www.sensorsportal.com

Sensors & Transducers

© 2013 by IFSA

http://www.sensorsportal.com

A Study and Analysis on a Perceptual Image Hash Algorithm Based on Invariant Moments

Hu Bin

School of Civil Engineering and Transportation, South China University of Technology
No. 381 Wushan Road, Tianhe District, Guangzhou 510640, P. R. China

Received: 7 November 2013 /Accepted: 20 November 2013 /Published: 30 November 2013

Abstract: In this paper, we firstly introduce the perceived perceptual image hash algorithms proposed by scholars in recent years. Then we classify the algorithms according to the different theories. At the same time, we also describe the general process of perceptual image hash algorithm and realize the two classic algorithms in the professional field. After all above, we make use of the invariant moment of image to the invariance of geometric transformation, and propose a new perceptual image of hash algorithm based on invariant moments with robustness and security. Through the test from different aspects, we compare the algorithm with the performance of two kinds of classic algorithms, then we can conclude that the new algorithm has better robustness and security. *Copyright © 2013 IFSA.*

Keywords: Perceptual image, Hash algorithm, Invariant moment.

1. Introduction

Perceptual image hash technology is an image description of image content or characteristics, and the hash function also can make them produce two hash sequences which are same or similar even if the two pictures with similar perception have different representation in value [1]. This is the remarkable difference between perceptual image hash technology and the hash function in traditional cryptography. In the traditional cryptography, hash function (such as MD5 and SHA-1) is used to produce the digital signature relying on the key the then verify the message's integrity [2]. When the authentication message of hash function input terminal happens a change of 1 bit in the content, the output bit strings of hash function output terminal will produce violent dramatic changes, so hash function in cryptography can be only applied to the identification in text data, and multimedia data authentication allow lossy descriptions (such as image compression as JPEG) [3]. At the same time, multimedia information may experience filtering, geometric distortion, noise pollution and other changes in the process of transmission, but the expression of the multimedia data content basically are unchanged, and the data are true and complete [4].

Perceptual image hash technology requires algorithm can resist some common image process operations, which may be said that it has rather strong robustness, and it also requires the digital signature generated by perceptual image hash technology is based on the key, then the same images will produce different hash strings after encrypted by different key [5]. So the algorithm has rather strong security. Here the requirement about the unidirectional and collision resistance of hash function in traditional cryptography can satisfy the perceptual image hash function [6]. At the same time, perceptual image hash technology needs the image produce different hash strings compared with the distorted image [7].

This paper is organized as follows. The first section is an introduction, and it briefly introduces the perceptual image hash technology and describes research background and significance of the technology. Section 2 is about the processes of image hash technology and the existing research condition, and it briefly introduces the perceived perceptual image hash algorithms, then implements two classic algorithms of them. Section 3 puts forward a perceptual image hash algorithm based on invariant moments, and it's the main part of the paper [8]. Then in the section 4, we test the algorithm from four aspects such as robustness, collision resistance, security and resolving power, and compare the algorithm with the performance of the two classic algorithms [9]. The conclusions are given in section 5. In section 5, we finally summarize the work that has been done in this paper and proposes the next research direction in the future [10]. This section focuses on the perceived perceptual image hash algorithms and they are briefly introduced in this paper. By summarizing the existing algorithms, we find that the perceived perceptual image hash technology can be summarized as a unified process. At the same time, this section also describes the main basic properties of perceptual image hash function in details. The goal of the part is to emphasize technical challenges that currently exist in perceptual image hash technology, and in later sections to solve. Finally, this part introduces and realizes two typical perceptual image hash algorithms.

2. Achievements of Two Classic Perceptual Image Hash Algorithms

2.1. Basic Properties of Perceptual Image Hash Function

This section focuses on the perceived perceptual image hash algorithms and they are briefly introduced in this paper. By summarizing the existing algorithms, we find that the perceived perceptual image hash technology can be summarized as a unified process. At the same time, this section also describes the main basic properties of perceptual image hash function in details. The goal of the part is to emphasize technical challenges that currently exist in perceptual image hash technology, and in later sections to solve. Finally, this part introduces and realizes two typical perceptual image hash algorithms [11].

We can assume that Ω represents a limited collection of images and Γ represents the space of key. Perceptual image hash function has two variables which are an original image called $I \in \Omega$ and a key called $K \in \Gamma$, and the end output is an L-bit binary hash string s. $I_{ident} \in \Omega$ means an image which is similar to the original image after some image process. $I_{diff} \in \Omega$ means another images, which are different from the original image by perception. We take θ_1 and θ_2 as two very small arithmetic numbers and it satisfies $0 < \theta_1, \theta_2 < 1$.

Three basic properties of perceptual image hash function can be summarized as follows.

1) Perceptual robustness:

$$probability(\mathrm{H(I,K)}) = \mathrm{H}(\mathrm{I}_{ident}, \mathrm{K}) \ge 1 - \theta_1 \tag{1}$$

2) Resolving power:

$$probability(\mathrm{H(I,K)}) \ne \mathrm{H}(\mathrm{I}_{ident}, \mathrm{K}) \ge 1 - \theta_2 \tag{2}$$

3) Security:

$$probability(\mathrm{H(I,K) = s}) \approx \frac{1}{2^L} M, \forall s \in \{0,1\}^L \tag{3}$$

We find the three properties above are mutual contradiction in essences. Inside, the first property of perceptual image hash function on the smaller image changes is with robustness. However, second property requires there exists small collision probability for the different perceptual input. The same as the third property, and it affects the realization of the first property. Considering the security, the latter two properties are important. The proposed algorithm proposed in this paper is very good compromised among the three properties above [12]. The basic process of the existing perceptual image hashing algorithm can be concluded as three steps just as shown in Fig. 1:

2.2. Basic Process of Perceptual Image Hash Technology

The basic process of the existing perceptual image hashing algorithm can be concluded as three steps just as shown in Fig. 1:

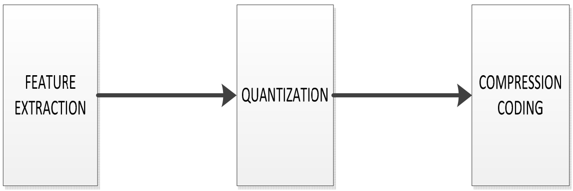

Fig. 1. Image hash process.

The first step is extracting the feature vectors from the image, second step is to get binary hash string by quantizing extracted feature vector, and the third step is to get the final hash sequence by the compressing binary hash string.

2.3. A perceptual Image Hash Algorithm Based on Iterative Filtering

Mihcak and Venkatesan have proposed a perceptual image algorithm based on iterative filtering. The algorithm gets hash value by the approximate component from the original image after binaryzation and wavelet transform [13]. The algorithm is mainly based on a phenomenon they found in a large number of experiments: after very little alteration, the original important geometric features of the original image do not change any more. We describe the process of algorithm in details as follows:

1) We make I represent the original image and operate L wavelet decomposition for I , then take I_A represent the approximate component from image after wavelet decomposition.

2) Set T as a threshold value, then take threshold operation to the approximate component I_A and get binary matrix M.

$$\mathrm{M}(\mathrm{i},\mathrm{j})=\{\begin{matrix} 1 & ifI_A(i,j)\geq T \\ 0 & otherwise \end{matrix} \quad (4)$$

In the formula (4) above, the threshold value T is selected by the self-adaption method and it should satisfy $\mathrm{W}(\mathrm{M})\approx \mathrm{q}$, where q is a parameter which is larger than 0 and less than 1 and W is the normalized Hamming weight.

3) Take $M_1=M$, $ctr=1$.

4) Take statistics and filter in order to M_1 and get $M_2=S_{[m,n],p}(M_1)$, where m, n, p are parameters and S is an order statistics filter.

5) Take $M_3(i,j)=AM_2(i,j)$, then get M_4 from M_3 after linear shift invariant filtering where A is a parameter.

6) Take the same operation on M_4 just as (2) and get M_5, then we can get $\mathrm{W}(\mathrm{M}_5)\approx \mathrm{q}$.

7) If $\mathrm{ctr}\geq \mathrm{C}$, we can end the iteration and go to (8), otherwise we should calculate $\mathrm{D}(M_5.M_1)$ and we will end the iteration if the value is less than threshold value ε and go to (8), or take $M_1=M_5$, $crt=crt+1$ and go to (4) to go on the iteration. $\mathrm{D}(M_5.M_1)$ is the Normalized Hamming distance.

8) $\mathrm{H}(\mathrm{I})=M_5$, where $\mathrm{H}(\mathrm{I})$ is the hash value of the original image I.

2.4. A perceptual Image Hash Algorithm Based on DCT Smoothing Template

Fridrich and Goljan proposed a perceptual image hash algorithm based on DCT smoothing template. The proposed algorithm is based on general experimental phenomena: if an image doesn't change in perception, then the image's low frequency DCT coefficients won't change obviously after discrete cosine transform (DCT). At the same time, to make sure the security of the algorithm and the production of the hash value relying on the key, firstly the algorithm produce N smoothing zero mean template $P^{(i)}$, i =1,2,3…. and then get N mapping values when take the coefficient of the image through DCT switch to the templates in the control of key. To different images, we can get different threshold values and compare the absolute values of N mapping values with threshold values, then we get an N-bits binary value as the final hash value. The process can be described as formulae below.

$$\text{if } \left|B\bullet P^{(i)}\right|<Th\text{ , }\quad b_i=0 \quad (5)$$

$$\text{if } \left|B\bullet P^{(i)}\right|\geq Th\text{ , }\quad b_i=1 \quad (6)$$

The process of the self-adaptive selection process about threshold value Th is based on the theory of maximizing information to determine, then we must ensure that half of the absolute values of N mapping values are greater than Th, half of them are less than Th. By this we can abstract the information in the N-bit binary string contains in maximum.

The process of generating zero mean value smoothing template is as follows: firstly, generate N pseudo random matrixes uniformly distributed between 0 and 1 in the control of the key; then the N pseudo random matrixes generate N pseudo random smooth templates $P^{(i)}$, i =1,2,3,…,N after repeatedly filtering through a low pass filter; in the end, every pseudo random smooth template $P^{(i)}$ minuses the mean value and get the final pseudo random zero-mean smooth template $P^{(i)}$.

3. The Proposed Perceptual Image Hash Algorithm Based on Invariant Moments

This section gives a perception image hash algorithm based on invariant moments. By using invariant property about the invariant moments features to geometry transform, the hash algorithm can be with the ability of geometric transformation, and then through the application of pseudo random rectangle produced by the key in the image, we can realize the generation of the hash values is in the control of the key, thus the algorithm's security can be ensured. Finally, we can compare the algorithm with two classic algorithms and we'll see that this algorithm is with higher robustness and security.

3.1. Invariant Moments and Analysis of its Characteristics

Many algorithms at present can be robust to general JPEG compression and filtering operation, but can't resist geometry transformation attacks. In order to make the algorithm able to resist geometric attacks, we want to extract image features which are invariant to affine transform. The invariant moment is a way to describe the regional image, and the invariant moments can remain unchanged in the main, with the image's affine transformation.

(p+q) order moment of area $f(x, y)$ is defined as

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{7}$$

And the central moment corresponding to the above is

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x, y) \tag{8}$$

In the two formulae above, p, q =1,2,3,…, and $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$, which is a real coordinate. (p+q) order normalized gravity moment $f(x, y)$ is defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \ p, q = 1,2,3,\ldots \tag{9}$$

In the formula (9), $\gamma = 1 + \frac{p+q}{2}$, p, q=2,3,4….

The following 7 two-dimensional invariant moments are derived from the two orders and three orders normalized central moments, they have invariance to translation, rotation, mirror and scaling.

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}{}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right]$$
$$+(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

$$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right]$$
$$+(3\eta_{2121} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

3.2 The Basic Steps of the Algorithm

The basic process is shown in Fig. 2 and steps of the algorithm are just as follows:

1) Our task is to extract the shorter sequence of bits which is the existence as the best response to the nature of the image of an image from a large number of redundant data. Since the wavelet transform can compress to express the image local time-frequency characteristics, we firstly take wavelet decomposition to the image.

2) Generation the pseudo random sequence with the key, and the image can be divided into N pseudo random overlapping rectangular areas by pseudo random sequences. N can be controlled by the key, and the generated hash sequences is in length of 56*N bits. While the image is divided into pseudo-random blocks, it makes up that moment invariants can only describe the global features of the image [14].

3) The 7 invariant moments formulae in section 3.1 can be used to calculate 7 invariant moments of each rectangular region. Though invariant moments calculation in each rectangular area, we get the local information of the image and improve the ability of algorithm to identify the different images.

4) The invariant moments of each rectangular region are combined into a column vector, and the column vector can be quantized for the final hash values.

5) In order to get hash sequence which is more compressed, Reed-Muller error-correcting decoder can be available to generate hash values in compression.

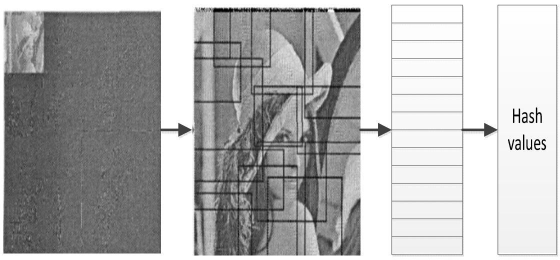

Fig. 2. Formation Process of hash values.

3.3 Similarity Measurement Criteria

We use the normalized Hamming distance $\mathrm{D}(\bullet,\bullet)$ to measure the similarity of two images, which is defined as shown in (10).

$$\mathrm{D}(h_1, h_2) = \frac{1}{L}\sum_{K=1}^{L}\left|h_1(k) - h_2(k)\right| \tag{10}$$

where X is the image, $\bar{X}$ represents the image after the operation of reserving contents, Y represents an image that looks different from X, and $H_k(\bullet)$ represents hash function controlled by the key. Our goal is to get hash values which satisfy the equations (11) and (12).

$$\mathrm{D}\left(H_k(X), H_k(\bar{X})\right) < T_1 \tag{11}$$

$$\mathrm{D}\left(H_k(X), H_k(\mathrm{Y})\right) < T_2 \tag{12}$$

In the equations above, $0 < T_1 < T_2 < 0.5$, here we hope that the more two numbers differ, the better. In ideal circumstance, normalized Hamming distance of images which seem similar is close to 0, and normalized Hamming distance of images which don't look similar is about 0.5.

4. Analysis and Compare on Algorithms' Performance with Normalized Hamming Distance

4.1. Test on Robustness

We test the algorithm on the performance of resistance on various attacks holding image content (such as filtering, noise adding and geometric transformation). We take the normalized Hamming distance average value of the 1000 images with they suffer attack as the output value. At the same time, we compare the performance of the algorithms with two typical algorithms proposed by Fridrich and Miheak.

The performances about three algorithms against geometric transform, filtering, noise, JPEG compression and shear attack are just shown as Fig. 3 to Fig. 11.

From Fig. 3 and Fig. 4 it can be seen after the image suffer geometric attacks, the normalized Hamming distance generated by this algorithm is more close to 0 than that of the other two algorithm, and until take the image achieve a 10 degree rotation and reduce to the original 1/10, the normalized Hamming distance is less than 0.125.

From Fig. 5, Fig. 6 and Fig. 7, we can see that this algorithm can better resist the attacks of mean and Gauss low-pass filter than the other two algorithms, and in the fight against median filtering attacks, it only slightly worse than Mihcak algorithm, with the normalized Hamming distance less than 0.1.

From Fig. 8 and Fig. 9, we can see that this algorithm's ability against additive uniformly distributed noise attack is less than the other two algorithms, but in resistance of additive Gauss noise distribution is superior to the other two algorithms, and when the uniform distribution noise is in the [0, 18] range, the normalized Hamming distance can be not larger than 0.125.

From Fig. 10, we can see that the JPEG compression resistance performance of this algorithm is much better than the other two algorithms, and the normalized Hamming distance of it is no more than 0.03. Fig. 11 shows when the pixels in the image is sheared off about 12 %, the normalized Hamming distance of it is less than 0.125, but when the image pixels suffer large area cut of a malicious attack, the normalized Hamming distance will increase. Then we can see the security of the algorithm.

Examples are shown as Table 1.

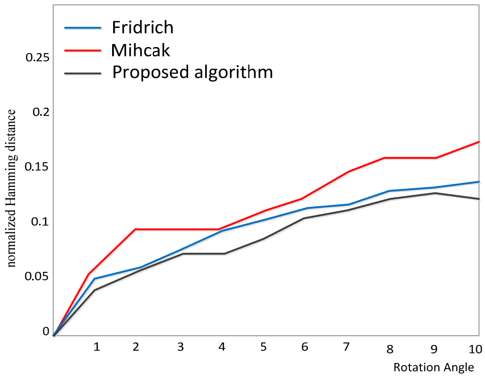

Fig. 3. Performance of resisting rotation attack.

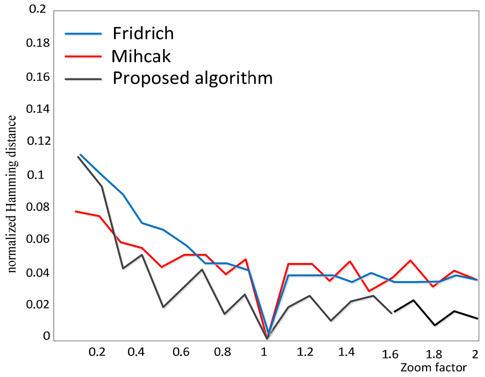

Fig. 4. Performance of resisting scale attack.

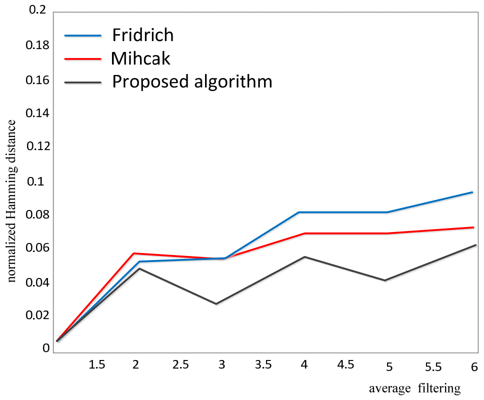

Fig. 5. Performance of resisting average filter attack.

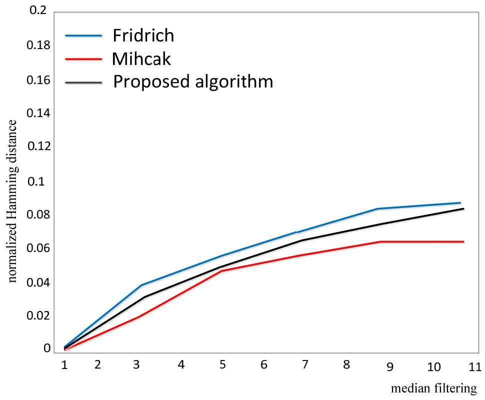

Fig. 6. Performance of resisting median filter attack.

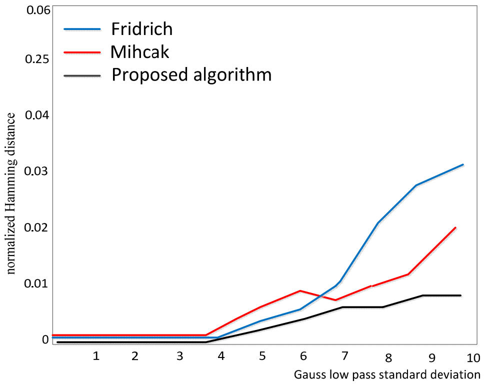

Fig. 7. Performances of resisting Gaussian filter attack.

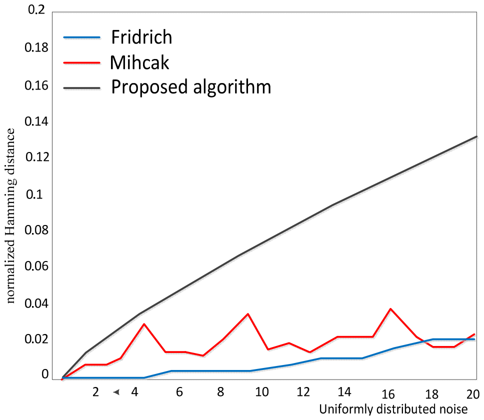

Fig. 8. Performance of resisting uniform distributing noise attack.

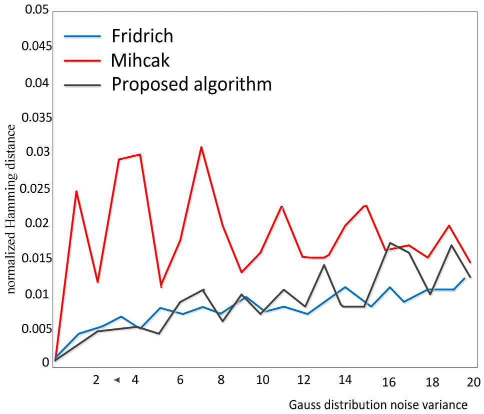

Fig. 9. Performance of resisting Gaussian distributing noise attack.

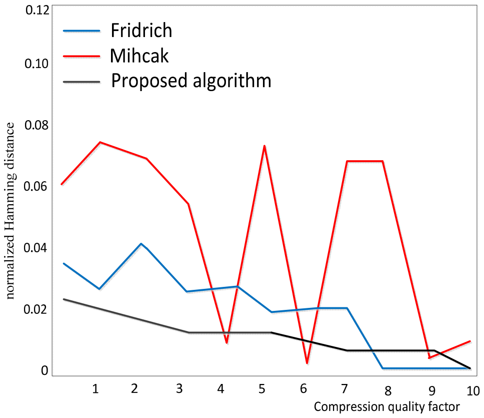

Fig. 10. Performance of resisting JPEG compression attack.

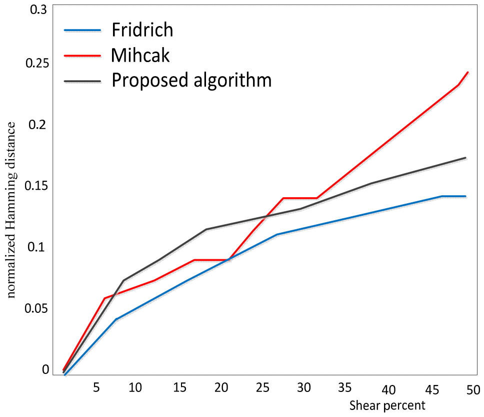

Fig. 11. Performance of resisting shearing attack.

Table 1. Comparison of invariant moments.

Invariant moments	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
Original image	6.6029	19.0687	27.6781	25.4155	56.5682	35.1682	51.9624
Rotate -4°	6.6029	19.0685	27.6782	25.4156	56.5881	35.1652	51.9626
Vertical mirror	6.6029	19.0687	27.6781	25.4156	56.5682	35.1625	52.0573
Reduce 1/2	6.6040	19.0721	27.6790	25.4206	65.7833	35.1691	51.9704

4.2. Test on Collision Resistance

The collision means different images generate similar hash values. We get 499500 matching results when we match 1000 extracted images' hash values and matching values histogram shows as Fig. 12. We can see that the normalized Hamming distances distribute between 0.1786 and 0.5589, and results can be approximate Gauss distribution $\mathrm{N}(\mu,\sigma)$, where the mathematical expectation is $\mu = 0.3270$, the standard deviation $\sigma = 0.0447$, so according to the normalized distribution of Hamming distance and testing results on robustness we can choose $T_1 = 0.125$, $T_2 = 0.15$. Therefore, the collision rate of images:

$$P_c = 1 - \int_{T_1}^{\infty} \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx \qquad (13)$$

The results can be seen that the collision rate of the algorithm is very small, so it can be with a high probability to ensure uniqueness of image hash value.

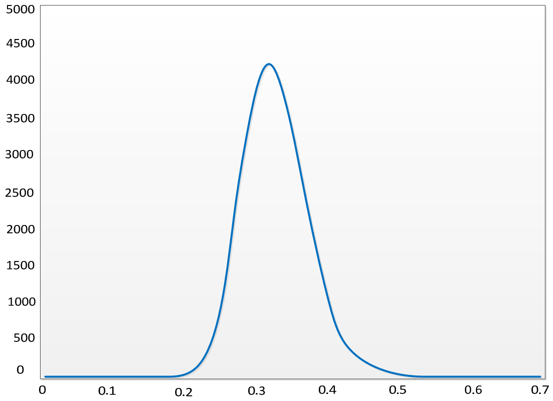

Fig. 12. Statistical histogram of 499500 matched values.

4.3. Test on Security

In order to prove that hash value generated by the algorithm depends on the key, we use the Lena image as a test image, generate 1000 different hash values with different keys, match them in pairs, and get 499500 matching results. Matched values histogram is shown as Fig. 13, and it can be seen from the chart that the normalized Hamming distance of different keys distribute between 0.1911 and 0.4071, after fitting Gauss distribution, the mean and standard deviation are 0.2994 and 0.0248, and according to the formula (13) we can get the probability of different key collision is 1.0161×10^{-12}, which proves the algorithm relies on the key with security. The robustness and security are two important properties of perceptual image hash algorithm.

The average values of Hamming distance between original image and the tampered image are as shown in Table 2 below.

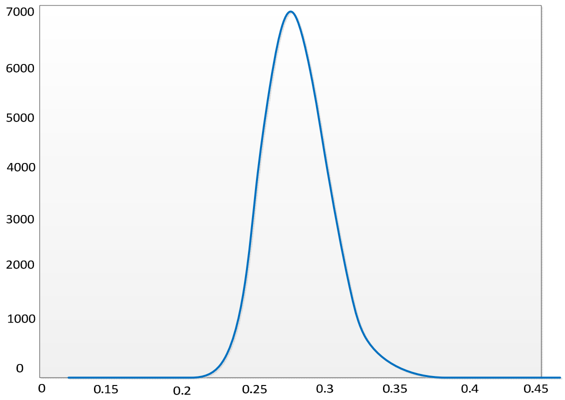

Fig. 13. Statistical histogram of diplo-matching of 1000 different secret keys.

Table 2. Average of normalized Hanmming distance of tampered images.

Hash algorithm	D1	D2
Mihcak	0.20	0.28
Fridrich	0.26	0.31
Proposed algorithm	0.28	0.32

5. Conclusion

This paper proposes a perceptual image hash algorithm based on invariant moments. Through a lot of experiments we see that the algorithm can resist shear, filtering, noise adding, JPEG compression, geometric attacks, and so on, especially it can resist 10° spin attack, so the algorithm has very strong robustness. Also the possibility of confliction between different images is very low, the generated hash values depend on the key, which makes the attacker can't tampered image and keep the hash value, even if he gets the image if he doesn't know the hash values. So we know the algorithm with high security. At the same time, the algorithm can generate pseudo random rectangle to control the length of the generated hash values, thus the robustness and security of the algorithm are compromised.

In conclusion, we propose a new algorithm hash algorithm; its robustness is stronger than two kinds of typical hash algorithm at present. The complexity of the algorithm is very low, running speed is very fast, and it is very suitable for fast retrieval of image database.

6. Conclusions

This paper proposes a perceptual image hash algorithm based on invariant moments. Through a lot of experiments we see that the algorithm can resist shear, filtering, noise adding, JPEG compression, geometric attacks, and so on, especially it can resist 10^{o} spin attack, so the algorithm has very strong

robustness. Also the possibility of confliction between different images is very low, the generated hash values depend on the key, which makes the attacker can't tampered image and keep the hash value, even if he gets the image if he doesn't know the hash values. So we know the algorithm with high security. At the same time, the algorithm can generate pseudo random rectangle to control the length of the generated hash values, thus the robustness and security of the algorithm are compromised.

In conclusion, we propose a new algorithm hash algorithm, its robustness is stronger than two kinds of typical hash algorithm at present. The complexity of the algorithm is very low, running speed is very fast, and it is very suitable for fast retrieval of image database.

References

[1]. Ashwin Swaminathan, Yinian Mao, Min Wu., Robust and Secure image hashing, *IEEE Transaction on Information Forensics and Security,* Vol. l, No. 2, 2006, pp. 215-230.

[2]. Jin A. T. B., Ling D. N. C., Goh A. Biohashing, Two factor authentication featuring fingerprint Data and tokenized random number, *Pattern Recognition*, Vol. 37, No. 11, 2004, pp. 2245-2255.

[3]. Monga V., Evans B. L., Perceptual image hashing via feature points: performance evaluation and tradeoffs, *IEEE Transaction on Image Processing*, Vol. 15, No. 11, 2006, pp. 3453-3466.

[4]. Fridrich J., Goljan M., Robust hash functions for digital watermarking, in *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, Las Vegas, Navada, USA, 2000, pp. 178-183.

[5]. Mihcak M. K., Venkatesan R., New iterative geometric techniques for robust image hashing, in *Proceedings of the ACM Workshop on Security and Privacy in Digital Rights Management,* Vancouver, BC, Canada, 2001, pp. 13-21.

[6]. Kozat S. S., Venkatesan R., Mihcak M. K., Robust perceptual image hashing via matrix invariants, in *Proceedings of the International Conference on Image Processing,* Genova, Italy, 2004, pp. 3443-3446.

[7]. Schneider M., Chang S. F., A robust content based digital signature for image authentication, in *Proceedings of the IEEE International Conference on Image Processing,* Lausanne, Switzerland, 1996, pp. 227-230.

[8]. Kailasanathan C., Naini R. S., Image authentication surviving acceptable modifications using statistical measures and k-means segmentation, in *Proceedings of the IEEE EURASIP Workshop on Nonlinear Signal Image Processing,* Baltimore, MD, 2001.

[9]. Venkatesan R., Koon S. M., Jakubowski M. H. et al., Robust image hashing, in *Proceedings of the IEEE International Conference on Image Processing,* Vancouver, BC, Canada, 2000, pp. 664-666.

[10]. Mihcak M. K., Venkatesan R., A tool for robust audio information hiding: a perceptual audio hashing algorithm, in *Proceedings of the 4th International Information Hiding Workshop*, Pittsburgh, PA, 2001, pp. 51-65.

[11]. Nalini K., Chikkerur R. S., Jonathan H. C., et al., Generating Cancelable Fingerprint Templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 20, No. 4, 2007, pp. 561-572.

[12]. Sutcu Y., Qiming L., Memon N., Protecting Biometric Templates with Sketch, Theory and Practice, *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 3, 2007, pp. 503-512.

[13]. Connie T., Teoh A., Goh M. PallnHashing, A novel approach for dual factor authentication, *Pattern Anal. Appl.,* Vol. 7, No. 3, 2004, pp. 55-268.