SICO: A SYSTEM FOR DETECTION OF NEAR-DUPLICATE IMAGES DURING SEARCH

Jun Jie Foo[§] *Ranjan Sinha*[‡] *Justin Zobel*[§]

[§]School of Computer Science & I.T.
RMIT University, Australia

[‡]Dept. of Computer Science & Software Eng.
University of Melbourne, Australia

ABSTRACT

Duplicate and near-duplicate digital image matching is beneficial for image search in terms of collection management, digital content protection, and search efficiency. In this paper, we introduce SICO, a novel system for near-duplicate image detection during web search. It accurately detects near-duplicates in the answers returned by commercial image search engines in real-time. We show that SICO — which utilizes PCA-SIFT local descriptors and adapts near-duplicate text document detection techniques — is both effective and efficient. On a standard desktop personal computer, SICO identifies clusters of near-duplicate images with 93% accuracy in under 30 seconds, for an average of 622 returned images for each query.

1. INTRODUCTION

Digital images indexed by commercial image search engines such as Yahoo![1] and Google Images[2] contain large numbers of duplicate and near-duplicate images [1]. This phenomenon is particularly pronounced for images of popular subjects, such as celebrities, music albums, and feature films. Storage and retrieval of such redundant image instances may be unnecessary, is inconvenient in lists of answers, and may represent issues of digital rights management.

Digital image copies are rarely identical at the bit level, but are still effectively duplicates or near-duplicates (henceforth referred to as near-duplicates) of one another as perceived by a typical user. For this reason, bit-level duplicate detection methods are useless. However, near-duplicate detection can allow simplification of answer lists, and can serve as an adjunct to digital watermarking techniques — which are ill-suited for retrieval applications [2] for protection against digital content pirating and for identification of copyright infringement in large image and video databases.

In this paper, we present SICO[3] (Similar Image COllator) a novel system that automatically identifies near-duplicate images during web search. To our knowledge, this is the first practical system that performs non-query-based automatic identification of near-duplicate images. We demonstrate that SICO can efficiently identify nearly all instances of near-duplicates within the sets of images retrieved by text-based commercial image search engines, with acceptable processing time. SICO represents a novel and effective amalgamation of algorithms for tasks such as improving the presentation of a set of image search results.

[1] `http://images.search.yahoo.com`
[2] `http://images.google.com`
[3] `http://sico.cs.rmit.edu.au`

2. BACKGROUND

The detection of near-duplicate instances can be broadly categorized into two groups: query-based and non-query-based. Query-based approaches require an example image for the detection of near-duplicate instances. This assumes that there is at least one example image against which a given collection can be compared. It is impractical to detect all instances of near-duplication in a collection with query-based approaches, as every image in the collection is potentially a query-example. Non-query-based approaches identify all near-duplicate instances given a collection of images, but there is little existing work in this area.

To retrieve near-duplicate images in response to a query image, Ke et al. [3] have demonstrated near-perfect accuracy using PCA-SIFT local descriptors [4] and locality-sensitive hashing (LSH) [5]. Qamra et al. [2] propose perceptual distance functions for query-based near-duplicate retrieval using colour and texture image features, but effectiveness is not high. This phenomenon is particularly apparent when visually similar (but not near-duplicate) images are present within the collection. For automatic detection of near-duplicate instances, Chang et al. [6] propose RIME, a system that uses a clustering-based approach for automated detection of near-duplicate images; although they report good results, the system was only tested on ten near-duplicate images with limited image variations. Recently, Zhang and Chang [7] use machine learning by graph matching, but observe limited effectiveness on a small collection of images; efficiency and scalability remain an issue. There is no system capable of detecting non-query-based near-duplicates for web search.

In previous work [8], we have shown that a combination of near-duplicate text document detection techniques, and a modified LSH using PCA-SIFT local descriptors can be used to automatically and effectively identify near-duplicates within a moderate-sized crawled image collection with modest pro-

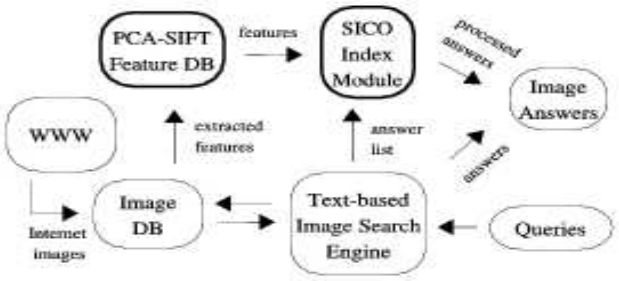

Fig. 1. The architecture of the SICO system consists of the highlighted components. The other components are typical of most image search engines.

cessing time. In recent work [1], we showed that the answer sets returned by commercial image search engines contain high levels of redundancy, especially for popular subjects. We also showed that of the two viable approaches to near-duplicate detection, perceptual distance functions (using colour and texture features) and robust local descriptors, the latter shows promise for such tasks even on unconstrained collections as in web image search.

3. OVERVIEW OF SICO

Our SICO (Similar Image COllator) system uses the SIFT interest point detector [9] and PCA-SIFT local descriptors. In earlier work we have shown that such an approach combined with a modified locality sensitive hashing [5, 8] index can accurately identify even severely edited (cropped or scaled) versions of images [3, 8, 10].

Figure 1 shows the different components of a typical image search engine; SICO comprises the highlighted components. Real-time feature extraction from each of the result images is time-consuming and may not be practical. Hence, in our system (as shown in the figure), these image features are extracted either when the images are first fetched during crawling or as an off-line batch process. Once the image features are extracted, SICO can serve as an auxiliary system to facilitate near-duplicate detection within the images returned by existing text-based image search engines; the *SICO index module* is used at query time. A summary of the process of a SICO-enabled image search engine (refer to figure 1) for automated identification of near-duplicate images is as follows:

For each image in the database (off-line):
- Extract PCA-SIFT local descriptors.
- Store each descriptor in a feature database.

Given an image query (on-line):
- Get answers from the image search engine.
- Build a SICO index of the answers.
- Collate image pairs using the SICO index.
- Return image answers from the image database.

The SICO components are as follows.

PCA-SIFT Descriptor Extraction

To generate PCA-SIFT descriptors, each image is first processed using the initial three of the four phases of the SIFT interest point detector, followed by PCA-SIFT, which is an alternative method of computing the fourth phase of the original SIFT algorithm.

Given an image, SIFT first identifies all local peaks, or *keypoints*, in various scales and locations using a difference-of-Gaussian (DoG) function simulating a pyramidal scheme [9], after which poorly localized and unstable points (below a threshold level) are discarded. Then, remaining keypoints are subsequently assigned a dominant orientation (for rotation invariance), which is computed using gradient patches centered around each keypoint.

Using the information from the SIFT detector, that is, location, scale, and dominant orientations, the PCA-SIFT algorithm generates local descriptors by concatenating the horizontal and vertical gradient maps from 41×41-pixel patches to produce a $2 \times 39 \times 39 = 3042$ element local descriptor (vector). Each vector is then reduced to 36 (a parameter empirically determined by Ke et al. [4]) feature spaces using principal component analysis; any two vectors are deemed a match within an Euclidean distance (L_2-norm) of $3,000$. We use the same settings in SICO. The output is a set of local descriptors that are likely to be preserved (robust) when an image is transformed (even severely cropped). Note that SICO is not limited to using PCA-SIFT local descriptors or SIFT interest points; it can be applied to other robust local descriptors and interest points such as those described in [11]. In this work, we limit our discussion to PCA-SIFT local descriptors — henceforth referred to as PCA-SIFT features.

SICO Index

The number of PCA-SIFT features extracted from each image depends on its complexity, typically ranging from hundreds to thousands. To efficiently index these features, SICO employs locality-sensitive hashing (LSH) — an approximate nearest-neighbour search scheme [5]. Given an image collection, the similarity of two images can be assessed by computing the number of matching features within an $L_2 - norm$ of $3,000$; a large number of feature matches reflects high similarity [4, 9]. SICO employs the hash function implementation of Ke et al. [3]; this is a family of hash functions in the Hamming space [5], which embeds the $L_1 - norm$. Details of the LSH implementation used in SICO appear elsewhere [8, 5].

In previous work [8], we used near-duplicate text document detection techniques to automatically detect near-duplicate images within a collection (without using a query image). The LSH-generated hash values can be used efficiently by hash-based probabilistic counting [12] to determine the ex-

istence of near-duplicate relationship of an image to other images within the collection. This process can also be described as the generation of a *relationship graph*, wherein each node represents an image, and an edge between two nodes indicate a probable near-duplicate relationship. The key idea of hash-based probabilistic counting is to refine and filter the number of near-duplicate pairs with the aim of quickly discarding false positives, since the number of image pairs to be considered grows quadratically with collection size.

Using this approach, each hash value (the post-indexing phase of a PCA-SIFT features) is treated as a token, akin to words in text documents. Thus, each image is converted into a series of representative tokens that are then indexed in an inverted file. Each token entry in the inverted file contains a list of images in which that token occurs. To generate a relationship graph, all possible image pairs (edges) in every postings list are accumulated using a hash-counter to quickly eliminate edges that do not have matching tokens above a certain threshold T in the first pass. Then the number of matching tokens of the remaining edges (a smaller pool) can be accumulated to reflect the actual number of PCA-SIFT feature matches between two images. The first-pass counter is a coarse approximation due to the occasionally spurious edges generated due to hash collisions, whereas the subsequent pass uses exact counting for accumulating matching tokens. All edges without at least T matching features are also discarded in the subsequent pass. SICO uses $T = 32$, as we empirically observed that this value yields high accuracy [8].

4. EVALUATION AND TESTBED

For our test collection, we use a list of 20 queries of celebrities, 15 of which are popular queries listed in Google Zeitgeist.[4] We also use an additional 5 queries that we have observed to contain large numbers of near-duplicates within returned answers. All queries are used to retrieve images from Google Image Search [5].

The Google search engine returns a maximum of $1,000$ answers for each query; some answer images are no longer accessible on the original server, resulting in an average of 622 answers per query, and a total of $12,443$ retrievable images overall. On average, 291 489 PCA-SIFT features are extracted from the answer image set of each query. We have observed that near-duplication is more prevalent in queries for celebrities and historical figures [1]; we limit the experiment described in this paper to such subjects. The queries used are: (1) *50 Cent*, (2) *Aaliyah*, (3) *Angelina Jolie*, (4) *Avril Lavigne*, (5) *Bill Clinton*, (6) *Bob Marley*, (7) *Brad Pitt*, (8) *Carmen Electra*, (9) *Donald Rumsfeld*, (10) *Edgar Allan Poe* , (11) *George W. Bush*, (12) *Keira Knightley*, (13) *Kurt Cobain*, (14) *Miles Davis*, (15) *Princess Diana*, (16) *Robbie Williams*, (17) *Terri Schiavo*, (18) *Tom Cruise*, (19) *Tupac*, and (20) *William Shakespeare*. The fifth, ninth, tenth, eleventh, and fourteenth queries are the five additional queries that we selected. All images were retrieved on 22nd December, 2006.[6]

To determine effectiveness, we manually assessed the accuracy of the machine identified groups (clusters) of near-duplicates, such that each cluster consists of images believed to be near-duplicates of each other (that is, derived from the same source); images not in any cluster are deemed *singletons*. This is relatively stringent, in that, a correctly identified cluster requires every image to be — in one form or another — a near-duplicate of all other images within this cluster.

This evaluation process is analogous to the *precision* metric in information retrieval, wherein only the ability of a system in identifying correct answers (image clusters) is assessed. We do not consider false negatives (images that are falsely identified as singletons), since that would require relevance assessment of every item in the answer sets of every query. Moreover, we have shown in previous work that our algorithm is highly accurate in identifying seeded relevant images within controlled collections, and that PCA-SIFT features are effective even for unconstrained web collections [1, 8].

We also report the timing results and memory requirements of SICO. The former is the elapsed time for the entire SICO process of identification of all image near-duplicates to corresponding clusters; this does not include feature extraction, which is accomplished off-line or during image crawling. The timing includes loading the PCA-SIFT features into memory, and building the SICO index. An in-memory index is used to minimize disk accesses. The memory usage we report includes all memory structures used by the SICO index module; this includes data structures for the PCA-SIFT features, the LSH index, and hash-based counters (each structure is detailed in our previous work [8] and that of Ke et al. [3]).

All experiments were performed on a Pentium IV 3 GHz machine with 1 GB main memory, running the Linux 2.4 kernel. The core SICO system is developed in C++, whereas the online demonstration system is built with AJAX.

5. RESULTS

As shown in Table 1, there are an average of 56 clusters identified by SICO within the returned image answers for a given query, with each cluster containing an average of approximately 6 suspected near-duplicate images. As shown in column 2, near-duplicate images comprises (on average) 22% of the image answers, reflecting the substantial level of near-duplication within those returned by image search engines. It takes a little less than 30 seconds, on average, for SICO to automatically identify image near-duplicates within the image results for each query. With further analysis, we find that over 70% of processing time is spent on the LSH component, whereas less than 30% is used to load image features into

[4] www.google.com/press/zeitgeist/archive.html
[5] http://images.google.com
[6] The list of image URLs (of each subject) used for retrieval can be found at http://sico.cs.rmit.edu.au/SICO_URLS.tgz

Table 1. Effectiveness of identifying near-duplicate instances within image answers using the queries are shown in column 1. Columns 2 and 3 show the ratio of near-duplicate images (NDI) in image answers, and the number of machine identified groups (clusters), respectively. Column 4 shows the correct (manually assessed) groups along with the percentages (within parentheses); column 5 shows the memory requirements of SICO for collation of near-duplicates. In the last column, the collation time on collections of approximately 622 (on average) images for each query is shown. Due to limited space, only fragments of the query keywords are shown.

Query subject	NDI ratio	Total groups	Correct groups (%)	Mem. (MB)	Time (sec)
50 Cent	31%	77	73 (95%)	249	30.9
Aaliyah	30%	80	75 (94%)	206	22.2
Jolie	14%	45	42 (93%)	192	15.8
Lavigne	26%	77	73 (95%)	222	24.4
Clinton	17%	59	54 (92%)	285	39.5
Marley	18%	60	56 (93%)	250	32.5
Pitt	15%	57	55 (96%)	220	25.9
Electra	20%	58	54 (93%)	218	23.8
Rumsfeld	20%	67	59 (88%)	244	29.4
Poe	25%	58	57 (98%)	248	30.9
Bush	15%	35	30 (86%)	261	34.8
Knightley	20%	56	53 (95%)	209	23.0
Cobain	31%	67	64 (96%)	264	33.2
Davis	19%	54	53 (98%)	217	24.8
Diana	14%	53	43 (81%)	274	37.3
Williams	25%	86	84 (98%)	240	31.2
Schiavo	33%	38	33 (87%)	230	26.9
Cruise	14%	63	60 (95%)	231	25.8
Tupac	28%	82	78 (95%)	245	29.7
Shakespeare	16%	37	32 (86%)	247	30.8

memory, and hash-based probabilistic counting, combined. This shows that the LSH hash tables can be further optimized for a more efficient system. The collation timings indicate that the processing time of SICO is practical, considering our modest experimental platform with large numbers of features (291 489) used for each image answer set.

Based on our evaluation, SICO accurately identifies approximately 93% of the clusters on average, where only 7% (of 100%) of the clusters are erroneous. An average of 238MB of memory is used during the processing. This finding is not surprising given that the number of image answers is typically limited to 1,000 for web search, which translates to modest memory requirements for the real-time SICO index. Overall, these results are pleasing, as they reflect the effectiveness of SICO for near-duplicate detection in web search using only modest processing and memory.

6. CONCLUSIONS

We have presented SICO, a novel system for automatic detection of near-duplicate instances within the results of a web image search. We have shown that our approach is accurate and efficient for this task, and allows images to be quickly reorganized based on the near-duplicate clusters using a modest experimental platform. Such a reorganization has a twofold benefit. First it allows suspect image copies within answer sets to be quickly identified. Second, near-duplicates are essentially redundant images that can be pruned to reduce the amount of repeated information in image answers. In future work, we intend to incorporate other robust interest points and local descriptors, and to expand SICO to allow image redundancy elimination to facilitate meaningful reorganization of image answers.

Acknowledgment. This work is supported by the Australian Research Council.

7. REFERENCES

[1] J.J. Foo, J. Zobel, R. Sinha, and S.M.M. Tahaghoghi, "Detection of image versions for web search.," in *Proc. ACM-CIVR*, 2007. To appear.

[2] A. Qamra, Y. Meng, and E.Y. Chang, "Enhanced perceptual distance functions and indexing for image replica recognition.," *IEEE Trans. PAMI*, vol. 27, no. 3, pp. 379–391, 2005.

[3] Y. Ke, R. Sukthankar, and L. Huston, "An efficient parts-based near-duplicate and sub-image retrieval system.," in *Proc. ACM-MM*, 2004, pp. 869–876.

[4] Y. Ke and R. Sukthankar, "PCA-sift: A more distinctive representation for local image descriptors.," in *Proc. CVPR*, 2004, pp. 506–513.

[5] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing.," in *Proc. VLDB*, 1999, pp. 518–529.

[6] E.Y. Chang, C. Li, J.Z. Wang, P. Mork, and G. Wiederhold, "Searching near-replicas of images via clustering.," *SPIE Multimedia Storage and Archiving Systems VI*, pp. 281–292, 1999.

[7] D.Q. Zhang and S.-F. Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning.," in *Proc. ACM-MM*, 2004, pp. 877–884.

[8] J.J. Foo, R. Sinha, and J. Zobel, "Discovery of image versions in large collections.," in *Proc. MMM*, 2007, pp. 433–442.

[9] D.G. Lowe, "Distinctive image features from scale-invariant keypoints.," *IJCV.*, vol. 60, no. 2, pp. 91–110, 2004.

[10] J.J. Foo and R. Sinha, "Pruning SIFT for scalable near-duplicate image matching.," in *Proc. ADC*, 2007, pp. 63–71.

[11] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors.," in *Proc. CVPR*, 2003, pp. 257–263.

[12] N. Shivakumar and H. Garcia-Molina, "Finding near-replicas of documents and servers on the web.," in *Proc. WebDB Workshop*, 1998, pp. 204–212.