Robust Hashing for Efficient Forensic Analysis of Image Sets

Martin Steinebach

Fraunhofer SIT, Darmstadt, Germany
steinebach@sit.fraunhofer.de

Abstract. Forensic analysis of image sets today is most often done with the help of cryptographic hashes due to their efficiency, their integration in forensic tools and their excellent reliability in the domain of false detection alarms. A drawback of these hash methods is their fragility to any image processing operation. Even a simple re-compression with JPEG results in an image not detectable. A different approach is to apply image identification methods, allowing identifying illegal images by e.g. semantic models or facing detection algorithms. Their common drawback is a high computational complexity and significant false alarm rates. Robust hashing is a well-known approach sharing characteristics of both cryptographic hashes and image identification methods. It is fast, robust to common image processing and features low false alarm rates. To verify its usability in forensic evaluation, in this work we discuss and evaluate the behavior of an optimized block-based hash.

Motivation

Detection of the presence of illegal media material like digital images containing child pornography seems to be trivial: Either one uses cryptographic hashes to recognize the images or one compares the images with identified copies in a given database. But media documents may undergo several processing steps during their lifetime; while these operations do not modify the visual content of a document, its binary representation does change.

For example, media files are usually stored and distributed in compressed form. Such compression methods are often lossy and will render the decompressed data slightly different from the original copy (e.g. the JPEG format does not store perceptually insignificant parts of an image). Besides compression, the data may also undergo other incidental distortions such as scaling.

Therefore, the binary representation of media documents cannot directly be compared to each other. Perceptual or robust hashes provide an automated way of deciding whether two media files are still "perceptually identical", for example whether one document is a copy of another one, which was processed without changing its semantics. A hash is a short digest of a message, which is sensitive to modifications: if a document is severely changed, the hash value will change as well in a random manner. Hashes can be used to identify an object if the hash of the

P. Gladyshev and M.K. Rogers (Eds.): ICDF2C 2011, LNICST 88, pp. 180–187, 2012.

original copy is stored in a database. During the detection of illegal material, the found media files are hashed and theses hashes are searched for in the database. If a very similar hash is found, illegal content has been detected.

To effectively use this technology within forensic scenarios, a low computational complexity is vital. Cryptographic hashes are designed to by computed efficiently. Robust hash functions need to be in the same region of complexity to be accepted as an alternative to cryptographic hashes. Otherwise the delay caused by robust hash calculation is a hindrance in evidence analysis and the user ends with two options: Running a test which is robust against post processing and may find more evidence but will take more time than acceptable for the case or run a test that will meet time requirements but will find less evidence. Both options are not satisfying, but the second one will at least provide the chance for finding evidence for a case.

In contrast to cryptographic hashes, perceptual hashes allow to compute a hash of a document that remains invariant under some distortions that do not alter the perceptual characteristics of the document. Processed documents can still be reliably compared to each other.

Several perceptual hashes for various media types are known, which provide different levels of robustness. For example, Roover et al. [3] provide an image hash algorithm which is robust against geometrical operations like scaling and rotation; the hash draws its robustness from the use of the Radon transform. Friedrich and Goljan propose an approach based on random noise similarity in [4]. Zhou et al. [5] propose a robust hash for video streams, which is based on the similarity between spatial and temporal adjacent blocks. More video hashing approaches using perceptual models are given in [6] and [7]. Examples for audio hashing can be found in [8] and [9].

Robust Block Hash

There are many approaches in the literature suggesting robust hash functions. They mainly compete with respect to robustness against various attacks like lossy compression, scaling, distortion, cropping or rotation. For our forensic application we analyzed their computational complexity and found most of them using complex transformations like DCT or wavelet. While these operations help to survive many attacks and increase the robustness, they slow down the hashing process. Therefore we compared a number of hash algorithms and compared them with respect to their robustness and complexity. In our work [11] we show that the performance of the algorithm with the lowest complexity compares well with those of higher complexity. Therefore we decided to proceed with this algorithm based on block mean computation [12].

This method is described as follows:

a) Convert the image to grey scale and normalize the original image into a preset size.

b) Let N denote the bit length (e.g. 256 bit) of the final hash value. Divide the pixels of the image I into non-overlapped blocks $I_1,I_2,...,I_N$.

c) Encrypt the indices of the block sequence $\{I_1,I_2,...,I_N\}$ using a secret key K to obtain a block sequence with a new scanning order. The authors specify no further details about what encryption algorithm to use. For our application encryption is not required, therefore this step is skipped.

d) Calculate the mean of the pixel values of each block. That is, calculate the mean value sequence $\{M_1,M_2,...,M_N\}$ from the corresponding block sequence. Finally obtain the median value Md of the mean value sequence.

e) Normalize the mean value sequence into a binary form and obtain the hash value h(i) as 0 if M_i <Md or 1 if $M_i \geq$ Md

Figure 2 shows the hashing process from an example image to a robust hash of size 16x16 or 256 bit. It must be noted that the whole process only requires simple operations like mean calculation and comparison of values, but no transformations.

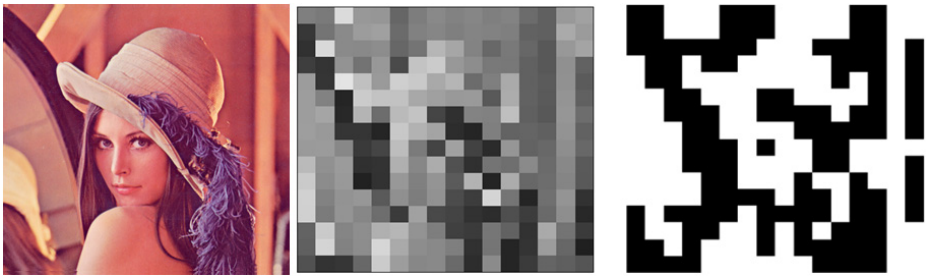

Fig. 1. Creation of Block Hash

Optimization

Evaluating the block mean value hash, we identified some drawbacks in its design leading to unnecessary error rates and fragility which were easy to counter

1. The mean calculation over the whole image is prone to produce very similar hash for images of a certain design. As an example, all images where half of the image is a dark area, like a forest, and the other half is a light area, like the sky, are hashed as a sequence of all 1s in the on half of the hash and 0s in the other half. To counter this, we introduced a segmentation of the hash in four subareas. The mean value of these subareas is then used to for the threshold decision. This leads to a much better distribution of 1s and 0s in critical images. In the example above now the lighter parts of the ground would be distinguished from the darker ones in the given subarea, leading to a salt and pepper hash being individual for each image of similar structure.
2. The algorithm is fragile against mirroring of the image. This is a serious drawback as especially horizontal mirroring often does not reduce the perceived quality of an image. Simply storing both the original and its mirrored counterpart in the hash database would increase the database which is desired to be as small as possible due to efficiency. Running two database lookups at the comparison stage is also not efficient. The most efficient method we identified is an automatic mirroring of the image during hash calculation in such a way that the darkest subarea is always

on the upper left. The darkest area can easily be identified by the lowest mean value of the four subareas which need to be calculated for the optimization step discussed above. By this, our hash algorithm is robust against any type of mirroring. As an additions feature, if the differences of the mean values are too low, more than one database lookup with different mirroring options is allowed during detection. This is necessary to prevent lossy compression to disable detection by randomly altering the relationship between two areas. The threshold for multiple lookups has been fixed at 10% difference between lowest and second lowest area.

Weighted Distance

After the two optimizations the block mean robust hash showed a very high robustness to lossy compression, scaling and mirroring. Error rates with the algorithm are about 1 percent. While this is a good overall performance for robust hashing, it still is too high for forensic evaluation. Interestingly, especially the false alarm rate is seen as problematic by users, while a false rejection rate of 1%, meaning 1% of all analyzed illegal images being known to the database are not recognized, is acceptable. The challenge of the false alarm rate is the required human observer needed to remove all false alarms from the evidence collection. A false alarm rate of 1% for one million images still means that 10,000 images need to be verified.

Therefore another decision method in addition to the hamming distance is applied. The idea here is looking at all hash bits of a test image which are not equal to the hash stored in the database. We calculate how far away from the mean of the subarea they are. If they are close to the mean they are much more likely to have flipped due to post production operations than those with a great distance to the mean value. We call this weighted distance as the hash bits can have different weights in the decision if two hashes are equal or not.

The calculation of the weighted distance is defined as follows: The variance of the distance to the mean value of the hash bits from the test image hash not equal to the database hash is divided by the variance of the equal bits, multiplied by the hamming distance and then multiplied by 1,000 to receive values in a range comparable to the hamming distance.

Decision

The decision process utilizing both hamming distance and weighted distance is executed as follows:

1. The robust hash h of the test image is calculated.
2. h is compared to all hashs stored in the database. The hash in the database with the smallest hamming distance d to h is identified
3. If $d \leq 8$, it is identified as the image in the database
4. If $d \leq 32$ but > 8, the weighted distance is calculated. If the weighted distance ≤ 16, the image is identified

In most cases test images will be rejected as their hamming distance is well over 32. In theory a random image hash of 256 bits in average will have a hamming distance of 128 to another randomly selected image as there is a 50/50 chance that bits are identical. Due to some typical image characteristics and their impact on the calculated hash in practice the average random distance is smaller.

From our experiments we noticed an average minimal hamming difference of 62 for images not in the database. Of course this cannot be compared with the average hamming distance of 128 as here we look at the smallest hamming distance found in our database. Still the number is surprisingly low. Our choice of hamming distance 32 as the threshold for a close investigation comes from this number. The idea here is that any image with a hamming distance half of the average hamming distance is worth further investigation.

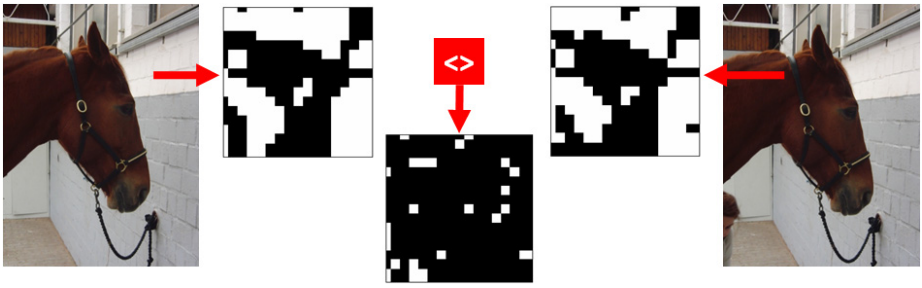

Fig. 2. Block Hash Differences in similar Images

A robust hash is still a hash function aiming to distinguish images from each other, therefore images which are quite similar still should feature sufficiently different hashes to hold each other apart. In figure 3 one can see that our block hash is able to do so. While the hamming distance of the similar images is below 32 as one can count in the lowest part of the image, due to the strong differences in the lower left of the two images caused by the human head appearing, the weighted distance is well above 16. So even very similar images can be distinguished which is important for forensic applications.

Evaluation

We evaluated our optimized and efficient robust hash with a test set aiming to be close to typical applications like detection of illegal pornographic images. Therefore we used a set of 4,400 images of a cheerleader team. The common characteristics are a large quantity of skin colors, very similar poses over all images and presence of one or more persons in all images. All 4,400 images where randomly divided into two groups. One group was added to a hash library already featuring 88,000 hashes of other images. This group was used as the test set to be recognized, simply called “known” images. The other group was used as a test set to evaluate the false alarm rate. As these images are “unknown” to the database, they must not trigger alarms.

Figure 4 shows four examples from the two sets. The two images with the black border are in the "known" group, the two images with the grey border are in the "unknown" group. For our evaluation is was important to have such similar images as only if the false alarm rate is low with such sets automated detection of illegal pornography can be executed efficiently due to the vast amounts of legal pornography that can be assumed to be present at image collections analyzed during forensic operations.

The evaluation procedure was as follows: All 4,400 images were scaled down by a factor that ensured that the larger edge was only 300 pixels long, meaning an average size reduction of 25% compared to the original images. The images were mirrored horizontally and stored with a JPEG quality factor of 20, producing small images of low quality. This procedure simulates rather strong changes occurring in usual image conversion.

Fig. 3. Test Set Examples (http://www.cheerleader-frankfurt.de/galacticdancers)

Figure 5 shows the detection success utilizing only the hamming distance as a feature to distinguish known and unknown images. One can see how well both groups can be distinguished by the robust hash. Still, in some cases false rejections and false acceptances occur. With hamming distance 32 as the threshold, the false acceptance rate is 1.1%, the false rejection rate is 0.4%. When using the hamming distance of 8 as a threshold, false rejection rises to 10% while false acceptance drops to 0%.

In figure 6 we show the advantage of using both hamming distance and weighted distance. Images are now only accepted if they either have a hamming distance of 8 or less or a weighted distance of 16 or less. Here the false acceptance rate drops to 0%, while the false rejection rate is only 0.2%.

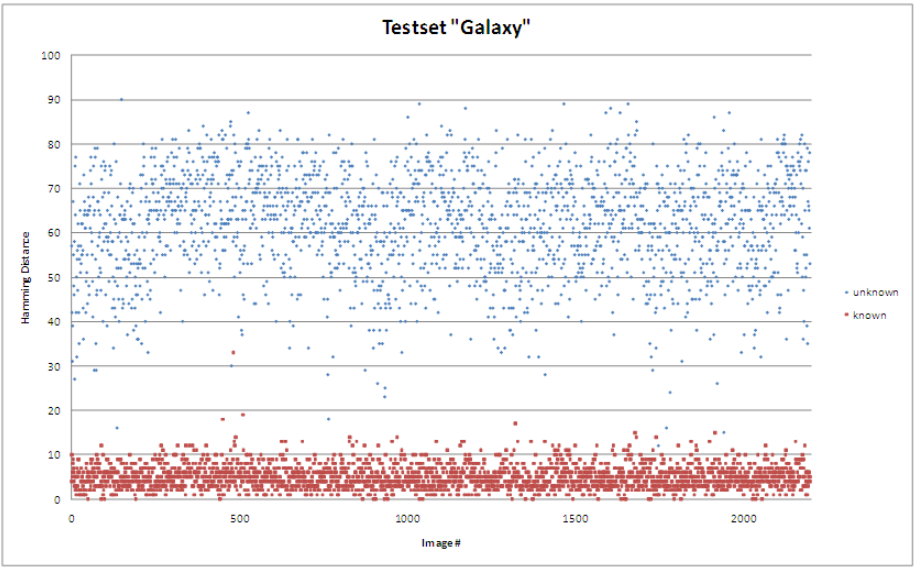

Fig. 4. Block Hash Distinction by Hamming Distance

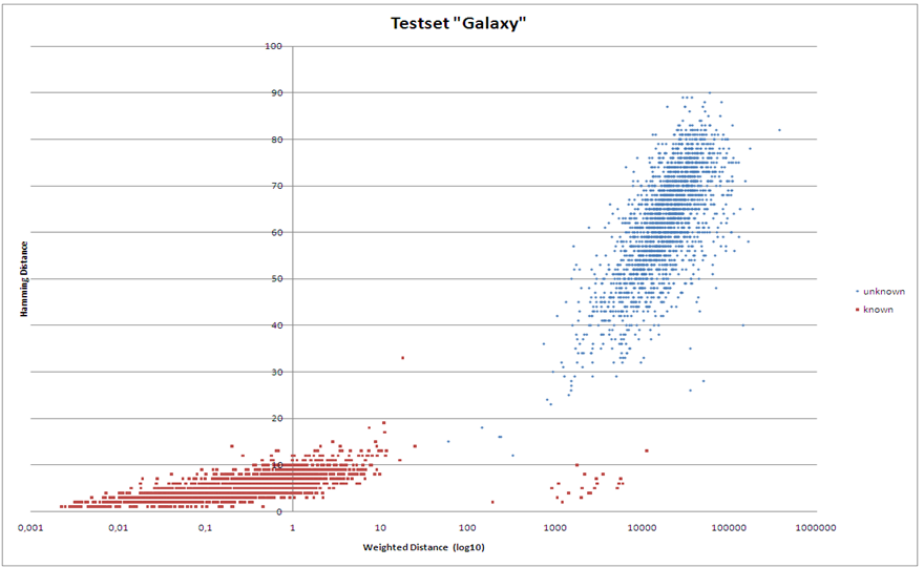

Fig. 5. Block Hash Distinction by Weighted Distance

Summary

In this work we show how robust hash function can be used in forensic image detections as an alternative to cryptographic hashes. To achieve an acceptable

performance of the detection process, only robust hash function of low computational complexity can be utilized. We show how such a robust hash function, the block mean hash can be optimized to perform at acceptable error rates. With a set of optimizations we achieve a false acceptance rate of 0% and a false rejection rate of 0.2%. This makes our hash function suitable for practical forensic image analysis.

This work was supported by the government of Hessen in LOEWE (Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz) as the funded project ForBild.

References

1. Cano, P., Batle, E., Kalker, T., Haitsma, J.: A review of algorithms for audio fingerprinting. In: Proceedings of the IEEE Workshop on Multimedia Signal Processing, pp. 169–173 (2002)
2. Boneh, D., Shaw, J.: Collusion-Secure Fingerprinting for Digital Data. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 452–465. Springer, Heidelberg (1995)
3. Roover, C.D., Vleeschouwer, C.D., Lefebvre, F., Macq, B.: Robust video hashing based on radial projections of key frames. IEEE Transactions on Signal Processing 10(10) Part 2, 4020–4037 (2005)
4. Fridrich, J., Goljan, M.: Robust hash functions for digital watermarking. In: Proceedings of the International Conference on Information Technology: Coding and Computing, pp. 178–183 (2000)
5. Zhou, X., Schmucker, M., Brown, C.L.: Video Perceptual Hashing Using Interframe Similarity. In: Sicherheit, pp. 107–110 (2006)
6. Liu, T., Zhang, H.-J., Qi, F.: A novel video key-frame-extraction algorithm based on perceived motion energy model. IEEE Transactions on Circuits and Systems for Video Technology 13(10), 1006–1013 (2003)
7. Oostveen, J., Kalker, T., Haitsma, J.: Visual hashing of video: application and techniques. In: Wong, P.W., Delp III, E.P. (eds.) IS&T/SPIE 13th Int. Symposium on Electronic Imaging San Jose, Security and Watermarking of Multimedia Contents. SPIE – The International Society for Optical Engeneering, CA, USA, p. 4314 (2001)
8. Haitsma, J.A., Oostveen, J.C., Kalker, A.A.C.: Robust audio hashing for content identification. In: Content based Multimedia Indexing (CBMI 2001), Brescia, Italy (2001)
9. Allamanche, H., Helmuth, F., Kasten, C.: Content-Based Identification of Audio Material Using MPEG-7 Low Level Description. In: Electronic Proceedings of the International Symposium of Music Information Retrieval (2001), `http://ismir2001.ismir.net/papers.html`
10. Schneier, B.W.: Applied Cryptography, 2nd edn., ch. 18. Wiley (1996)
11. Zauner, C., Steinebach, M., Hermann, E.: Rihamark: Perceptual Image Hash Benchmarking. In: Proceeding of Electronic Imaging 2011 - Media Watermarking, Security, and Forensics XIII (2011)
12. Yang, B., Gu, F., Niu, X.: Block mean value based image perceptual hashing. In: Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Multimedia Signal Processing (IIHMSP), pp. 167–172. IEEE (2006) ISBN 0-7695-2745-0