Digital Investigation 11 (2014) S27–S35

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Fast indexing strategies for robust image hashes

Christian Winter*, Martin Steinebach, York Yannikos

Fraunhofer Institute for Secure Information Technology SIT, Rheinstr. 75, 64295 Darmstadt, Germany[1]

Keywords:
Robust hashing
ForBild
Indexing
Vantage point tree
Locality-sensitive hashing

ABSTRACT

Similarity preserving hashing can aid forensic investigations by providing means to recognize known content and modified versions of known content. However, this raises the need for efficient indexing strategies which support the similarity search. We present and evaluate two indexing strategies for robust image hashes created by the ForBild tool. These strategies are based on generic indexing approaches for Hamming spaces, i.e. spaces of bit vectors equipped with the Hamming distance. Our first strategy uses a vantage point tree, and the second strategy uses locality-sensitive hashing (LSH). Although the calculation of Hamming distances is inexpensive and hence challenging for indexing strategies, we improve the speed for identifying similar items by a factor of about 30 with the tree-based index, and a factor of more than 100 with the LSH index. While the tree-based index retrieves all approximate matches, the speed of LSH is paid with a small rate of false negatives.

Introduction

The forensic research community has created various tools for similarity search over the past decade. All these tools follow a two-step approach for identifying pairs of similar files: First they calculate short digests ("hashes") of the files, and then they compare the digests for similarity. Hence the hash function must map similar files to similar digests, and there must be a similarity function for the digests.

In the domain of multimedia data, the forensic research has adapted methodologies developed for multimedia retrieval and other multimedia applications. In particular, our *ForBild* tool for robust image hashing (Steinebach, 2012; Steinebach et al., 2012) has been developed based on the evaluation of different perceptual hashing methods (Zauner et al., 2011). The algorithm employed in ForBild is an improved version of *block mean value based hashing* (Yang et al., 2006). The choice of this algorithm is justified with its hash calculation speed and its low error rates. These properties are important requirements for forensic applications with huge amounts of data.

While the hash algorithm of the ForBild tool is based on the evaluation of various approaches, the search algorithm has not been considered by now. The ForBild tool searches for similar hashes in a naive way by comparing each query hash to each hash in the reference database. Although the hash comparison uses the Hamming distance, which can be calculated very efficiently, the naive brute force search requires a significant amount of time for databases with hundred thousands or even millions of images. Suitable indexing strategies should perform much better than brute force by restricting the search to a subset of the reference hashes for each query. Due to the computationally cheap Hamming distance only very effective (size of subset) and efficient (time needed for subset selection) indexes will be faster than a brute force search.

This paper presents two suitable indexing strategies we identified during the analysis of various approaches. These strategies can be applied to ForBild hashes as well as any

* Corresponding author. Tel.: +49 6151 869 259; fax: +49 6151 869 224.
E-mail addresses: christian.winter@sit.fraunhofer.de (C. Winter), martin.steinebach@sit.fraunhofer.de (M. Steinebach), york.yannikos@sit.fraunhofer.de (Y. Yannikos).

[1] http://www.sit.fraunhofer.de.

http://dx.doi.org/10.1016/j.diin.2014.03.004

other type of block mean value based hashes. Moreover, these strategies should also be suitable for other types of similarity hashes which are compared with the Hamming distance. Our first indexing strategy is based on a metric tree and presented in Sect. Tree-based index, and the second strategy is an LSH approach presented in Sect. LSH-based index. Evaluation results of these strategies are contained in Sect. Evaluation.

Block mean value based hashing

Block mean value based (BMB) hashing divides an image into a fixed number of blocks and calculates one hash bit for each block. ForBild uses $16 \times 16 = 256$ blocks; Yang et al. (2006) do not specify the number of blocks used in their work.

The hash bits are calculated according to the following procedure:

1. Convert the image to grayscale, i.e. remove the color information and retain the brightness information.[2]
2. Calculate the mean brightness of each block. This is an intuitive approach for scaling the image into the grid of blocks. The result is a tiny grayscale version of the image, which has one pixel per block. We call this result the intermediate hash of the image.
3. Determine the median value of the previously calculated mean values.
4. Set the final hash bit for each block according to whether its mean value is above the median or not. Hence the hash is a tiny bi-tonal version of the original image. For most images (very simple graphics are an exception) the hash has no visually recognizable content.

ForBild made two improvements for this approach: It calculates a separate median for each quadrant of the image to increase the hash collision resistance, and it has an automatic flipping mechanism to produce hashes robust against mirroring. Additionally, it inherits the robustness against image scaling (even non-proportional scaling), lossy compression, Gaussian filtering, noise adding, gamma correction, color adjustments, etc. from the original approach. These robustness properties and a low collision rate make it well-suited for identifying modified versions of known images. In the forensic domain blacklisting of child sexual abuse images is an obvious application for the ForBild tool.

Match decision

In order to check a given hash against a database of reference hashes (e.g. a blacklist), the Hamming distance is employed, which counts the number of non-matching bits. Hashes of versions of the same image have mostly identical bits while hashes of unrelated images should share on average half of the bits (128 in the case of ForBild) by chance. The procedure of selecting the closest hash from a reference list reduces the average Hamming distance to 62 for images unknown to the database (Steinebach, 2012). At a first glance, this is an unexpectedly low distance. A naive calculation under the assumption of independent and identically distributed (i. i. d.) bits implies that the distance of unrelated images should be above hundred with very high probability. However, the assumption of i.i.d. bits is not satisfied because neighboring bits of BMB hashes are strongly correlated. Consequently, the distribution of distance values is wider than expected, and hence the average of the best distance is lower than expected. This observation is important for our optimization in Sect. Choice of vantage points.

ForBild declares hashes with Hamming distance of at most 8 as good match. A distance above 8 and below 33 indicates a potential match. Such potential matches are reexamined by calculating a mismatch penalty, originally called “weighted distance” (Steinebach et al., 2012, Sect. 2.3) and credited to a “quantum hash” method developed by Jin and Yoo (2009). The calculation of the mismatch penalty requires as additional input the intermediate hash of one of the images.[3] Each non-matching bit of the two hashes is penalized based on the heuristic that a hash bit is less stable if the according intermediate value is closer to the median. The penalty for the mismatch of an unsteady bit (i.e. small difference between intermediate value and median) is small while the penalty for the mismatch of a reliable bit (i.e. large difference) is high. If the mismatch penalty falls below a threshold, the potential match is declared as match.

Query performance

The time needed for checking a query hash against a reference list obviously depends on the size of this list. As ForBild performs a naive linear search through the list, the required time is linear in the number of reference hashes. We evaluated the running time of the original ForBild tool using our workstation, and we measured an average time of about 5.0 ms for checking one pre-calculated hash against our reference list containing approximately 130,000 hashes (see Sect. Evaluation for details about the experimental environment). The hash calculation required on average 46 ms for an image from the reference image collection.[4] Thus the hash comparison needs about 10% of the total time in the present setting.

Advanced ForBild variants

While the ForBild hash is robust against many image operations, the underlying BMB approach does not

[2] None of the existing papers specifies a conversion method. Hence the retained “brightness” might be for example the luma, (gamma compressed) relative luminance, or perceptual lightness. ForBild actually uses luma.

[3] Hence the penalty function is asymmetric, which does not comply with the term “distance”.

[4] Our initial evaluation of the ForBild tool resulted in an average hashing time of 12.5 ms because a different image set with smaller average image size was used (Steinebach et al., 2012, Sect. 3.2). The figures presented by Breitinger et al. (2013, Table 2) conform to a linear dependency between image size and hashing time.