

An Improved Histogram-Based Image Hashing Scheme Using K-means Segmentation ^{*}

Yang Ou¹, Chul Sur², and Kyung Hyune Rhee³

¹ Department of Information Security, Pukyong National University,
599-1, Daeyeon3-Dong, Nam-Gu, Busan 608-737, Republic of Korea

`ouyang@pknu.ac.kr`

² Department of Computer Science, Pukyong National University

`kahlil@pknu.ac.kr`

³ Division of Electronic, Computer and Telecommunication Engineering,
Pukyong National University

`khchee@pknu.ac.kr`

Abstract. The perceptual image hashing is an emerging technique which can be used in image authentication and content-based image retrieval. Design of an image hashing scheme which has both robustness and fragility properties is becoming a new challenge. Recently, Xiang *et al.* proposed a histogram-based image hashing scheme which achieves a good robustness to perceptually insignificant attacks. However, the fragility to malicious attack is not optimistic. In this paper, we propose an improved histogram-based image hashing scheme by using K-means algorithm, which obtains a better fragility result than original one. In particular, we firstly segment the image into two segments by using K-means algorithm, then extract histogram in each segment. Finally, we combine two bin groups and generate the image hash value. The experimental results confirm that our improvement not only preserves the robustness of original one, but also enhances the fragility to malicious attacks.

Key Words: Perceptual Image Hashing, K-means Algorithm, Histogram Intensity

1 Introduction

With the spreading use of multimedia information, security of media contents has become an important concern by attracting large research attentions. Multimedia authentication techniques have emerged to verify content integrity and prevent forgery. Traditional data integrity issues are addressed by cryptographic hashes (e.g. MD5, SHA-1) or message authentication functions, which are very sensitive to every bit of the input message. However, the multimedia data, such

^{*} This work was supported by the Korea Research Foundation Grant Funded by the Korea Government (MOEHRD) (KRF-2006-211-D00289)

as digital images, always undergo various acceptable manipulations such as compression, image enhancement or other common signal processing operations. The sensitivity of traditional hash functions could not satisfy these perceptual insignificant changes. Nowadays, perceptual image hashing (also called image hashing or perceptual hashing) is emerging rapidly [1, 2] which takes into account changes in the visual domain. Perceptual hash functions are designed to produce the same hash value as long as the input has not been perceptually modified. Particularly, the image hash should be robust against perceptually unchanged modifications while highly sensitive to different digital representations.

1.1 Related Work

The early image hashing approaches [3, 4] usually based on DCT/DWT use the low frequency components to extract important image features from a reduced set of the original image data. However, performance of these techniques is not very satisfactory as the relation between the low frequency transform domain coefficients and the essential contents of the image is not clear.

More attention has been concentrated on the robustness of image hashing against geometric transformation, such as rotation, scaling and cropping for an acceptable percentage. Lu *et al.* proposed a mesh-based image hashing scheme [5] by identifying meshes and normalizing meshes into standard size blocks to generate the hash. This scheme showed a good resistance to geometrical distortions, however, the authors claimed that it is somewhat complex because of the mesh normalization. In [6], an image hashing method based on discrete polar Fourier transform was proposed to make the hash resilient to geometric and filtering operations. Meanwhile, Monga *et al.* [7] proposed the scheme by using end-stopped wavelet to capture feature points in the image, then the feature vector is quantized to form a binary string as the hash value. However, even though most image hashing approaches provide a satisfactory robustness property, the sensitivity to perceptual significant modifications, i.e, whether they can distinguish certain malicious distortions, is still somewhat indefinite. Li *et al.* [8] have claimed that some state-of-the-art image hash algorithms [6, 7, 9] could not detect small malicious attack and some authentic distortions. Therefore, the fragility of an image hash should be also considered deliberately.

More recently, a novel histogram shape-based image hashing scheme [10] is proposed by employing the insensitivity of histogram shape to geometric distortions. The experimental results in [10] depicted that it is much more robust than previous approaches [6, 7] against geometric changes. Nevertheless, the histogram extracted from the whole image cannot exhibit local information. Thus, this scheme could not achieve a satisfactory result to fragility of malicious attacks.

1.2 Our Contributions

In this paper, aiming to overcome the limitation of Xiang *et al.*'s scheme [10], we propose an improved image hashing scheme based on K-means algorithm which

is a simple unsupervised learning algorithm popularly used in image segmentation. Our proposed image hash function achieves a higher fragility to perceptual significant attacks on the image, as well as preserving a similar robust level of original scheme. In order to robustly capture features from the whole image but not lose local information, we employ K-means clustering algorithm: firstly, we segment the image into two segments; secondly, we combine the histogram bins which are extracted from each segment, then finally generate the hash value by comparing any two bins in the bin group. Moreover, we provide comparative experimental results to confirm that our improvement obtains a better sensitivity to perceptual significant result than original scheme.

The remainder of the paper is organized as follows. In Section 2, we describe the basic requirements of image hashing. The original histogram shape-based image hashing scheme is briefly reviewed in Section 3. Our improved histogram-based image hashing scheme is presented in Section 4. Following, we show the experimental results on both original and improved schemes. Finally, we conclude the paper in Section 5.

2 Requirements of Image Hashing

Perceptual image hashing must be both perceptually robust and secure. An image hash should capture the essential attributes of the image so that insignificant changes to the human eyes will not substantially alter the hash value. Generally, the image hashing involves a secret key in order to introduce the randomization of final hash value. The following notation is used to describe the requirements of image hashing:

- I : the input image, which is the hash value generated from;
- k : the key involved in image hash generation;
- $P(*)$: a certain probability;
- $H(*)$: the image hash function;
- θ_1, θ_2 : two given parameters where $\theta_1, \theta_2 \in (0, 1)$.

Three desirable properties of a perceptual image hash are identified as follows [2]:

- (1) Perceptual robustness:

$$P(H(I, k) = H(I_{ident}, k)) \geq 1 - \theta_1, \text{ for a given } \theta_1,$$

- (2) Fragility to visually distinct images:

$$P(H(I, k) \neq H(I_{diff}, k)) \geq 1 - \theta_2, \text{ for a given } \theta_2,$$

- (3) Unpredictability of the hash:

$$P(H(I, k) = v) \approx \frac{1}{2^q}, \forall v \in \{0, 1\}^q$$

where I_{ident} is an image visually identical or very similar to I , I_{diff} is different from, or a tampered version of I , and q is the length of the binary hash sequence.

Property-1 requires that for any pair of "perceptually identical" images and a secret key, the hash values must be identical with high probability. Whereas property-2 requires that for "perceptually different" images, their hash values should be also different from each other. The third property requires that as the secret key k is varied over a possible key set for a fixed input image, the output hash value must be approximately uniformly distributed among all possible q -bit outputs.

The second and third requirements may collectively be considered as the security of image hashing. It is clear that these three basic properties are in conflict with one another. A hash based on crude image features may be very robust since modifications to the image cannot affect the hash value. In this case, however, collision between perceptually different images will likely happen. Further, to meet the third requirement, perfect randomization, i.e. uniform distribution of the hash value over the key space, is desirable although this will inevitably sacrifice to some extent the robustness property without taking appropriate measures.

From the practical point of view, both robustness and security are important. Lack of robustness makes an image hash useless as explained in the previous section, while security means it is extremely difficult for an adversary to modify the essential content of an image yet keep the hash value unchanged. Thus tradeoffs must be sought, and this usually forms the central issue of the image hashing research.

3 Histogram Shape Based Image Hashing

Xiang *et al.* proposed an image hashing scheme robust against geometric deformations [10]. The image histogram shape invariance to geometric distortions is exploited for image hashing. The histogram shape is represented as the relative relations of groups of two different bins. Figure 1 illustrates the procedures of image hash generation.

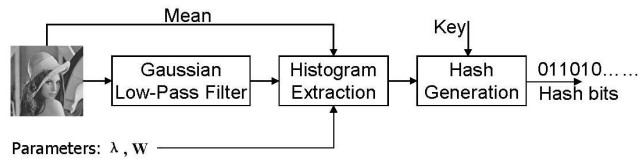


Fig. 1. The histogram shape-based image hashing scheme [10]

The input image is firstly filtered with a low-pass Gaussian filter, where the 2-D convolution mask has the size $(2 * k * \sigma + 1) \times (2 * k * \sigma + 1)$. Here σ

is the standard deviation of the distribution and k is the truncation point for preserving more energy of Gaussian distribution. The authors set $\sigma = 3, k = 3$ in this Gaussian filter to achieve a tradeoff between robustness and sensitivity of the following computed hash value.

After low pass filtering, the histogram is extracted from the preprocessed image by referring to the mean value of the original image. The histogram of a gray-level image of size $X \times Y$ with bins of the width W can be described by

$$H_W = \{h_W(i) | i = 1, \dots, L\}, \quad (1)$$

where $h_W(i)$ denotes the number of pixels in the i^{th} bin and satisfies $\sum_{i=1}^L h_W(i) = X \times Y$. In general, the first or last several bins hold less samples and they are more sensitive for only slight modifications on the image. Therefore, the histogram is suggested to be computed from a selected gray range by referring to the mean of the input image. The selected gray range is defined as

$$R = [(1 - \lambda)M, (1 + \lambda)M], \quad (2)$$

where $\lambda \in [0.4, 0.6]$ is selected to remove the bad bins by multiplying the mean M . On the other hand, to gain the robustness against the interpolation errors, the bin width W should be limited in the range $[2, 5]$.

Now, a binary sequence is computed according to the relative relations in the number of pixels among groups of two different bins. Denote the number of pixels in the i^{th} bin as $h(i)$. Let two different bins be a group denoted by $\{h(i), h(j)\}$ satisfying the condition $1 \leq i < L$ and $i < j \leq L$. Hence, the number of groups is totally $C_L^2 = L(L-1)/2$. For the group $h(i), h(j)$, a binary sequence is obtained by comparing $h(i)$ and $h(j)$, formulated as

$$\text{bit} = \begin{cases} 1 & \text{if } h(i)/h(j) \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Finally, the key-dependent hash is obtained by randomly permuting the resultant binary sequence is. The permuted binary sequence denoted by $\text{hash} = \{\text{hash}(k) | 1 \leq k \leq C_L^2\}$.

The corresponding parameters are given as: $L = 30, \lambda = 0.55$ and $W = 4$. Thus, for gray-level images, each image hash has a length of 435 bits extracted from the histogram with 30 bins.

Discussions. The geometric attacks respect the rules that some or all of the pixels are displayed at a random amount under the constraint of visual coherence. Actually, the histogram extracted from one image is independent of the shifting of the pixels on the image plane. The authors exploit this merit to generate a robust image hash. However, generating the same histogram from a different image is not difficult. Obviously, this image hashing scheme could not detect these malicious modifications on the image, such as face morphing. More details are discussed in Section 5.

On the other hand, as we mentioned before, the image hashing can be applied to image authentication and image retrieval fields. Even though the original scheme could not be sensitive to some malicious attack, it can also be used to retrieve similar images in image databases, since in this application the distinction requirement is not very strict. However, regarding the image authentication, the sensitivity to malicious attacks should be considered deliberately.

4 Improved Image Hashing Scheme Using K-means Segmentation

The histogram of the entire image only reflects the global information, and also there is no spatial/local information about the image intensities. A direct solution of this problem is to divide the image into blocks and separately calculate histogram for each block [11]. However, the intensity of pixels in blocks may not be stable under some geometric attacks. One example of original and rotated image blocking is shown in Figure 2. Clearly, the histogram in each block of Figure 2(b) must be largely different from Figure 2(a). Hence, using image blocking to capture local information is not robust against some geometric deformations.

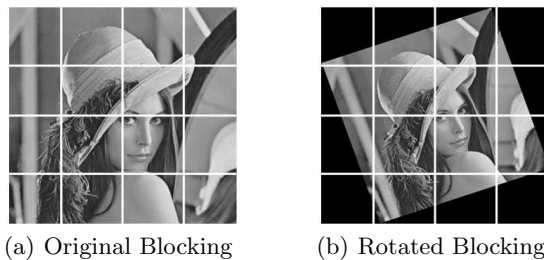


Fig. 2. Example of Blocking Images

In order to robustly capture local information in the image, we propose an improvement of image hashing scheme in [10] by employing K-means segmentation [12]. The block diagram of our enhanced scheme is shown in Figure 3. After Gaussian Low-pass filtering, the filtered image is segmented by using K-means algorithm. Then, the histogram in each segmentation is calculated separately. Finally, we concatenate the bins in each segmentation and generate the final hash value.

4.1 Image Segmentation Using K-means Algorithm

K-means algorithm is one of the simplest unsupervised learning algorithms that solving the clustering problem. It is popularly used in pattern recognition [13] and image retrieval [14] area. Briefly speaking, suppose observations are $\{x_i :$

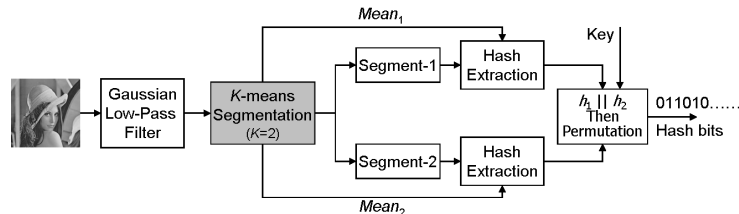


Fig. 3. Block Diagram of Proposed Improved Scheme

$i = 1, \dots, L\}$, the goal of K-means algorithm is to partition the observations into K groups with means $\hat{m}_1, \hat{m}_2, \dots, \hat{m}_K$ such that

$$D(K) = \sum_{i=1}^L \min_{1 \leq j \leq K} (x_i - \hat{m}_j)^2 \quad (4)$$

is minimized, where $D(K)$ is the sum of squares of distances between observations and the corresponding means (also called cluster centroid).

In image segmentation application, the observations are the pixels in image plane. Therefore, after K-means segmentation, the image is segmented into K non-continuous regions. Each of them has its own mean as the cluster centroid. Since in [10] the histogram is extracted by referring the mean value of the image, here we propose to extract bins in each segment depending on the segment mean.

Because the robustness against perceptual insignificant attacks on the image should be considered carefully, we suggest that K is only set to 2. For larger K , extracted bins will be too sensitive and not suitable for this histogram-based image hashing.

Note that, in practice there are some geometric modifications resulting in a number of redundant pixels. We remove these pixels in order to keep the stability of the means. On the other hand, considering this kind of changes, since the segmentation is based on pixel values but not pixel coordinates, the problem we mentioned in image blocking (refer Figure 2) can be avoided in our K-means algorithm-based scheme. Moreover, the local information is also captured as shown in Figure 4, which are the segmentation masks of both original Lena and face-attacked Lena images (The original face-attacked image is shown in Figure 5(d)).

4.2 Hash Generation

After K-means image segmentation, the procedures of bins extraction from each segment are similar as introduced in Section 3. In each segmentation, the first and last several bad bins are also needed to be removed by referring the segment mean value. We present two slight modifications to make the bins extraction more appropriate on segments:

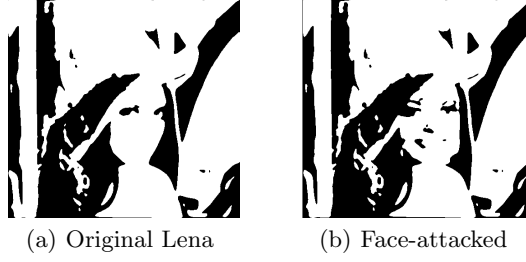


Fig. 4. K-means Segmentations of Original and Face-attacked Lena Images

- (1) Since the mean values in each segment from one image are not the same, the selected gray range could not be defined by Eq.(2). In order to extract a fixed number of bins, we firstly find the "center" bin which is the closest to the segment mean value. Then the same number bins before and after the previous "center" bin are extracted.
- (2) We set the bin width $M = 4$, but the number of available bins is reduced in each segment because of K-means clustering. Therefore, in each segment, we suggest only 12 through 15 bins be extracted. For example, if 13 bins are selected in each segment, it means after finding the "center" bin, 6 bins before and after it will be extracted.

Let $h_1(k)$ and $h_2(k)$ be the number of pixels in the k^{th} bin in two segments, separately. We combine all bins extracted from segments and form the final bin group $\{h(i)|i = 1, \dots, L\}$. L is the total bin number. Then, a binary sequence is generated by using Eq.(3). The key-dependent hash is obtained by randomly permuting the resultant binary sequence.

5 Experimental Results

In this section, we simulate both our proposed and original schemes in terms of fragility and robustness testing. All test images are selected in grey level with size 512×512 .

The Hamming distance and Euclidean distance are commonly used metrics for comparing the similarity of two sequence. We use the Hamming distance to compare hash value since it is more suitable for binary sequence. For a pair of image hashes, the Hamming distance between them is defined as

$$Dis(hash_1, hash_2) = \frac{1}{l} \sum_{k=1}^l |hash_1(k) - hash_2(k)| \quad (5)$$

which is expected to be close to 0.5 for different ones and zero for similar images. l is the length of hash sequence, which used to normalize the distance in the range $[0,1]$.

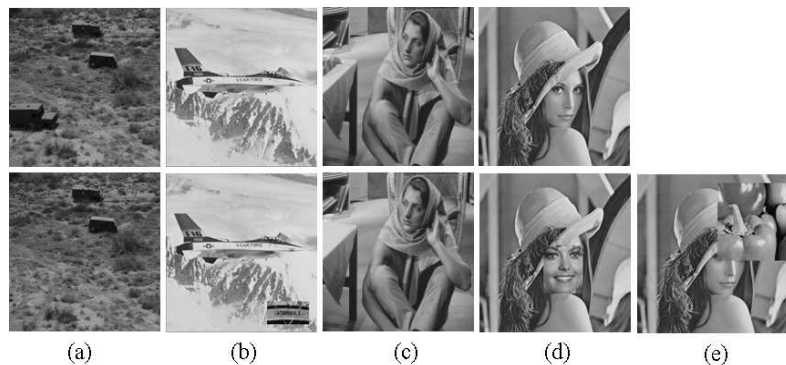


Fig. 5. Examples of Malicious Attacks Images

5.1 Fragility Testing of Original Scheme

The image attacks can be classified into two types: perceptual insignificant attack and perceptual significant attack. The former includes geometric modifications and common signal processing operations on the image. Whereas some malicious modifications, such as object removal, object changing and object insertion, can be regarded as perceptual significant attack. Figure 5 shows some examples of malicious attack on images, where the above images are original ones and the below images are attacked ones.

Table 1. Relative Hamming Distances by Various Attacks

Attacks	Hamming Distance
Object changing	0.0161
Object removal-1	0.0207
Object removal-2	0.0270
Object insertion	0.0345
JPEG, QF=50	0.0084
Gaussian filter, $\sigma^2 = 5$	0.0368
Gaussian noise, $\sigma^2 = 6$	0.0056
Rotation, 10°	0.0178
Cropping, 5%	0.0541

To achieve good robustness to perceptual insignificant changes and the fragility to collision, tampering and forgery are serious challenges. Here we show some experimental results of different attacks on original histogram-based image hashing scheme. The Hamming distances of image hashes between original Barbara and attacked Barbara images are listed in Table 1. Obviously, from the right column data, we can find that there is no largely different between perceptual insignificant attack and malicious attacks. In other words, the original histogram-based

Table 2. The Perceptual Insignificant Attacks in Figure 6

	Attacks	Parameters
JPG	JPEG compression	QF=50
J2K	JPEG2000 compression	Ratio: 10
GN	Gaussian noise	$\sigma^2 = 5$
UN	Uniform noise	$\sigma^2 = 5$
GF	Gaussian filter	$\sigma^2 = 6$
GB	Global bending	Factor: 5
RO	Rotation	Degree: 10°
SC	Scaling	Percent: 5%
CR	Cropping	Percent: 12.5%
HFB	High frequency bending	Factor: 0.5

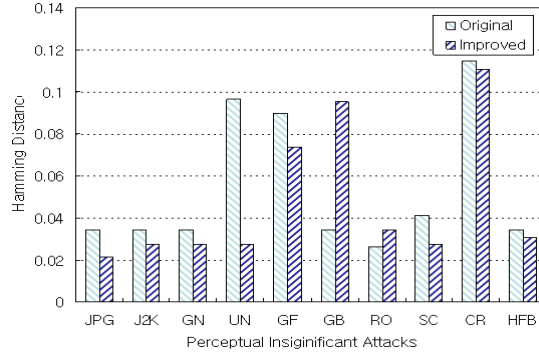
**Fig. 6.** Comparisons of Perceptual Insignificant Attacks

image hashing scheme could not distinguish that which image is maliciously modified and which one is perceptually unchanged. Therefore, even though this approach achieves a good robustness to geometric deformations, the fragility to malicious modifications could not be guaranteed.

5.2 Improvement Testing of Proposed Scheme

In this testing, we show the comparative results of our improved scheme on both robustness and fragility.

Firstly, we select ten popular perceptual insignificant attacks including geometric changes and some image processing operations, such as compression and low pass filters. The detailed information about attacks is listed in Table 2. Figure 6 illustrates the Hamming distances of image hashes between original images and perceptual insignificant attacked images, by simulating both original and improved scheme. The goal of this comparison is to confirm that the good robustness property of original scheme is preserved in our improved scheme.

Fortunately, we can observe that most of the hamming distances of improved approach are similar or lower than original one, thanks to the K-means clustering algorithm. Note that, less hamming distance implies the scheme is more robust against perceptual insignificant attacks. The Global bending (GB) introduces wave-like bending in image plane, which may result a tiny instability during image segmentation. However, the Hamming distance is also lower than 0.1 under this kind of attacks.

On the other hand, enhancing the fragility to perceptually significant attacks is another goal in our proposed scheme. We simulate some representative malicious attacks listed in Table 3. The first five attacks can be found in Figure 5. Figure 7 depicts the Hamming distances of image hashes between original images and perceptual significant attacked images, obtained by both original and improved scheme. Obviously, our improved scheme achieves relative larger Hamming distance than original one, which depicts that there is an obvious discrimination between perceptual changed and unchanged images. It is worth to note that, in fragility testing, larger hamming distance indicates that the scheme is more sensitivity to malicious attacks. Therefore, our approach is more fragile to perceptual significant attacks. The attack (e) is an arbitrarily cropping and paste attack which changes almost one quarter of an image. Hence, it is not strange that the original one also obtains a high Hamming distance.

Table 3. The Perceptual Significant Attacks in Figure 7

	Image	Attacks
a	Trucks	Object removal
b	Airplane	Object insertion
c	Barbara	Object removal
d	Lena	Face morphing
e	Lena	Arbitrary cropping and paste
f	Satellite1	Object insertion
g	Barbara	Object removing + face morphing
h	Satellite2	Object insertion

As we mentioned in Section 2, a good image hashing scheme should be robust to perceptually insignificant changes, as well as fragile or sensitive to perceptually significant changes which mean that an adversary can not obtain the same hash value by modifying the image content. According to the above experimental results, we can observe that our proposed scheme does not destroy the good robustness property of Xiang’s approach. Furthermore, it achieves a higher security level since the fragility to visually distinct images is largely increased.

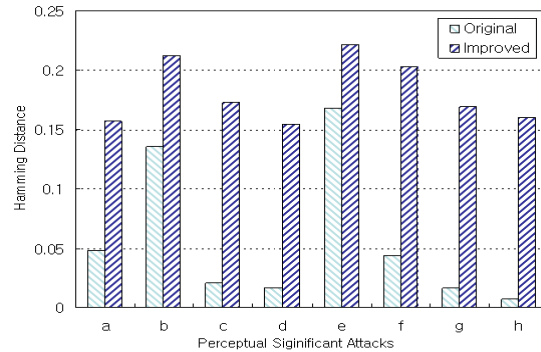


Fig. 7. Comparisons of Perceptual Significant Attacks

6 Conclusion

In this paper, we have proposed an improved histogram-based image hashing scheme by using K-means algorithm. Our proposed scheme is simple but has an excellent simulation results than original one. Our scheme overcame the limitation of original one by robustly capturing local information in the image. The experimental results showed that our improved scheme achieves robustness performance for perceptual insignificant attacks, and also satisfactory fragility performance for perceptually significant attacks.

References

1. S.Wang, X.Zhang: Recent development of perceptual image hashing. *Journal of Shanghai University (English Edition)* **11** (2007) 323–331
2. V.Monga: Perceptually based methods for robust image hashing. PhD thesis, University of Texas (2005)
3. J.Fridrich, M.Goljan: Robust hash functions for digital watermarking. In: *IEEE International Conference on Information technology: coding and computing*. (2000) 178–183
4. M.K.Mihcak, R.Venkatesan: New iterative geometric technique for robust image hashing. In: *ACM workshop on security and privacy in digital rights management*. (2001) 13–21
5. C.Lu, C.Hsu: Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication. *Multimedia Systems* **11** (2005) 159–173
6. A.Swaminathan, Y.Mao, M.Wu: Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security* **1** (2006) 215–230
7. V.Monga, B.L.Evans: Perceptual image hashing via feature points: performance evaluation and tradeoffs. *IEEE Transactions on Image Processing* **15** (2006) 3453–3466
8. F.Li, P.Bart: Attacking some perceptual image hash algorithms. In: *IEEE International Conference on Multimedia and Expo*. (2007) 879–882

9. L.Yu, M.Schmucker, C.Busch, S.Sun: Cumulant-based image fingerprints. In: SPIE-IS&T Electronic Imaging. (2005) 68–75
10. S.Xiang, H.Kim, J.Huang: Histogram-based image hashing scheme robust against geometric deformations. In: MM&Sec '07: Proceedings of the 9th workshop on Multimedia & security, ACM (2007) 121–128
11. M.Schneider, S.F.Chang: A robust content based digital signature for image authentication. In: IEEE International Conference on Image Processing. (1996) 227–230
12. J.A.Hartigan, M.A.Wong: A k-means clustering algorithm. Applied Statistics **28** (1976) 100–108
13. S.Theodoridis, K.Koutroumbas: Pattern recognition. Academic Press (2006)
14. J.Zhang, C.W.Yoo, S.W.Ha: ROI based natural image retrieval using color and texture feature. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery. (2007) 740–744