A Short Survey on Perceptual Hash Function

Arambam Neelima[1], Kh Manglem Singh[2]

[1]Department of Computer Science & Engineering, NIT Nagaland,
Chumukedima, Dimapur- 797103, Nagaland. INDIA.
neelimaarambam@yahoo.co.in

[2]Department of Computer Science & Engineering, NIT Manipur,
Takyelpat , Imphal - 795001,Manipur,INDIA
manglem@gmail.com

Abstract: ***The authentication of digital image has become more important as these images can be easily manipulated by using image processing tools leading to various problems such as copyright infringement and hostile tampering to the image contents. It is almost impossible to distinguish subjectively which images are original and which have been manipulated. There are several cryptographic hash functions that map the input data to short binary strings but these traditional cryptographic hash functions is not suitable for image authentication as they are very sensitive to every single bit of input data. When using a cryptographic hash function, the change of even one bit of the original data results in a radically different value. A modified image should be detected as authentic by the hash function and at the same time must be robust against incidental and legitimate modifications on multimedia data. The main aim of this paper is to present a survey of perceptual hash functions for image authentication.***

Keywords: Hash function, image authentication.

**Cite as: Arambam Neelima, Kh. Manglem Singh, "A Short Survey on Perceptual Hash Function" ADBU J.Engg Tech, 1(2014) 0011405(8pp)*

1. Introduction

A hash function may be defined as any algorithm that maps data of variable length to a data of fixed length called as hash value. A hash function takes data of finite variable bit length as input and produces a value of fixed size as output. The output of such a function is called hash value, hash or message digest. The input data is often termed message. Hash functions are categorized into keyed and unkeyed hash functions. An unkeyed hash function generates a hash value from an arbitrary input, whereas a keyed function generates a hash value from an arbitrary input using a secret key. Keyed hash functions take the data x to be hashed and a secret key k as input. It also provides authentication in addition to data integrity. Hence, by involving a secret key k to the hash calculation not only the integrity of the message x can be proved, but also the origin authenticity is verifiable. However, an unkeyed hash function is designed only to provide data integrity. It takes only the data to be hashed as input. A hash function should have the following properties.

- Uniqueness: Perceptually distinct images should produce unique hash values
  $H_k(I) \neq H_k(I')$
- Procedure Compactness: The size of the hash value should be smaller than the size of the original image size($H_k(I)$)<size(I)
- Perceptual Robustness: Perceptually identical image should have the same hash value $H_k(I) \approx H_k(I_d)$, if I and I_d are perceptually identical.
- One way function: The hash generation should be invertible.

Most of the perceptual hash functions try to extract features of the digital image, which are invariant under significant modification in order to meet the above desired properties. When developing and designing robust hash functions the main challenge while developing and designing a robust hash function is to be able to distinguish modifications on the content that changes the perception from those

that do not change the perception. Most robust hash algorithms are based on the extraction of features relevant for the human perception. After the extraction, the hash of just these features is calculated. Apart from the above points a perceptual hash function should satisfy some additional properties which are listed below [2].

- Perceptual Similarity/Robustness**:** Hash algorithm has to be robust, i.e. incidental and legitimate post processing operations on the media must not render an entirely different hash value. Hence, perceptually similar respectively indistinguishable media objects have to exhibit the same or a similar hash value.
- Distinction**:** The property distinction predicates the computational/practical infeasibility of determining two perceptually different media data exhibiting the same or a similar hash value.
- Security**:** Security against active and malicious attacks on the feature extraction and the steps (like preprocessing stage, partitioning, transformation, feature extraction, post processing) should be provided. Furthermore, it is important to point out that the properties "pairwise independence" for perceptually different media data and "uniform distribution" for all media objects should also apply to robust hashes. Finally, it is recommendable to extract the features of the multimedia content dependent on a secret key between the sender and the verifier.
- Localization**:** The applied system should be able to localize the tampered regions of the inspected multimedia content.

So, a robust image hash function normally consists of a pre-processing stage, partitioning, transformation, feature extraction, post processing and lastly hash value generation. But it is not compulsory to have all these stages for perceptual image hash function. Pre-processing stage converts the data into a general format. Partitioning consists of partitioning the image into block or frame. Transformation stage transforms each partitioned block into some frequency domain as they may be sensitive to modifications. Meaningful and content relevant features are extracted in feature extraction stage. Features may include coefficients of the transformed domain, block based histogram, image edge information etc. Postprocessing stage consists of a quantization of the extracted features in order to gain more robustness against distortion and reduce the memory requirement.

2. Hashing Function

In this section, we shall discuss various hashing methods. A hashing function should tolerate the image processing operations for transmission, enhancement or restoration and at the same time should detect any significant changes in the image content.

2.1 Histogram based Hash Functions

Xiang et al developed a histogram based image hashing scheme [4]. The images were filtered with a low pass Gaussian filter followed by histogram extraction and then a binary sequence are generated from the histogram extracted using the formula given in Equation 1, which is afterwards randomly permuted with a secret key giving the hash bits

$$\text{bit} = \begin{cases} 1, & \text{if } \frac{h(i)}{h(j)} \geq 1 \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

where $h(\text{i})$ and $h(j)$is the number of pixels in the i^{th} and j^{th} bin respectively

The proposed method achieved satisfactory robustness performance for most common signal processing operations and geometric attacks, however it could not detect malicious modification on the image such as face morphing. These limitations were overcome by using *k*-means algorithm. They proposed an improved histogram based image hashing scheme where the image is passed through a low pass Gaussian filter, then they segment the image into two segments by using *k*-mean clustering algorithm, and then extract the histogram in each segments. Binary sequences are generated from the histogram using the similar approach [4], which is further combined to generate

the hash value. Even though this approach achieves a good robustness to geometric deformations it still shows weakness to global bending.

Choi and Jung Park proposed an image hash scheme based on hierarchical histogram [5]. It hierarchically changes the width of the bin by merging several bins and empowers a weighting factor into the hash at each level. A hash string is generated from the histogram of the image by using the following formula.

$$\text{hash}(k) = \begin{cases} 1, \text{if } p(k) - p(k+1) > Th \\ 0, \text{otherwise} \end{cases} \quad (2)$$

where *p(k)* is the number of pixel in the k^{th} bin and *Th* is a threshold defined to minimize the error generated by similar population of neighbouring bins.

The proposed method is robust to all noise addition, rotation and cropping attacks, but sill shows weakness to histogram equalization. Xiang and Kim proposed another histogram based image hashing method for searching content preserving copies. The images are filtered with a Gaussian filter, and then histogram is extracted from the preprocessed image referring to the mean value. The hash bit is generated from the histogram similarly as in [3].

2.2 Discrete Cosine Transform based Hash Functions

The primitive image hashing approaches are usually based on discrete cosine transform (DCT), discrete wavelet transforms (DWT) and fast Fourier transform (FFT). DCT expresses a function or signal in terms of sinusoids with different frequencies and amplitudes using only cosine functions whereas DFT uses both cosine and sine functions.

Fridrich and Goljan [6] observe that the magnitude of a low-frequency DCT coefficient cannot be changed easily without causing visible changes to the image. They construct a robust hash using the block DCT coefficients.

Lin and Chang [7] proposed a method based on DCT coefficient, which is at the same position of two different block of the image. They extract some features of an image based on the relationship between DCT coefficients, which are at the same position of two different block of the image. These features are then encrypted using a public key encryption method to from the hash value.

Sun and Chang [8] also proposed a DCT based hash function. They used only three DCT AC coefficients for every 8×8 image block along with DC coefficients.

2.3 DWT Hash Functions

Mihcak and Venkatesan [9] proposed a method that employs DWT on the image followed by an iterative filtering. However, the proposed method does not involve any pseudo randomness as it does not use any secret key. Later they proposed another method in order to overcome the earlier limitations. They randomly chose some regions of the image using a secret key and formed a sub-image of the input image and applied the earlier algorithm in the sub-image formed.

Lu and Hsu [32] proposed a geometric distortion resilient image hashing method. They passed the image through a DWT and generate the meshes of the images i.e. decompose the image into set of disjointed triangles by taking few salient points from the lowest frequency sub bands. Then, a mesh normalization process is used to transfer the decomposed meshes to meshes of fixed size using affine transformation and interpolation. Then a binary string is generated as a hash code using the following equation

$$H_k(s) = \begin{cases} 1, \text{ if } \left| AC_k^s \right| \text{ belongs to the first 32} \\ \qquad \text{largest AC coefficient} \\ 0 \text{ , otherwise} \end{cases} \quad (3)$$

where, $AC_k^s(1)$ is the first coefficient located at the lowest frequency sub band in a block of a normalized mesh and $H_k(s)$ is a hash bit in a binary hash sequence.

2.4 SVD based Hash Functions

Singular Value Decomposition (SVD) is often used in image processing fields. It decomposes the image into three components U, S and V. Suleyman et al [22] pseudo randomly extracted the features to construct a secondary image through SVD method. Final features are extracted from the secondary image using SVD method to form the final hash value.

Ricardo et al [11] proposed a new image hashing function, which consists of image normalization, SVD decomposition, quantization and compression. First they resized the image into a standard size followed by division of image into sub-image. SVD decomposition was applied on randomly selected sub-images, and after rearranging *U* and *V*, SVD was applied again and finally quantized and compressed through decoding stage of an order-3 Reed Muller decoder to generate the final binary hash value.

2.5 Local Color Features based Hash Functions

In [25], Zhenjun et al extracted the color features of the input image after resizing it to a standard size and obtained a color feature vector. These color features were then compressed to form the final hash value.

2.6 Radon Transformation based Hash Functions

Lei et al [17] proposed a method based on Radon Transformation. The input image was passed through a Radon Transform and moment features were calculated. Then DFT was applied on the moment features. The significant DFT co-efficient were normalized and quantized to form the final perceptual hash.

Wu et al [26] used Radon transformation as a pre-processing stage for their hash function and use a 2 level wavelet decomposition to extract the features from Radon coefficients. After the feature extraction, FFT was applied on the high frequency wavelet coefficients to make the feature values distribution sparser, and finally the FFT coefficients were transformed into a binary string taking the mean of coefficients as the quantization threshold.

GuoD and Hetzinakos [27] also proposed a method which was a combination of discrete wavelet transform and Radon Transform.

2.7 Other Methods

Ahmed [28] proposed a wavelet based hashing scheme. The image is transformed by modulating each pixel of the input image with the help of a secret key. This transformed image then underwent wavelet transformed to generate an intermediate hash value, which is then permuted using another secret key to produce the final hash value. This method is robust against JPEG compression, low pass and high pass filtering. However the proposed system is not robust to other operations like change in brightness, contrast enhancement.

Ashwin et al [3] developed a hash scheme based on Fourier Mellin transform (FMT). They applied a low pass filter to the image, which was then scaled to a predetermined size followed by histogram equalization. FFT was applied to the resulting image and was converted. Image pixels in polar coordinates were summed up along the x-axis. The resulted sum was then quantized and compressed to generate the final hash value using Reed-Muller decoder. Later they improved the scheme by making the quantization and compression stage key dependant.

Tang et al scaled the input image to a square image by bilinear interpolation method. A Gaussian low pass filter is then applied to the square image followed by a division of the image into different rings. Entropies are calculated from each ring and combine to form the hash value [31], image texture was used to characterize the entropy in images. The length of the hash generated will be equal to the number of rings divided.

Azhar [29] et al proposed a method based on quantization step analysis. The input image is

divided into non overlapping blocks, mean of each blocks are extracted as the features from the transformed image. These features are quantized to form the quantized intermediate perceptual hash vector. Uniform quantization scheme is used for the quantization stage. The quantized intermediate perceptual hash vector is compressed and encrypted by the cryptographic hash function SHA-1 to form the final hash value.

Liu and Chang [30] proposed a hashing scheme based on the wave atom transform. The input was decomposed into multiscale coefficients with tilings using the wave atom transform. The perceptual hash was then extracted from the mean and variance of tilings.

3. Conclusions

Various perceptual hash functions have been developed in the recent years authenticating the multimedia data. However there are drawbacks associated with these schemes like large hash value etc. This paper gives a short survey of image hashing functions used for authenticating images.

References

[1] W. Stalling, Cryptography and Network Security, 4 Edition. vol. 3, Ed. Pearson Education India, 2006.

[2] B. Canova , A Survey of Security Mechanisms to Verify the Integrity and Authenticity of Multimedia-based Data,2011

[3] A Swaminathan, Y.N. Mao, M. Wu, "Robust and secure image hashing," IEEE Trans. Inf. Forensics Security. 1 (2) pp 215–230, 2006

[4] S Xiang, HJ Kim, J Huang, "Histogram-based image hashing scheme robust against geometric deformations". In proceeding of the *9th ACM* Multimedia and Security Workshop (Dallas, Texas, USA), pp. 121–128, 2007.

[5] Y S Choi and J H park, "Image hash generation method using hierarchical histogram" Multimedia Tools and Applications, Multimedia Tools and Applications , Springer, Vol.61, Iss.1, pp.181-194, 2012.

[6] J Fridrich, M Goljan, Robust hash functions for digital watermarking. In the proceeding of IEEE International Conference on Information Technology: Coding Computing (Las Vegas, NV , USA), pp. 178–183,2000.

[7] C.Y.Lin,S.-F.Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," IEEE Transactions on Circuits and Systems for Video Technology 11(2) ,pp. 53–168, 2001.

[8] Q. Sun, S. F Chang "A Robust and Secure media signature scheme for JPEG images," Journal of VLSI Signal Processing , Vol. 41, Issue 3 pp. 305-317,2005.

[9] M.K. Mihcak and R. Venkatesan. "New iterative geometric methods for robust perceptual image hashing". In ACM Workshop Security and Privacy in Digital Rights Management, Philadelphia, PA, Nov. 2001.

[10] V. Monga, M.K. Mihcak, "Robust and secure image hashing via non-negative matrix factorizations," IEEE Trans. Inf. Forensics. 2(3) pp. 376–390, 2007.

[11] R.A. Parrao-Hernandez, M.N. Miyatake and B. M. Kurkoski, "Robust Image Hashing Using Image Normalization and SVD Decomposition," 54rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS) ,2011.

[11] A Swaminathan, Y Mao, M Wu, "Image hashing resilient to geometric and filtering operations." In the proceeding of IEEE Workshop on Multimedia Signal Processing (Siena, Italy), pp. 355–358,2004

[12] C.Y.Lin,S.-F.Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," IEEE Transactions on Circuits and Systems for Video Technology 11(2), pp.53–168,2001

[13] R. Venkatesan, S.M. Koon, M.H. Jakubowski, and P Moulin. "Robust image hashing," In IEEE International Conference Image Processing, Vancouver, BC, Canada, Sep. 2000, pp 664-666, 2000.

[14] V.Monga, M.K.Mihcak, "Robust and secure Image hashing via non-negative matrix factorizations," IEEE Trans. Inf. Forensics Security, vol. 2, no. 3, pp.376– 390, Sep 2007.

[15] Z.Tang, Y.Dai ,X.Zhang, "Perceptual hashing for color images using invariant moments," Applied Mathematics & Information Sciences (2012)643S 650S.

[16] Y. Lei, Y. Wang, J. Huang, "Robust image hash in Radon transform domain for authentication," Signal Processing: Image Communication Vol. 26, pp.280–288,2011

[17] F. Lefebvre, B. Macq, J.D. Legat, "Rash: Radon Soft Hash Algorithm," In Proceedings of the European Signal Processing Conference (EUSIPCO'02), Toulouse, France, September 2002.

[18] Lefebvre, F., Czyz, J., Macq, B, "A robust soft hash algorithm for digital image signature." In Proceedings of the IEEE International Conference on Image Processing. Vol. II, Barcelona,Spain, pp. 495–498,2003

[19] F. Ahmed, M.Y. Siyal, V.U. Abbas, "A secure and robust hash-based scheme for image authentication," Signal Processing Vol. 90 pp. 1456–1470, 2010.

[20] Tang, S. Wang, X. Zhang, W. Wei, S. Su, Robust image hashing for tamper detection using non-negative matrix factorization, Journal of Ubiquitous Convergence and Technology Vol. 2 , pp.18–26, 2008.

[21] S. S. Kozat, M. K. Mihcak and R. Venkatesan, "Robust Hashing via Matrix Invariances," *IEEE Transactions on Image Processing*. Kozat S.S, Venkatesan R, Mihcak M.K., "Robust perceptual image hashing via matrix invariants," International Conference on Image Processing, ICIP '04, vol.5, pp.3443-3446 Vol. 5, 24-27 Oct. 2004

[22] Lin Y, Chang SF " A robust image authentication system distinguishing JPEG compression from malicious manipulation." IEEE Transactions on Circuits System Video Technology, Vol 11(2), pp.153–168, 2001.

[23] Monga V, Evans BL "Perceptual image hashing via feature points: performance evaluation and trade-offs." IEEE Transaction on Image Process Vol. 15(11), pp. 3453–3466.

[24] Zhenjun Tang, Xianquan Zhang, Xuan Dai,Jianzhong Yang, Tianxiu Wu, "Robust Image Hash Function using local color features," Int. J. Electron. Comm.(AEU)67, pp.717-722,2013

[25] Di Wu, Xuebing, Xiamu Niu, "A novel mage hash algorithm resistant to print scan" Signal Processing Vol. 89, pp. 2415-2424,2009.

[26] VC Guo and D Hatzinakos, "Content based image hashing via wavelet and Radon transform," Advances in Multimedia Information Processing PCM. Vol 4810, pp.755-764, 2007

[27] F Ahmed,M Y Siyal and Vali Uddin Abbas, "A Secure and Robust hash-based scheme for image authentication" Signal Processing Vol. 90, pp. 1456-1470,2010

[28] Azhar et. Al "A Robust and secure perceptual hashing system based on a quantization step analysis" Signal processing Volume 28, Issue 8, , pp. 929–948, September 2013.

[29] F.Liu and L.M Chang, "Perceptual image hashing via wave atom transform," Digital Forensic and Watermarking pp. 468-478. 2011

[30] Zhenjun Tang , Xianquan Zhang, Liyan Huang, Yumin Dai, "Robust Image hashing using Ring based Entropies," Signal Processing **,**Vol. 93, Issue 7, pp. 2061–2069, July 2013

[31] Chun-Shien Lu and Chao-Yong Hsu, "Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication," Multimedia Systems Vol.11, Issue 2 , pp. 159-173.

Author Profile

Neelima Arambam, Neelima is working as an Assistant Professor in the Department of Computer Science & Engineering, National Institute of Technology, Nagaland, India. She received her M.Tech degree in Infrmation Technology from Tezpur University (India) in 2012.

Kh. Manglem Singh, Manglem is working as an Associate Professor in the Department of Computer Science & Engineering, National Institute of Technology Manipur. He did PhD from IIT Guwahati.