Radon Transform-Based Secure Image Hashing

Dung Nguyen, Li Weng, Bart Preneel

Radon Transform-Based Secure Image Hashing

Dung Q. Nguyen, Li Weng, and Bart Preneel*

Katholieke Universiteit Leuven, ESAT/COSIC-IBBT
dungtobe@gmail.com,
li.weng@esat.kuleuven.be,bart.preneel@esat.kuleuven.be

Abstract. This paper presents a robust and secure image hash algorithm. The algorithm extracts robust image features in the Radon transform domain. A randomization mechanism is designed to achieve good discrimination and security. The hash value is dependent on a secret key. We evaluate the performance of the proposed algorithm and compare the results with those of one existing Radon transform-based algorithm. We show that the proposed algorithm has good robustness against content-preserving distortion. It withstands JPEG compression, filtering, noise addition as well as moderate geometrical distortions. Additionally, we achieve improved performance in terms of discrimination, sensitivity to malicious tampering and receiver operating characteristics. We also analyze the security of the proposed algorithm using differential entropy and confusion/diffusion capabilities. Simulation shows that the proposed algorithm well satisfies these metrics.

1 Introduction

In order to efficiently identify digital images, perceptual hash techniques have been used [1–3]. A hash value, typically a short binary string, is generated to act as a unique identifier of the corresponding image. Since an image can be stored under different digital representations, a *perceptual* hash value is expected to be resilient to content-preserving manipulations, such as JPEG compression, filtering, etc. Additionally, perceptual hash algorithms are also useful for secure applications, e.g., image content authentication. Currently, many effective signal processing tools are available to modify image content. Therefore, an image hash algorithm is also required to make the hash output dependent on a secret key [2]. Only the entity knowing the key can generate the corret hash value. It helps to ensure that image information is not tampered with during transmission.

* This work was supported in part by the Concerted Research Action (GOA) AM-BioRICS 2005/11 of the Flemish Government and by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy). The second author was supported by the IBBT/AQUA project. IBBT (Interdisciplinary Institute for Broad-Band Technology) is a research institute founded in 2004 by the Flemish Government, and the involved companies and institutions (Philips, IPGlobalnet, Vitalsys, Landsbond onafhankelijke ziekenfondsen, UZ-Gent). Additional support was provided by the FWO (Fonds Wetenschappelijk Onderzoek) within the project G.0206.08 Perceptual Hashing and Semi-fragile Watermarking.

The performance of a perceptual image hash algorithm primarily consists of robustness, discrimination, and security. Robustness means the algorithm always generates the same (or similar) hash values for similar image contents. Discrimination means different image inputs must result in independent (different) hash values. The security of a perceptual image hash algorithm has two aspects. The first is the ability to detect malicious tampering. Another aspect is the difficulty of deriving a hash value without knowing the key.

In this paper, an image hash algorithm is proposed. It exploits the invariance of the Radon transform to rotation and scaling. Our work is inspired by the RAdon Soft Hash algorithm ($RASH$) [4]. This algorithm has good robustness, but its discrimination capability is worth improving. Moreover, it does not incorporate a secret key. In the proposed new algorithm, we strengthen the capability of the original algorithm by improving its discrimination and security properties. A special randomization scheme is introduced to maintain the robustness, and meanwhile improve the overall performance.

The rest of this paper is structured as follows: Section 2 presents the Radon transform properties and the proposed algorithm in detail. Section 3 provides performance evaluation of the proposed algorithm in comparision with the RASH algorithm. Section 4 discusses security of the proposed algorithm. Section 5 concludes the work.

2 The Proposed Algorithm

The proposed algorithm aims to be robust against content-preserving manipulations. It is also expected to improve discrimination in comparison with the RASH algorithm.

2.1 Radon Transform and Properties

The Radon transform is computed by taking the line integrals of a two-dimensional image $f(x, y)$ along a set of directions. The line integral along a particular direction θ is called a projection. The line integral of the function $f(x, y)$ along the line L defined by the direction θ and the distance x' from the origin in the coordinates (x', y') [4] is given by

$$R_\theta(x') = \int_L f(x'cos\theta - y'sin\theta, x'sin\theta + y'cos\theta)dy' \ . \tag{1}$$

The expression (1) leads to two noticeable properties.
Scaling: $f(ax, ay) \leftrightarrow \frac{1}{a}R_\theta(ax')\ ,\ \ where\ \ a > 0$
The Radon transform of a scaled image $f(ax, ay)$ is proportional to the Radon transform of the image $f(x, y)$ with the same scaling factor a, or $\frac{R_\theta(ax')}{R_\theta(x')} = a$.
Rotation: if an image $f(x, y)$ is rotated by ω degrees, the Radon transform of the rotated image is

$$f(x.cos\omega - y.sin\omega, x.sin\omega + y.cos\omega) \leftrightarrow R_{\theta+\omega}(x')\ , \tag{2}$$

i.e., it can be obtained by circularly shifting the transform coefficients of the image $f(x, y)$ according to ω.

2.2 Proposed Algorithm

The proposed algorithm has a hash generation part and a hash verification part. The former consists of three stages: image preprocessing, feature extraction and quantization. In the latter, the hash values may first undergo a rotation detection; then the normalized Hamming distance is computed to measure their similarity. The resultant distance is compared with a threshold to decide if the two values correspond to the same content.

The rotation detection stage enhances robustness against rotation. However, it also decreases overall discrimination. Therefore, we come up with two schemes for practice. *Scheme 1* involves the rotation detection stage. *Scheme 2* skips the rotation detection stage.

A. Hash generation

Stage 1: Image Preprocessing. The input image $I(x, y)$ is first converted to gray and down-sampled to the canonical size 512×512 pixels. Next it is smoothed by a low-pass filter. Histogram equalization is applied to the filtered image.

Stage 2: Feature Extraction. We introduce a new approach to extract invariant image features in the Radon transform domain. In detail, this stage includes two steps:

a. Radon transform. We apply the Radon transform to the preprocessed image for the projection angles $\theta = 0, 1, ..., 179$ to obtain a set of projections $\{R_\theta(x'_i)\}$. The projection along each angle θ is a vector of line integrals along the lines L_i (*projection paths*) defined by the distance x'_i to the origin.

b. Feature randomization. We next calculate a weighted sum of selected projection paths along each angle θ. An intermediate hash vector of 180 elements is obtained.

$$h_\theta = \sum_{i=1}^{N_p} \alpha_i R_\theta(x'_i), \ \theta = 0, 1, ..., 179 \tag{3}$$

where N_p is the number of selected projection paths; $\{\alpha_i\}$ are normally distributed pseudorandom numbers with mean m and variance σ^2.

Stage 3: Quantization. We uniformly quantize the 180-element intermediate vector to generate a 360-bit hash value.

B. Hash verification

The input hash values first go through rotation detection. This stage is only applied in Scheme 1. The hash value of a possibly rotated image (h_2) is compared with that of the original image (h_1) to estimate the rotation angle by means of maximum cross-covariance

$$R_{h_1,h_2}(m) = \sum_{n=0}^{N-m-1} \left(h_1(n) - \overline{h_1}\right)\left(h_2(n+m) - \overline{h_2}\right) \tag{4}$$

where $\overline{h_1} = \frac{1}{N}\sum_{i=0}^{N-1} h_1(i)$, $\overline{h_2} = \frac{1}{N}\sum_{i=0}^{N-1} h_2(i)$ are the means of the hash values h_1, h_2 respectively; $N = 360$ is the hash length; and $m = 0, 1, ..., 359$. The rotation angle is determined by

$$\varphi = 360 - \underset{m}{argMax}(R_{h_1,h_2}(m)) \ . \tag{5}$$

After the hash values are aligned by φ, their normalized Hamming distance (NHD) is computed as

$$d_{h_1,h_2} = \frac{1}{N}\sum_{i=1}^{N} |h_1(i) - h_2(i)| \, . \tag{6}$$

3 Performance Evaluation

The proposed algorithm is evaluated on a database of 618 different natural scene images. The types of images include architecture, sculpture, humanoid, landscape, food, and vehicle. The image sizes vary from 640×480 to approximately 3000×2000.

In the feature extraction stage, the number of selected projection paths is set as 5 and the distance between them is set as 50. The pseudorandom numbers are normally distributed with the mean 1 and standard deviation 2, controled by a secret key. The robustness of RASH and the proposed algorithm is verified under

Table 1. Set of manipulations

Type of manipulation	Manipulation parameter
Legitimate manipulations	
Gaussian filtering	Filter size: 11×11, 21×21
Median filtering	Filter size: 3×3, 5×5
JPEG compression	Quality factor: 20, 10
Gaussian noise	Standard deviation: 0.04, 0.08
Salt and Pepper noise	Noise density: 0.04, 0.08
Rotation	Angle: 2, 4 degrees
Cropping	Percentage: 2%, 4%
Malicious manipulation	Object size after preprocessing:
Object insertion	32×32, 64×64

various manipulations listed in Table 1. We apply legitimate manipulations to each original image and generate 14 perceptually similar images. The hash value of each manipulated image is computed and compared with that of the original image. The NHD between two hash values is expected to be close to zero. The algorithms are also tested under object insertion to measure their sensitivity to malicicious local modifications. In total 16 manipulated images are generated for each original image.

Table 2. Normalized Hamming distances for manipulations.

Manipulation	Parameter	Normalized Hamming distance		
		RASH	Scheme 1	Scheme 2
Gaussian filtering	11×11	0.0031	0.0033	0.0029
	21×21	0.0031	0.0032	0.0029
Median filtering	3×3	0.0094	0.0095	0.0086
	5×5	0.0198	0.0195	0.0197
JPEG compression	20	0.0079	0.0085	0.0084
	10	0.014	0.0144	0.0146
Gaussian noise	0.04	0.0531	0.054	0.055
	0.08	0.0818	0.0793	0.0798
Salt&pepper noise	0.04	0.027	0.0273	0.0298
	0.08	0.0446	0.0442	0.0445
Rotation	2°	0.047	0.0427	0.1091
	4°	0.0781	0.0722	0.1762
Cropping	2%	0.0233	0.0456	0.0435
	4%	0.0376	0.0797	0.0788
Object insertion	32×32	0.0149	0.0232	0.0233
	64×64	0.0515	0.0652	0.0661

Table 2 shows the average NHDs for the manipulations on the image database. All the algorithms are strongly robust to Gaussian filtering with the NHDs on the order of 10^{-3}. They also perform well under Median filtering and JPEG compression with the NHDs on the order of 10^{-2}. For Gaussian noise as well as salt and pepper noise, the NHDs of the algorithms are smaller than 0.1 and comparable to each other. This is partially due to the low-pass filtering in the preprocessing stage.

For rotation, the NHDs of Scheme 1 are smaller than those of RASH. This is attributed to the rotation detection. On the other hand, Scheme 2 without the rotation detection is less robust to rotation angles larger than 3°. For cropping, the proposed methods show lower performance than RASH. This is because after the cropped image is re-scaled to the canonical size, the selected projection paths at the same distances to the origin have changed; while in the RASH algorithm the medium projection path remains unchanged.

Regarding object insertion, the proposed methods have higher NHDs than the RASH algorithm. Therefore, the proposed algorithm has stronger sensitivity to malicious modifications, meaning better ability for content authentication. This is because in the proposed algorithm, the selected projection paths cover a larger area of the image – an advantage of the proposed algorithm. While in the RASH algorithm only the medium path is selected. In our experiment, the NHD for the proposed methods is higher than 0.1 when the object size is larger than 96×96 pixels.

In order to test the discrimination capability, each original image and its maniplulated images are put into a group. Hash values from different groups are pair-wise compared. There are $\binom{618}{2} \times 17^2 = 55098717$ hash pairs in the test. The

resultant NHD is expected to be close to 0.5, because hash values of different image contents are independent.

The average NHDs between different image contents for the RASH, the proposed scheme 1 and scheme 2 are 0.317, 0.329 and 0.477 respectively. The proposed methods show better discrimination than the RASH algorithm. This is due to the randomization procedure. Scheme 2 has better discrimination than Scheme 1. This is because in Scheme 1 the rotation detection reduces the randomness achieved in the feature extraction stage. In Scheme 2 the NHD is computed directly from two random hash values. Hence there is a tradeoff between robustness and discrimination in the proposed methods.

We use the receiver operating characteristics (ROC) to compare the overall performance. The ROC curves (in enlarged scale) are shown in Fig. 1. Given a false positive rate (P_f), the proposed methods have higher probability of correct detection (P_d) than the RASH algorithm. Hence the proposed methods achieve better overall performance. It is also observed that Scheme 2 achieves the best ROC curve.

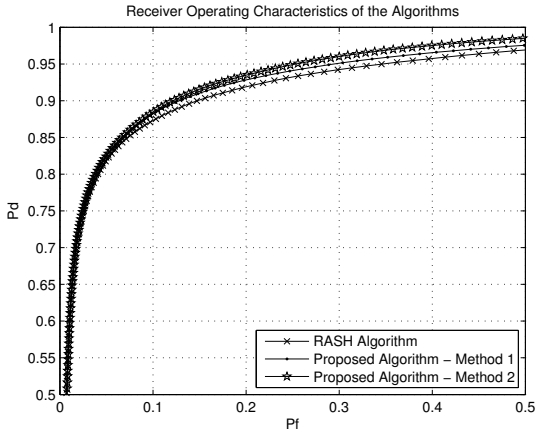

Fig. 1. Receiver operating characteristics of the algorithms.

4 Security Analysis

The security of image hashing is still an open area. A well-known security metric is the differential entropy of the hash value, proposed by Swaminathan *et al.* [2]. It measures the effort of an attacker to estimate the hash value without knowing the secret key. Larger entropy means increasing amount of randomness. In our algorithm, the differential entropy of the hash value increases when the variance of normally distributed pseudorandom numbers becomes larger or when the number of sample points is larger.

Following the approach in [2], we derive the differential entropy expression of each hash element for the proposed algorithm

$$H(h_\theta) = \frac{1}{2} log_2 \left((2\pi e)\sigma^2 \sum_{i=1}^{N_p} R_\theta^2(x_i') \right). \tag{7}$$

Table 3 shows the differential entropy of some image hash algorithms (cf. [2]). The differential entropy of the proposed algorithm is in the range $13.89 - 14.92$. It is quite stable, compared with those of Swaminathan's Scheme-1 and Venkatesan's algorithm. It is greater than Fridrich's, Venkatesan's and Mihcak's algorithms, and smaller than Swaminathan's Scheme-2.

Table 3. Differential entropy of different hash algorithms [2].

Hash algorithm	Differential Entropy		
	Lena	Baboon	Peppers
Proposed algorithm	14.57 - 14.76	13.89 - 14.19	14.72 - 14.92
Swaminathan's scheme-1 [2]	8.2 - 15.6	13.58 - 16.18	8.76 - 15.46
Swaminathan's scheme-2 [2]	16.28	16.39	16.18
Fridrich's algorithm [5]	8.31	8.32	8.14
Venkatesan's algorithm [1]	5.74 - 11.48	5.96 - 11.70	5.65 - 11.39
Mihcak's algorithm B [3]	8	8	8

Coskun *et al.* [6] defined diffusion and confusion capabilities as security metrics. They measure the difficulty of revealing the relationship between the secret key and the hash value (confusion), and the relationship between the input image and the hash value (diffusion).

A hash algorithm with good confusion generates statistically independent hash values using different keys for the same image input. In our test, 100 hash values are generated for each image using 100 different keys. The average NHD of $\binom{100}{2} = 4950$ hash pairs is computed for some images and shown in Table 4. The proposed scheme 2 has higher NHD than the proposed scheme 1 for all the tested images. This means that Scheme 2 shows better confusion capability. A hash algorithm with good diffusion generates different hash values for different image contents, corresponding to the discriminative capability. The discrimination test before implies that Scheme 2 has better diffusion capability than Scheme 1.

5 Conclusion and Discussion

In this work, we propose a robust and secure image hash algorithm. The algorithm extracts image features in the Radon transform domain. A randomization

Table 4. Confusion capability of two proposed methods.

Proposed algorithm	Lena	Baboon	Boat	Peppers
Proposed scheme 1	0.3382	0.3047	0.3228	0.3350
Proposed scheme 2	0.4575	0.4442	0.4489	0.4214

mechanism is incorporated to make the hash output dependent on a secret key. It is resilient to filtering, JPEG compression, and noise addition. It is also robust to moderate geometrical distortions including rotation and cropping. The proposed algorithm achieves significant improvement to the well-known RASH algorithm. It has better discrimination and higher sensitivity to malicious tampering than RASH, which leads to a better operating characteristic. The key-dependent feature also makes it suitable for a wider range of applications. The security of the algorithm is evaluated in terms of differential entropy and confusion/diffusion capabilities. Good security is confirmed by both metrics.

There is a tradeoff between discrimination and robustness in the proposed methods. Scheme 1 takes advantage of rotation detection to improve its robustness against rotation. However, this decreases its discrimination and subsequently lowers the overall performance. Since Scheme 2 achieves better results in the security evaluation than Scheme 1, there is also a tradeoff between robustness and security.

In the future, we plan to improve the proposed algorithm by detecting several geometric distortions (e.g. scaling and cropping) before computing the hash distance. This will further enhance robustness. More security metrics will be taken into account. It is interesting to evaluate the maximum number of key re-uses, see [7, unicity distance].

References

1. Venkatesan, R., Koon, S., Jakubowski, M., Moulin, P.: Robust image hashing. In: Proceedings of IEEE International Conference on Image Processing Vol.3, pp. 664-666. (2000)
2. Swaminathan, A., Mao, Y., Wu, M.: Robust and secure image hashing. In: IEEE Transactions on Information Forensics and Security, Vol.1, No. 2. (June 2006)
3. Mihcak, K., Venkatesan, R.: New iterative geometric methods for robust perceptual image hashing. In: In Proceedings of the Workshop on Security and Privacy in Digital Rights Management. (2001)
4. Lefebvre, F., Macq, B., Legat, J.: Rash: Radon soft hash algorithm. In: Proceedings of the European Signal Processing Conference, Toulouse, France. (Sep. 2002)
5. Fridrich, J., Goljan, M.: Robust hash functions for digital watermarking. In: Proceedings of the The International Conference on Information Technology: Coding and Computing. (2000)
6. Coskun, B., Memon, N.: Confusion/diffusion capabilities of some robust hash functions. In: Proceedings of 40th Annual Conference on Information Sciences and Systems. (2006)
7. Mao, Y., Wu, M.: Unicity distance of robust image hashing. In: IEEE Transactions on Information Forensics and Security, Vol. 2, No. 3. (Sep. 2007)