A Fast Quantization Tree Based Image Retrieval Method

X.L. Wang\textsuperscript{1}, X. Wang\textsuperscript{2}, and A.N. Hou\textsuperscript{2}
\textsuperscript{1}School of Information Engineering, Changan University, Xi’an, Shaanxi, China
\textsuperscript{2}School of Software Engineering, Xi’an Jiaotong University, Xi’an, Shaanxi, China

Abstract - Traditional content-based image retrieval technology expresses the content of each image by feature vectors. Then image retrieval process begins by calculating the similarity between the image to search and the images in the database in terms of the corresponding feature vectors, next ranks the images in the database by a descending order of similarity, and finally outputs the desired top ones. This method shows good accuracy and high efficiency when the image databases are not very big. For modern large image databases, however, these methods will not satisfy users’ requirements for retrieval time and accuracy. To meet these challenges, in this paper, a new content-based image retrieval method is proposed which is similar to the Scalable Vocabulary Tree (SVT) image retrieval method but with important variations. Experimental results show our method can more efficiently deal with larger image databases while with similar retrieval accuracy.

Keywords: Content-based Image Retrieval, Visual Vocabulary, Vocabulary Tree, Clustering Algorithm, Random Quantization Tree

1 Introduction

The development of information technology gives an impetus to the development of multimedia technology, and image databases grow rapidly in an exponential scale. People begin to care about how to manage these images scientifically, legimitately and effectively, and how to identify needed images quickly and accurately from huge image databases. Therefore, image retrieval technology becomes an active research area.

Content-based image retrieval is a high-level learning task and can be pursued at different levels of visual properties such as appearance and shape etc. Since recognizing an object by its shape is computationally much more expensive, researchers try to represent object by learning appearance features of objects, such as color and texture, in the form of local descriptors which are extracted from image local regions and form the basis of many appearance-based object discovery applications.

Published by David Lowe in 1999, Scale-Invariant Feature Transform (SIFT) is an algorithm in computer vision to detect and describe local features in images which are invariant to color, rotation, and translation etc. \cite{1}. With its receptive field-like image operations, SIFT based image descriptors have opened up an area of research on image-based matching and recognition with numerous application areas. Being based on theoretically well-founded scale-space operations or approximations thereof, these approaches have been demonstrated to allow for robust computation of image features and descriptors from real-world image data.

For modern large databases, traditional sequential brute force retrieval methods are of low efficiency and, therefore, unable to meet the requirements of users. Different retrieval methods have been proposed to improve retrieval efficiency. For fast image retrieval, Scalable Vocabulary Tree (SVT) is a way of both improved retrieval accuracy and better retrieval efficiency. In this mechanism, SIFT descriptors as image characteristics are first extracted as the content features to depict an image. Next, the method of hierarchical \(k\)-means clustering algorithm is applied on the extracted image features to generate a visual word vocabulary tree. Finally, a hierarchical Term Frequency Inverse Document Frequency (TF-IDF) scoring using hierarchically defined visual words that form the vocabulary tree is proposed to allow much more efficient lookup of visual words. The image retrieval based on vocabulary tree saves time for image matching, especially for large image libraries. However, due to the “curse of dimensionality” problem of high-dimensional space, the establishment of the vocabulary tree consumes a considerable amount of time, and, therefore, can not meet the users’ requirements. To partially circumvent this problem, through studying the vocabulary tree, we propose a random quantization tree (QT) method to retrieve similar images. The experimental results show that the random quantization tree is significantly better than vocabulary tree in the tree construction efficiency while maintaining similar retrieval accuracy to that of SVT.

The rest of the paper is organized as follows. The vocabulary tree method as a solution to the content-based image retrieval problem is described in Section 2. An introduction to our proposed quantization tree-based method is given in Section 3. Experiments are designed in Section 4 to explore the suitability of our method as a solution for the current image retrieval problem. Finally, conclusions are made in Section 5. Before proceeding, a brief introduction to the SIFT image descriptor is given in the below.

1.1 Scale-Invariant Feature Transform

To recognize any object in an image, interesting points (which are termed as keypoints in the SIFT framework) on an object can be extracted from a training image to provide a
feature description of the object, which can then be used to identify the object when attempting to locate it in a test image containing that object. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination.

Lowe's method for image feature generation transforms an image into a large collection of feature vectors, each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. There are mainly four steps involved in the SIFT algorithm.

In the first step, that is, Scale-Space Extrema Detection, an image is convolved with Gaussian filters at different scales, and the difference of successive Gaussian-blurred (DoG) images is taken. Keypoints are then identified as local maxima/minima of the DoG images that occur at multiple scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint. Scale-space extrema detection can produce too many keypoint candidates, some of which are unstable. In the second step, that is, Keypoint Localization, the SIFT algorithm discards low contrast keypoints and filters out those locating on edges. To do so, Taylor series expansion of the scale space is used to get more accurate location of extrema, and if the intensity at this extrema is less than a threshold value, it is rejected. Further, DoG has higher response for edges, so edges also need to be removed. By eliminating any low-contrast keypoints and edge keypoints, strong interest points remain. It can happen that keypoints have the same location and scale, but different directions. In the third step, that is, Orientation Assignment, an orientation can be assigned to each keypoint to achieve invariance to image rotation. In a neighbourhood taken around a keypoint location depending on the scale, the gradient magnitude and direction are calculated so as to generate an orientation histogram with 36 bins covering 360 degrees. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. Finally, in the fourth step, Keypoint Descriptor, a 16x16 neighbourhood around the keypoint is taken. It is divided into 16 sub-blocks of 4x4 size. For each sub-block, 8-bin orientation histogram is created. So a total of 128-bin values are available. It is represented as a vector to form a keypoint descriptor. Additionally, several measures are taken to achieve robustness against illumination changes and rotation etc.. These steps ensure that the keypoints are more stable for matching and recognition.

The SIFT descriptor is based on image measurements in terms of receptive fields [2][3][4][5] over which local scale invariant reference frames [6][7] are established by local scale selection [8][9][7]. These features share similar properties with neurons in the inferior temporal cortex that are used for object recognition in the primate vision [10]. A general theoretical explanation about this is given in the Scholarpedia article on SIFT [11].

![Fig. 1. SIFT-based image matching](image)

For content-based image retrieval, as illustrated in Fig.1, keypoints between the input image and the images in the databases are matched by identifying their similarity. However, for modern large databases, these methods do not scale well with the size of the image databases, and can not select a small number of matching images out of the databases in acceptable time. To speed up, Lowe used a modification of the K-d tree algorithm called the Best-bin-first (BBF) search method [12] that can identify the nearest neighbors with high probability using only a limited amount of computation. However, due to the “curse of dimensionality” problem, its search efficiency could degenerate to that of a sequential search in high-dimensional space. Therefore, rapid image retrieval algorithm is in immediate need.

## 2 Scalable recognition with a VT

Trees present an efficient way to index local image regions. Scalable Vocabulary Tree (SVT) is a two-stage recognition scheme that can handle a large number of objects and enable extremely efficient retrieval. The scheme builds upon popular techniques of indexing descriptors extracted from local regions which are hierarchically quantized in a vocabulary tree and scales efficiently to a large number of objects.

Being highly distinctive in performance evaluation [13], SIFT keypoints of objects are first extracted from a set of training images in large databases and stored. An object is recognized in a new image by individually comparing each feature from the new image to those from databases and finding candidate matching images based on similarity measure of their feature vectors.

In the offline training stage, SVT provides a more efficient training of the tree by defining a hierarchical quantization that is built by hierarchically k-means clustering on a large set of representative descriptor vectors. Instead of k defining the final number of clusters or quantization cells, k defines the branch factor (i.e., the number of children of each node) of the tree. More specifically, an initial k-means process is run on the training data, defining k cluster centers. The training data is then partitioned into k groups, where each group consists of the descriptor vectors closest to a particular cluster center. The same process is then recursively applied to each newly generated group of descriptor vectors, recursively
defining quantization cells by splitting each quantization cell into \( k \) new parts. The tree is determined level by level, up to some maximum number of levels, and each division into \( k \) parts is only defined by the distribution of the descriptor vectors that belong to its parent quantization cell.

In the online phase, each descriptor vector is simply propagated down the tree by comparing it to the \( k \) candidate cluster centers (represented by \( k \) children in the tree) at each level and choosing the closest one. This is a simple matter of performing \( k \) dot products at each level which is very efficient if \( k \) is not too large. The path down the tree can be encoded by a single integer and is then available for use in scoring. The nodes of an SVT are the centroids determined by hierarchical \( k \)-means clustering of database feature descriptors. The determination of the relevance of a database image to the query image is based on how similar the paths down the vocabulary tree are for the descriptors from the database images and for those from the query image. Using the SVT, information about a feature set is condensed for the descriptors from the image to the query image is based on how similar the paths down the tree are for the descriptors from the database images and for those from the query image. Using the SVT, information about a feature set is condensed to node visit counts. Suppose an SVT has \( N \) nodes. During the training phase, all feature descriptors extracted from the \( i \)-th database image are classified through the SVT, using a greedy nearest-neighbor search. After classification, it is known how many times, \( n_i(j) \), the \( j \)-th node in the SVT is visited by the \( i \)-th database feature set, resulting in a vector of node visit counts \( d_i = [n_i(1), n_i(2), \ldots, n_i(N)] \). Similarly, when a query feature set is received on the server, its descriptors are classified through the SVT, generating its vector of node visit counts \( q = [n_q(1), n_q(2), \ldots, n_q(N)] \). The dissimilarity between \( d_i \) and \( q \) is given by,

\[
D_i(d_i, q) = \frac{W \cdot d_i}{\|W \cdot d_i\|} - \frac{W \cdot q}{\|W \cdot q\|}
\]  

where \( W = \text{diag}(w_1, \ldots, w_N) \) is a matrix used to assign different entropy-related weights to different nodes. The Euclidean distance can be the dissimilarity measure to use. The database feature sets with the smallest \( D_i \) values are further considered for a more elaborate comparison.

Scalable image retrieval systems have the merit of retrieval efficiency and usually give impressive results, but the resulting visual vocabulary representation usually faces two crucial problems. First of all, hierarchical quantization errors and biases exist in the generation of “visual words”. Secondly, because of involving hierarchical quantization of local image descriptors, they are computationally expensive owing to the cost of hierarchical \( k \)-means clustering during training. Particularly, when the backend database is updated incrementally, the vocabulary tree model has to be re-generated entirely from the overall dataset. To lessen system computational cost, in the following, we describe an unsupervised optimization strategy in generating the hierarchy structure of visual vocabulary, which produces a more effective and adaptive retrieval model for large-scale search. By our approach, efficient and effective transfer of a retrieval model across databases of different sizes is feasible.

3 Scalable recognition with a QT

The most significant property of the SVT scheme is that the tree directly defines the quantization in the sense that the quantization and the indexing are fully integrated, essentially being one and the same. Different from SVT, our approach defines a visual vocabulary non-hierarchically, and then devises an approximate nearest neighbor search scheme in order to find similar visual words efficiently.

3.1 Visual vocabulary generation

In order to understand and manipulate the world, labeling local descriptors extracted from training images to target categories through machine learning methodology is often a major task in the building of intelligent systems. Each feature vector appears as a point in the feature space and patterns pertaining to different classes will fall into different regions of the feature space. The label-learning process is to partition the feature vectors into sensible clusters based on some properties in common. To ease the labeling task by grouping unlabeled feature vectors, unsupervised learning (also known as clustering in data mining literature) have gained prominence and many approaches have been developed for it, including hierarchical, partition-based, density-based, model-based, and graph-based approaches. An extensive study and comparison of the state-of-the-art unsupervised object discovery techniques are reported in [14]. As illustrated in Fig.2, being a graph-based approach, MST-based clustering algorithms can outperform the classic methods such as \( k \)-means clustering algorithm when the boundaries of the clusters are irregular. To this end, a fast MST-based clustering algorithm, proposed in [15], has obtained impressive segmentation results for challenging image data obtained from an indoor environment [16]. We use this method in this research for an outdoor environment based on SIFT image local descriptors.

![Fig. 2. Illustration of MST-based clustering](image-url)
3.2 Nearest neighbor based matching

Matching an object model against object data from some other source, object recognition capability is fundamental in many practical robotic systems and can be performed by nearest neighbor search. An image retrieval algorithm based on the plane nearest neighbor matching of feature sets is described in the following as,

1) label manually all $U$ training images in a database with known $W$ categories, and extract their SIFT descriptors to form two sets, $F = \{ F_o, 1 \leq o \leq U \}$, and the corresponding label set, $L = \{ L_i, 1 \leq i \leq U \}$; $L_i \in L_o$, $1 \leq j \leq W$;
2) given a coming new image, extract its SIFT feature set of size $V$, $T = \{ t_m, 1 \leq m \leq V \}$;
3) for each SIFT feature, $t_m$, calculate its Euclidean distance with every feature in $F$, if its nearest neighbor is in $F_o$, $1 \leq o \leq U$, assign $F_o$’s label, $L_o$, to $t_m$;
4) repeat Step 3) until all the feature vectors are processed;
5) a label set is obtained for the query image, and use the majority of the labels in the set as the label for this image.

Though simple and elegant, nearest neighbor search in the presence of a large database of object models is not very efficient. To support efficient image matching, multidimensional index structures, especially various types of tree-based index structures, have been developed to reduce the computation cost.

3.3 Quantization tree

Hierarchies are a natural way to organize concepts and data. For a long time, research on object recognition has aimed at building hierarchical models [17]. As a very interesting alternative to ANN search algorithm, decision tree construction is a classic technique for classification of a set of data records of known classes. Standard splitting rules usually involve a single attribute (e.g., applying a threshold to obtain locally optimal decisions at each node) [18]. Branches from the root to the leaf nodes represent conjunctions of features that lead to those class labels. To organize objects in the real world into a hierarchy based on their semantic meaning, in our approach, a tree-structured vector quantization method is used to generate a 3-way approximate nearest neighbor search tree, which is illustrated in Fig.3. Given the obtained training set, a tree root node exists at the first level which includes all the indices to the database feature elements. Then an initial set of 3 representative patterns, called centers, are randomly selected from the elements of the root as its children cluster centers (or tree nodes) at the second level, and the whole set of the elements is clustered into three subsets by assigning each feature vector to its closest representative cluster center according to a selected distance measure (say Euclidean distance). At the third level, for each of the 3 clusters obtained at the second level, three feature vectors are selected randomly as its children cluster centers and elements are clustered correspondingly, resulting in 9 cluster tree nodes at the level. This procedure continues until either all the elements in a node belong to the same object class (a pure node) or the number of the elements in a node is below some limit, say, one hundred. Every feature vector in the leaf nodes has a percept landmark associated with it.

Next, to search through the tree, given a new feature vector, its distances to the randomly chosen cluster centers at the second level are calculated and the winner is the center that the feature vector is nearest to. At the third level, the feature vector’s distances with the three children centers of the winner at the second level are computed and the corresponding new winner is selected. This procedure continues until coming to a leaf node. If it is pure, then stop. Otherwise, do a nearest-neighbor search, and the winner is the training feature vector which gives minimum distance according to the chosen metric. Our final scoring mechanism follows the same procedure as that of the SVT’s.

4 Experiments

In this section, we present the results of experiments conducted to evaluate the performance of our proposed recognition method. The method was tested by performing queries on a database consisting of 2794 images with 9 known visual categories. A subset of 114 images is chosen as the training set, for which SIFT descriptors are extracted and clustered. 72 images are chosen as the test set. The database is queried with every image in the test set. We implemented all the algorithms in C++. All the experiments were performed on a computer with Intel Core i5-3470 3.20 GHz CPU and 4 GB RAM. The operating system running on this computer is Windows 7. We use the timer utilities defined in the C standard library to report the CPU time.

Fig.4 shows some typical scenes of images used in the experiments. The query results by our method and the corresponding query efficiency are compared with those of two other methods, namely, brute force (BF, named for nearest neighbor based matching) method and SVT, are summarized in Fig.5 and Table 1, respectively. For retrieval effectiveness performance measure, we focus on the right images returned at the top for the query and especially how many percent of the other images in each category are found perfectly.

Fig.5 shows the retrieval results for the test images. There are nine sets of three lines in the graph, representing
the performances of three methods on nine categories of the images. In each three-vertical-line set, the left vertical line denotes the results for brute force method, the middle line denotes the results for our method, and the right line denotes the results for the SVT method. Clearly, the retrieval effectiveness of our algorithm agrees with the other two methods for 5 out of 9 cases and has retrieval accuracy rate above 80% for 8 out of 9 cases. Overall, it can be seen that our algorithm performs similarly well as the other two retrieval methods.

![Fig. 4. Some typical scenes of images used.](image)

The running time performances of our algorithm, the brute force method (BF), and the SVT method are summarized in Table 1. There are three rows in the table, representing the average running time performance of tree construction, single query image retrieval, and their sum, respectively, over 10 trial runs of three algorithms. From the table, we can see that our algorithm outperforms the other two algorithms and is faster than SVT by a factor of 30 in running time performance.

**Table 1. Running time performances.**

<table>
<thead>
<tr>
<th></th>
<th>BF (hour)</th>
<th>SVT (s)</th>
<th>KQTree (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree Construction</td>
<td>0</td>
<td>15122</td>
<td>465</td>
</tr>
<tr>
<td>Image Retrieval</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>4</td>
<td>15126</td>
<td>468</td>
</tr>
</tbody>
</table>

5 Conclusions

Content-based image retrieval demands for efficient search structures and algorithms. Design of searching methods that scale well with the size of the database and the dimensionality of the data is a challenging task. Trees present an efficient way to index local image regions, making the nearest neighbor search more efficient by pruning. A fast image retrieval approach with an indexing scheme significantly more efficient than the state-of-the-art SVT method has been presented. The approach is built upon a quantization tree that hierarchically organizes labeled descriptors from image keypoints. Experimental results have demonstrated the improved efficiency of the proposed method while with a similar effectiveness to that of the state-of-the-art SVT.

Acknowledgments

The authors would like to thank the Chinese National Science Foundation for its valuable support of this work under award 61473220 and all the anonymous reviewers for their valuable comments.

References


