LARGE-SCALE DUPLICATE DETECTION FOR WEB IMAGE SEARCH*

Bin Wang[1], *Zhiwei Li*[2], *Mingjing Li*[2], *Wei-Ying Ma*[2]
[1]University of Science and Technology of China, Hefei 230026, China
[2]Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China

ABSTRACT

Finding visually identical images in large image collections is important for many applications such as intelligence propriety protection and search result presentation. Several algorithms have been reported in the literature, but they are not suitable for large image collections. In this paper, a novel algorithm is proposed to handle the situation, in which each image is compactly represented by a hash code. To detect duplicate images, only the hash codes are required. In addition, a very efficient search method is implemented to quickly group images with similar hash codes for fast detection. The experiments show that our algorithm can be both efficient and effective for duplicate detection in web image search.

1. INTRODUCTION

Image is one of the most popular media types on the Internet. With the profusion of digital cameras and camera cell phones, the number of online images increases quickly in recent years, so more people can search for their desired images on the web. To meet those needs, many image search engines have been developed and are commercially available in the market. For instance, both Google and Yahoo have indexed over one billion images. In addition to great abundance, another important fact of the web images is that there are many duplicates, that is, one image can be copied for many times and each copy has a different URL. Although visually identical, those images are recognized as different ones by current image search engines, which identify the images by their URLs. So when a user seeds a query, the returned list of images may contain many duplicates. An example is presented in Figure 1. Those duplicates, shown in numbered boxes, obviously downgrade the user's perceptibility and should be purged to improve the search experience.

There have been many methods in the literature dealing with the detection of duplicate or near-duplicate images [3, 4, 5, 7]. They are reasonable solutions for their designated problems. Yet, for very large image collections, those methods suffer from either intensive computation complexity or degraded performance.

Figure 1: Search result of query "Paris Hilton"

In this paper, we propose a fast and effective method to detect all visually duplicate groups in large image collections. Each image is converted to a *K*-bit hash code according to its content. The concise representation of image content can greatly facilitate the quick search and grouping process. "Grouping" means to categorize the images into different groups so the images within each group are duplicates of each other. The experiments illustrate that the proposed method can efficiently group the duplicate images with high precision. For 100,000 images, less than 0.4 second is required to group all images. The storage cost of hash codes is negligible comparing to the size of image file itself.

In this paper, the evaluation of our algorithm is mainly in the context of presentation of web image search results. However, it can be applied into other areas including copyright (intelligence propriety) protection, product search in e-commerce and so on. The applications of proposed algorithm in those scenarios will be explored in future work.

The rest of the paper is organized as following. In Section 2, we clarify the task of duplicate image detection. Then the proposed algorithm is detailed in Section 3. Section 4 shows the experimental results. Finally, we give the conclusion and possible future work in Section 5.

2. TASK CLARIFICATION

The algorithm is designed to quickly and precisely find all visually duplicate image groups for a given set of images.

* This work was performed at Microsoft Research Asia.

"Visually identical" isn't restricted by the exact match. There can be great differences in many aspects such as scale and compressed file format. At first, we clarify the concept of "duplicate" in the scope of this paper. Duplicate is a kind of relationship for an image pair (two images), which means two images are visually identical. In our notation, three kinds of variations between images are deemed as the reason of "duplicate":

- Scale: this includes both the change in height or width and the ratio of width/height. Images can be stretched horizontally or vertically, or they can be zoomed in or out, such as thumbnails.
- Color/grayscale: it is common to convert a color image into a grayscale one. The images under such processing are deemed as duplicates.
- Storage format: there are many image storage formats on Internet, e.g. JPEG, GIF, PNG and so on. When an image is transformed into a different storage format, slight difference can be introduced though visual appearance can be maintained.

Other types of image deformation, including luminance changes, translation, rotation and non-uniform scaling, can introduce visually similar but not identical images. Because the detection of similar images is a challenging topic out of this paper's scope, we will work on that as our future work.

For web image search task, duplicate detection may be conducted only in a small subset of whole collection. The term "detection scope" refers to the number of images in this subset, which is more directly influential than dataset size.

In addition, there are several factors which can influence the algorithm design, such as storage cost and speed of execution. The storage cost should be small, and the speed should be fair for both searching and index building.

3. DUPLICATE DETECTION ALGORITHM

Traditional methods often need $O(n^2)$ pair-wise comparisons for detection of all duplicate groups in n images. For large n (e.g. thousands or millions), such computation is infeasible. Besides, the high-dimensionality of images' content representation exacerbates the problem. We propose an effective method to exploit the advantages of hash codes, which greatly improves the speed and retains high precision. Our algorithm consists of four parts: image feature extraction, dimension reduction, hash code generation, and grouping of hash codes. Each image is first converted into a high dimensional feature vector. Then the feature vector is projected into a low-dimensional sub-space and mapped to a K-bit hash code. All the above processes can be completed when building the image database. The last part of the algorithm is the hash code grouping, which may be an interactive process. For the purpose of improved recall, similar hash codes should be grouped instead of identical ones. The whole process is depicted in Figure 2. Only the "Duplicate Detection via Hash Code Grouping" (dark marked block) may need to be performed interactively. The hash code generation and grouping method provides the invariance of image content. All parts are detailed in the following sub-sections.

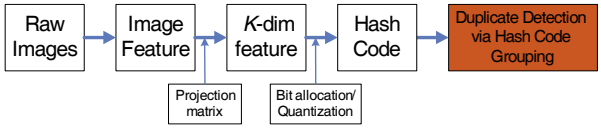

Figure 2: Flowchart of duplication detection

3.1. Image features

Features are the concise representation of the image's content. The appropriate feature should be able to represent the images' content and structure well simultaneously, and be robust to three kinds of variations discussed in Section 2. Therefore, we propose to use the gray block feature. The experiments indicate the feasibility of gray block feature.

In the calculation of gray block feature, each image is regularly divided into n by n blocks. For each block, the average luminance is calculated. The *k-th* dimension value of the feature is calculated as

$$f_k = \frac{1}{N_k} \sum_{i,j \in B_k} I(i,j) \quad k = 1,2,\ldots,n^2$$

where B_k corresponds to block k, N_k is the number of pixels in B_k and $I(i,j)$ is the pixel luminance at the coordinate (i,j). So, an image is represented by a vector $F_i = (f_1, f_2, \ldots, f_{n*n})^T$. The feature vector of different n can be concatenated to represent the image content in more details. For example, the feature vectors of gray block 7x7 and gray block 6x6 can be unified into one single feature vector.

Obviously, the gray block features can be seen as small thumbnail of original images. It maintains the primary content and structure information, and is invariant to the scale change. Each component of the feature is the mean value of many pixels, which makes it robust to the small variance in pixel values. Finally, the feature is calculated on the luminance so it is robust to the change in color. So, the gray block feature has the desired invariance property.

3.2. Dimension reduction

The goal of dimension reduction is two-fold. One is to get a compact representation while maintaining as much original information as possible. The other is to reduce the small noise and potential value drifting by omitting the least significant dimensions. Such projection can be implemented as $G_i = AF_i$ by using a projection matrix A. To get A, principle component analysis (PCA) is conducted on the feature matrix of a sufficient large image collection. The first several principle vectors corresponding to the largest eigenvalues are retained to form A. For all the images, A is the same.

3.3. Hash code generation

The hash code generation is essentially a vector quantization (VQ) process. As the final quantized vector has K bits, how to allocate the bits to each dimension is an important issue [1, 6]. A simple and effective method is to allocate *1* bit for each of first K dimension as following

$$H_{i,k} = \begin{cases} 1 \ if \ G_{i,k} > mean_k \\ 0 \ if \ G_{i,k} \le mean_k \end{cases}$$

where $mean_k$ is the mean value of dimension k. In this way, the K-dimension feature vector is transformed into a K-bit binary string, which is the image's hash code. K is constrained to be no more than 32.

3.4. Duplicate detection via hash code grouping

Our target is to quickly categorize the hash codes of duplicate images into groups. One problem for the vector quantization is the threshold operation. A little drifting near the boundary can completely changes the quantized value. To improve the performance, similar hash codes with small difference should be grouped together instead of identical ones. Such kind of difference can be indicated by Hamming distance between two binary strings: Due to the nature of PCA, the drifting is more likely to occur in less significant dimensions than in more significant ones. Therefore, to improve the recall as well as retain the precision, the criterion for similar hash codes is that the most significant L bits should be identical while small variance is allowed in least significant K-L bits. It can also be depicted in a mathematical way:

H_i and H_j is similar iff

$$\sum_{k=1}^{L}(H_{i,k} \oplus H_{j,k}) = 0 \text{ and } \sum_{k=L+1}^{K}(H_{i,k} \oplus H_{j,k}) \le T$$

where $H_{i,1}$ is the most significant bit while $H_{i,K}$ is the least significant one. The parameter L and pre-defined threshold T are tunable parameters for different application scenarios.

4. EXPERIMENTAL RESULTS

4.1. Dataset

We collected top 1,000 queries from a typical image search. After the elimination of invalid and redundant ones, 995 queries remain. We submitted each query to an image search engine, and downloaded its top 1,600 returned images. As there may not be so many images for every query, 1,443,066 images are collected in total.

To evaluate the performance, we manually labeled the results of four queries, and the ground truths of 995 valid queries are automatically generated using pair-wise comparison of the above-mentioned gray block feature, which is quite effective in detecting duplicate images with low efficiency. The duplicate image detection is performed within each query's scope. Our experiments were conducted on a computer with Intel P4 3.1GHz CPU.

4.2. Measurements

Like the problems for evaluating clustering algorithms, traditional precision and recall measures cannot be applied here directly. The number of detected duplicate groups may differ from that of the ground-truth groups. Besides, one ground-truth group may be split into multiple detected groups. Or, a detected group can contain images from different ground-truth groups. Here we propose some measures suitable for evaluating the duplicate detection task.

If a detected group is a subset of a ground-truth group, it is called a "correct" group. Then, group precision (GP) and group recall (GR) are calculated as

GP = (# of correct groups)/(# of detected groups)*100%

GR = (# of correct groups)/(# of ground-truth groups)*100%

Duplicate is a kind of relationship for an image pair (two images). Thus two natural measures are the precision and recall of duplicate image pairs. A "correct" image pair means it belongs to the intersection of a detected group and a ground-truth group. Then, the image pair precision (IPP) and image pair recall (IPR) are calculated as

IPP = (# of correct pairs)/ (# of detected pairs)*100%

IPR = (# of correct pairs)/ (# of ground-truth pairs)*100%

Figure 3 shows the distribution of group size and cumulative distribution for the detection scope of 1,000. The distributions for different detection scopes are very similar. This figure shows that most of the ground-truth groups have less than 3 images. So GP and GR are reasonable measures. Besides, the number of image pairs is $O(n^2)$ for a group of size n. When a large ground-truth group is split into small detected groups, the IPR will be low even with high IPP. As mentioned above, large groups are rare and these measures are reasonable.

Usually F-measure [F=2*precision*recall/ (precision + recall)] is used as a concise indication, where the precision and recall receives equal weight. For our scenario of presentation of image search results, the correct groups give users good experience. But a wrong group will upset users GREATLY. It implies that the precision is much more important than the recall, and high group precision should be ensured with high priority.

4.3 Performance

Table 1 and Table 2 show the performance of proposed algorithm on manual labeled data set. Four labeled queries are representative: two ("Angelina Jolie" and "Britney Spears") have many duplicates because both are celebrities, while the other two ("Anime" and "Cartoon") have very few duplicates. The tables show that the proposed algorithm is

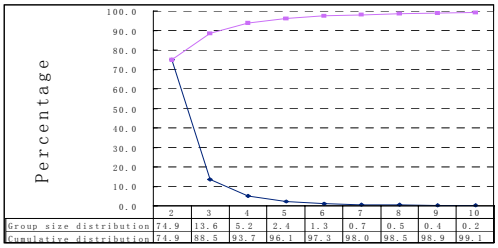

Figure 3: Distribution of ground-truth group size

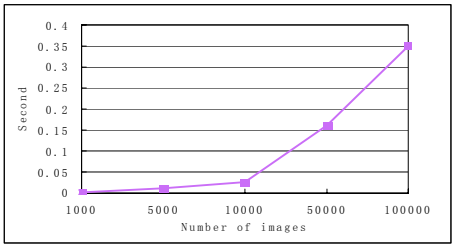

Figure 4: Hash code grouping speed

very effective to find the duplicate image groups.

Table 1: Performance on manual labeled data

Query Words	Detect Group	Correct Group	Ground truth Group	GP (%)	GR (%)
Angelina Jolie	177	176	256	99.4	68.8
Anime	21	21	44	100.0	47.7
Britney Spears	167	164	245	98.2	66.9
Cartoon	59	59	96	100.0	61.5
(total)	424	420	641	99.1	65.5

Table 2: Performance on manual labeled data (cont'd)

Query Words	Detected Duplicate Images	Detected Image Pair	Correct Image Pair	IPP (%)
Angelina Jolie	276	424	423	99.8
Anime	22	23	23	100.0
Britney Spears	230	327	315	96.3
Cartoon	61	63	63	100.0
(total)	589	837	824	98.4

The performance for different detection scopes of all 995 queries is presented in Table 3. The proposed method achieves more than 90% precision and more than 55% group recall. For the detection scope of 1,600 images, the GP is 92.2% and the GR is 55.9%. Unlike the GP and GR, both the IPP and IPR decrease when the detection scope increases. The decrease in pair recall is partly because of the appearance of more large groups. For the detection scope of 1,600 images, the IPP is 92.0% and the IPR is 26.5%.

Table 3: Performance for different scopes

Detection scope	100	200	300	400	500	600	700	800	900	1000
GP	95.7	95.0	94.5	94.1	93.9	93.7	93.5	93.3	93.2	93.0
GR	55.4	55.9	56.4	56.4	56.6	56.7	56.6	56.8	56.8	56.6
IPP	96.2	95.4	94.3	93.9	93.6	93.2	93.2	93.1	93.0	92.8
IPR	35.4	32.7	31.2	30.6	30.2	30.0	29.8	29.4	29.3	28.8

We also conduct experiments on a dataset of 2.4 million images. We get the group precision of over 90%. Since it is impossible to label all the duplicate images, the group recall is not reported.

4.4. Speed test

The grouping speed is a very important factor in the search. Because only hash codes are required, the group process is very fast. Usually, image search engines return thousands of images. Figure 4 shows the average grouping time of proposed algorithm for different number of images. The results are averaged for 100 runs. For 10,000 images, the grouping operation costs less than 0.1 second, which is imperceptible. When the number of images grows to 50,000 and 100,000, the consumed time will not exceed 0.4 second, whereas pair-wise comparison may take several minutes.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose an algorithm to efficiently and effectively detect visually duplicate images in a large set of images. We first calculate a *K*-bit ($K \leq 32$) hash code for each image and conduct the duplicate image detection with only the hash codes. Because the hash codes are very compact representation of the image content, the detection process is very fast. The experiments show the proposed algorithm can find duplicate images with high precision and the time cost for grouping 100,000 images is less than 0.4 second.

Since we can represent images in such a compact form, one future work is to cluster the similar images to further present more organized results to users.

6. REFERENCES

[1] H. Ferhatosmanoglu, and E. Tuncel, "Vector Approximation based Indexing for Non-uniform High Dimensional Data Sets," *Proceedings of 9th CIKM*, McLean, USA, pp 202-209, 2000

[2] C. Herley, "Why Watermarking is Nonsense," *IEEE Signal Processing Magazine*, pp. 10-11, September 2000

[3] A. Jaimes, S-F. Chang., and A.C. Loui, "Detection of Non-Identical Duplicate Consumer Photographs," *Proceedings of 4th IEEE PCM*, Singapore, pp. 16-20, Dec. 15-18, 2003

[4] Y. Ke, R. Sukthankar, and L. Huston, "Efficient Near-duplicate Detection and Sub-image Retrieval," *Proc. ACM Intl. Conf. on Multimedia*, New York, pp. 869-876, Oct. 10-16, 2004

[5] S. Lin, Ozsu, M.T., Ozsu, V Oria, et al., "An Extendible Hash for Multi-Precision Similarity Querying of Image Databases," *Proceedings of VLDB 2001*, Roma, Italy, pp. 221-230, 2001.

[6] E. A. Riskin, "Optimal Bit Allocation via the Generalized BFOS algorithm," *IEEE Trans. on Information Theory*, Vol. 37, No. 2, pp. 400-402, March 1991

[7] D-Q. Zhang, and S-F. Chang, "Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning," *Proceedings of ACM International Conference on Multimedia*, New York, pp. 877-884, Oct. 10-16, 2004