Copyright Detection System for Color Based Image Retrieval

Zin Win Nge Nge Htwe, May Thu Lwin
Computer University, (Mandalay)
zwngenge@gmail.com

Abstract

Nowadays, a logo is the visual representation and forms the foundation of company and organization. Most of the companies want their logos are private and detect copyright rules. They do not want the logos of their companies duplicate. All the logos that have been registered are stored in an image database. This system intends to see if the new logo of an organization is very similar to any of the existing logos in the image database in order to avoid copyright infringements. Among image retrieval methods, color based image retrieval method is applied in this system. This system analyzes and implements three image retrieve methods to extract the desired image. These methods are K-D Tree, Vocabulary Tree and Locality Sensitive Hashing (LSH). Based on implementation and analysis, the system compares running time and accuracy percentage between these three image retrieve methods.

1. Introduction

Image processing is the used of automated or manual techniques to provide the means of accessing, preprocessing, extraction features, classifying and display the original or processed imagery for subjective evaluation, interpretation with data and retrieving data.

Image retrieval is retrieving similar images from an image database. The image database retrieval performance using the color set features causes retrieval effectives.

This paper intends to analyze and implement the comparison of query time and accuracy percentage of three query image retrieve methods such as K-D Tree, Vocabulary Tree and Locality Sensitive Hashing.

This paper is organized with six sections. The first section deals with introduction of the system. Section two explains related work for the system. The third section explains three retrieve methods used in this system. Section four describes the implementation of the system. Experimental result is explained in section five and the next section is conclusion of the system.

2. Related Work

There are a few number of work have been done for image retrieval methods. In this paper describes other systems presented by many researchers.

Shashank J, Kow shik P, Kannan Srinathan and C.V. Jawahar apply K-D tree, Vocabulary tree, and Locality Sensitive Hashing methods for the image contains private information. It uses for private content based image retrieval [1] with retrieving similar images from an image database without revealing the content of query image.

Yaniv Masler uses Quard tree, K-D tree, R-tree and Locality Sensitive Hashing methods in order to search nearest neighbor of the image [2].

Mei-Chen Yeh describes Multimedia Content Explosion by using Vocabulary tree method [3].

3. Copyright Detection System

To retrieve similar logo images in the image database, firstly the registered logos maintained in the image database. The images in the database and query image are indexed by using color features. By comparing the index structure of query image and the images in the image database, decide the similarity assessment of the query image.

In this system, the query image and the images in the database are composed of 150 pixels width and 100 pixels height. There are many image retrieval methods. Among them color based image retrieval methods are applied in this system by using K-D, vocabulary and LSH methods

The proposed copyright detection system includes two main fractions as shown in Figure 1. The images have been registered that can save in the database. The querying image is to retrieve the exactly or approximate similar images that the companies want to check whether their new logo images that have been kept in the database or not. If the images have already been registered but these do not have in the database, the registered images can be saved in the database.

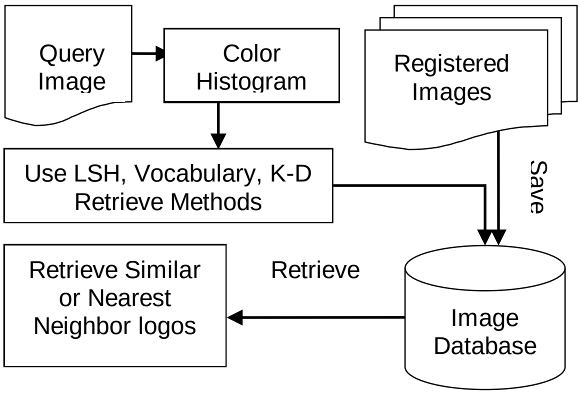

Figure 1. Overview of Proposed System Architecture

The query image is indexed by using color histogram. When retrieving the query image from the database, the system apply LSH, Vocabulary tree and K-D tree image retrieve methods. After retrieving, the system informs the most similar images that depend on the user defined threshold value.

3.1. Color Based Image Retrieval

Color indexing is a process in which the images in the database are retrieved on the basis of the image's color content. There are two color contents. These are global color and regional color. Indexing images by global color distribution has been achieved by using color histogram [4] [5]. The color histogram provides a good approach for retrieving images that have similar overall color content. Therefore, it is essential to utilize color indexing that captures regional color information.

Color regions are extracted through steps that transform, quantize and filter the image morphologically such that insignificant color information is lost and prominent color regions are emphasized. The color regions remaining in the image are extracted and represented using color sets that define the color content of the regions.

Firstly, the large single color regions are extracted and followed by multiple color regions. Then, utilize binary color sets to represent the color content which also allows for very efficient indexing. By removing single color region first, and then capture color content that is the most intuitive.

3.2. K-D Tree

The central data structure is a tree structure which forms the hierarchical representation of the mass distribution. The K-D tree is a balanced binary tree. The root-cell of this tree represents the entire simulation volume.

To build the K-D tree, start from the root-cell and bisect recursively the cells through their longest axis, so that an equal number of particles lie in each sub-volume. The simplicity of the structure and the availability of fast median finding algorithms allow for a very efficient tree construction.

Pointers are unnecessary since each node in the tree can be indexed so that the finding of children, parent and sibling nodes are simple bit shift operations. The K-D tree is a data structure for storing a set of points from a k dimensional space. Each intermediate node contained the split dimension and split value.

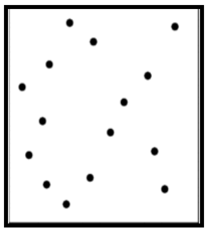

Figure 2. Input Image

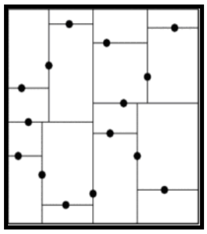

Figure 3. Output with K-D tree

The image in Figure 2 is a set S of n points in k dimensions. In Figure 3, K-D tree can constructs by partitioning point sets based on the points in the input image. Each node in the tree is defined by a plane through one of the dimensions that partitions the set of points into left/right (or up/down) sets, each with half the points of the parent node. These children are again partitioned equal halves, using planes through different dimension. Partitioning stops after reaching n levels, with each point in its own leaf cell. K-D tree constructs by inserting points incrementally and dividing the appropriate cell. The images were indexed by K-D tree using color histogram as the feature.

3.2.1. Color Histogram. Color histogram is popular method for characterizing image content. The color histogram for an image is constructed by counting the number of pixels of each color. The extraction of similar color progression includes selection of a color space, quantization of the color space, computation of histograms, derivation of the histogram distance function and identification of indexing shortcuts. The images in the image database were segmented into fixed blocks and each block was indexed separately.

3.3. Locality Sensitive Hashing

Locality Sensitive Hashing (LSH) [6] techniques are widely used for image retrieval. In the hashing technique employs a hash function: *h* which is used to divide the images in a database into bins. All the images with similar color value for the hash function are placed in a single bin. Given a query image, the hash function is applied to it and is mapped to a bin depending on the output value. Only the images in that bin are retrieved as the results of the query and thereby increasing the performance.

The data in the bin to which the query image is mapped by the hash function is to be retrieved. LSH is also similar to the general hashing technique except that it uses multiple hash functions, so as to ensure that for each function, the probability of collision is much higher of objects which are close together. The query image is hashed using all hash functions and the *k* nearest neighbors is computed from the images retrieved by each hash.

LSH is used for an approximated calculation of distances between the tuples of a table by using randomized hash functions. Each hash function produces a partition of the set of tuples and each block of this partition consists of tuples that collide under that hash function. It is used to enhance the agglomerative hierarchical clustering of the single link method. The close objects are likely to collide under a high number of randomly chosen hashing functions. Both of these techniques compute the real distances between objects residing in the same blocks.

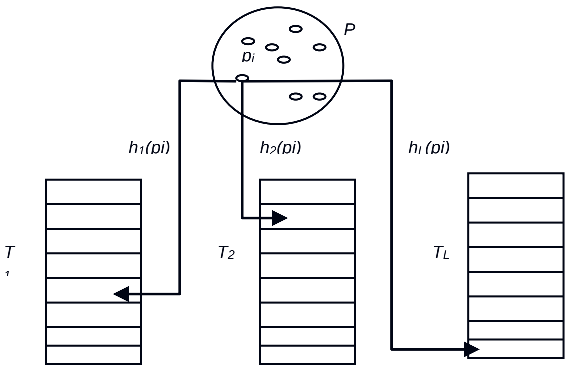

Figure 4. Locality Sensitive Hashing Algorithm

Figure 4 demonstrates the algorithm for LSH which composed of four stages. Firstly, choose several hash function from 1 to *l*, where *l* is the number of hash table *T*, in order to store the RGB color pixels for the images in the relative hash table. And then initialize a hash array for each h_i. Store each point *p* in the bucket $h_i(p)$ of the ith array, *i*=1 to *l*. Finally, answer query for each *i*=1 to *l* and return the closest point found.

3.3.1. Hash Tree Based Retrieval. LSH is also similar to the general hashing technique except that it uses multiple hash function, so as to ensure that for each function, the probability of collision is much higher of objects which are close together. The hash function is applied within each bin to construct a hash tree. The obtained hash bins are further hashed, thus forming a hash tree.

3.4. Vocabulary Tree

The vocabulary tree can be defined the quantization. Feature space is quantized through K-means clustering and builds into a means clustering and builds into a vocabulary tree. The vocabulary tree is used to index the database. Each high-dimension feature vector is quantized into an integer which corresponds to a path in the vocabulary tree. Query image is hierarchically quantized in a vocabulary tree using K-means clustering.

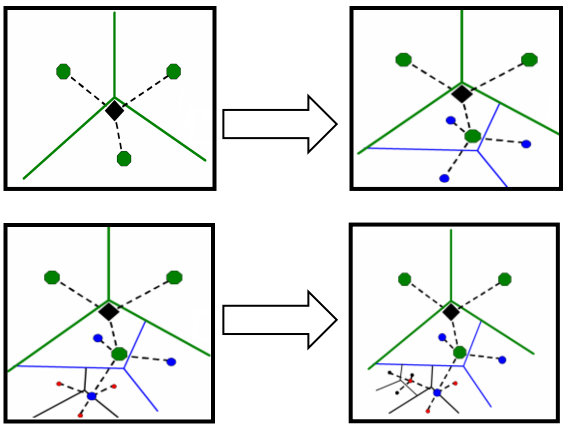

Figure 5. Vocabulary Tree Construction

3.4.1. K-Mean Clustering. Firstly, an individual K-Mean 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 descriptors vectors closest to a particular cluster center.

The same process is then recursively applied to each 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 L, and each division into k parts is only defined by the distribution of the descriptor vectors that belong to the parent quantization cell. This process is illustrated in Figure 5 which shows the example for the hierarchical quantization at each level by k center where k is equal to 3.

3.4.2. Vocabulary Tree Indexing. In an incremental dataset, the vocabulary tree is used to generate the vector of query image. However, the distribution of a new image is diverse comparing to original dataset. At this time, the performance would be continuing to decrease with coming of new images. In order to deal with incremental dataset, scalable recognition model is applied to adapt data increase and removal in an incremental dataset. To maintain an effective recognition model over renewed dataset, vocabulary tree adaptation is necessary. Scale-invariant feature transform (SIFT) features of a new data batch are reindexed using the original vocabulary tree.

3.4.3. SIFT Feature. Scale-invariant feature transform (SIFT) feature in an image was used to obtain a vocabulary of visual words. SIFT feature algorithm plays four stages. These are scale-space extrema detection, key point localization, orientation assignment and key point descriptor. The first stage searches over scale space in order to identify potential interest points. Key point localization is the location and scale of each candidate point is determined and keypoints are selected based on measures of stability. One or more orientations are assigned to each keypoint based on image gradient in the orientation assignment stage. A keypoint descriptor is generated for each keypoint from local image gradients information at the scale found in keypoint localization stage.

4. Implementation of the System

This paper describes the comparison of query time and accuracy percentage of three image retrieval methods. The query image is quantized by using color histogram. Size can easily be normalized so that different image histogram can be compared. But the system limits the size of images which depicts in Figure 5. This figure describes the query image with 150 pixels width and 100 pixels height and the RGB color value for 150x100 dimensions.

By using the color value, the system retrieves images from the database that are absolutely or relatively similar to the query image The images in the database are registered by the other organizations in order to detect copyright rules. The number of images increase in the image database can cause an increase in the retrieval times. The system can retrieve at least fifty percent similar to the query image and search at most ten nearest neighbor images from the database.

After retrieving, the organization knows their logo has been registered in the image database or not. If the query image is in the database, it cannot be duplicated because of detecting copyright rules. If it is not in the database, it can be saved and protected the logo image not to be used by other organization.

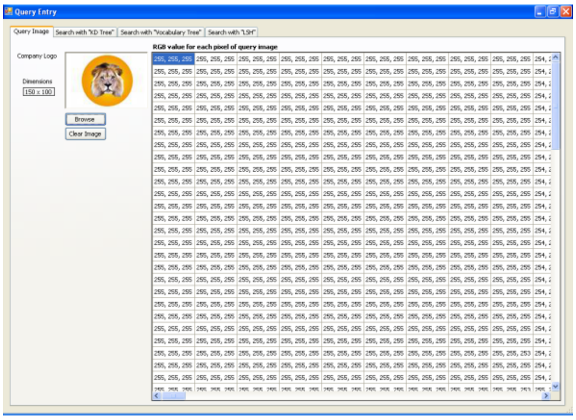

Figure 5. The RGB Color Value for Query Image

The copyright detection system aims to avoid copyright infringement among organizations, to know the concept of three image retrieval methods and to compare query times and accuracy percentage of these three methods.

5. Experimental Result

The experimental results of query time and accuracy percentage by using K-D tree, Vocabulary tree and Locality Sensitive Hashing methods are shown in Table 1, Table 2, Figure 1 and 2. These query time and accuracy percentage are tested over fifty registered images dataset.

The query time for each of the three retrieve methods have been reported in Table1. But the queries times can vary depend on the database size. When one to ten nearest neighbor images are indicated, these three retrieve methods obtain the following milliseconds to retrieve the similar images with query image. In Table1 demonstrates the results of comparing query time when searching at most ten nearest neighbor images based on fifty percent similar threshold value.

To evaluate the test results we use the values precision and recall. The used parameters are describes as:

RCR= relevant cases retrieved; **RCI**= irrelevant cases retrieved; **RC**= retrieved cases; **RCB**= relevant cases in case base.

Precision P=**RCR**/**RC**, where **RC**=**RCR**+**RCI**
Recall R=**RCR**/**RCB**, where **RCB** $\subset$ **RCR**

Table 2 describes the accuracy percentage of the query image based on similarity threshold value (k=1,2,.....,10), but such result depends on the dataset in the database. This table illustrates for only one tested input query image. For the single query image, the system can compare at most ten nearest neighbor images. When the one who is searching over one nearest neighbor image, the accuracy percentage may be hundred percent similar with the tested query image then the query image is situated in the database. Otherwise, two to ten nearest neighbor images are explored the percentage may be altered depend on the datasets. Figure 1 shows the comparison of query times and Figure 2 shows the comparison of accuracy percentage by chart which is depending on the recall method.

Table1. Query Time Comparison

Similar Img.	K-D millisecond	Vocabulary millisecond	LSH millisecon d
2img	48930	1562	47488
4img	48730	1562	48059
6img	48489	1672	48119
8img	76409	4697	67448
10img	102527	4852	67948

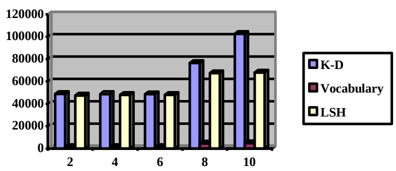

Figure1. Query Time Comparison Chart

Table2. Accuracy Percentage Comparison

Similar Img	K-D		Vocabulary		LSH	
	P %	R %	P %	R %	P %	R %
1img	2	100	2	100	2	100
2img	4	50	4	67	4	67
3img	6	42	6	60	6	60
4img	8	57	8	57	8	57
5img	10	71	10	50	10	50
6img	12	85	12	50	12	50
7img	14	46	14	47	14	47
8img	16	43	16	47	16	47
9img	18	40	18	45	18	45
10img	20	56	20	45	20	45

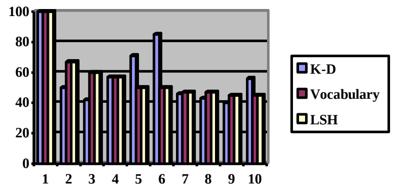

Figure2. Accuracy Percentage Comparison Chat (Recall)

6. Conclusion

In this paper, the system briefly describes the comparisons of three image retrieve mechanisms. After making testing and comparing of three methods, the query time for retrieving at most ten nearest neighbor images are distinctly different. The query time for K-D tree takes long time by comparing with LSH and Vocabulary tree. Among them LSH is not as fast as Vocabulary tree. So that Vocabulary tree method is the fastest one. But the accuracy percentage of the K-D tree method is able to be flexible and its results are not stable which is depending on the datasets. So, the Vocabulary tree and LSH are nearly equal in accuracy percentage. These accuracy percentages are tested according to only one query image.

This paper can be used in any large organizations that ask for the privacy of the companies' logos. The new organization can know their new logo has been registered by other organization. It can also be registered image with a little different color value. The system was tested on images with 150 pixels width and 100 pixels height resolution. So, it can be extended for the image with any pixels width and height resolution and also can retrieve any number of nearest neighbor images from the image database.

7. References

[1] J. Shashank, P. Kow shik, Kannaro Srinathan and C.V Jawahar, "Private Content Based Image Retrieval", India.

[2] Mei-Chen Yeh, "Approximately Sequence Matching for Fast Visual Retrieval", California, 2009.

[3] Yaniv Masler, "Nearest Neighbor and Locality Sensitive Hashing", IDC, 16-03-2008.

[4] Faloutsos C., Flickner M., Niblack W., Petkovic D., Equitz W., and Barber R, "Efficient and Effective

Querying by Image Content", IBM Research Journal, No.9453 (83074), August 3, 1993.

[5] Swain M., and D. Ballrad, "Object Recognition from Local Scale- Invariant Feature".

[6] Mayur Datar, Nicole Immorlica, Piotr Indyk, Vahab Mirrokni, "Locality Sensitive Hashing Scheme Based on p-Stable Distributions".

[7] Aimin Feng, Xuejunliu, Bin Chen, "Structured learning from Data for Novelty Detection by Linear Programming", Hong Kong, March18-20 2000.