Scalable Similar Image Search by Joint Indices*

Jing Wang[§] Jingdong Wang[†] Xian-Sheng Hua[‡] Shipeng Li[†]
[§] Peking University [†] Microsoft Research Asia [‡]Microsoft Bing
cis.wangjing@pku.edu.cn {jingdw, xshua, spli}@microsoft.com

ABSTRACT

Text-based image search is able to return desired images for simple queries, but has limited capabilities in finding images with additional visual requirements. As a result, an image is usually used to help describe the appearance requirements. In this demonstration, we show a similar image search system that can support the joint textual and visual query. We present an efficient and effective indexing algorithm, neighborhood graph index, which is suitable for millions of images, and use it to organize joint inverted indices to search over billions of images.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Retrieval Models

General Terms

Algorithms, Experimentation

Keywords

Similar image search, neighborhood graph index, joint inverted index

Introduction. Commercial image search engines, e.g., Microsoft Bing image search and Google image search, are able to return relevant images for simple textual queries. To help find images with certain visual requirements, a lot of recent efforts have been made to utilize visual contents such as color sketch [3], concept sketch [6], and so on. In this demonstration, we present an efficient and effective indexing algorithm, neighborhood graph index, which is suitable for millions of images, and use it to organize joint inverted indices to search over billions of images. We justify the effectiveness by applying them to a similar image search system that supports joint textual and visual queries.

System. The snapshot of the similar image search system is shown in Figure 1. The system initially displays the images in the database and shows text-based search results after typing in a textual query. The user can click the link "similar" to find similar images in the database, referred as similar image search with an in-index query. The user can also upload an image or paste a URL that points to an image to find similar images, referred as similar image search with an out-of-index query, and in this case, the user can additionally type in a textual query to perform the search jointly using the visual and textual queries. Figures 2 and 3 show several examples of similar image search.

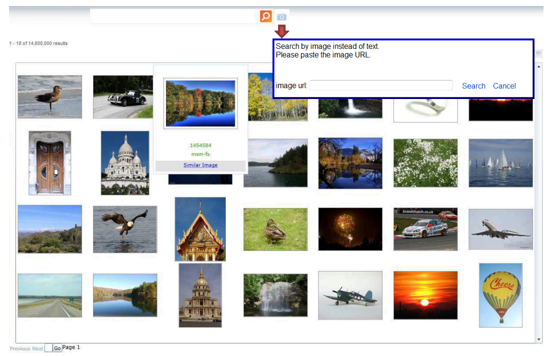


Figure 1: The snapshot of the interface.

Technologies. The image database consists of N images $\{I_n\}_{n=1}^N$, and each image I_n is associated with a set of textual words $\mathcal{T}_n = \{w_{n_1}, \cdots, w_{n_m}\}$. We extract three features to describe an image: GIST (a 384-dimensional vector), spatial color histogram (SCH, a 768-dimensional vector), histogram of oriented gradients (HOG, a 512-dimensional vector), which correspond to the texture, color, and shape characteristics, respectively. GIST and HOG are widely-used, and not described in detail in this demo. The spatial color histogram feature is extracted as follows. We first divide the image into 9×9 cells. For each cell, we extract a 12-dimensional color histogram by softly mapping each RGB color into 12 quantized colors. Then we transform 9×9 cells into 8×8 blocks, where each block is formed by 4 cells. At last we aggregate the histograms over 4 cells to describe each block, and concatenate the histograms over all blocks, forming a 768-dimensional feature vector. We concatenate three feature vectors together to get a single vector $\mathbf{f}_n$ for image I_n.

We present an efficient and effective indexing algorithm, neighborhood graph index, and then use it to organize joint inverted indices. Neighborhood graph index is suitable for middle scale image database (e.g., 5 millions) as the neighborhood graph construction is computationally expensive. Using it to organize inverted indices can make similar image search feasible for billions of images.

The neighborhood graph is constructed by random spatial partitioning followed by neighborhood propagation [5], which is shown to get the best performance in terms of both efficiency and accuracy. To exploit the textual information, the neighborhood propagation scheme splits some edges if the images connected by one edge do not share any common

*This work was done when Jing Wang was an intern at Microsoft Research Asia.

MM'12, October 29–November 2, 2012, Nara, Japan.
ACM 978-1-4503-1089-5/12/10.

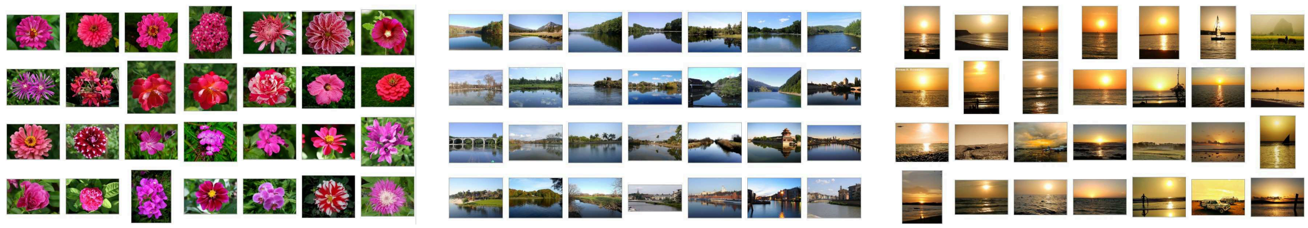
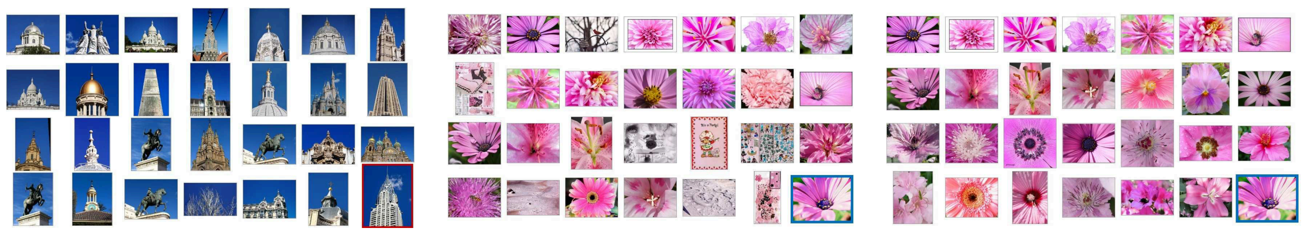


Figure 2: Similar image search results for in-index queries. For each figure, the first image is the query. The first result is from neighborhood graph index and the remaining two are from inverted indices.



(a) (b) Without textual queries (c) With the textual query, "flower"

Figure 3: Similar image search results for out-of-index queries. The bottom right image (with a bounding box) in each figure is the query image. All results are from neighborhood graph index.

textual words. In the search strategy, if the query image is in-index, similar images are found by recursively expanding higher and higher order neighborhoods in a best-first manner. If the query image is out-of-index, we propose to use an iterative local search scheme [4] that conducts the search process alternatively over trees [1] and the neighborhood graph, and filter out images that do not share any common textual words with the query.

An inverted index consists of a set of inverted lists, each corresponding to a set of points, obtained through hashing, clustering, or spatial tree partitioning. The points in the same inverted list are regarded as candidate similar points. To boost the performance, multiple inverted indices are usually generated. We build multiple complementary PCA (principal component analysis) trees to create the inverted indices [2], motivated by the complementary hashing approach in [7]. Furthermore, to efficiently and effectively organize the inverted indices, we compute the centers for each inverted list, and build a neighborhood graph over the centers, then reassign each point to multiple nearby centers by the iterated local search [4], yielding an updated inverted file. Besides, the image database is also organized using the associated texts though an inverted index. In the search strategy, a query is first mapped to a leaf node in each spatial partitioning tree, or to several centers when using a neighborhood graph to update the inverted lists, and all the points in the inverted lists are collected together as the candidate images. If the query consists of a textual component, we filter out some candidate images that are not in the inverted list corresponding to the textual query. Finally, the feature vectors of the final candidate points are used to compute the similarities with the query image.

To avoid returning the duplicate images, we propose to remove the duplicate images using a simple but effective scheme. We compute the similarities of all the candidate images with the query, and sort them according to the similarity scores. The idea is that if the similarity scores of several images with the query images are almost the same, they are most likely to be duplicate. The images with almost the same similarity scores can be found in a linear time. Then, we can check if those images are indeed duplicate by computing the similarities, which takes square time cost but is efficient as the number is usually very small.

Experiments. We conduct experiments to check the approximate nearest neighbor (ANN) search performance with 1M GIST features being the reference database and 10K GIST features being the query database. Neighborhood graph index gets a 87% accuracy within 8 milliseconds when searching for 20 nearest neighbors, and more results can be found from [4]. When using the neighborhood graph to organize the inverted lists, which are constructed by the centers from the adaptive forward selection scheme [2], the accuracy reaches 83%.

Conclusions. In this demonstration, we show similar image search systems, which are developed using our proposed two indexing algorithms: neighborhood graph indexing and inverted indices. The former one is suitable for millions of images, e.g., vertical image search including shopping images search, and the latter one is suitable for billions of images.

References

[1] Y. Jia, J. Wang, G. Zeng, H. Zha, and X.-S. Hua. Optimizing kd-trees for scalable visual descriptor indexing. In *CVPR*, pages 3392–3399, 2010.

[2] W. Tu, R. Pan, and J. Wang. Similar image search with a tiny bag-of-delegates representation. In *ACM Multimedia*, 2012.

[3] J. Wang and X.-S. Hua. Interactive image search by color map. *ACM TIST*, 3(1):12, 2011.

[4] J. Wang and S. Li. Query-driven iterated neighborhood graph search for large scale indexing. In *ACM Multimedia*, 2012.

[5] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li. Scalable k-nn graph construction for visual descriptors. In *CVPR*, pages 1106–1113, 2012.

[6] H. Xu, J. Wang, X.-S. Hua, and S. Li. Image search by concept map. In *SIGIR*, pages 275–282, 2010.

[7] H. Xu, J. Wang, Z. Li, G. Zeng, S. Li, and N. Yu. Complementary hashing for approximate nearest neighbor search. In *ICCV*, pages 1631–1638, 2011.