

DISTRIBUTED ARCHITECTURE FOR LARGE SCALE IMAGE-BASED SEARCH

Yu Zheng, Xing Xie, Wei-Ying Ma

Microsoft Research Asia, 4/f Sigma Building, No. 49, Zhichun Road, Haidian District, Beijing, 100080, P.R. China

ABSTRACT

In recent years, some computer vision algorithms such as SIFT (Scale Invariant Feature Transform) have been employed in image similarity match to perform image-based search applications. However, with the increasing scale of image databases, centralized image retrieval system no longer provide adequate prompt search. In this paper, we design a scalable distributed architecture, which is analog to web search engine, for efficient large-scale image retrieval. In our distributed architecture, images are partitioned to multiple servers and an index is built. Administrated by a controlling server, each distributed server matches query image in its own image sub-collection in parallel and returns the intermediate search results, a list of images similar to query image, to the controlling server for further re-ranking and merging. An evaluation of the results shows that our distributed architecture removes the limitation of a centralized image retrieval system. By performing reasonable indexing, merging and ranking strategies, the precision level of search is near to that performed on stand-alone retrieval systems indexing all images.

1. INTRODUCTION

Finding information based on an object's visual appearance is useful when specific keywords for the object are not known [1]. In past years, several content-based image retrieval approaches have been proposed for object categorization [2] and location recognition [3]. Subsequently, text retrieval approach has been introduced into image retrieval systems in [4][5][6] to deploy a more comprehensive search application with image query. In these methodologies, salient regions of images are detected as local features and respectively presented with 128 dimensional vectors by SIFT (Scale Invariant Feature Transform) descriptor [7]. Each image is regarded as a document and its SIFT features are deemed as separate virtual words. Since it is an expensive operation to compare each individual salient region in a query image with those in the database, building an index is necessary for efficient retrieval. However, the scale of images keeps on increasing rapidly. In this case, current image retrieval approaches no longer achieve prompt image matches on a single machine in view of constraints of computer resources.

In [8], distributed architectures have been presented for content-based image retrieval to overcome the limitations of

centralized search by allowing distributed image matches. In these scenarios images are presented by just one description vector containing color histogram, texture, and shape information etc. Each server builds a local index individually for these feature vectors it holds and calculates similarities between images by distance between two vectors. Thus, even without a global index the retrieved images from multiple servers can be ranked simply by their distance to a query image's vector to generate the final results. But in SIFT descriptor-based image retrieval system,

- Every image has hundreds of local feature vectors and different image holds different number of features.
- The distance between feature vectors is only used to find neighbor features and the number of neighbor features found on image is leveraged to rank the similarity between two images.

In other words if the neighbor features are retrieved from different local index directly without any compensational merging and ranking strategies, the number of matched features on every image is less comparable, hence the final search results are hard to be merged reasonably from multiple intermediate results.

In this paper, we aim to present scalable distributed architecture for large scale image retrieval system over which the conception of web search engine is employed to perform efficient similarity match between images using SIFT features. Reasonable indexing, search results ranking and merging strategies have also been proposed and compared on the designed distributed system, all of which improve the search speed on large scale images and maintain its precision as accurately as that implemented on a stand-alone machine.

2. OUR DISTRIBUTED ARCHITECTURE

In current distributed system for text retrieval, there are two kinds of data partitioning methods. They include *partitioned by index* and *partitioned by document*. In the former, all the features extracted for whole images should be indexed in single machine in advance and then the divided portion of index can be deployed to multiple servers. However, this approach can lack promise in image retrieval using SIFT features in view of resource limitations of stand-alone servers in large scale indexing. What's more, the time-consuming re-indexing process will be implemented again when new images are added to existing databases which

causes distributed systems to lose its scalability. Therefore, we have chosen the later method in our distributed architecture to partition images to multiple servers and build a local index on each server. The local index will be loaded to the server's main memory to promote search speed by avoiding disk access. In this way, increases of image scale can be handled by adding servers to original distributed system. However, since we do not have a global indexing, reasonable results ranking and merging strategies become more essential for distributed systems using images *partitioned by document* to minimize the bias caused by inconsistencies of local indexes built in different servers.

As depicted in Fig. 1, much like distributed system for text retrieval, our distributed system includes a controlling server and several distributed search servers that are connected via a network.

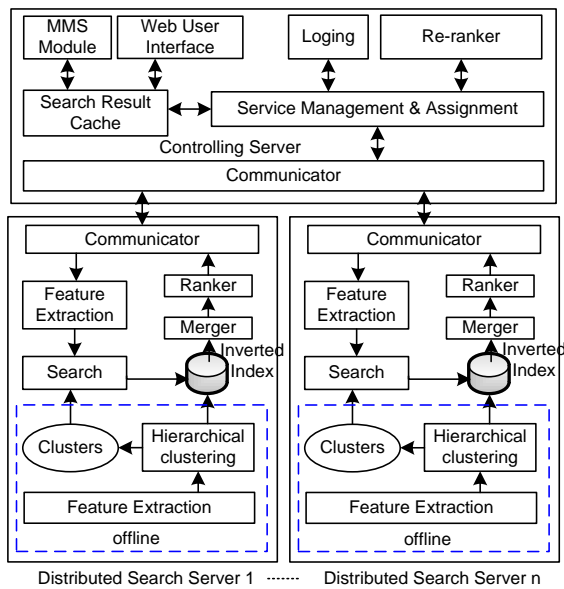


Figure 1 Framework of distributed system

The controlling server is responsible for accepting image queries from client via Multimedia Message Service (MMS) or web user interface, distributing the query to search servers, collecting intermediate results from the servers and combining them into a final result for the client. A cache that stores previous search results is held in controlling server to improve the performance of retrieval by avoiding replicated searches. In each search server, just as centralized image retrieval system presented in [4][5], local features are extracted offline on each image and represented using SIFT descriptor, which are processed later by a hierarchical cluster-based approach to build a local index for fast online retrieval. The local features of received query image are first extracted using the same detector and descriptor as those used in creating the index. Then, for each feature extracted from the query image, a number of near neighbors will be retrieved from hierarchical clusters according to the distance between two features. In terms of inverted indexing, several similar images are retrieved and processed by merging and

ranking function, usually ranked by the number of neighbor features found on them. Hereafter, a list of images similar to the query image will be sent from search servers to controlling server for further re-ranking to generate the final search results. Finally, the search result will be returned to client via MMS modem or a web UI and saved in the search result cache simultaneously.

The difference between our distributed architecture and related systems for image retrieval application include:

- We partition images to distributed servers and build indexes for portions individually on servers.
- We implement merging and ranking functions in distributed servers to determine top n images similar to query image and perform re-ranking function in the controlling server to generate the final search results.
- We take both the number of neighbor features found on retrieved image and the distance between matched features into account when designing ranking function.

3. MERGING AND RANKING STRATEGY

As depicted in Fig. 2, we design four merging functions and two ranking functions for our distributed architecture. At first, the retrieved image list will be merged via one out of four merging functions and then ranked by a ranking function in each search server. Subsequently, one ranking function will be performed again in the controlling server to re-rank search results from distributed search servers. Hence 16 ($4 \times 2 \times 2$) approaches can be selected to process the retrieved image lists and generate final search results.

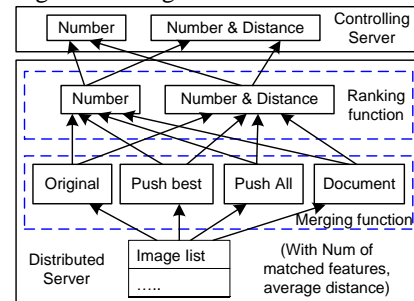


Figure 2 Results merging and ranking strategy for distributed architecture

Before describing the merging and ranking functions in detail, we propose two kinds of concepts to define a document when employing approaches of web search engine in our distributed architecture. Besides regarding one image as a document like existing methods, we deem five continuous images about a similar scene as one document and name each image an identification of the document it belongs to and its place in this document. The document appearing hereafter denotes the second concept. We evaluate two kinds of document definitions in Section 4. The number of retrieved neighbor features and the distance between two matched features are two points to design the ranking function. The distance between features is determined by their normalized L2 distance. Suppose the 128-dimension

feature can be denoted as $X=(x_1, \dots, x_i, \dots, x_{128})$, the normalized L2 distance can be calculated as equation (1):

$$D(X, Y) = \sqrt{\sum_{i=1}^{128} \left(\frac{x_i}{\|X\|_2} - \frac{y_i}{\|Y\|_2} \right)^2} \quad (1)$$

where $\|X\|_2$ and $\|Y\|_2$ are the L2 norms of feature X and Y .

The average distance between query images (Img_q) and i th image (Img_i) is calculated as equation (2), where n is the number of matched features found on Img_i . The average distance of a document can be calculated as equation (3), where m is number of retrieved images from this document. Meanwhile the number of matched features of a document can be calculated by adding up all the number of matched features from retrieved images belonging to this document.

$$D_I(Img_i, Img_q) = \frac{\sum_{k=1}^n [D(X_k, Y_k)]}{n} \quad (2)$$

$$D_D = \frac{\sum_{i=1}^m [D_I(Img_i, Img_q)]}{m} \quad (3)$$

Four merging functions:

- *Original image based*: Every image holds its own number of matched features and average distance calculated as equation (2).
- *Push All*: Add up the number of matched features of every image from the same document and set the number of matched features of all these images with the sum. Every image still holds its average distance calculated as equation (2).
- *Push Best*: From the end of retrieved image list, every image only adds its number of matched features to the nearest image which is listed in front of it and from the same document. Every image still holds its average distance calculated as equation (2).
- *Document based*: Retrieved Images from same document will be merged and represented by the document they belong to with sum of matched features as well as average distance computed as equation (3).

Two ranking functions:

- *Ranked by number (N)*: the more matched features of query image found on an image or a document, the topper the image or document should be ranked.
- *Ranked by number and distance (ND)*: first rank the image or document by number mentioned above. If two images or documents have the same number of matched features, the smaller the average distance of an image or document holds, the topper the image or document should be ranked.

Hereafter, we use “*merging function - ranking function on distributed server - ranking function on controlling server*” to describe a strategy to generate the search result for convenience. For instance, *Push Best-N-ND* means we first merge the retrieved image list via *Push Best* function and then rank it *by number (N)* on distributed system and re-rank it *by number and distance (ND)* on controlling server.

4. IMPLEMENTATION AND EXPERIMENTS

4.1 Experimental Setting

Image dataset: We select 1.2 million continuous street scene images of Seattle offered by Virtual Earth of Microsoft and scale down all of them to 400x300 pixels. All 6 adjacent images about same street scenes are organized as one document in which five images are used as dataset and another one is selected as a query image. Therefore, we hold a 1 million image dataset and corresponding 0.2 million image query set. Meanwhile, five servers with Quad 2.4 GHZ AMD CPU, 8GB memory and running Windows server 2003 are leveraged to build a tiny distributed system to test proposed merging and ranking strategies. Since 8GB memory can only hold an index of 100,000 images with SIFT features, we randomly pick continuous 100,000 images out of 1 million images dataset (20,000 corresponding query set) and distribute them to five servers in the following two ways so that we can compare their performances with that of a stand-alone server.

- *Partition images by servers*: 100,000 images are partitioned by one to five separate servers. For instance, each server will hold 50,000 images when we partition all images by two servers.
- *Add server one by one*: partition 100,000 images to five servers equally, i.e. every server holds 20,000 images, and we add server to the distributed system one by one.

Local features are extracted by DoG (Difference of Gaussian) detector [7] and SIFT descriptor and indexed by hierarchical Growing Cell Structures (GCS) cluster method on every server in advance. When a cluster has more than 1,600 features points, it will be re-clustered into five sub-clusters.

Evaluation criterion: As shown in equation (4), m is the total number of query images and C_{ni} denotes whether the i th query image finds at least one right match within its top n search results. If matched, $C_{ni}=1$. Otherwise $C_{ni}=0$. Thus P_n represent the precision of top n search results.

$$P_n = \frac{\sum_{i=1}^m C_{ni}}{m} \quad (4)$$

4.2 Partition Images by Servers

In this subsection we select 4,000 images evenly from the 20,000 query set. Fig. 3 and four present that with the same ranking function, *document based* merging function provide the best accuracy of match in the scenario of *partition images by servers*, next to it is *Push Best*, *Original* and *Push All*. Since more images similar to the query image will usually be retrieved from the same document in a correct match as compared to a negative match, the sum of matched features found on images from same document will improve the rank of correct match. As depicted in Figure 4, there is little difference among the precision of *document based-N-N* performed on 1 to 5 servers, which denotes that our architecture is robust to diverse image partition methods.

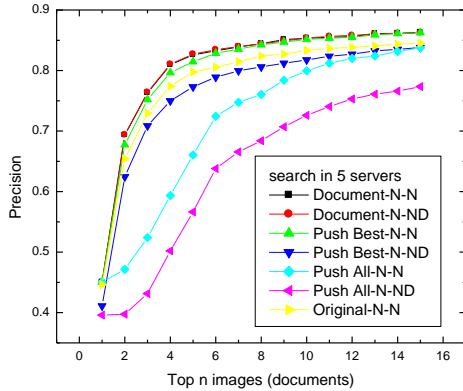


Figure 3 Search images with *ranked-by-number* function in 5 servers

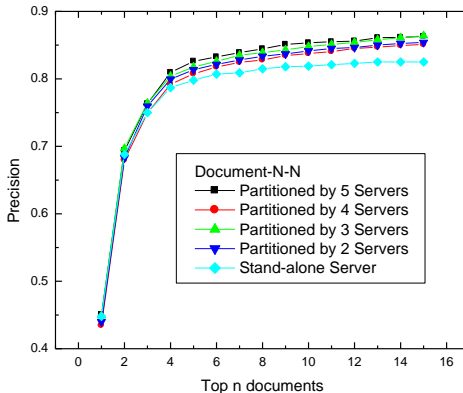


Figure 4 Comparison on precision of *document-N-N*

4.3 Extend Image Scale by Adding Server

In this subsection, from the 20,000 image query set we select the first 4,000 images as queries and add the other four servers step-by-step to evaluate the scalability of our distributed architecture. As illustrated in Fig. 5, when the *document-ND-N* strategy is employed in distributed architecture, the precision of image retrieval only degrades somewhat with the increase of server. Meanwhile Figure 6 also proves that *document based* merging function associated with distance information is very effective to handle the increasing scale of images and support scalability.

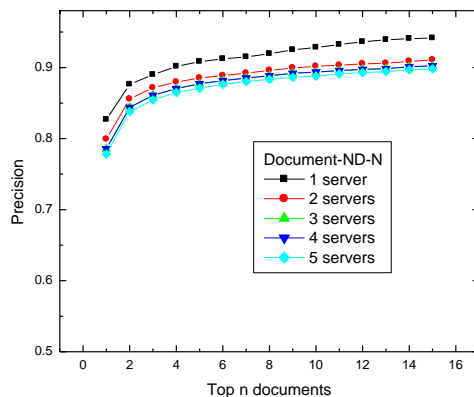


Figure 5 Performance of *Document-ND-N* with adding servers

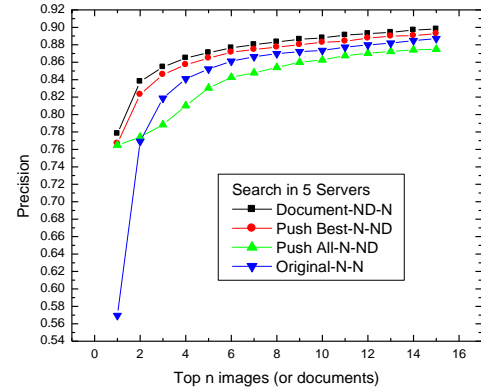


Figure 6 Comparison on precision among different strategies

5. CONCLUSION

In this paper, a scalable distributed architecture-- analog to web search engine-- has been proposed for large-scale image retrieval. We discuss image partitioning, indexing, search results merging and ranking strategies to provide efficient image-based search and maintain its precision close to that of non-distributed image retrieval systems. The proposed architecture is robust to different image partition methods and increasing scale of images through

- merging images from same scene as a document while
- taking both the number of matched features and average distances of matched image or document into account as criteria to design the ranking function.

In the future, we will employ a stop list and TF-IDF into distributed systems and deploying 1 million images of Seattle street scenes to ten servers to build an effective location recognition system. Meanwhile, we will design load balance algorithm and data redundant backup strategy to build a more robust distributed system for image retrieval.

6. REFERENCES

- [1] T. Yeh, K. Grauman, K. Tollmar and T. Darrell, "A Picture is Worth a Thousand Keywords Image-Based Object Search on a Mobile Platform". CHI, April 02-07, 2005, Portland, OR, USA
- [2] Y. X. Chen, and J. Z. Wang. "Image Categorization by Learning and Reasoning with Regions". *Journals of Machine Learning Research*, (5) 2005, pp. 913-939.
- [3] T. Yeh, K. Tollmar, T. Darrell. "Searching the Web with Mobile Images for Location Recognition", Conf. on Computer Vision and Pattern Recognition (CVPR'04), Vol. 2 pp. 76-81, 2004
- [4] J. S. Hare and P. H. Lewis, "Content-based image retrieval using a mobile device as a novel interface". *Storage and Retrieval Methods and Applications for Multimedia 2005*, Volume 5682, pp. 64-75 (2004).
- [5] J. Sivic, A. Zisserman. Video Google: "A Text Retrieval Approach to Object Matching in Videos". IEEE International Conference on Computer Vision, 2003
- [6] D. Nistér, H. Stewénius. "Scalable Recognition with a Vocabulary Tree". Proc. of the Conf. on Computer Vision and Pattern Recognition, Vol. 2, Washington, DC, USA, pp. 2161-2168, 2006.
- [7] D. Lowe. "Distinctive image features form scale-invariant key points". *IJCV*, 2(60), pp.91-110, 2004.
- [8] I. King, C. H. NG, and K. C. SIA. Distributed Content-Based Visual Information Retrieval System on Peer-to-Peer Networks. *ACM Trans. on Info. Systems*, Vol. 22, No. 3, pp. 477-501, 2004.