IPSJ Transactions on Computer Vision and Applications Vol. 2 253–261 (Dec. 2010)

Research Paper

Distance-based Multiple Paths Quantization of Vocabulary Tree for Object and Scene Retrieval

Heng Yang,[†1] Qing Wang[†1] and Ellen Yi-Luen Do[†2]

The state of the art in image retrieval on large scale databases is achieved by the work inspired by the text retrieval approaches. A key step of these methods is the quantization stage which maps the high-dimensional feature vectors to discriminatory visual words. This paper mainly proposes a distance-based multiple paths quantization (DMPQ) algorithm to reduce the quantization loss of the vocabulary tree based methods. In addition, a more efficient way to build a vocabulary tree is presented by using sub-vectors of features. The algorithm is evaluated on both the benchmark object recognition and the location recognition databases. The experimental results have demonstrated that the proposed algorithm can effectively improve image retrieval performance of the vocabulary tree based methods on both the databases.

1. Introduction

We are interested in the issue of image retrieval in large-scale databases. Given a query image which contains either a particular object or a scene from an interest place, our motivation is to return from the large database a set of high content-related images in which that object or scene appears. However, this problem becomes difficult when one requires a search in a very large database in acceptable time. In general, the standard approach to solve this problem is firstly to represent the images by high-dimensional local features and then to match images by dealing with millions of feature vectors.

Several successful approaches for image alignment, video retrieval and object recognition have been recently reported [1)–7)]. These methods mimicked the text-retrieval approaches using the analogy of "visual words" which are actually defined by specific feature vectors and can be considered as the division of the high-dimensional Euclidean space. Sivic and Zisserman [1)] firstly employed a text retrieval approach for video object recognition. The feature vectors are quantized into bags of visual words, which are defined by k-means clustering method on feature vectors from the training frames. Then the standard TF-IDF (Term Frequency-Inverse Document Frequency) weighting scheme, which down-weights the contribution of the commonly occurred words, is used for scoring the relevance of an image to the query one. Nister and Stewenius [2)] further developed the method of visual word representation [1)]. They designed a hierarchical vector quantization method based on a vocabulary tree and thus a much larger and more discriminatory vocabulary can be used efficiently which can improve the image searching quality dramatically. Similarly, Moosmann, et al. [3)] employed a forest of random trees to rapidly and distinctively assign descriptors to clusters. Schindler, et al. [4)] used the same data structure as the vocabulary tree for large-scale location recognition application. In particular, they presented a Greedy N-Best Paths (GNP) algorithm to improve the retrieval performance of the traditional vocabulary tree algorithm by considering more candidates instead of one at each level of the tree. Chum, et al. [7)] brought the query expansion technology, which is a standard method in text retrieval system, into the visual domain for improving the retrieval performance. They first utilized the spatial constraint between query image and each returned image and then used these verified images to learn a latent feature model which controlled the construction of expanded queries. Philbin, et al. [8)] introduced the soft-assignment technology so that a high-dimensional descriptor can be mapped to a weighted combination of visual words, rather than hard-assignment to a single word. This method improved the performance of retrieval since it allowed the inclusion of features which were lost in quantization step based on the hard-assigned methods. More recently, Jegou, et al. [9)] proposed an approximate nearest neighbor search method based on product quantization, producing short codes and corresponding distance estimators approximating the Euclidean distance between the original vectors. They presented the distance between a vector and code, which is actually an alternative way for sub-vector representation. The feature space is decomposed into a Cartesian product of low dimensional subspaces and the feature vector is quantized in each subspace separately. This method outperforms traditional methods

†1 School of Computer Science and Engineering, Northwestern Polytechnical University
†2 College of Computing, Georgia Institute of Technology

significantly in terms of search quality and memory usage.

In a word, there are mainly four key stages in current successful image retrieval systems based on the bag-of-visual-words model:

(1) Build a large and discriminatory visual vocabulary using the training data;

(2) Quantize the feature vectors of the database into their corresponding visual words respectively;

(3) Use the TF-IDF scheme to score the similarity between images;

(4) Employ well-known technologies to further refine the retrieval results, such as spatial verification and query expansion.

In this paper, we only focus on the first two stages and we improve the image retrieval quality via two novel contributions in two fundamentally different ways – a more sophisticated quantization method for higher retrieval performance and a way to build vocabulary tree by sub-vectors of features for more efficiency.

The remainder of this paper is organized as follows. Section 2 briefly reviews the traditional vocabulary tree algorithm and Section 3 presents our approach in details. The experimental results and related discussions are given in Section 4. Finally, the conclusion is summarized in Section 5.

2. Traditional Vocabulary Tree Algorithm

The traditional vocabulary tree [2] is built by hierarchical k-means clustering algorithm on SIFT [10] feature vectors from the training data. A vocabulary tree is a kind of full k-way tree of depth L. Therefore, there are k^L leaf nodes (visual words) at the bottom of the tree. For a SIFT feature, it is down from the root node to a leaf node by comparing at each level the feature vector to the k candidate cluster centers and choosing the closest one. Then the path down the tree is encoded by an integer which represents a specific visual word. Finally, the relevance score between a query image and a database image is computed by using the standard TF-IDF scheme.

However, there exist two drawbacks in the traditional vocabulary tree. The first and most significant drawback is the inaccurate quantization scheme which decreases the retrieval quality. At every level of the tree, the quantization loss will be inevitable when the feature vectors are located on the boundaries that are defined by the cluster centers (see **Fig. 1** for illustration). In Fig. 1, points

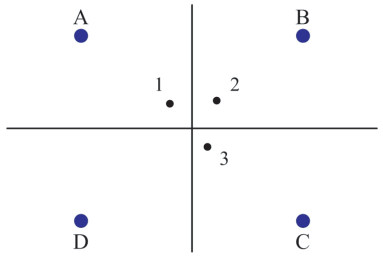


Fig. 1 Illustration of quantization loss in traditional vocabulary tree algorithm.

A, B, C and D represent cluster centers and points 1, 2, 3 are feature vectors, respectively. The cross lines are the bounds defined by cluster centers A to D together. Points 1, 2, 3 will be assigned to different visual words despite them being close to each other. In addition, the memory cost for loading a large vocabulary tree is too high. In the work of Nister, et al. [2], loading a vocabulary tree with 10 branches and 6 levels occupies as much as 143 MB of memory.

3. Our Approach

On the analysis of the drawbacks of the traditional vocabulary tree, our approach is designed to solve two issues of quantization loss and low efficiency in this section. The proposed approach has three steps. Firstly, a large vocabulary tree is built based on sub-vectors. Then, feature vectors are assigned to visual words by the proposed quantization algorithm. Finally, the standard TF-IDF scheme is employed to give the relevance score between database image and the query one.

3.1 Building Vocabulary Tree Using Sub-vectors of the Features

There is an intuitional assumption in our approach, which is,

Assumption: If some of the corresponding sub-vectors of the features are close respectively, the whole feature vectors should be close to each other.

This assumption has been employed in nearest neighbor search in high dimensional space in our previous work [11]. The sub-vector consists of the randomized chosen dimensions of the whole feature vector and sub-vector based hashing algorithm for high dimensional image feature matching outperform traditional nearest neighbor search algorithms [10),12].

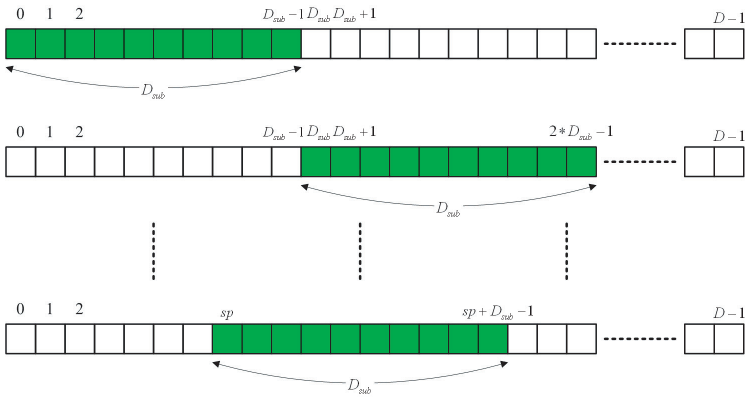


Fig. 2 Illustration of the procedure of constructing D_{sub} dimension sub-vector from D dimension feature.

In this paper, for the simplicity of implementation, we just construct sub-vector by using the consecutive dimensions. **Figure 2** illustrates the procedure of constructing sub-vectors. We pre-set a start position sp, e.g., 0, for sub-vector and extract D_{sub} dimensional vector items continuously. In subsequent levels of vocabulary tree, we shift the start position to $(sp + D_{sub})$ MOD $(D - D_{sub} + 1)$, where MOD is the modulo operator to ensure the sub-vector is not beyond the scope of the whole vector. Intuitionally, the correct probability of this assumption will increase with the growing of D_{sub}. The experimental result (see Fig. 3 in Section 4.1) has validated this assumption and the gain of retrieval performance could be negligible when $D_{sub} \geq 60$. That means the vocabulary tree built by the sub-vectors is as discriminatory as the one built by the whole-vectors. The procedure of building vocabulary tree is described in *Algorithm 1*.

It is worth mentioning that a bisecting k-means algorithm [13] is applied in our method instead of the regular k-means algorithm. Bisecting k-means algorithm has some advantages over k-means, such as more efficient and producing clusters with smaller entropy. Furthermore, the most important merit is that it tends to produce clusters of similar sizes, while k-means is known to produce clusters of widely different sizes. This merit is very important for training a large vocabulary tree. One purpose for building such a tree is to create leaf nodes (visual words)

Algorithm 1: *Building Vocabulary Tree Based on Sub-Vectors*

Initialization: All the training features are loaded in the root node, set current level $l = 0$, start position of sub-vector $sp = 0$, $D_{sub} = 60$ and SIFT vector length $D = 128$, respectively.

(1) Use bisecting k-means approach [13] to partition the features in the current node into k clusters according to the sub-vector at the position $[sp, sp + D_{sub} - 1]$ in each feature;
(2) Assign features of the k clusters to k children of the current node respectively;
(3) $l = l + 1$;
(4) $sp = (sp + D_{sub})$ MOD $(D - D_{sub} + 1)$;
(5) The same process (1)∼(4) is applied to each child of current node recursively, and it is ended if l equals to maximum number of level L.

as more as possible, whereas k-means often produces null clusters when k is large and the number of features in current node is small.

The total memory cost for a vocabulary tree is linear with the dimensionality of feature vector [2]. Therefore, building a vocabulary tree based on sub-vectors ($D_{sub} = 60$) can save more than half size of memory cost of the traditional method ($D = 128$).

3.2 Distance-based Multiple Paths Quantization

The reason that results in the quantization loss in traditional vocabulary tree algorithm is that it only chooses the closest candidate at each level. An improved method is the Greedy N-Best Paths (GNP) algorithm [4] which chooses the closest N nodes at each level by comparing the $k \times N$ candidates. GNP algorithm can reduce the quantization loss since it considers more nodes in traversing a tree, which could decrease the risk that the feature vectors, near the bounds but close to each other, are quantized to different visual words. However, there still exist two problems in GNP algorithm. On one hand, the candidate paths number N is a constant, which is not flexible. For example, in the case that a feature vector is much nearer to its nearest cluster center than to its second nearest one, it is only need to consider the nearest center as the candidate instead of N candidates for the nearest one is discriminatory enough. On the other hand, there still exists risk for assigning those close feature vectors to different visual words since GNP algorithm finally returns one closest candidate at the last level of the tree.

To address the above mentioned issues of quantization loss, we propose the distance-based multiple paths quantization (DMPQ) algorithm to quantize the

Algorithm 2: *Distance-based Multiple Paths Quantization (DMPQ)*

Initialization: A given feature q, level $l = 1$, maximum paths number M, threshold t_c for choosing candidates, and threshold t_d for discarding the ambiguous words.

(1) Compute Euclidean distances from the corresponding sub-vector of q to all children nodes of the root;
(2) While ($l < L$) {
(3) Find m ($1 \le m \le M$) closest candidates at the level l, and make them satisfy the following inequalities at the same time,
$$\frac{d^l_{q,nn\_1}}{d^l_{q,nn\_2}} \ge \frac{d^l_{q,nn\_1}}{d^l_{q,nn\_3}} \ge \cdots \ge \frac{d^l_{q,nn\_1}}{d^l_{q,nn\_m}} \ge t_c > \frac{d^l_{q,nn\_1}}{d^l_{q,nn\_(m+1)}}$$
(4) $l = l + 1$;
(5) Compute distances from the corresponding sub-vector of q to all $k \times m$ candidates at the level l;
(6) }
(7) If ($\frac{d^L_{q,nn\_1}}{d^L_{q,nn\_2}} > t_d$) then discard q;
(8) Else q is quantized to the closest candidate at the level L.

feature vectors. The proposed DMPQ algorithm improves vocabulary tree based methods in two aspects. First, instead of choosing one candidate or the constant N candidates at each level of the vocabulary tree, DMPQ algorithm considers dynamic m candidates which are chosen by their distances to the query vector. Second, at the leaf level of the tree, DMPQ algorithm throws off the ambiguous features which are located around the bounds. Our approach is described in *Algorithm 2*, where d^l_{q,nn_m} denotes the distance from the corresponding sub-vector of the feature q to its m-th nearest neighbor cluster center at the level l. DMPQ algorithm dynamically chooses the discriminatory candidates at each level, which is determined by the parameter t_c. Furthermore, DMPQ algorithm throws off the ambiguous features at the leaf level of the tree, which are determined by the parameter t_d. In this way, DMPQ algorithm can even play a role of filter to remain the unambiguous feature vectors that have low risk of quantization loss.

The proposed algorithm is mainly represented by two important parameters t_c and t_d. These two parameters make our algorithm generalize the traditional vocabulary tree and GNP algorithms. The traditional vocabulary tree is just the specific case of our approach with $t_c = 1$ ($m = 1$ in this case) and $t_d = 1$; and so is GNP with $t_c = 0$ and $t_d = 1$. The setting of parameters t_c and t_d in DMPQ algorithm will be discussed in detail in Section 4.1.

3.3 Scoring Scheme

The standard TF-IDF scheme has been successfully applied to infer the relevance score between images. We also follow the TF-IDF scheme, and the j-th element of the query visual word q and database visual vector d are given as follows,

$$q_j = \frac{n_{jq}}{n_q} \log \frac{N}{N_j}, \qquad d_j = \frac{n_{jd}}{n_d} \log \frac{N}{N_j} \tag{1}$$

where n_{jq} and n_{jd} denote the number of the j-th visual word in query image and database image respectively; n_q and n_d are the number of features in query image and database image respectively; N_j is the number of images in database where the j-th visual word occurs, and N is the number of images in database.

According to Eq. (2), the relevance score that gives the similarity between query image $\boldsymbol{I}_q$ and database image $\boldsymbol{I}_d$ can be calculated by the scalar product between the normalized $\boldsymbol{q}$ and $\boldsymbol{d}$ (terms the normalized vectors as $\tilde{\boldsymbol{q}}$ and $\tilde{\boldsymbol{d}}$, respectively),

$$Score(\boldsymbol{I}_q, \boldsymbol{I}_d) = \tilde{\boldsymbol{q}} \cdot \tilde{\boldsymbol{d}} = \sum_{j|\tilde{q}_j \neq 0, \tilde{d}_j \neq 0} \tilde{q}_j \tilde{d}_j \tag{2}$$

4. Experimental Results and Analysis

To evaluate the performance of the proposed algorithm, we use two challenging image databases. One is the benchmark images ⋆1 provided by Nister, et al. [2] for object recognition, which contains 10,200 images in groups of four that belong together. In each group, the same object is taken from different positions or under varying illumination conditions. Another database is Ljubljana urban images ⋆2 provided by Omercevic, et al. [14] for location recognition. It consists of 612 images of urban environment covering an area of 200×200 square meters. At each of the 34 standpoints, 18 images were captured at 6 orientations and 3 tilt angles. SIFT [10] algorithm is used for both local feature detection and description, which has been widely used nowadays.

Our experiments are divided into three aspects. First, we test the parameters

⋆1 http://www.vis.uky.edu/~stewe/ukbench/data/
⋆2 http://vicos.fri.uni-lj.si/LUIS34/

setting of our approach on the training data – a subset from the object recognition database. Second, we evaluate the retrieval quality of our method on the whole object recognition database. Third, we employ our algorithm on the Ljubljana urban database to further examine its performance in location recognition application. In the second and third parts of experiments, we use three algorithms as baselines, which are the traditional vocabulary tree [2], traditional vocabulary tree with GNP algorithm [4] and the soft-assignment algorithm [8]. All these experiments are performed to verify the effectiveness of reducing quantization loss for vocabulary tree by the proposed DMPQ algorithm.

All the experiments are executed on a PC with Pentium IV dual-core 2.0 GHz processor and 2 GB memory.

4.1 Parameters Setting on Training Data Set

In our approach, there are three important parameters, which are the dimension D_{sub} of sub-vector in building vocabulary tree, and two thresholds t_c and t_d in DMPQ algorithm for quantization. We choose the first 2,000 images of the object recognition database as both the off-line training data set for building a vocabulary tree and the testing benchmark for parameters setting. From the 2,000 images, 1.5 M SIFT feature vectors are generated. In all experiments in this subsection, the vocabulary trees using either whole-vectors or sub-vectors are both built with 10 branches and 6 levels, which will result in about 10^6 visual words. The retrieval quality can be measured by *Average Retrieval Accuracy* (ARA) which can be computed by Eq. (3), where cnt_i denotes how many of the first four most similar images in the same group as the i-th image (including the i-th image itself), n is the number of the database images.

$$ARA = \frac{1}{n}\sum_{i=1}^{n}\frac{cnt_i}{4} \tag{3}$$

Different vocabulary trees can be built with different D_{sub} using the *Algorithm 1* described in Section 3.1. **Figure 3** gives the ARA curves on the 2,000 training images with different value of D_{sub}. The quantization method is the same as the traditional vocabulary tree. We can see that the average retrieval accuracy increases along with the growth of D_{sub}. However, the gain becomes negligible when $D_{sub} \geq 60$. That is to say, using *Algorithm 1* with $D_{sub} = 60$ can efficiently

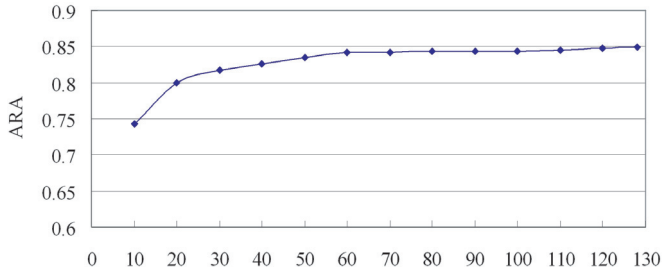
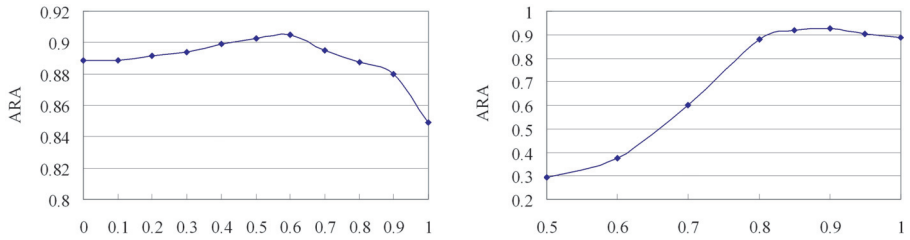


Fig. 3 Average Retrieval Accuracy (ARA) on the changing of D_{sub}.

build an effective and discriminatory visual vocabulary as that using the whole-vectors of features.

Considering the unavoidable quantization error of vocabulary tree for high dimensional feature search, the sub-vector based vocabulary tree can hold the approximate performance as traditional tree built from the whole feature vector. From the simple statistics, if we use $D_{sub} = 60$ dimensional sub-vector to construct vocabulary tree, we can save 53% memory occupation for 128 D SIFT features and reduce more than a half computation of Euclidean distance.

As mentioned in Section 1, Jegou, et al. [9] proposed to use product quantizers to produce short codes and corresponding distance estimators approximating the Euclidean distance between the original vectors. The feature space is decomposed into a Cartesian product of low dimensional subspaces. A feature vector is then represented by a short code composed of its subspace quantization indices. Apart from the traditional distance metric between the quantized codes of original vectors, the asymmetric distance between a vector and a code can increase the accuracy of nearest neighbor search. This has further verified the assumption mentioned in Section 3.1 and the effectiveness of the sub-vector representation. Comparing to Jegou, et al.'s method [9], this paper actually focus on the construction of a hierarchical visual word tree and the quantization is carried in the subspace derived from sub-vector at each level by different strategies, e.g., greedy N-best path (GNP) in Schindler, et al.'s work [4], a weighted combination of visual words (soft-assignment) in Philbin, et al.'s work [8] and our distance-based multiple path quantization (DMPQ). We now do not consider using all the sub-



(a) ARA vs. t_c (b) ARA vs. t_d

Fig. 4 ARA versus the changing of t_c and t_d in DMPQ algorithm.

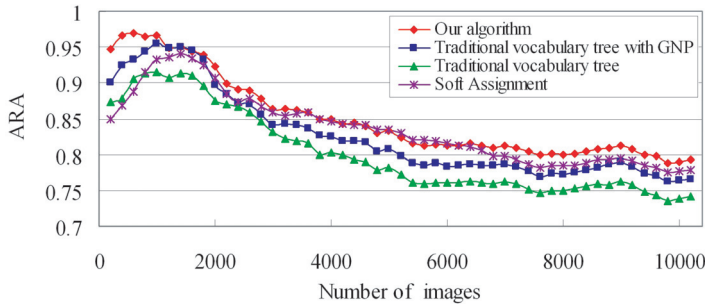


Fig. 5 The object image retrieval accuracy of our algorithm compared to traditional vocabulary tree [2], traditional vocabulary tree with GNP [4] and soft-assignment technology [8] on the whole object recognition database.

spaces spanned by corresponding sub-vectors and only focus on the one subspace in the tree node at each level. Since it is obvious that product quantizer can strengthen the search precision and ensure the adaptation of the code book to the data distribution to represent, it is a good trend to improve the quantization for feature vector matching.

Apart from the setting of D_{sub}, it is also important to discuss other two thresholds of t_c and t_d in DMPQ algorithm, which determine the accuracy rate of feature matching. The maximum of path number M in DMPQ algorithm is set to 10 and $D_{sub} = 128$ (using the whole vector).

Figure 4 shows the object image retrieval accuracy with different settings of t_c and t_d respectively. From Fig. 4 (a) (fixed $t_d = 1$), we can find that when t_c varies from 0 (in GNP case) to 1 (in traditional vocabulary tree case), there is a peak where t_c is around 0.6. The reason is that t_c gives our algorithm more chances to consider multiple candidates rather than one; and at the same time it prevents some candidates being examined in GNP algorithm so that it makes the searched candidates more distinctive and thus obtains more accurate results. From Fig. 4 (b) (fixed $t_c = 0$), we can find the best result is reached when t_d is set to around 0.9. Because t_d helps DMPQ algorithm to discard the ambiguous feature vectors that are located near the bounds at the leaf level as much as possible, accurate representation of the visual word vector of an image can be obtained. The value of t_d should be set close to 1; otherwise, it makes the algorithm discard too many discriminatory feature vectors that are located far from the bounds.

In summary of this subsection, the important parameters of our method are set as $D_{sub} = 60$, $t_c = 0.6$ and $t_d = 0.9$ in all subsequent experiments.

4.2 Performance Comparison on the Whole Large-scale Database

We have already built vocabulary tree using the SIFT features from the subset (first 2,000 images) of the object recognition database. Now we evaluate the performance of our algorithm with the traditional vocabulary tree [2], traditional vocabulary tree with GNP algorithm [4] and the soft-assignment algorithm [8] on the whole database which contains 10,200 images. The vocabulary trees of the three algorithms have the same shape with 10 branches and 6 levels. Parameter N in GNP algorithm and M in DMPQ algorithm are set to 10 equally. For the soft-assignment algorithm, the parameters are chosen by referring the literature [8], which are the spatial scale σ = 6,250 and the number of nearest visual words $r = 3$.

The object image retrieval accuracy of the four algorithms along with the changing number of images is shown in **Fig. 5**, from which we can see that the GNP algorithm indeed improves the performance of the traditional vocabulary tree algorithm. The soft-assignment algorithm can achieve better retrieval performance than the GNP and vocabulary tree algorithms when the number of image exceeds 2,000 and it is as good as our DMPQ algorithm when the number of image is less than 6,200. However, the performance of soft-assignment drops when the size of dataset exceeds 6,200. In particular, our algorithm gives the best results.

Fig. 6 Four retrieval examples on the object recognition database by our algorithm.

Even when the database size is up to 10 K, our algorithm can obtain about 80% average retrieval accuracy.

Apart from the search accuracy, it is also important to evaluate the performance in terms of the processing time and memory requirement. It is no doubt that traditional vocabulary tree is the fastest and lowest cost one among these four algorithms. Since there are $k \times L$ times distance comparisons during quantization, where k and L are the branch and the depth of the tree respectively. In the experiment, the vocabulary tree is a full 10 branch tree with 6 levels so that the total comparison times of traditional tree is 60. Since GNP algorithm has to reserve fixed N best candidates from the root to $L-1$ level, the comparison times is $k + (L-1) \times k \times N$, i.e., 510 in this experiment when $N = 10$. Our DMPQ algorithm dynamically selects M best candidates at most so that the comparison times is $k + (L-1) \times k \times m$, where $1 \leq m \leq M$. The upper bound is $k + (L-1) \times k \times M$. It is obvious that DMPQ is faster than GNP to a certain extent. DMPQ and GNP algorithms have the same memory requirements as traditional tree. In soft-assignment process, we have to compute a weighted combination of r visual words so that we need to enlarge the size of memory space for r times for index [8]. As a result, its time complexity is based on the size of parameter r.

In addition, **Fig. 6** shows most five similar retrieved images for four queries using the proposed algorithm. The query samples are listed in the first column and the top five similar images are listed from second to sixth columns respectively. The images within blue dashed rectangle denote they are not in the same groups of the query ones respectively. The performance ARA (denoted by the numbers listed on the most right column) is computed by the front four similar retrieval results, but here we list the top five results to show the effectiveness of our algorithm. Although the images within the blue dashed rectangle do not belong to the same group of the query ones, e.g., the first and third rows of Fig. 6, they are content-related to the query images to some extent.

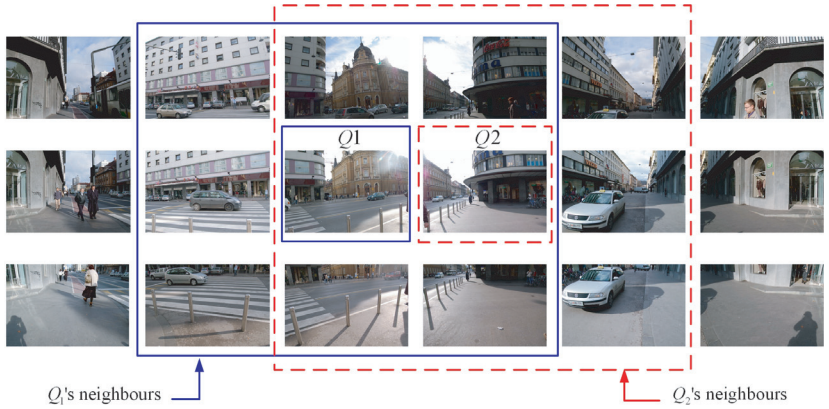


Fig. 7 Example images taken in one standpoint.

4.3 Performance Comparison on Ljubljana Urban Database

One may doubt the proposed algorithm whether the vocabulary tree learnt on one database could be applicable to another database. To validate this, we apply our vocabulary tree trained from the object recognition database to Ljubljana urban database. **Figure 7** gives example images taken at the first standpoint. For every standpoint, we choose the middle two as query images (such as Q_1 and Q_2 in Fig. 7) and define their neighbors as the ground truth of retrieval. For example to Q_1, nine nearest images (include itself) within the big blue rectangle are defined as its neighbors. Similarly, the images within the big red dashed rectangle are Q_2's neighbors. Therefore, there are totally 68 query images.

Figure 8 depicts the scene image retrieval accuracy of the proposed algorithm, traditional vocabulary tree [2], improved quantization GNP algorithm for vocabulary tree [4] and the soft-assignment algorithm [8]. The parameters setting for the

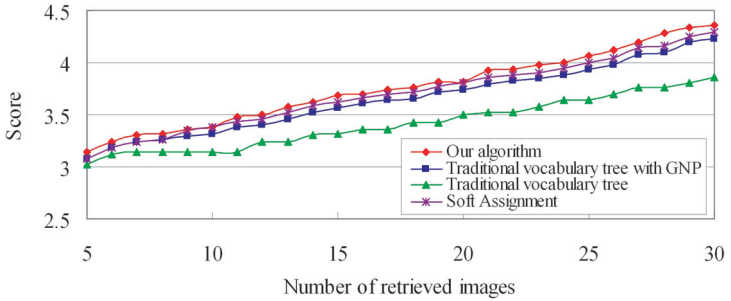


Fig. 8 The scene image retrieval accuracy of four algorithms on Ljubljana urban database.

Fig. 9 Three retrieval examples on the Ljubljana urban database using the proposed algorithm.

four algorithms are the same as in Section 4.2. The curves in Fig. 8 show the average number of ground truth images in n top-ranked retrieved images and n is changing from 5 to 30. It can be seen that the average scores of all the four algorithms do not exceed 4.5 due to the weak bounds between the database images. Although the query images and their neighbors are taken in the same standpoint, the content between them are sometimes low related. However, all of the four algorithms can get the satisfactory performance in this database – averagely more than 3 correct location images could be returned. Furthermore, our algorithm can achieve the best result compared to the other three methods.

As retrieval examples, **Fig. 9** lists the most four similar retrieved images for three query samples using the proposed algorithm. In Fig. 9, the query samples are listed in the first column and the top four similar images retrieved by our algorithm are listed from second to fifth columns respectively. The first example shows the perfect performance of our algorithm for all the retrieved images are the neighbors of the query one and they are indeed high content-related. The second example actually also shows the perfect retrieval quality as the first example. But the image within the blue dashed rectangle (content-related to the query in fact) is considered as the wrong result for it is taken in different standpoint to the query one. The third example shows the worst performance of the three examples. The last two retrieved images (within the blue dashed rectangles) are wrong results which are neither content-related nor taken in the same standpoint to the query image.

5. Conclusion

The main contribution of this paper is to propose a distance-based multiple paths quantization algorithm to solve the issue of quantization loss in traditional vocabulary tree methods. The proposed DMPQ (Distance-based Multiple Paths Quantization) algorithm dynamically chooses the discriminatory candidates at each level according to the distance ratio and throws off the ambiguous features which are located around the bounds at the leaf level of the tree. In addition, a more efficient way to build vocabulary tree based on sub-vectors is introduced, which not only creates large and discriminatory visual vocabulary but also saves much memory cost. Experimental results have proved the efficiency and effectiveness of our algorithm in image retrieval based applications, such as object recognition and scene recognition. In future work, we will consider more on the product quantizer for more precious quantization in huge scale feature vector set and plan to apply our method towards the internet-scale image databases, such as the photo-sharing website Flickr [15].

Acknowledgments This work is supported by National Natural Science Fund (60873085) and National Hi-Tech Development Programs under grant No.2007AA01Z314, P.R. China.

References

1) Sivic, J. and Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos, *ICCV2003*, Vol.2, pp.1470–1477 (2003).
2) Nister, D. and Stewenius, H.: Scalable Recognition with a Vocabulary Tree, *CVPR2006*, Vol.2, pp.2161–2168 (2006).
3) Moosmann, F., Triggs, B. and Jurie, F.: Randomized Clustering Forests for Building Fast and Discriminative Visual Vocabularies, *NIPS2006* (2006).
4) Schindler, G., Brown, M. and Szeliski, R.: City-Scale Location Recognition, *CVPR2007* (2007).
5) Jegou, H., Harzallah, H. and Schmid, C.: A Contextual Dissimilarity Measure for Accurate and Efficient Image Search, *CVPR2007* (2007).
6) Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.: Object Retrieval with Large Vocabularies and Fast Spatial Matching, *CVPR2007* (2007).
7) Chum, O., Philbin, J., Sivic, J., Isard, M. and Zisserman, A.: Total recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval, *ICCV2007* (2007).
8) Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.: Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases, *CVPR2008* (2008).
9) Jegou, H., Douze, M. and Schmid, C.: Searching with Quantization: Approximate Nearest Neighbor Search Using Short Codes and Distance Estimators, *Technical Report 7020*, INRIA, August (2009).
10) Lowe, D.G.: Distinctive Image Features from Scale Invariant Keypoints, *IJCV*, Vol.60 (2004).
11) Yang, H., Wang, Q. and He, Z.: Randomized Sub-Vectors Hashing for High-Dimensional Image Feature Matching, *ACM MM2008*, pp.705–708 (2008).
12) Slaney M. and Casey M.: Locality-sensitive Hashing for Finding Nearest Neighbors, *IEEE Signal Processing Magazine*, Vol.25, pp.128–131 (2008).
13) Li, Y. and Chung, S.M.: Parallel Bisecting K-means with Prediction Clustering Algorithm, *The Journal of Supercomputing*, Vol.39, pp.19–37 (2007).
14) Omercevic, D., Drbohlav, O. and Leonardis, A.: High-Dimensional Feature Matching: Employing the Concept of Meaningful Nearest Neighbors, *ICCV2007* (2007).
15) http://www.flickr.com

(Received February 18, 2010)
(Accepted September 13, 2010)
(Released December 15, 2010)

(Communicated by *Rin-ichiro Taniguchi*)

Heng Yang received his Master and Ph.D. degrees from Northwestern Polytechnical University in 2006 and 2010, respectively. He worked as visiting student in Georgia Institute of Technology, USA, from 2008 to 2009, supported by China State Scholarship fund. He has published about 20 papers in the International journals and conferences. His current research interests are content based image retrieval and large-scale image classification.

Qing Wang received his Master and Ph.D. degrees from Northwestern Polytechnical University in 1998 and 2000, respectively. He is currently an professor at Northwestern Polytechnical University. His current research interests are large-scale 3D scene modeling and rendering, image based lighting, light field photography and application and web scale image classification. He is a member of IEEE and ACM.

Ellen Yi-Luen Do is an associate professor the College of Computing, at Georgia, USA. Her research work focuses on the development of computer aided design tools to support freehand drawing as an interface to knowledge based tools. She has also worked in the areas of computer based visual analysis tools, collaboration and annotation, physical computing and tangible media, creative toys and the home of the future.