https://doi.org/10.2352/EI.2024.36.4.MWSF-340

Efficient Hash Lookup for PhotoDNA

Martin Steinebach; Fraunhofer SIT|ATHENE: Darmstadt, Germany

Abstract

PhotoDNA is a widely used method for generating robust image hashes. It is widely used today for the detection of CSAM. This results in large numbers of images that need to be compared. This is done over a Euclidean distance, which requires relatively expensive computations. We present an approach that allows the comparison of these images to be performed significantly more efficiently. We also show that both robustness and resistance to false positives are not compromised. Our approach is based on converting the PhotoDNA hash from 144 bytes to 300 bits, which can be compared using Hamming distance. An advantage is that the existing hashes can be converted directly, so no new calculation of hashes from reference images is necessary.

Motivation

Robust (also called perceptual) hashing plays a crucial role in the ongoing effort to combat Child Sexual Abuse Material (CSAM) by facilitating the identification and removal of such content from online platforms. CSAM encompasses various forms of visual or digital media, including images, videos, or computer-generated content, depicting the sexual abuse or exploitation of children.

The process of robust hashing involves creating a distinct digital 'fingerprint', or hash, for an image or video. This hash serves as a unique identifier, enabling the comparison of content with known instances of CSAM.

By employing robust hashing, online platforms gain the ability to proactively detect and eliminate CSAM without solely relying on user reports. This proactive approach contributes significantly to reducing the spread of such content.

Furthermore, robust hashing proves instrumental in assisting law enforcement in forensics investigations. Its is used to search large quantities of images for relevant hits. This speeds up the analysis and makes it possible to handle the large amounts of data that frequently occur.

Efficiency

Due to the high prevalence of the method, one property is of particular interest: The hash comparison is done over a Euclidean distance: two hashes are compared by forming 144 differences and then squaring them. The root is then taken from the sum of these resulting values. This is the distance between the two hash values. For two points P and Q in an n-dimensional space the following formula is used:

$$d = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2} \qquad (1)$$

This calculation is comparatively complex for a hash. Often the much easier to compute Hamming distances are used, where differences in binary sequences are simply counted. The Hamming distance is a measure of the difference between two strings of the same length. It is calculated by counting the number of positions at which the corresponding symbols are different. If you have two strings A and B of the same length, the Hamming distance can be defined as follows:

$$H(A,B) = \sum_{i=1}^{n} [a_i \neq b_i] \qquad (2)$$

In this work, we want to consider whether existing hashes can be converted to a binary representation and compared by Hamming distance without degrading the recognition rates of the method. This could produce a much more resource-efficient solution.

The PhotoDNA Algorithm

The full mechanisms of PhotoDNA have not been disclosed beyond some basic papers by the creators [7] and a presentation by Microsoft. Nevertheless, there have been some attempts to recreate the algorithms from the known facts[1]. PhotoDNA is included in forensic tool sets.

From the available information, we assume the following algorithm:

1. Normalization: Convert to grayscale and downscale to 26x26 pixels. Note: Both operations can affect the hash result due to their handling of edges and textures, so reimplementations may produce values different from the leaked library.
2. Segmentation: The 26x26 pixels are divided into 6x6 quadrants with an overlap of 2 pixels. There are 6 quadrants per row, starting at 1, 5, 9, 13, 17, and 21. There are 36 quadrants
3. Gradients: Sobel gradients are computed for each quadrant. This results in four values representing horizontal and vertical positive and negative sums. The range of values is 0 to 255. In some papers it is mentioned that the value 255 means "255 or more".
4. comparison: There are several ways to compare two hashes. The most common seems to be the Euclidean distance [8]. As far as we know, there are no official thresholds for deciding whether two images are identical or not. The choice of threshold will control the likelihood of false positives or false negatives [19].

In table 1 we provide an example of a PhotoDNA hash as a sequence of 144 byte values derived from the image in figure 1. The values are not structured with respect to gradiant directions.

[1] https://www.hackerfactor.com/blog/index.php?archives/931-PhotoDNA-and-Limitations.html

Figure 1. *Example image from the coco [12] dataset.*

Tab. 1: Hash of image figure 1

	1	2	3	4	5	6	7	8	9	10	11	12
0	0	36	0	42	0	56	0	39	7	33	0	37
12	0	118	49	245	193	255	16	2	181	255	26	0
24	0	32	0	43	0	52	0	44	6	36	0	27
36	255	2	75	125	255	158	132	235	10	255	164	255
48	0	45	0	40	0	44	0	52	3	12	0	36
60	166	3	13	144	224	203	9	255	18	136	0	150
72	0	49	0	44	0	38	0	45	4	7	0	32
84	12	0	0	39	25	6	0	86	42	2	0	43
96	0	41	0	36	2	20	0	26	3	3	0	16
108	0	0	0	0	0	0	0	0	0	0	0	0
120	0	26	1	5	1	12	0	2	1	1	0	5
132	0	0	0	3	1	0	0	4	13	0	0	13

State of the Art

Hash-based algorithms are used in various application areas, such as image search, duplicate or near-duplicate detection, or image authentication. [5] [14] [6] [24]. In this paper, we assume that the difference between cryptographic and robust hashing is known. Briefly, cryptographic hashing is not robust and will generate hashes with no similarity between versions of an image after lossy compression or scaling. Many robust hashing algorithms use perceptual features of images [31, 30, 29, 28]. With advances in deep learning, neural network-based approaches have also been explored for robust hashing. These approaches use deep neural networks to learn feature representations that capture image content and generate compact hash codes for similarity comparison. [3] [17] [2].

Security vs. Robustness

It is often overlooked that content recognition methods are often not designed to be secure. The task of robust hashing methods and classifiers is to recognize or classify content. It is not assumed that an attacker will directly target the methods to prevent this recognition. In the field of multimedia security, a distinction is made between robustness and security. Robustness addresses changes to content that are caused by processing that is normally expected, such as scaling or lossy compression [11] [32]. Robust hashing methods are resistant to this, and classifiers should not exhibit any serious drops in performance here either.

Security, on the other hand, means that an attacker deliberately targeting the algorithms[10] [4] [15] . For example, robust hashing methods can be used to make local changes to the image that cause the hash to change significantly, even though the image itself is not or only slightly disturbed.

This also applies to modern hashing methods based on machine learning, such as NeuralHash from Apple[27]. These attacks can potentially be carried out in both directions: The hash of an image is changed so that it is no longer recognized. Or the hash of another image is changed in such a way that it is mistakenly considered to be stored in a database.

A	B	A	B	A	B	A	B	A	B	A	B
C	D	C	D	C	D	C	D	C	D	C	D
A	B	A	B	A	B	A	B	A	B	A	B
C	D	C	D	C	D	C	D	C	D	C	D
A	B	A	B	A	B	A	B	A	B	A	B
C	D	C	D	C	D	C	D	C	D	C	D
A	B	A	B	A	B	A	B	A	B	A	B
C	D	C	D	C	D	C	D	C	D	C	D
A	B	A	B	A	B	A	B	A	B	A	B
C	D	C	D	C	D	C	D	C	D	C	D
A	B	A	B	A	B	A	B	A	B	A	B
C	D	C	D	C	D	C	D	C	D	C	D

Figure 2. *Structure of PhotoDNA values. 6x6 cells with four values A,B,C,D.*

Attacks on PhotoDNA

In [16], preimage attacks on PhotoDNA (as well as facebook PDQ) are shown. By accepting a certain amount of noise, it is possible to generate image pairs with matching hashes. In [13] the privacy of the hashes was verified. It was argued that it is not possible to derive the original images from their hashes using machine learning. Recent experiments, however, show that images can be reconstructed from the hashes[2]. The question is whether the re-created images rely more on the hashes or on the training data of the re-creation system.

Own Previous Work

An alternative robust hash of ours is the ForBild block hash presented in [18] [23]. It is the result of an evaluation of image hashing methods [32]. Based on this hash, we have added segmentation countermeasures based on face detection [25], watershed image segmentation [24] and machine learning[20][21]. Beyond image recognition, we also addressed the possibility of combining privacy and robust hashing in [1] [9] [26]. As an alternative to robust hashing, we also evaluated feature-based montage detection using SIFT and SURF in [22].

Binarization of PhotoDNA

The original PhotoDNA hash consists of 144 byte values. These are the result of 6x6 overlapping squares, for each of which the edge intensity is calculated in four directions (vertical positive, vertical negative, horizontal positive, horizontal negative). This results in 6*6*4 values ranging from 0 to 255. Figure 2 illustrates the structure of the original PhotoDNA hash, with A to D representing the four gradient directions.

We suggest a simple conversion to a binary representation: We compare values of individual directions in adjacent cells. One bit of our hash results form the output of the comparison between

[2] https://www.anishathalye.com/2021/12/20/inverting-photodna/

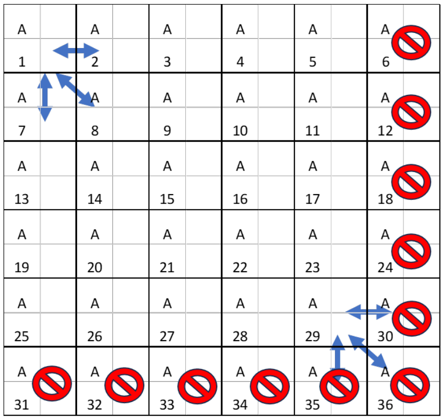

Figure 3. *Illustration of value comparison*

two values x and y, e.g. A1 and A2.

$$\text{HashBit} = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{else} \end{cases} \tag{3}$$

We perform these comparisons for neighboring cells in horizontal, vertical and diagonal direction. Since there are 6*6 cells, but we always need a direct neighbor, the comparison is reduced to 5*5, i.e. we get 3*5*5=75 bits.

Figure 3 shows the selection of comparison values for direction A: We compare neighbors in three directions shown by the arrows. The marked blocks at the right column and bottom row are not used as starting points. The last starting point is A29, comparing with A30, A35 and A36.

We do the same for directions B to D. This means that we get 300 bits as a hash from the comparison. These 300 bits are our new binary representation of the PhotoDNA hash. Now that we have a binary representation, we can efficiently calculate the Hamming distance between two hashes.

Evaluation

The test material is 10,000 images from the Coco dataset [12], which contain a wide collection of different motifs. The longest edge of each of these images is 640 pixels. In addition, we used 99.999 different images from Coco for creating a hash data base. This data base is used for evaluating the collision resistance.

Robustness

Since PhotoDNA is a robust hash, the first thing we did was to investigate the effect of robustness on the hash after converting it into a binary sequence. The following attacks were applied by Irfanview[3]:

- 80: convert=jpg80
- 70: convert=jpg70
- 30: convert=jpg30
- c2: crop=(2,2,1000,1000) and jpgq=80
- c10: crop=(10,10,1000,1000) and jpgq=80
- 300: resize_long=300 and jpgq=80
- 500: resize_long=500 and jpgq=80

Cropping is done by setting the starting point of the crop area to the top left. Crop(2,2,....) means that the image with a maximum length of 640 pixels has been cropped at position (2,2) by removing one pixel row and column from the top left. The crop boundary (.....,1000,1000) can be seen as an neutral setting as it is beyond the size of the image.

Our first investigations show that the robustness of the hash does not suffer by converting it to a binary representation. The following two tables first show the percentage mapping to bins of Hamming distances in steps of five. The attacks considered are JPEG compression with quality factor 80, 70 and 30 (80,70,30), cropping at the upper left corner by 2 and by 10 pixels (C2 and C10), and scaling to 300 and 500 pixel page length (R300, R500).

While the original calculation has a potentially huge result space, the binary hash is limited to a maximum Hamming distance of 300 due to its 300 bit length. We have limited our investigation to 25% of this maximum distance, so we only show the range between 0 and 74 as well as all higher valuescombined. For the Euclidean distance, we consider the range from 0 to 425 as well as all higher values in steps of 25.

It can be clearly seen that the robustness of both comparison methods is similar. In both cases, only cropping C10 by 10 pixels is a challenge to robustness. The other attacks show results that are all very low distances.

Tab. 2: Robustness of original hash comparison (in percent)

Range		80	70	30	C2	C10	R300	R500
0	25	22.75	22.75	22.22	7.27	0.00	17.02	20.57
26	50	36.77	36.76	37.11	45.31	0.05	39.93	38.04
51	75	21.82	21.85	21.89	26.43	2.02	23.46	22.40
76	100	10.58	10.57	10.63	12.07	23.98	11.17	10.82
101	125	4.57	4.56	4.60	5.08	39.18	4.76	4.61
126	150	1.84	1.85	1.85	2.06	21.48	1.91	1.85
151	175	0.85	0.86	0.88	0.91	8.24	0.89	0.88
176	200	0.40	0.39	0.41	0.45	2.95	0.44	0.41
201	225	0.24	0.24	0.22	0.23	1.14	0.23	0.23
226	250	0.08	0.07	0.08	0.08	0.50	0.08	0.08
251	275	0.05	0.05	0.05	0.05	0.21	0.04	0.05
276	300	0.03	0.03	0.03	0.03	0.10	0.03	0.03
301	325	0.02	0.02	0.02	0.02	0.06	0.02	0.02
326	350	0.01	0.01	0.01	0.01	0.04	0.01	0.01
351	375	0.00	0.00	0.00	0.00	0.01	0.00	0.00
376	400	0.00	0.00	0.00	0.00	0.02	0.00	0.00
401	425	0.01	0.00	0.01	0.00	0.01	0.01	0.01
426	99999	0.00	0.00	0.00	0.00	0.01	0.00	0.00

Collision resistance

Results based on comparison of hash values from randomly selected images show a very low false error rate that is comparable to the original PhotoDNA results discussed in our previous paper [19]. Comparing each of the 10,000 image hashes with the other 99,999 hashes show an average minimal Hamming distance of 144, which is close to the coin flip average of 150 for 300 bits. Figure 4 and table 4 show the results for all 10,000 images. No hash was below the hamming distance threshold of 75. The observed minimal distance is 83. Figure 5 provides a sorted view on all 10,000 minimal distances.

Discussion

The innovation of our work lies in demonstrating that existing and widely used methods such as PhotoDNA can be made

[3] https://www.irfanview.com/, version 4.60

Tab. 3: Robustness of binary hash comparison (in percent)

Range		80	70	30	C2	C10	R300	R500
0	4	34.88	34.38	32.29	17.58	0.02	25.69	31.22
5	9	39.12	39.56	39.73	46.81	1.83	42.78	40.46
10	14	16.87	16.76	17.77	23.6	13.47	20.7	18.52
15	19	5.85	5.96	6.67	8.05	31.07	7.04	6.26
20	24	2.1	2.17	2.26	2.49	30.13	2.35	2.25
25	29	0.73	0.68	0.74	0.9	15.5	0.89	0.78
30	34	0.28	0.32	0.32	0.33	5.63	0.37	0.33
35	39	0.08	0.08	0.09	0.14	1.62	0.08	0.07
40	44	0.03	0.02	0.04	0.04	0.44	0.04	0.05
45	49	0.02	0.02	0.02	0.02	0.13	0.01	0.02
50	54	0.02	0.02	0.03	0.02	0.07	0.01	0.01
55	59	0.01	0.01	0.01	0.01	0.04	0.03	0.01
60	64	0	0	0.01	0	0.02	0	0.01
65	69	0	0.01	0.01	0	0.01	0	0
70	74	0	0	0	0	0	0	0
75	300	0.01	0.01	0.01	0.01	0.02	0.01	0.01
401	425	0,01	0,00	0,01	0,00	0,01	0,01	0,01
426	99999	0,00	0,00	0,00	0,00	0,01	0,00	0,00

Tab. 4: Minimal Hamming Distances distribution

Range		Percent	Range		Percent
150	154	36.04	110	114	1.31
145	149	12.5	105	109	0.59
140	144	13.58	100	104	0.25
135	139	11.65	95	99	0.15
130	134	9.33	90	94	0.05
125	129	7.22	85	89	0.02
120	124	4.64	80	84	0.01
115	119	2.66	75	79	0

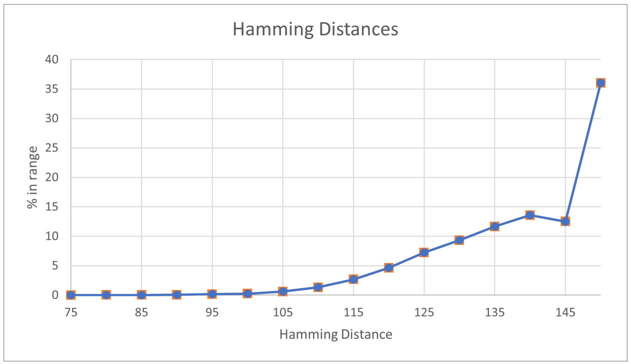

Figure 4. *Minimal Hamming Distance distribution*

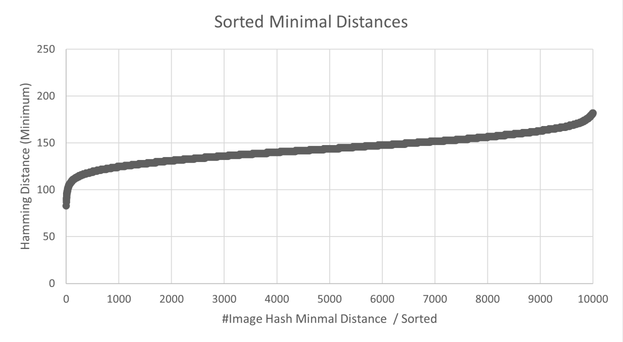

Figure 5. *Sorted minimal distances*

much more efficient using simple approaches without reducing their performance. Our approach should allow a significantly faster hash lookup and requires only 38 bytes of memory instead of 144 bytes. Existing hash collections can be easily converted, there is no need to recompute the hash. Especially when large data collections have to be stored and compared, our approach has a massive advantage over the established method.

Our evaluation with 10,000 images shows that with a threshold of Hamming Distance 75 (25% of the maximum distance) we achieve zero false positives and 0.01% false negatives.

In conclusion, it is imperative to underscore the significance of enhancing the efficiency of perceptual hashing algorithms. The primary justification for this lies in the ability of optimized algorithms to expedite the processing of extensive data volumes, thereby diminishing the temporal and computational resources requisites. Such an enhancement is particularly pivotal in scenarios necessitating real-time or near-real-time processing, exemplified by content moderation on digital social platforms or in video surveillance frameworks.

A reduction in computational demands, coupled with a decrease in requisite data transmission, unequivocally leads to a lower overall energy consumption. This aspect is of paramount importance in the context of sustainable computing methodologies, especially within the ambit of large-scale systems and data centers. Furthermore, an efficient perceptual hashing system is inherently more capable of adapting to scalability demands. This adaptability is critical in an era where digital content is proliferating at an exponential rate, necessitating efficient processing and comparison mechanisms.

Moreover, the reduction of the perceptual hash length significantly minimizes the volume of data necessitating storage. This attribute is particularly beneficial in database systems where substantial quantities of images or videos are archived for subsequent comparison or retrieval.

Finally, it is essential to acknowledge that a reduction in energy consumption transcends mere cost savings; it also substantially lessens the environmental footprint. This consideration is increasingly crucial in contemporary discourse, where the focus on sustainability is paramount. Therefore, advancing the efficiency of perceptual hashing algorithms not only presents technical and economic benefits but also aligns with broader ecological objectives.

Acknowledgments

This research work has been funded by BMBF and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

[1] Uwe Breidenbach, Martin Steinebach, and Huajian Liu. Privacy-enhanced robust image hashing with bloom filters. In Melanie Volkamer and Christian Wressnegger, editors, *ARES 2020: The 15th International Conference on Availability, Reliability and Security, Virtual Event, Ireland, August 25-28, 2020*, pages 56:1–56:10. ACM, 2020.

[2] Olena Buchko et al. Classification of confidential images using neural hash. *NaUKMA Research Papers Computer Science*, 5:68–71, 2022.

[3] Veena Desai and DH Rao. Image hash using neural networks. *International Journal of Computer Applications*, 63(22), 2013.

[4] Brian Dolhansky and Cristian Canton Ferrer. Adversarial collision attacks on image hashing functions. *arXiv preprint arXiv:2011.09473*, 2020.

[5] Andrea Drmic, Marin Silic, Goran Delac, Klemo Vladimir, and Adrian S. Kurdija. Evaluating robustness of perceptual image hashing algorithms. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 995–1000. IEEE, 2017.

[6] Ling Du, Anthony T.S. Ho, and Runmin Cong. Perceptual hashing for image authentication: A survey. *Signal Processing: Image Communication*, 81:115713, 2020.

[7] Hany Farid. Reining in online abuses. *Technology & Innovation*, 19(3):593–599, 2018.

[8] Hany Farid. An overview of perceptual hashing. *Journal of Online Trust and Safety*, 1(1), 2021.

[9] Marius Leon Hammann, Martin Steinebach, Huajian Liu, and Niklas Bunzel. Predicting positions of flipped bits in robust image hashes. *Electronic Imaging*, 35:375–1, 2023.

[10] Qingying Hao, Licheng Luo, Steve TK Jan, and Gang Wang. It's not what it looks like: Manipulating perceptual hashing based applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 69–85, 2021.

[11] Shubham Jain, Ana-Maria Crețu, and Yves-Alexandre de Montjoye. Adversarial detection avoidance attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2317–2334, 2022.

[12] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[13] Muhammad Shahroz Nadeem, Virginia N. L. Franqueira, and Xiaojun Zhai. Privacy verification of photodna based on machine learning. In *Security and Privacy for Big Data, Cloud Computing and Applications*, pages 263–280. IET, August 2019. I've uploaded the final proof of the chapter.

[14] Dat Tien Nguyen, Firoj Alam, Ferda Ofli, and Muhammad Imran. Automatic image filtering on social networks using deep learning and perceptual hashing during crises.

[15] Jonathan Prokos, Neil Fendley, Matthew Green, Roei Schuster, Eran Tromer, Tushar Jois, and Yinzhi Cao. Squint hard enough: attacking perceptual hashing with adversarial machine learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 211–228, 2023.

[16] Jonathan Prokos, Tushar M. Jois, Neil Fendley, Roei Schuster, Matthew Green, Eran Tromer, and Yinzhi Cao. Squint hard enough: Evaluating perceptual hashing with machine learning. Cryptology ePrint Archive, Paper 2021/1531, 2021.

[17] Chuan Qin, Enli Liu, Guorui Feng, and Xinpeng Zhang. Perceptual image hashing for content authentication based on convolutional neural network with multiple constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(11):4523–4537, 2020.

[18] Martin Steinebach. Robust hashing for efficient forensic analysis of image sets. In Pavel Gladyshev and Marcus K. Rogers, editors, *Digital Forensics and Cyber Crime*, volume 88 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 180–187. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[19] Martin Steinebach. An analysis of photodna. In *Proceedings of the 18th International Conference on Availability, Reliability and Security, ARES 2023, Benevento, Italy, 29 August 2023- 1 September 2023*, pages 44:1–44:8. ACM, 2023.

[20] Martin Steinebach, Tiberius Berwanger, and Huajian Liu. Towards image hashing robust against cropping and rotation. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–7, 2022.

[21] Martin Steinebach, Tiberius Berwanger, and Huajian Liu. Image hashing robust against cropping and rotation. *Journal of Cyber Security and Mobility*, pages 129–160, 2023.

[22] Martin Steinebach, Karol Gotkowski, and Hujian Liu. Fake news detection by image montage recognition. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–9, New York, NY, USA, 2019. ACM.

[23] Martin Steinebach, Huajian Liu, and York Yannikos. Forbild: Efficient robust image hashing. In *Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 195–202. SPIE, 2012.

[24] Martin Steinebach, Huajian Liu, and York Yannikos. Efficient cropping-resistant robust image hashing. In *2014 Ninth International Conference on Availability, Reliability and Security*, pages 579–585. IEEE, 2014.

[25] Martin Steinebach, Huajian Liu, and York Yannikos. Facehash: Face detection and robust hashing. In Pavel Gladyshev, Andrew Marrington, and Ibrahim Baggili, editors, *Digital Forensics and Cyber Crime*, volume 132 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 102–115. Springer International Publishing, Cham, 2014.

[26] Martin Steinebach, Sebastian Lutz, and Huajian Liu. Privacy and robust hashes: Privacy-preserving forensics for image reidentification. *Journal of Cyber Security and Mobility*, pages 111–140, 2020.

[27] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. Learning to break deep perceptual hashing: The use case neuralhash. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 58–69, 2022.

[28] Rui Sun and Wenjun Zeng. Secure and robust image hashing via compressive sensing. *Multimedia Tools and Applications*, 70, 06 2012.

[29] Zhenjun Tang, Lv Chen, Xianquan Zhang, and Shichao Zhang. Robust image hashing with tensor decomposition. *IEEE Transactions on Knowledge and Data Engineering*, 31(3):549–560, 2019.

[30] Zhenjun Tang, Fan Yang, Liyan Huang, and Xianquan Zhang. Robust image hashing with dominant dct coefficients. *Optik*, 125(18):5102–5107, 2014.

[31] Zhenjun Tang, Xianquan Zhang, Xuan Dai, Jianzhong Yang, and Tianxiu Wu. Robust image hash function using local color features. *AEU - International Journal of Electronics and Communications*, 67(8):717–722, 2013.

[32] Christoph Zauner, Martin Steinebach, and Eckehard Hermann. Rihamark: perceptual image hash benchmarking. In Nasir D. Memon, Jana Dittmann, Adnan M. Alattar, and Edward J. Delp III, editors, *Media Watermarking, Security, and Forensics III*, SPIE Proceedings, page 78800X. SPIE, 2011.

Author Biography

Martin Steinebach is the manager of the Media Security and IT Forensics division at Fraunhofer SIT. From 2003 to 2007 he managed the

Media Security in IT division at Fraunhofer IPSI. He studied computer science at the Technical University of Darmstadt and finished his diploma thesis on copyright protection for digital audio in 1999. In 2003 he received his PhD at the Technical University of Darmstadt for this work on digital audio watermarking. In 2016 he became honorary professor at the TU Darmstadt. He gives lectures on Multimedia Security as well as Civil Security. He is Principle Investigator at ATHENE and represents IT Forensics and AI security. Before he was Principle Investigator at CASED with the topics Multimedia Security and IT Forensics.