FaceHash: Face Detection and Robust Hashing

Martin Steinebach(✉), Huajian Liu, and York Yannikos

Fraunhofer SIT, Rheinstrasse 75, 642955 Darmstadt, Germany
martin.steinebach@sit.fraunhofer.de

Abstract. In this paper, we introduce a concept to counter the current weakness of robust hashing with respect to cropping. We combine face detectors and robust hashing. By doing so, the detected faces become a subarea of the overall image which always can be found as long as cropping of the image does not remove the faces. As the face detection is prone to a drift effect altering size and position of the detected face, further mechanisms are needed for robust hashing. We show how face segmentation utilizing blob algorithms can be used to implement a face-based cropping robust hash algorithm.

Keywords: Robust image hash · Face detection · Blob detection

1 Motivation

Today we see a rapid increase of hard disk drive capacities, which allows storing a huge amount of data, including media files. This leads to sometimes massive collections of files, especially in the case of images. These can be private or third party images from the Internet. The content of the images goes from legal personal, natural to synthetic but also to illegal (e.g. child pornographic).

If an investigation of such a collection occurs, manual scanning and classification parts of these collections is very time consuming, ethically questionable and practically unfeasible. In digital forensics, searching in over one terabyte of images for evidence against an offender is common and needs to be very fast and accurate.

This problem is addressed by several methods in image forensics, such as content classification and image identification [1]. One common approach is the use of cryptographic or robust hashing. While robust hashing is resistant against many operations, many of the known methods only take the whole picture into account. To prevent evidence from being detected, offenders can crop out regions of interests of their illegal image and save them, removing unwanted environmental parts of the image. Then, the above methods cannot recognize these saved parts, since the rest of the image is absent. The result is missing evidence against the offenders.

We introduce a new approach named FH (Face Hash), which aims at solving this problem. FH contains a combination of digital forensics and computer vision methods. The focus region of this approach is the face. Identifying and distinguishing faces is an ability every human possesses and masters with growing age over time [2]. By looking at a picture every human detects known people, recognizes and judges about their emotional reaction at that moment. Hence, if someone modifies an image by cropping, the region of interest not to be removed will contain the face.

P. Gladyshev et al. (Eds.): ICDF2C 2013, LNICST 132, pp. 102–115, 2014.
DOI: 10.1007/978-3-319-14289-0_8

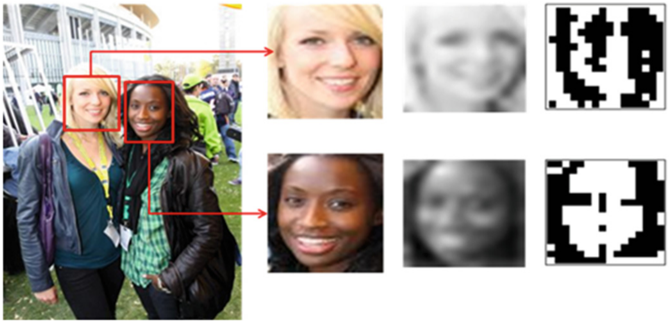

Fig. 1. Face detection and robust hashing (© http://www.cheerleader-frankfurt.de/galacticdancers)

Also in child pornography face expressions are an important part, which the offender uses as proof of their entertainment to normalize the process of collecting [3]. An offender would therefore keep the face as part of his region of interest from the image. Therefore the FH approach uses a combination of face detection and image identification methods to quickly recognize illegal images. The result would be a manageable set of binary hash values (face hashes), which describe the victims or offenders within an illegal content in specific images. Comparing those face hashes against others from a police database, enables classifying them as legal or known illegal.

2 Robust Hashing

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 a comparison of hash algorithms with respect to their robustness and complexity is important. In [5] the authors show that the robustness of the Block Mean Value algorithm featuring the lowest complexity compares well with those of higher complexity. An optimization of this algorithm based on block mean computation improves robustness, speed and error rates [6].

2.1 Block Mean Value Hash

In 2006, Yang et al. proposed a block mean value based perceptual image hash function in [4]. Four slightly different methods are proposed. The latter two additionally incorporate an image rotation operation to enhance robustness against rotation attacks. This increases the computational complexity of the latter two methods. Due to our low complexity requirement we focus on the simplest method.

This method is described as follows:

1. Convert the image to grey scale and normalize the original image into a preset size.
2. Let N denote the bit length (e.g. 256 bits) of the final hash value. Divide the pixels of the image I into non-overlapped blocks $I_1, I_2, \ldots, I_N$.
3. Encrypt the indices of the block sequence $\{I_1, I_2, \ldots, I_N\}$ 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.
4. Calculate the mean of the pixel values of each block. That is, calculate the mean value sequence $\{M_1, M_2, \ldots, M_N\}$ from the corresponding block sequence. Finally obtain the median value M_d of the mean value sequence.
5. Normalize the mean value sequence into a binary form and obtain the hash value h(i) as 0 if $M_i < M_d$ or 1 if $M_i \geq M_d$.

Yang's analysis against cropping attacks on the mean block perceptual hash showed that the original and cropped images differ after 10 % cropping up to 24 % and after 20 % cropping up to 33 %. So while it is robust against many common attacks, cropping remains a major challenge for this algorithm.

2.2 Optimization

In [6] the algorithm is improved by a number of additional features: To make it robust against mirroring, it automatically flips the image to be hashed in such a way that the brightest corner always lies in the upper left. It calculates the hash by dividing the 16 × 16 area into four 8 × 8 subareas and computes the median for each of these areas to achieve a higher difference of images featuring a similar structure. And it uses a weighted hash comparison to increase the rate of correct rejections. In the following, only the optimization steps one and two are utilized.

3 Face Detection

Face detection today is a common mechanism in security and entertainment. Photo software uses face detection for tagging and organizing image collections. Facebook and Google+ use face detection and recognition for tagging and searching. In this section we briefly describe the face detection we use in our face hash. Both are based on the work of Lienhart and Mayds [7], one is the OpenCV implementation, the other one the face detection of the Scilab Image and Video Processing Toolbox.

At this stage of our work we only consider frontal face detectors, due to the lack of open source implementation of multi view face detector.

3.1 Viola and Jones Rapid Face Detector

One of the fastest state of art algorithms with real time processing abilities is Viola and Jones Face Detector [8, 10]. It uses Haar-like features, which can be computed very fast. Afterwards classifiers are trained with a machine learning technique called AdaBoost. Detecting faces with only these two techniques is possible but still the computation costs are too high for real time face detection. The speed of the face detector is increased with a so-called “cascade” structure of classifiers. It is a degenerated tree with n stages. The Viola and Jones detector has five stages. At each stage negative examples are filtered out. The detection rate of the detector is around 95 % by 50 false positives on the MIT and CMU test set.

Lienhart and Maydt extended the feature set from the Viola and Jones face detector with 45° rotated Haar-like features. For calculating these features, a new rotated integral image version was created. This face detector implemented in OpenCV is used for FH. Also the face detector of the Scilab Image and Video Processing Toolbox implemented by Shiqi Yu and Jia Wu quotes the work of Lienhart and Maydt as its base.

4 Challenge

As introduced in Sect. 1, we combine a face detector with a robust hashing algorithm. One of the major challenges for robust hashing still is cropping. While in theory it is simple to deal with cropping by using sections and redundancy, this will lead to very complex and time consuming search strategies in the forensic analysis. Therefore we utilize the detection of face areas and calculate hashes of these areas. As robust hashes are robust against scaling and lossy compression, it can be assumed that as long as the face detector always provides the same area of the image to the hashing software, the detection will be successful.

But this is a challenge for the face detectors. As we show in Fig. 2, detected face areas differ from version to version of an image. For a face detector it is not important to provide exactly the same area in every time as long as it successfully identifies the area containing the face. Cropping the image leads to a drift of the area’s size and position. Detailed test results are given in Sect. 6.1.

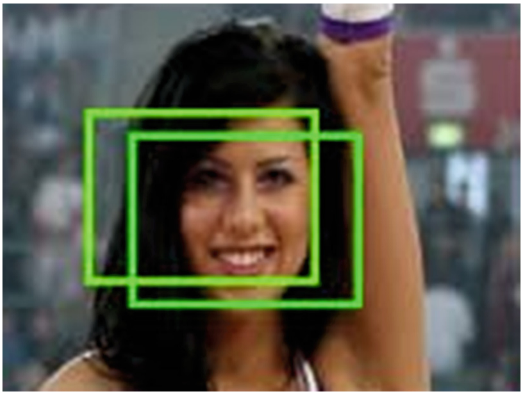

Fig. 2. Face detector inconsistency (© http://www.cheerleader-frankfurt.de/galacticdancers)

5 Face Blob Hashing

To counter the challenge of face drifting, we propose to utilize sub partitioning of the faces to be analyzed. As the faces are still rather similar even under the effect of drifting, our concept is to use individual objects within the face which can be detected and hashed. The position of these objects should move together with the detected face. Face detection and blob segmentation work well together as face detection is very robust except its drift against cropping and lossy compression. Without face detection, cropping would have a serious impact on the blob analysis. Only the relatively stable structure of a face can be utilized for blob analysis and robust hashing.

Our approach uses blob segmentation. For our test implementation, we apply the SearchBlobs and AnalyzeBlobs algorithms of IPT [12]. Blob thresholds are also calculated by IPT based on Otsu [11]. The Face Blob Hashing algorithm can be summarized as follows (see also Fig. 3):

1. Detect faces in an image using Viola and Jones Rapid Face Detector.
2. For each face, analyze its blobs.
3. For each blob, calculate its bounding box. A minimum of pixels (1 % of all face pixels) must be present in the bounding box, otherwise the blob is discarded.
4. For each bounding box, calculate its robust block hash.

In the training phase, all images to be recognized run through the four steps above. Each robust hash of each blob is stored in a database together with the name of the original image it is retrieved from.

As a result, a set of robust hashes for the individual blobs derived from all detected faces in the image is created. If more than one face is found in the image, the blob hashes of all face blobs are stored in the database as the goal of the algorithm is to identify images and not individual faces. It is not necessary to distinguish between the individual faces of an image.

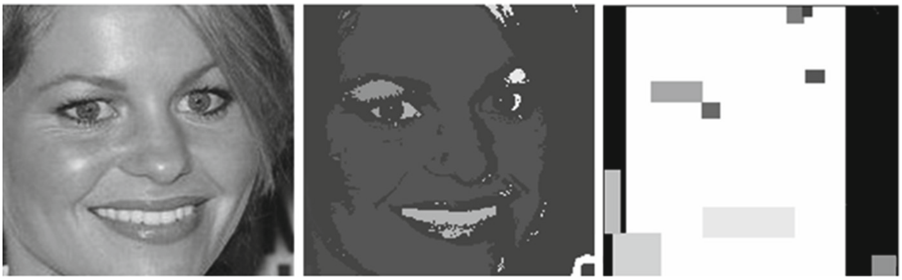

Fig. 3. Face (left), Blobs (middle) and Blob Areas (right)

In the detection phase, for an image to be analyzed also all four steps above are executed. Then the minimum hamming distance for each blob hash to all blob hashes in the database is calculated. For all blobs featuring a reference with a hamming distance equal to or smaller than a given threshold (usually 32) the reference to the image in the database and the hamming distance is stored in a result set. The blob hashes can be discarded at this point.

Now the image in the database with the most blobs similar to the analyzed image is derived from the result set. Only if at least two blobs in the result set point to the same image, the result is trusted and the image is tagged is identified. By this, false positives are reduced significantly. Random occurrences of small hamming distances between blobs and database entries happen in some cases, but as they are random, they do not share the same reference image and thereby can easily be filtered.

The average hamming distance of all result set entries pointing to that image is calculated to provide a quality rating for the detected match.

6 Test Results

In the following we provide our current state of test results, both for the motivating drift and for the blob hash detector. We also provide performance information comparing face detection and hash calculation.

6.1 Face Drift

To better understand the individual behavior of the face detection algorithm, we executed a deep analysis of the drift of the detected face area from version to version.

We used 10 images with one or more faces in it, and executed an automatic cropping starting in the upper left of the image. Both in the x and y axis between 0 and 20 pixels were cropped, resulting in 441 cropped versions per image. Images for this stage were taken from stock photo catalogues (Fig. 5) and model set cards (Fig. 7). Therefore a very high image quality was given for all 10 test images.

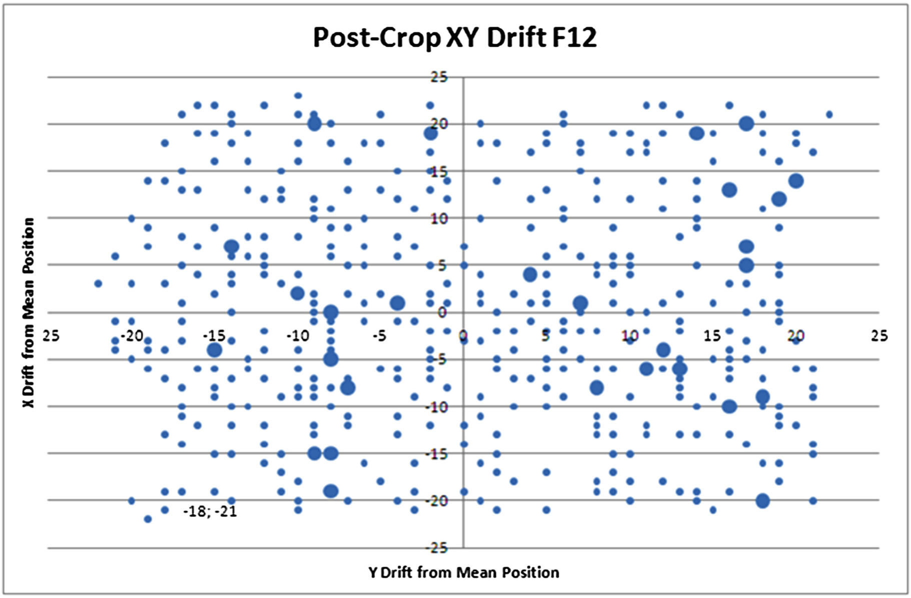

Fig. 4. Drift of example images

We determined the individual position of the identified face area and matched it to the original area of the image before cropping. After doing this for each of the 441 version of a face, we calculated the mean identified face position. Then we calculated the individual distance of all versions to this mean value. Figure 4 shows the result of this analysis. The positions of the detected face drift almost randomly in a range from ±22 in horizontal and vertical direction. A drift of this strength means that the detected face will not have robust hash equal or similar to the one retrieved from the original image.

Figure 6 shows examples of area drifts. Figure 5 was cropped several times and these three face areas have been among the ones detected by the face detector. While the face is equal for a human observer, for the robust hash algorithm they differ largely.

6.2 Face Blob Hamming Distance Results

To evaluate our face blob robust hash, we took 4600 images and tried to extract all faces within. These images are the same test set we already used in [13], so test results can be directly compared. The set consists of the online gallery of a cheerleader team, showing one or more person on almost any photo. We use these photos as the inter-photo similarity due to similar poses and dresses is very high, resulting in a challenge similar to distinguishing legal and illegal pornographic images. Figures 1 and 2 show examples of the set.

Due to the behavior of the face extractor, also false positives, meaning regions of the image wrongly identified as a face, have been extracted. Altogether, 6411 faces and false positives were detected. For each of these, the blob hash set as discussed in Sect. 5 was calculated and stored in a database.

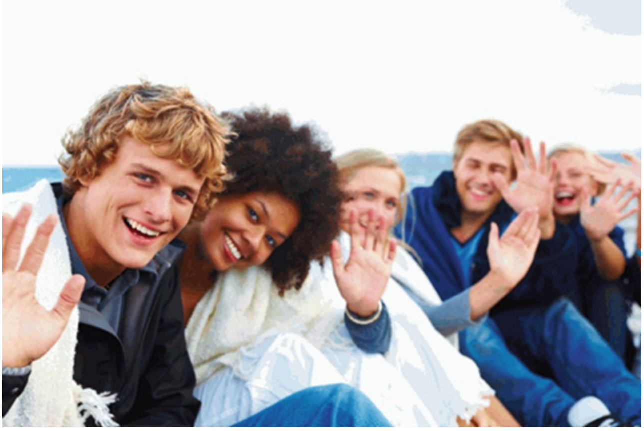

Fig. 5. Original Photo

Then an automated cropping attack was executed, cropping an area of 300 × 300 pixels at position (20, 20) out of each of the 4600 images (see Fig. 7). From these cropped copies, the faces again were extracted. This time, 3237 faces were found, the rest was lost due to the cropping.

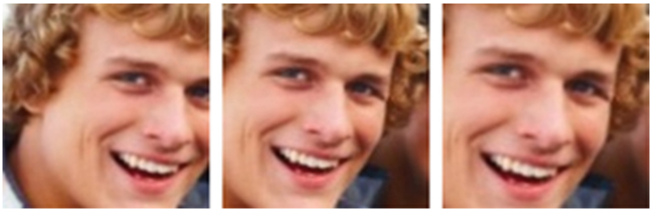

Fig. 6. Real world drift results

Now for each face the overall blob hamming distance was calculated to verify that the correct images have been found by the face blob robust hash. Figure 8 shows the result of our evaluation. The average hamming distance of all blobs for the correct faces ranges from 0 to 32 as could be expected. At the end, half of all faces could be correctly identified after cropping. False positives did not occur.

In Fig. 9 we can see that for most images, more than one face blob with a hamming distance below the threshold of 32 could be found. This is helpful when it comes to identifying false positives. For test images not stored in the database the average amount of blobs coming from a single image (a false positive) in the database is 1.03, while for known images it is 2.79. Therefore, combining a low hamming distance threshold for the individual blobs with a minimum requirement of two detected blobs is a reliable strategy for detecting images containing faces.

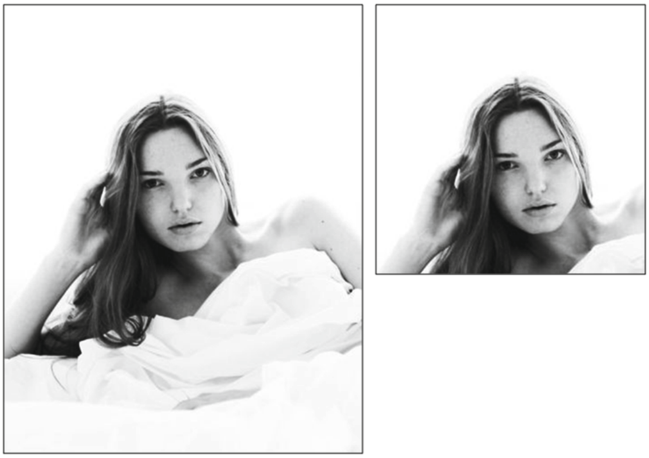

Fig. 7. Left: Original, Right: Cropped (http://models.com/newfaces/dailyduo/3624)

To verify that false positive rates can be expected to remain low, in Fig. 10 we provide the histogram of blob hamming distances of blobs derived from 100 faces not stored in the database. The average hamming distance is 52 in this case. This is the average hamming distance of a blob found in the database matching the unknown blob most closely.

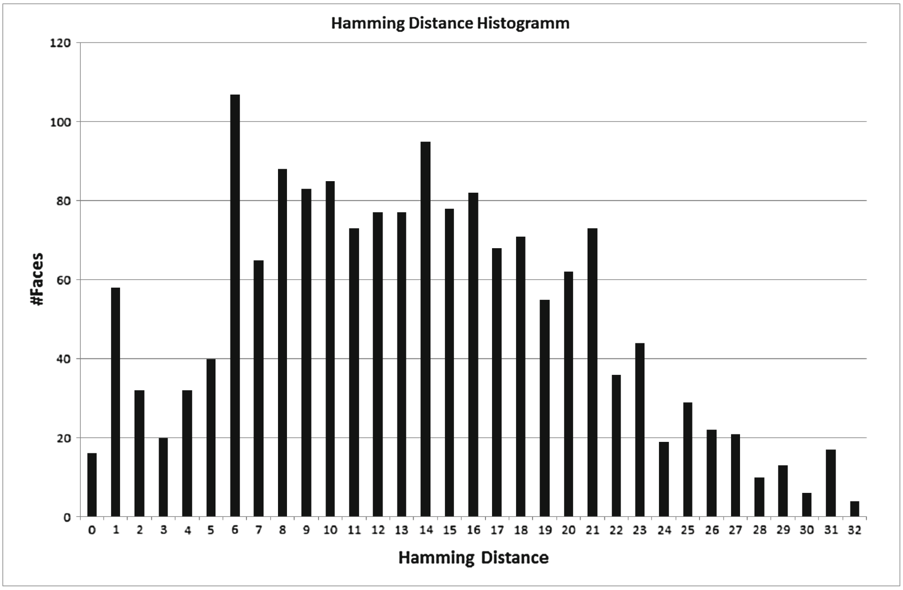

Fig. 8. Hamming Distance Histogram of 1658 faces

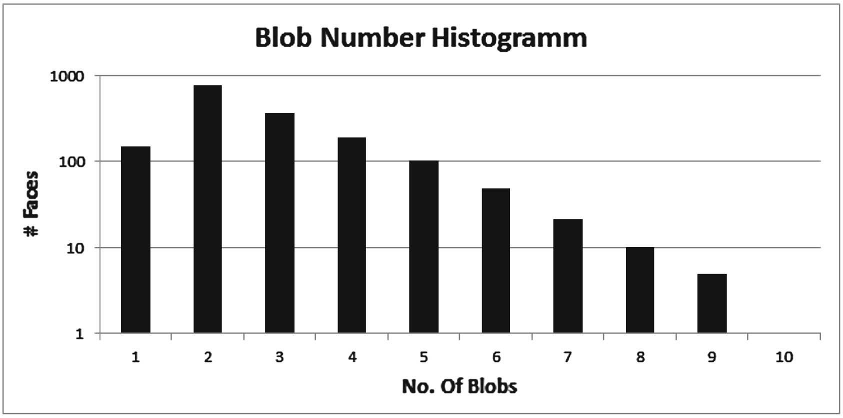

Fig. 9. Correct elements (Blobs) found per image

In Fig. 11 we provide an example of our decision strategy. Here in addition to cropping and JPEG compression also horizontal mirroring has been executed on the test image. For better visualization, we select 10 example images. The black rectangles show the average hamming distance calculated from all blob candidates. This means: Only if at least 2 blobs share the same face reference with a hamming distance below 33 they are taken into account. At face#6 one of the blobs pointing to the correct face has a hamming distance of 44 and therefore is ignored as the hamming distance

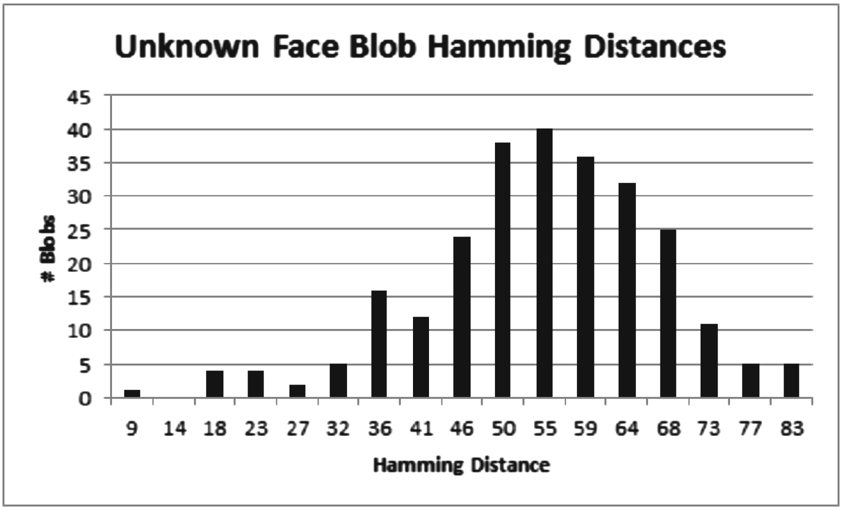

Fig. 10. Hamming Distance Histogram for 100 unknown faces

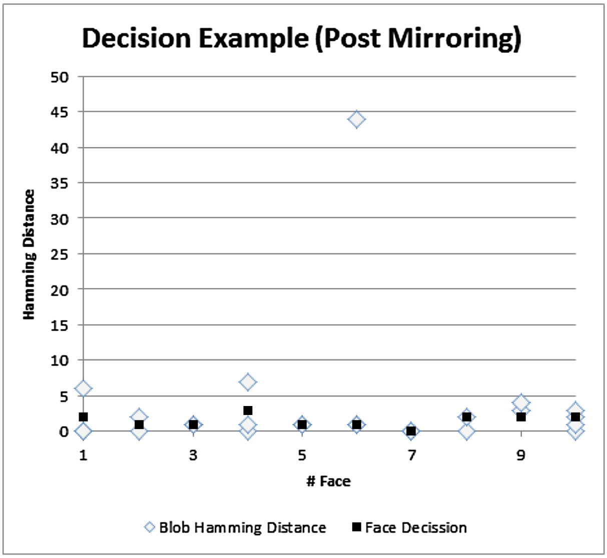

Fig. 11. Decision example for 10 faces

threshold is 32. As still 2 other blobs pointing to face#6 with a hamming distance of 1 were detected, the face is nevertheless identified correctly.

In Figs. 12 and 13 an example of the blob hash robustness against scaling, a common attack besides cropping is given. In this case, 100 images were either scaled down by 20 % or scaled up by 100 %. The histograms show that the hamming distance is below the threshold of 32 in most cases. Overall, similarly to only cropping the image, after adding scaling the images could be identified in half of the cases.

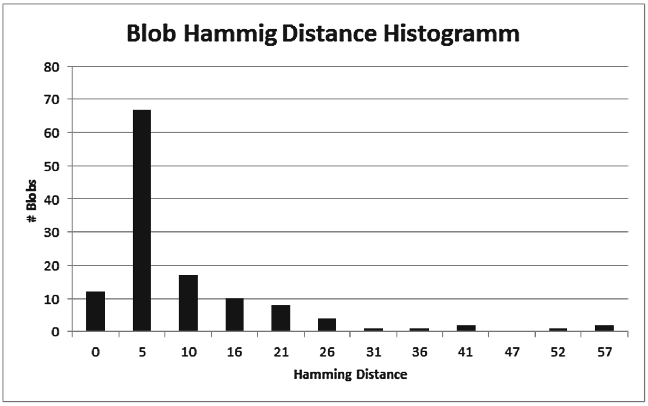

Fig. 12. Hamming Distance Histogram after downscaling by 20 %

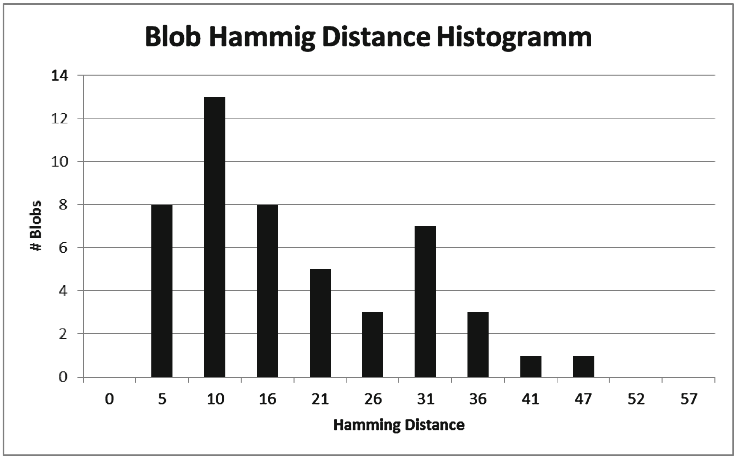

Fig. 13. Hamming Distance Histogram after upscaling by 100 %

6.3 Processing Speed

Using robust hashing in forensic application requires a processing speed high enough not to become a hindrance in examinations. Therefore in this section we provide a first impression of the performance of our approach. As stated above, a detailed view on its performance cannot be given as parts of the application have only been implemented in a prototypic manner.

We compare the speed of robust hash generation and face detection. As a test base, we use 198 images in which 340 faces are detected. The implementation of Lienhart's face detector in OpenCV is used for face extraction. The test is performed on a PC with Intel Core 2 Duo T9400 CPU (2.53 GHz) and 4 GB RAM.

The face detection takes 128.5 s. Hashing these faces only requires 0.57 s. Processing one image with face detection takes an average of 649 ms; hashing one face takes 1.6 ms. When applying the robust hash directly on an image, average hashing will take 4 ms. The speed-up in face hash calculations comes from smaller face areas compared to full images, resulting in lower costs for scaling and JPEG decoding (Figs. 14 and 15).

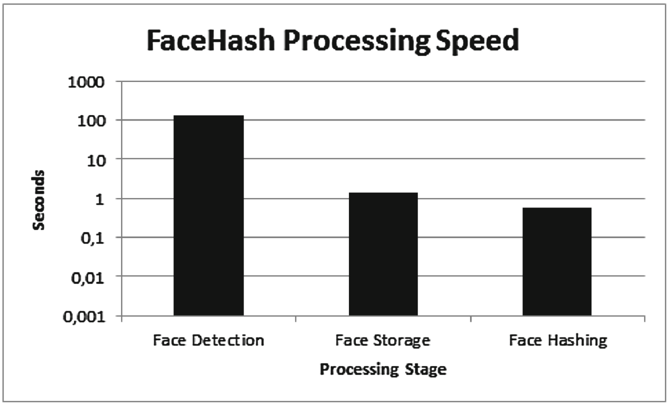

Fig. 14. FaceHash Processing Speed for 198 Images

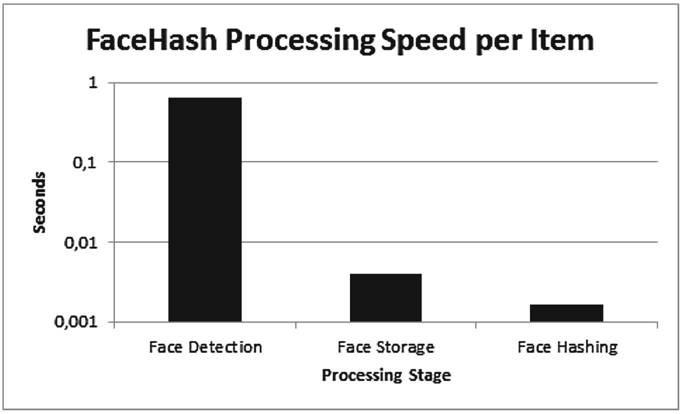

Fig. 15. Average Processing Speed for 198 images (Face Detection) and 340 faces (Face Storage, Face Hashing)

7 Conclusion and Future Work

In this work we present a novel approach to achieve robustness of robust image hashing against cropping. We show that a combination of face detector, blob segmentation and robust hashing provides promising test results and is able to re-identify images even after strong cropping combined with scaling and lossy compression. Our evaluation focuses on cropping, as most other image modifications like brightness and contrast changes or strong lossy compression have been successfully addressed by existing robust hash algorithms including the one we utilize as the robust hash base function.

The approach successfully detected half of all faces after a significant amount of cropping. Mirroring and scaling did not cause a serious reduction in detection quality. Utilizing the algorithm therefore provides a good chance to detect illegal images featuring human faces even after cropping. The computational complexity is kept as low as possible using only efficient algorithms for the individual steps. A real-world benchmark currently is not possible due to the prototypic implementation in Scilab. The resulting speed would be unrealistically low. Still, a comparison of our hash and a face detection mechanism shows that the difference between both mechanisms regarding computation time is huge. On the average, face detection is 200x slower than hashing.

Future work needs to evaluate how robust our approach is if a combination of cropping and other image modifications occurs. Another known challenge is image rotation. Here we will first need to evaluate the robustness of face detection. After successfully detecting a face, a normalization of the face angle should be possible, enabling image identification. Another line of future research is dropping the need of face detection and utilizing image segmentation directly on the images and not on extracted faces.

Acknowledgments. This work has been supported by the projects ForSicht and CASED, both funded by the State of Hessen.

References

1. Poisel, R., Tjoa, S.: Forensics investigations of multimedia data: a review of the state-of-the-art. In: 2011 Sixth International Conference on IT Security Incident Management and IT Forensics (IMF), pp. 48–61, May 2011
2. Schwarzer, G., Massaro, D.W.: Modeling face identification processing in children and adults. J. Exp. Child Psychol. **79**(2), 139–161 (2001)
3. Quayle, E., Taylor, M., Holland, G.: Child pornography: the internet and offending. Isuma Can. J. Policy Res. **2**(2), 94–100 (2001)
4. 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 (IIH-MSP), pp. 167–172. IEEE (2006). (ISBN 0-7695-2745-0)
5. Zauner, C., Steinebach, M., Hermann, E.: Rihamark: perceptual image hash benchmarking. In: Proceeding of Electronic Imaging 2011 - Media Watermarking, Security, and Forensics XIII (2011)
6. Steinebach, M.: Robust hashing for efficient forensic analysis of image sets. In: Gladyshev, P., Rogers, M.K. (eds.) ICDF2C 2011. LNICST, vol. 88, pp. 180–187. Springer, Heidelberg (2012)
7. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: IEEE ICIP 2002, pp. 900–903 (2002)

8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2001, CVPR 2001, vol. 1, pp. I–511. IEEE (2001)
9. Viola, P., Jones, M.: Robust real-time object detection. Int. J. Comput. Vis. **57**(2), 137–154 (2001)
10. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
11. Otsu, N.: A threshold selection method from grey level histograms. IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979). ISSN 1083-4419
12. Galda, H.: IPD - image processing design toolbox version 2.0, Scilab Toolbox (2009)
13. Steinebach, M., Liu, H., Yannikos, Y.: ForBild: Efficient robust image hashing. In: Media Watermarking, Security, and Forensics 2012, SPIE, Burlingame, California, United States (2012). ISBN,978-0-8194-8950-02012