maine, Orono. His research interests are image processing, pattern recognition, image compression, remotely sensed image analysis, space-filling curve and fractals. Prof. Kamata is a member of the IEEE, and the ITE in Japan.