Crawling, Indexing, and Similarity Searching Images on the Web

(Extended Abstract)

M. Batko[1] and F. Falchi[2] and C. Lucchese[2] and D. Novak[1] and R. Perego[2] and F. Rabitti[2] and J. Sedmidubsky[1] and P. Zezula[1]

[1] Masaryk University, Brno, Czech Republic
[2] ISTI-CNR, Pisa, Italy

Abstract. In this paper, we report on our experience in building an experimental similarity search system on a test collection of more than 50 million images, to show the possibility to scale Content-based Image Retrieval (CBIR) systems towards the Web size. First, we had to tackle the non-trivial process of image crawling and descriptive feature extraction, performed by using the European EGEE computer GRID, building a test collection, the first of such scale, that will be opened to the research community for experiments and comparisons. Then, we had to develop indexing and searching mechanisms which can scale up to these volumes and answer similarity queries in real-time. The results of our experiments are very encouraging for future applications.

1 Introduction

With the widespread use of digital cameras more than 80 billion photographs are taken each year, and a large part of it will be put on the web. The management of digital images promises to emerge as a major issue in the next years. In this context, the interest for Content-Based Image Retrieval (CBIR) systems is rapidly growing since they are a possible answer in the management of such data. A recent survey [1] reports on 56 systems, most of them exemplified by prototype implementations where the typical size database is counted in thousands of images. However, to be able to be relevant with respect to the new challenges posed by the vast amount of images on the web, CBIR systems should scale up their target, shifting from small scale, often highly specific, datasets to a much larger scale.

The scalability challenge is the focus of the European project SAPIR (Search on Audio-visual content using Peer-to-peer Information Retrieval)[3] that aims at finding new ways to analyze, index, and retrieve the tremendous amounts of speech, image, video, and music that are filling our digital universe, going beyond what the most popular search engines are still doing, that is, searching using text tags that have been associated with multimedia files.

[3] SAPIR European Project, IST FP6: http://www.sapir.eu/

In this work, we show that the techniques we are proposing are able to scale to the desired volume of images while answering similarity queries in real-time. Since our approach in similarity searching is based on metric spaces, all the index and search mechanisms can be straightforwardly applied to various data types, for example, face recognition, music, video clips, etc. We have established our scalability challenge targeting a test image collection of 50 million images, aiming at a search time of about 1 s.

The rest of the paper is organized as follows. Section 2 describes the process of building the test collection. Section 3 describes our indexing and searching mechanisms, starting from a centralized solution towards a sophisticated distributed system for 50 million images. Finally, Section 4 analyzes the results obtained, discussing also future research directions, opened by our achievements.

2 Building the Image Collection

Collecting a large amount of images for investigating CBIR performance issues is not an easy task, at least from a technological point of view. Since the absence of a publicly available collection of this kind has probably limited the academic research in this interesting field, we tried to do our best to overcome these problems. The main issues we had to face were the identification of a valid source, the efficient downloading and storing of such a large collection of images and the efficient extraction of metadata (MPEG-7 visual descriptors and others) from the downloaded images.

In choosing the Data Source, crawling the Web is the obvious solution if you are looking for a practically unlimited source of data. However, the time and bandwidth required can be huge: according to [2] to gather 50 million images, we would have to download and parse 325 million to 0.75 billion Web pages. Moreover, a large portion of these image can be decoration elements like buttons, bullet list icons, and many other are very small images or photo thumbnails.

For the need of high-quality data, we finally decided to crawl one of the most popular photo sharing sites born in the last years with the goal of providing permanent and centralized access to user-provided photos. This approach has several advantages over the general Web approach for image quality, collection stability, richness of metadata (i.e., considering user provided tags and comments). Among the most popular photo sharing sites, we chose to crawl Flickr, since it is the one with the richest additional metadata and provides an efficient API[4] to access its content at various levels.

It is well known that the graph of Flickr users, similarly to all other social media applications, is scale free [3]. We thus exploited the small-world property of this kind of graphs to build our huge photo collection. By starting from a single Flickr user and following friendship relations, we first downloaded a partial snapshot of the Flickr graph. We then exploited the Flickr API to get the whole list of public photo IDs owned by each of these users. In this way we have created a 4.5 GB file with 300 million distinct photo IDs.

[4] http://www.flickr.com/services/api/

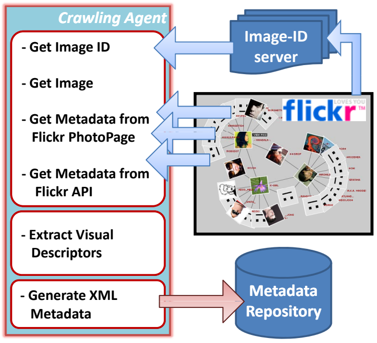

Fig. 1. Organization of the crawling and feature extraction process.

Since the purpose of the collection is to enable a general experimentation on various CBIR research solutions, we decided to retrieve almost all information available. Thus, for each photo: title and description, identification and location of the author, user-provided tags, comments of other users, GPS coordinates, notes related to portions of the photo, number of times it was viewed, number of users who added the photo to their favourites, upload date, and, finally, all the information stored in the EXIF header of the image file. Naturally, not all these metadata are available for all photos. In order to support content based search, we extracted several MPEG-7 *visual descriptors* [4] from each image. A visual descriptor characterizes a particular visual aspect of the image. They can be, therefore, used to identify images that have a similar appearance. Visual descriptors are represented as vectors, and the MPEG-7 group proposed a distance measure for each descriptor to evaluate the similarity of two objects [5]. Finally, we have chosen five MPEG-7 visual descriptors Scalable Colour, Colour Structure, Colour Layout, Edge Histogram, Homogeneous Texture.

Unfortunately, the extraction of MPEG-7 visual descriptors from high-quality images is very computationally expensive: this software running on a AMD Athlon XP 2000+ box takes about 4 seconds to extract the above five features from an image of size 500×333 pixels. Therefore, even without considering the time needed to download the image and all additional network latencies involved, we can estimate that a single standard PC would need about 6 years to process a collection of 50 million images. It was thus clear that we needed a large number of machines working in parallel to achieve our target collection of 50 million images in a reasonable amount of time. For this reason, we developed an application that allows to process images in parallel on an arbitrary (and dynamic) set of machines. This application is composed of three main components: the *image-id server*, the *crawling agents*, and the *repository manager* as shown in Figure 1.

We have considered GRID to be the right technology to obtain large amount of computing power we needed. GRID is a very dynamic environment that allows to transparently run a given application on a large set of machines. In particular, we had the possibility to access the EGEE European GRID infrastructure provided to us by the DILIGENT IST project[5]. We were allowed to use 35 machines spread across Europe. We did not have an exclusive access to these machines and they were not available all the time. Moreover, they were very different in both hardware and software installed.

In a GRID infrastructure, the user does not have a full control on the time and location where the job runs. In our case, the job always first downloads a totally self-contained crawling agent package from our repository-manager, and then runs the software contained in the package. The GRID provides a best-effort service, meaning that a job submitted to the GRID may be rejected and never executed: out of 66,440 jobs submitted, only 44,333 were successfully executed that means that 33,3 % of the jobs failed for GRID resources unavailability. Not all of the GRID machines were available through the crawling period and, therefore, we also used a set of local machines in Pisa which processed the images during the GRID idle time. We thus reached the total of 73 machines participating in the crawling and feature extraction process.

The result of this complex crawling and image processing activity is a test collection that served as the basis of our experiments with content-based image retrieval techniques and their scalability characteristics, in the context of the SAPIR project. Given the effort required in building such test collection, and the potential interest of the international research community to make experiments in large-scale CBIR, we decided to make it available outside the SAPIR project scope as the CoPhIR (Content-based Photo Image Retrieval) Test Collection, managed by ISTI-CNR research institute in Pisa[6]. The data collected so far (about 54 million images and related metadata) represents the world largest multimedia metadata collection available for research purposes, with the target to reach 100 million images till the end of 2008.

3 Content-based Image Retrieval

Building a large-scale system for efficient image similarity search is a tough and exciting challenge to face. Such system would put to test proclamations about the theoretical scalability of solutions on which the system is built. This section maps the route we followed from thousands of images towards tens of millions.

We perceive the content-based retrieval problem as a triangle: the type of *data* we have and the way of assessing the similarity, the *indexing techniques* that are used to enhance the efficiency, and the *infrastructure* the system is running on. The data we index and search is formed by the five MPEG-7 features described above, modeled as a single metric space [6].

All index and search techniques are based on the metric space model, written in Java and implemented over the same framework - the Metric Similarity Search Implementation Framework (MESSIF) [7]). The MESSIF contains the encapsulation of the metric space model and it provides a number of modules from a

[5] http://www.diligentproject.org/

[6] http://cophir.isti.cnr.it

basic storage management, over a wide support for distributed processing, to automatic collect performance statistics.

100K: Centralized Searching. Our first indexing experience with the Flicker images was with a rather small dataset of 100,000 images. We have used the M-Tree [8] technique which is de-facto a standard for similarity searching based on metric technology. We have decided to implement the Pivoting M-Tree (PM-Tree) extension [9], which employs additional filtering by means of precomputed distances between all objects stored in the structure and a fixed set of pivots, which improves search space pruning.

To establish a baseline, we have compared our results with a sequential scan algorithm. Our implementation confirmed the well-known fact that a query evaluation spend most of the time in metric function computations. M-Tree enhances the search performance significantly but the response times grow linearly as we increase the size of the dataset. It means that we can use the M-Tree structure for indexing several tens of thousands images, but we cannot go much farther if we require on-line responses.

1M: Distributed Searching A natural way to overcome the limitations of the centralized resources is to shift towards distributed environments. Such approach allows to exploit parallelism during the query processing and also provides an easily enlargeable and virtually unlimited storage capacity. Our similarity distributed structures [10] are based on the paradigm of Structured Peer-to-Peer (P2P) Networks. The system consists of *peers* – nodes which are equal in functionality. Each peer manages a part of the overall dataset and maintains a piece of navigation information which allows to route a query from any peer to the one with relevant data. In the case of similarity indices, the query is typically routed to multiple peers which search their local data and compute partial answers. The originating peer then gathers all the partial answers and merges them into the total result of the query.

The M-Chord [11] is a P2P data structure for metric-based similarity search which is used in our system. The M-Chord applies any one-dimensional P2P protocol in order to divide the data domain among the peers and to provide navigation within the system. The search algorithms exploit the features of the data-mapping to navigate the queries only to relevant peers. Every active node of the system takes over responsibility for an interval of keys. More information about M-Chord can be found in [11, 12]. From our experiments, we can see that the search times increase only slightly with the number of nearest objects (k). The response times about 500 ms are achieved.

10M: Approximate Distributed Searching We tried to increase the utilization of the hardware building a 10M network on the same HW configuration. We have observed that majority of the answering peers contribute to the answer result only negligibly or not at all and they are accessed only to proof the preciseness of the answer. We decided to allow a moderate approximation and thus prune the number of accessed peers significantly. We have reached this goal and thus managed to run the 500-peers network with 10M images on a relatively modest hardware configuration.

The approximate evaluation-strategy for the *kNN* queries in M-Chord [12] is based on the *relaxed branching* policy [13]. The basic idea is to examine highly-promising data partitions only and ignore those with low probability of containing qualifying data. Thus, the most promising parts of the most promising clusters are explored in M-Chord. The clusters are selected using a tunable heuristic that takes into consideration the distance between the query and the cluster's pivot. We can tune the query costs versus the quality of the approximation by specifying the number (or percentage) of peers visited within each cluster [12].

We have shifted the number of stored objects by one order of magnitude, which in terms of hardware means ten times higher demands on space. Fortunately, our 16-CPU infrastructure offers 64 GB RAM in total so we can still keep all the 500 peers in main memory. The M-Chord approximation algorithm picks 1–7 clusters to be accessed by the query (2.5 clusters on average); 40 % of peers are visited in each of these clusters. Our experiments show that the approximated search reduced the number of visited peers to 25 peers and thus kept the response times about 500 milliseconds while our approximation setting gives more than 80% of the precise answer.

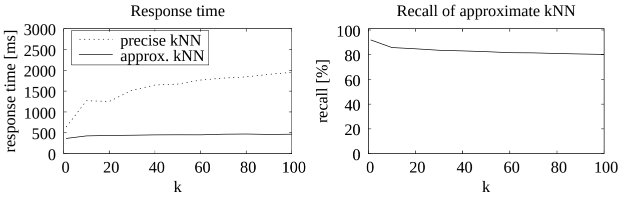

Fig. 2. Response times and recall for the 10M network.

50M: Disk-oriented Distributed Searching Currently, the final scale we have practical experience with is a distributed structure indexing 50 million images. The network consists of 2,000 peers each with approximately 25,000 objects. In order to test the scalability of our system, we temporarily gained access to a hardware infrastructure five times larger than for the 10M network (i.e. 80 CPUs and about 250 GB RAM). Simultaneously, we experimented with a persistent storage to cut the memory demands. In this implementation, the peers keep the leaf nodes of the M-Trees on disk, reducing the overall memory usage of the 2,000-peers network to 80 GB RAM; recall that the implementation is written in Java, which is relatively memory demanding and the programmer does not have a full control over the memory administration. Experiments with the disk-oriented were conducted on infrastructure with 32 physical disks and 32 CPUs. The objects stored on the disks occupied about 100 GB.

As the 50M data space is denser than the 10M set, we can afford to use more restrictive settings of the approximation – the query visits only 20 % of the most promising peers of each cluster. The left graph in Figure 3 depicts the recall (i.e., the percentage of the precise answer that is returned by the approximate query) of this approximation settings as the k grows. We can see, that the recalls are only slightly lower than for 10M, where we accessed 40 % of each cluster.

The results of the memory-based and the disk-oriented implementation differ only in the response times achieved – the comparison is in the right graph in Figure 3. With the memory-based implementation, we have reached practically the same response times as with the 10M network. We can see that the disk-oriented implementation has approximately triple response times. Let us realize, that these times can be further improved as they are strongly influenced by the hardware settings (the number and speed of the disks, usage of disk arrays, etc.) and also by the quality of the implementation.

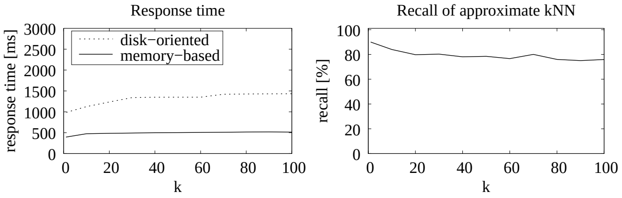

Fig. 3. Response time and recall for the 50M network.

4 Conclusions

In this paper we have tackled the scalability problem in the management of the fast growing digital image collections. We focus on two strictly related challenges of scalability: (1) to obtain a non-trivial collection of images with the corresponding descriptive features, and (2) to develop indexing and searching mechanisms able to scale to the target size.

Using a GRID technology, we have crawled a collection of over 50 million high-quality digital images, which is almost two orders of magnitude larger in size than existing image databases used for content-base retrieval and analysis, and extracted five descriptive features for each image. We have also proved that our distributed content-based retrieval system can scale to tens of millions of objects. The system offers a search for visually-similar images giving answers in approximately one second on our current database of 50 million images. The presented technology is highly flexible as it is based on the metric space model of similarity and on the principles of structured peer-to-peer networks. These results are summarized in Table 1. For convenience, we provide also some estimated values (marked with star). In the future, we plan to continue in the process of crawling and indexing images in order to reach a boundary of 100 million objects. We also plan to investigate application of additional descriptive features of the images and research the relevance feedback to further improve effectiveness. In addition, we would like to test the technology also on other multimedia types such as video, speech, and music.

Table 1. Response times of kNN with $k = 50$ using various techniques on different dataset sizes.

Technique	**CPU**	**100k**	**1M**	**10M**	**50M**
Sequential scan	1	4.3s	43.4s	7.2m*	36m*
M-Tree	1	1.4s	12s	1.8m*	-
Parallel seq. scan	16 80	- -	2.7s -	27s* 5.4s*	2.3m* 27s*
M-Chord	16	-	0.45s	1.7s	-
M-Chord with approximation	16 80	-	-	0.44s	0.45s
M-Chord with approx. and disk	32	-	-	0.87s	1.4s

References

1. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys (2008) To appear.
2. Baeza-Yates, R.A., del Solar, J.R., Verschae, R., Castillo, C., Hurtado, C.A.: Content-based image retrieval and characterization on specific web collections. Volume 3115 of LNCS. (2004) 189–198
3. Kumar, R., Novak, J., Tomkins, A.: Structure and evolution of online social networks. In: KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowedge Discover and Data Mining, ACM Press (2006) 611–617
4. MPEG-7: Multimedia content description interfaces. Part 3: Visual. ISO/IEC 15938-3:2002 (2002)
5. Manjunath, B., Salembier, P., Sikora, T., eds.: Introduction to MPEG-7: Multimedia Content Description Interface. John Wiley & Sons, Inc. (2002)
6. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach. Volume 32 of Advances in Database Systems. Springer-Verlag (2006)
7. Batko, M., Novak, D., Zezula, P.: MESSIF: Metric similarity search implementation framework. In: Proc. of DELOS Conference. Volume 4877 of LNCS. (2007) 1–10
8. Ciaccia, P., Patella, M., Zezula, P.: M-Tree: An efficient access method for similarity search in metric spaces. In: Proceedings of VLDB'97, August 25–29, 1997, Athens, Greece. (1997) 426–435
9. Skopal, T., Pokorný, J., Snásel, V.: PM-tree: Pivoting metric tree for similarity search in multimedia databases. In: Proc. of ADBIS, Budapest, Hungary. (2004)
10. Batko, M., Novak, D., Falchi, F., Zezula, P.: On scalability of the similarity search in the world of peers. In: Proc. of INFOSCALE, Hong Kong, New York, NY, USA, ACM Press (2006) 1–12
11. Novak, D., Zezula, P.: M-Chord: A scalable distributed similarity search structure. In: Proc. of INFOSCALE, Hong Kong, ACM Press (2006) 1–10
12. Novak, D., Batko, M., Zezula, P.: Web-scale system for image similarity search: When the dreams are coming true. In: CBMI-2008, Sixth International Workshop on Content-Based Multimedia Indexing, 18-20th June, 2008. To appear.
13. Amato, G., Rabitti, F., Savino, P., Zezula, P.: Region proximity in metric spaces and its use for approximate similarity search. ACM Transactions on Information Systems (TOIS) **21**(2) (2003) 192–227