An Empirical Study on Large-Scale Content-Based Image Retrieval

Yuk Man Wong, Steven C. H. Hoi, and Michael R. Lyu

Department of Computer Science & Engineering, The Chinese University of Hong Kong
Shatin, Hong Kong S.A.R., China, Email:{ymwong, chhoi, lyu}@cse.cuhk.edu.hk

ABSTRACT

One key challenge in content-based image retrieval (CBIR) is to develop a fast solution for indexing high-dimensional image contents, which is crucial to building large-scale CBIR systems. In this paper, we propose a scalable content-based image retrieval scheme using locality-sensitive hashing (LSH), and conduct extensive evaluations on a large image testbed of a half million images. To the best of our knowledge, there is less comprehensive study on large-scale CBIR evaluation with a half million images. Our empirical results show that our proposed solution is able to scale for hundreds of thousands of images, which is promising for building web-scale CBIR systems.

1 INTRODUCTION

Multimedia data including images and videos have been dramatically increased in our life due to the popularity of digital devices and personal computers. Huge volumes of media data bring a lot of critical challenges of retrieving contents from large-scale multimedia databases effectively. In recent years, multimedia information retrieval has attracted more and more attention in research community. One of the most popular and fundamental research topics is content-based image retrieval (CBIR) [1]. Although CBIR has been extensively studied in both academia and industry for many years, there are still a number of challenging issues, which do not have very effective solutions until now. These issues are often related to some long-standing challenges among several interdisciplinary research areas, including database, computer vision, information retrieval, and machine learning.

In general, a CBIR system at least consists of four modules: data acquisition and processing, feature representation, data indexing, query and feedback processing. Among them, an efficient data indexing solution is critical to making CBIR systems scalable to large-scale real-world applications. Although data indexing techniques have been well studied in database community, traditional indexing solutions are still not efficient enough for indexing image contents, which are often represented in high dimensional feature spaces. How to develop an efficient high dimensional data indexing solution is challenging and significant for building large-scale CBIR systems.

In this paper, we propose a scalable CBIR scheme by applying an emerging data indexing technique, locality-sensitive hashing (LSH) [2, 3], which enjoys some significant advantages for indexing high dimensional data. In addition to the proposed scheme, one main contribution in this paper is the comprehensive empirical evaluation of CBIR over a half million images. To the best of our knowledge, there is very limited empirical study for performance evaluation on large-scale CBIR systems. Moreover, we also address some challenges for building scalable CBIR systems on large-scale data and suggest some innovative ideas for tackling these issues.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 presents our proposed scalable CBIR solution with LSH. Section 4 discusses our feature representation methods. Section 5 gives our extensive evaluations on a large-scale image testbed. Section 6 sets out our conclusion.

2 RELATED WORK

Content-based image retrieval has been a very active research topic in multimedia community since the early 1990s. A lot of CBIR systems have been built in both industry and academia. Some famous systems includes IBM's QBIC [4], Virage [5], MIT's Photobook [6], VisualSEEK [7], and MARS [8], etc. Numerous papers have been published in the past decades. It is impossible to cite all related work here. Readers can refer to two comprehensive surveys on CBIR research in [1, 9].

For CBIR database indexing, a lot of traditional multidimensional indexing techniques, such as k-d tree, quad-tree, R-tree and its variant R^+-tree and R^*-tree, have been applied to CBIR. Some reviews and comparisons of these traditional indexing techniques in CBIR can be found in [10]. However, these traditional indexing techniques were usually not scalable to dimensions higher than 20 [10].

Recently, a promising indexing technique, locality-sensitive hashing, was proposed for solving the near neighbor search in high dimensional spaces efficiently [2, 3]. LSH has been attracted more and more attention in multimedia applications, such as video retrieval [11] and image replica detection [12]. However, there is still little empirical study of its performance on large-scale CBIR applications.

3 A SCALABLE CBIR SCHEME

3.1 Overview of Our Solution

Due to the curse of dimensionality, traditional fast similarity searching techniques are not better than exhaustive linear search when the dimensionality of the data is high. Locality-sensitive hashing is proposed to attack this problem. Previous empirical results have shown that LSH significantly outperformed tree-based techniques such as kd-tree and SR-tree for indexing high dimensional data. M. Datar et al. reported that LSH searches for near neighbors 30 times faster than k-d tree does when the dimensionality of the dataset goes beyond 200 [3]. For CBIR, image contents are often represented in

high dimensional spaces. Therefore, we suggest to employ LSH for indexing image contents in our CBIR solution.

3.2 Locality-Sensitive Hashing

As we known, the time complexity of database lookup of a hashing algorithm is $O(1)$ on average, which means a database lookup usually takes constant amount of time independent of the size of the database. Thus, hashing algorithm is commonly employed for indexing large-scale database. However, traditional hashing algorithms are never adopted for similarity indexing. This is because traditional hashing algorithms build index for searching identical copy of the query key, but not for searching near neighbors. Locality-sensitive hashing, an emerging new indexing algorithm, is recently proposed to solve high-dimensional near neighbor searching problem in Euclidean space l_2 and therefore is very suitable for indexing high dimensional data in CBIR applications.

LSH was first proposed by Indyk & Motwani [2] and further improved in [3]. LSH can answer queries in sublinear time with each near neighbor being reported with a fixed probability. LSH assumes that the probability that two points share the same hash value decreases when the distance between them increases. By using LSH, given a query, we can search its near neighbors very fast simply by computing its hash and examining only the points with which it collides. This enable us to build large-scale scalable CBIR systems.

For building a scalable system, we employ the E^2LSH (Exact Euclidean LSH) package [1], an implementation of LSH that solves exact R-near neighbors problem, to build the index of system's database. The exact R-near neighbor problem is formulated that given a query point q, each point p satisfying $||p - q|| \leq R$ has to be returned with a certain probability. This approach ensures that the K nearest neighbors of any query point can be found at some certain probability. The probability of finding near neighbors can be controlled by two parameters L and k, where L is the number of independently chosen hash functions, and each of these functions is a concatenation of k hash functions. Roughly speaking, a larger k reduces the chance of hitting data points that are not R-near neighbors while a larger L increases the probability of finding all R-near neighbor.

3.3 Our Scalable Implementation

The application of LSH for solving large-scale CBIR is not straightforward. One main problem is that E^2LSH is a main memory based implementation. It can answer a given query very efficiently. But it requires all the data points and the R-near neighbor data structures to be stored in the main memory for fast memory access. Therefore, the maximum database size it can handle at the same time is limited by the the amount of free main memory available in the machine. This bottleneck limits its direct application for large-scale database.

To enable our CBIR system scalable to large-scale data, we propose a multi-partition indexing approach to tackle the above issue. Specifically, we divide the whole database into multiple partitions so that each of them is small enough to be loaded into the main memory. Then we process queries on each partition respectively. The details of our multi-partition indexing and query processing solution are given as follows:

1. Divide the database into n partitions, where the number of partitions $n = \lceil \frac{database\ size}{max.\ partition's\ size} \rceil$.
2. Run E^2LSH indexing on each of the partitions to create the hash tables;
3. Given a query q, load the pre-built hash tables T of one partition into the main memory;
4. Run E^2LSH query on T to retrieve top k ranking images with respect to q;
5. Repeat (3) and (4) steps until all partitions are processed.
6. Collect the results from all partitions and return top k ranking results with respect to the query q.

Remark. The above solution is able to handle very large-scale database without any size limit. However, there are yet some critical issues to apply the above solution for huge web-scale applications. One main limitation of the above approach is the disk-access overhead for loading the hash tables into the main memory. When the database size is very large, the disk-access time will be a critical problem. To tackle this issue, we can consider some parallel solutions, such as cluster machines techniques, to overcome the disk-access overhead and speedup the overall solution in future work.

4 FEATURE REPRESENTATION

Feature representation is a key challenging step for building CBIR systems, which has been widely studied. In our solution, three types of visual features are used: color, shape and texture [13, 14].

For color, we use Grid Color Moment (GCM). Each image is partitioned into 3×3 grids and three types of color moments are extracted for each grid. Thus, an 81-dimensional color moment is adopted for color feature.

For shape, we employ an edge direction histogram. A Canny edge detector is used to get the edge image and then the edge direction histogram is computed. Each histogram is quantized into 36 bins of 10 degrees each. An additional bin is used to count the number of pixels without edge information. Hence, a 37-dimensional edge direction histogram is used for shape feature.

For texture, we adopt Gabor feature. Each image is scaled to 64×64. Gabor wavelet transformation is applied on the scaled image with 5 scale levels and 8 orientations, which results in 40 subimages. For each subimage, three moments

[1] http://web.mit.edu/andoni/www/LSH/index.html

are computed: mean, variance, and skewness. Thus, a 120-dimensional feature vector is adopted for texture feature.

In total, a 238-dimensional feature vector is employed to represent each image in our image database.

5 EMPIRICAL EVALUATION

5.1 Experimental Testbed

To examine the scalability of our proposed scheme, we have collected a testbed containing 500,000 images crawled from Web. Further, to enable an objective evaluation automatically, an additional set of 5,000 images from COREL image CDs are engaged as the query set for performance evaluation. This image set contains 50 categories. Each category in the dataset consists of exactly 100 images that are randomly selected from relevant examples in the COREL image CDs. Every category represents a different semantic topic, such as *antique*, *antelope*, *aviation*, *balloon*, *botany*, *butterfly*, *car*, *cat*, *dog*, *firework*, *horse* and *lizard*, etc.

The motivation for selecting the COREL images in semantic categories for evaluation is twofold. First, it allows us to evaluate whether the proposed approach is able to retrieve the images that are not only visually relevant but also semantically similar. Second, it allows us to evaluate the retrieval performance automatically, which will significantly reduce the subjective errors relative to manual evaluations. This evaluation methodology has been widely adopted in previous CBIR research [14].

5.2 Performance Metrics

We evaluate the retrieval performance of our CBIR system based on two standard performance metrics: *precision* and *recall*. The *precision* is calculated by the ratio of the number of relevant images in the returned images and the total number of returned images, while the *recall* is calculated by the ratio of the number of relevant images in the returned images and the total number of relevant images in the database.

The returned image is judged as the relevant image with respect to a query image if it is one of the 100 COREL images in the database which share the same category with the query image. Since each selected COREL category contains 100 images only, we assume total number of relevant images for each query is equal to 100 in the database. To evaluate the efficiency of our CBIR system, we measure the average CPU time elapsed on a given query.

5.3 Experimental Setup

To choose a query set for performance evaluation, we randomly select 20 images from each of the 50 COREL image categories, and form the query set of 1000 image examples. To evaluate how our CBIR system performs with respect to different database scales, we prepare 10 image databases of different sizes ranging from 50,000 images to 500,000 images. For a database of size N contains 5,000 COREL images and N-5000 other images selected from our testbed.

Table 1: Time Performance of LSH over ES on different databases

Size	ES (ms)	LSH (ms)	CPU Speedup
50000	32.3	7.6	4.246
100000	64.7	14.3	4.507
150000	96.9	22.8	4.244
200000	129.7	30.3	4.271
250000	160.7	37.3	4.306
300000	192.9	43.6	4.417
350000	224.8	49.8	4.508
400000	256.6	55.3	4.637
450000	288.4	62.4	4.617
500000	320.0	68.9	4.640

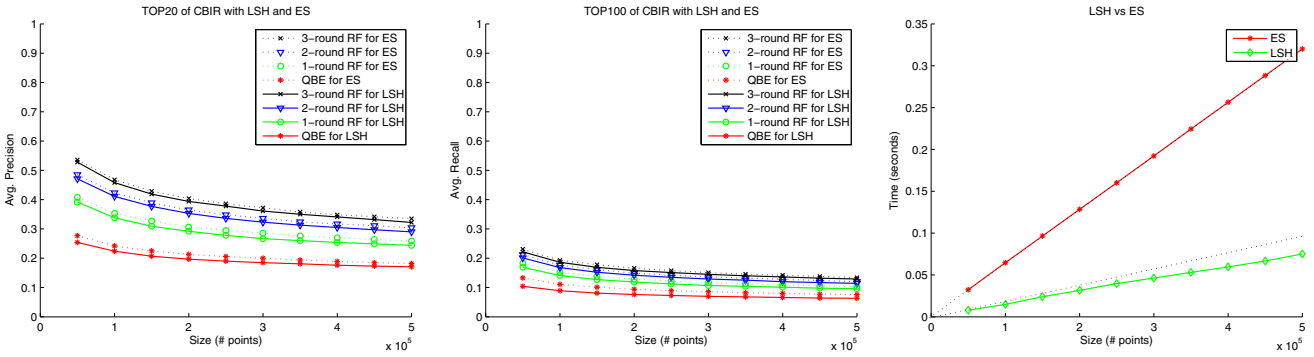
We employ the techniques described in section 4 to extract the image features (color, shape and texture) of the query and database images, in which each image is represented by a 238-dimensional feature vector.

For each of the 10 databases, we perform two sets of experiments to compare our LSH solution with the traditional methods. One set of the experiments employs our proposed LSH solution with parameters $L = 550$ and $k = 34$, while the other set uses an exhaustive linear search (ES) method based on Euclidean distance similarity measure. In each experiment, we query the database with a set of query images. For each query image, we retrieve the top 20 and top 100 ranking images. The returned ranking images are checked against the ground truth to figure out the relevant images. Based on the number of relevant images, we measure the precision and recall rates. The recall and precision rates for all queries are averaged for final performance comparison. In each experiment, we simulate two rounds of relevance feedback. Relevance feedback is based on a k-Near Neighbors approach [14] by re-querying the database with the relevant examples with respect to the given query.

All of our experiments are conducted on a 3.2GHz Intel Pentium 4 PC with 2GB memory running Linux kernel 2.6. All implementations are programmed by C++ language.

5.4 Experimental Results

Fig. 1(a) shows the experimental results of average precision by using LSH and ES respectively in our CBIR system. In this figure, top 20 ranking images are returned for evaluation. The "Query-By-Example" (QBE) curve represents the average precision without any relevance feedback. The "1-round RF" curve represents the average precision when one round of relevance feedback is applied to refine the QBE result. The "2-round RF" curve shows the average precision when another round of relevance feedback is applied to refine the "1-round RF" result. From the figure, we can observe that the results of LSH is very close to the ES results. Their maximal difference is no more than 5% at any database size. Fig.1(b) shows the average recall of our CBIR system using LSH and ES respectively, in which top 100 ranking images are returned for evaluation. From Fig.1(a) and (b), we can



(a) Avg. Precision of TOP 20 (b) Avg. Recall of TOP 100 (c) Avg. Query Time

Fig. 1*:* Graphs showing the average recall, the average precision, and the query time of the CBIR system.

see that the average recall and average precision of our CBIR system decrease with the database size, yet the decreasing rate diminishes when the database size increases. Fig.1(c) shows the average query time performance of our CBIR system with LSH versus that with ES. From the results in Fig.1(c), we can see that the query time for ES is linear to the database size, while the one for LSH is sublinear. To examine the details, we also show the time performance comparison in Table 1. We can see that the LSH approach is much faster than the ES solution with an average speedup greater than 4 times. And the gap of time performance between them grows even faster when the database size increases. In our future work, we will consider some parallel solutions, which can further improve the efficiency of our current solution.

6 CONCLUSION

In this paper we proposed a scalable content-based image retrieval scheme based on a fast high dimensional indexing technique, locality-sensitive hashing. We conducted extensive empirical evaluations on a large testbed of a half million images. The promising empirical results show that our scalable CBIR system is more efficient than traditional exhaustive linear searching methods and can be scalable to large-scale CBIR applications. We also addressed some limitations and challenges in our current solution and suggested some innovative solutions for future research. We believe these empirical experiences are valuable for future research and developments on web-scale CBIR applications.

7 ACKNOWLEDGMENTS

We appreciate Alexandr Andoni, the author of the E^2LSH package, for his valuable comments on the LSH implementations. The work described in this paper was fully supported by a grant from the Shun Hing Institute of Advanced Engineering (SHIAE), CUHK.

8 REFERENCES

[1] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, 2006.

[2] Piotr Indyk and Rajeev Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proc. 13th ACM Symposium on Theory of computing*, USA, 1998, pp. 604–613.

[3] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 20th Annual Symposium on Computational Geometry*, New York, NY, USA, 2004, pp. 253–262.

[4] W. Niblack, R. Barber, and et al., "The QBIC project: Querying images by content using color, texture and shape," in *SPIE Storage and Retrieval for Image and Video Databases*, 1994.

[5] J. R. Bach, C. Fuller, and et al, "The virage image search engine: An open framework for image management," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996, vol. 2670, pp. 76–87.

[6] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *IJCV*, vol. 18, no. 3, pp. 233–254, 1996.

[7] John R. Smith and Shih-Fu Chang, "Visualseek: A fully automated content-based image query system," in *ACM Multimedia*, 1996, pp. 87–98.

[8] S. Mehrotra, Y. Rui, O.-B. Michael, and T. S. Huang, "Supporting content-based queries over images in mars," in *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, 1997.

[9] Y. Rui, T. Huang, and S. Chang, "Image retrieval: current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 4, pp. 39–62, Apr. 1999.

[10] David A. White and Ramesh Jain, "Similarity indexing: Algorithms and performance," in *Storage and Retrieval for Image and Video Databases (SPIE)*, 1996, pp. 62–73.

[11] Shiyan Hu, "Efficient video retrieval by locality sensitive hashing," in *Proc. IEEE ICASSP*, 2005, vol. 2, pp. 449–452.

[12] Arun Qamra, Yan Meng, and Edward Y. Chang, "Enhanced perceptual distance functions and indexing for image replica recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*, vol. 27, no. 3, pp. 379–391, 2005.

[13] Steven C. H. Hoi and Michael R. Lyu, "A novel log-based relevance feedback technique in content-based image retrieval," in *Proc. ACM Multimedia Conference*, New York, US, Oct. 10–16 2004.

[14] Steven C. H. Hoi, Michael R. Lyu, and Rong Jin, "A unified log-based relevance feedback scheme for image retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 509–524, 2006.