Detecting similarity in color images based on perceptual image hash algorithm

Nada Hussein M. Ali 1, Marwa Emad Mahdi 1

1University of Baghdad – College of Science – Computer Department
Email: nada_husn@yahoo.com

Abstract. Due to the ever-increasing digitization, the authentication of digital media content is becoming more and more important. Authentication, in general, means deciding whether a digital media is authentic or not, that is, if it matches a given original image. The authentication depends heavily on the type of the digital media, in other words, it is important that every single bit exactly matches the original digital media. Cryptographic hash functions are adequate for such tasks; in this case, a robust hashing is a technology as a change tolerant alternative to cryptographic hashes. Since normal cryptographic hashing methods are error prone to image processing techniques, perceptual hashing is a promising solution to image content authentication. However, conventional image hash algorithms only offer a limited authentication level for overall protection of the content. When intensity components of the color image are used, it is meant that the image is converted to only hashing functions image to produce strong adaptive hashes and that is lead to inadequate recognition abilities.

In this paper, a hash function for color images has been introduced and it is considered robust against global changes inside the image contents. A good discrimination achievement has been obtained since it takes all constituents of the color images into account. Firstly, the input image is resized or re-scaled into a fixed size that is predefined earlier. Secondly, converting the RGB image to HSI and YCbCr color spaces, respectively. The purpose of this process is to extract local color features of the RGB image. Thereafter, the YCbCr image is divided into blocks for computing mean and variance of each block component. Finally, the hash values for both source and attack images are calculated using Euclidian distances formula to measure the difference between them.

Experiments are lead to validate the efficiency of the suggested perceptual robustness hash function. Different image operations; brightness, darkness and rotating; were used to test the robustness of the image hash functions. Besides, different evaluated metrics, such as Mean Square Error (MSE) and Peak to Signal to Noise Ratio (PSNR), were used to measure the performance of the hash function. The obtained results from the perceptual image hash algorithm matched up to 98% between the source image and their corresponding attack image.

Keywords
RGB, YCbCr, Hash function, Features, MSE, PSNR.
1- Introduction

Information security has become one of the main goals in the communication technology and data transmission, since the raw data is exposed to the enormous risk in World Wide Web (WWW) and copyright rules [1]. Perceptual image hashing also called an image hash function that maps the input image to a short string. Many applications applied using hash functions such as: tamper detection [2], authentication of image, image indexing, digital watermarking [3], retrieval of image [4], detection to copy image, reduced-reference quality assessment of image and digital forensics. For example, the editing tools image often used by people, such as ACDSee software and Photoshop for converting and the images are saved in JPEG format with various file names. Consequently, the image may have several copies in the computer. Visual contents of the copies have the same visual contents with the original image, but there are differences in the digital representations from the original one. Image hash function used by people in this case can benefit and to efficiently search for all versions that are similar (including to the image copies and the original image one) to the image from large-scale image database [5].

The classical encryption systems have restrictions in encrypting such as low efficiency, bulky data, and the high correlation between samples and so on [6]. However, the MD5 and the SHA-1 are cryptographic hashing functions, the image hash function does not depend or is not sensitive to a digital representation of the image. The visually identical images produce the same or very similar hash values and it does not matter whether their digital representations are same or not. In general, there are two properties an image hash function must have: (1) Discriminative capability; image hash function must be controlled by one or several keys, (2) Perceptual robustness; the hash function must be robust against the changes that might be occurring against image content such as image JPEG compression and noise removal process. In other words, the value for image hashes for both the source image and its related attack image should be equivalent.

2- Problem Statement

An image is considered as a multimedia object, e.g., the image can have different forms in the digital representations from one type to another, nevertheless they all look the same to the human perception. To authenticate any modification that may be occurring to the images one can use a cryptographic hash function, but it does no good to use it. Therefore, another hash function could be used; also called perceptual hash function; to establish the perceptual quality of multimedia content to verify the integrity of the recipient image.

3- Related Work

Researches have been devoted in the last decade to develop image hash functions with high performance and it is used in several ways as shown in the following points:

- **(Qin et al. (2012))** proposed an algorithm that successfully characterizes the main contents of the original image. In this algorithm, firstly the features are extracted and then a secret key is used for re-permutating the image contents. The obtained results have a good and robust performance for image hashes [7].

- **(Zhao et al. (2013))** suggested a robust hash algorithm for image tamper, forgery, changing and other operations. The proposed algorithm depends on extracting local features inside image blocks which are sensitive against aforementioned operations. The results show the effectiveness of the above algorithm [8].

- **(Wang et al. (2015))** presented a technique for preserving the contents against different image geometric distortions and tampering. The implemented results show a good performance to detect content tamper even in local regions [9].

- **(Freitas et al. (2016))** presented an algorithm for retrieving the image or video contents using a secure watermarking. This algorithm can detect the tampered region and has the capability to restore the missing information [10].

- **(Ur-Rehman and Zivic (2018))** introduced algorithm that is able to identify and balance between the major and minor changes in the modified image. The concluded
results of the proposed algorithm depend on localized region to locate the modification depending on a specified threshold [11].

4- Color Image Domains
Different domains for color images are available as demonstrated in the following:

- RGB Color Model
  The color image captured devices in this model exploit the three main colors in the lights; red, green and blue; depend on how much light is absorbed or reflected to obtain the output color [12].

- HSI Color Model
  The hue, saturation, and luminance are represented in the features of a color image. The hue represents the color classification, e.g., red, green, and blue, and the color form. The saturation or Chroma represents the amount of white in the color. The luminance represents the intensity or the brightness of the color [13].

- YC\textsubscript{b}C\textsubscript{r} Color Model
  YC\textsubscript{b}C\textsubscript{r} Color Model was defined in the ITU-R BT.601 standards of ITU (International Telecommunication Union) that is the mainly used European TV signal which symbolizes the encoding form of non RGB signal. The constituents of the YC\textsubscript{b}C\textsubscript{r} color model are; luminance Y, Chroma C\textsubscript{b} and C\textsubscript{r} components stand for the change between the red and blue value, respectively [14][15]. The YC\textsubscript{b}C\textsubscript{r} model is used for image compression, as well as it can also be used in video representation [16].

5- Perceptual Image Hash and Features
The perceptual image hash function should take into account the variations in the visual area of the image to yield hash values. The resultant hash value is used to recognize images, especially in compressed images. Another application used for perceptual hash is image authentication, where any variation or tamper is identified especially when watermarking information is embedded [17]. Figure 1 represents the main steps to extract the perceptual image hash.
The Proposed Image Hash Function

The perceptual image Hash algorithm’s main phases are listed in the following for comparison between two images:

1. Resize the RGB image: This means that the image may be different in the length or size and their hashes image must have the same length or the same (size) and is intended to resize them to control the size of the image. In this paper the image size used is 128 * 128.

2. Convert the RGB image into two types of color space: HSI and YCbCr.

\[ I = \frac{R + G + B}{3} \]  

\[ S = 1 - \left[ \frac{3}{R + G + B} \right] \times \min(R, G, B) \]  

\[ H = \cos^{-1} \left( \frac{1/2 \left| (R-G) + (R-B) \right|}{\sqrt{(R-G)^2 + (R-B)^2}} \right) \]  

\[ \frac{Y}{C_b} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \end{bmatrix} \]  

3. Feature extraction by computing mean and variance to each component in the HIS and YCbCr.

\[ m = \frac{1}{b^2} \sum_{j=1}^{b^2} B_i(j) \]  

\[ u = \frac{1}{b^2-1} \sum_{j=1}^{b^2} [B_i(j) - m]^2 \]  

\[ V = [m_H, u_H, m_S, u_S, m_V, u_V, m_{C_b}, u_{C_b}, m_{C_r}, u_{C_r}]^T \]  

\[ V = [V_1, V_2, \ldots, V_N] \]  

4. Find the hash value using the Euclidian distance metric. The hash value needs to be as short as possible. This process is started by compressing the feature matrix V. The first conducted is data normalization. Find the row of V by supposing \( r = [r(1), r(2), \ldots, r(N)] \). Then, convert the row of V to P by the following formulas.

\[ P(i) = \frac{r(i) - \mu}{\sigma} \]  

\[ \mu = \frac{1}{N} \sum_{i=1}^{N} r(i) \]  

\[ \sigma = \frac{\sqrt{\sum_{i=1}^{N} [r(i) - \mu]^2}}{N-1} \]  

\[ h = [h(1), h(2), h(3), \ldots, h(N)] \]  

5. Compute the Similarity measurements between the two images [19].

\[ D = \sqrt{\sum_{i=1}^{N} |h_1(i) - h_2(i)|^2} \]  

7- Results Analysis and Discussion

Three RGB test images of size 128 × 128 pixels will be used to test the image hash algorithm. Different processes are implemented on these images to compute their hashes and it is compared with the corresponding image attack. Besides, various affect operations are applied to the tested images using Photoshop software (2018) to measure the similarity between the attack images and the source image that they belong to, as demonstrated later. The source image is changed by three kinds of attacks that are altering the source image: Lina, Pepper and Baboon as shown in Figure (2), Figure (3) and Figure (4) as clarified in Table (1), Table
The brightness and darkness are mean operations representing increase and decrease in the light intensity, respectively, on the source image. While the rotation attack only rearranges the pixel position in the image rather than change the color values for the source image.

![Source image](image1)

![Attack image -1](image2)

![Attack image -2](image3)

![Attack image -3](image4)

**Figure 2: Lina's attack images in RGB**

**Table 1: Attack types at Lina's image**

| Name image       | Size image | Operation | Percentage change |
|------------------|------------|-----------|-------------------|
| Source image     | 128*128    | No operation | 0%                |
| Attack image-1   | 128*128    | Brightness | 65%               |
| Attack image-2   | 128*128    | Darkness   | 50%               |
| Attack image-3   | 128*128    | Rotation   | 100%              |

![Source image](image1)

![Attack image -1](image2)

![Attack image -2](image3)

![Attack image -3](image4)

**Figure 3 Pepper’s attack images in RGB**

**Table 2. Attack types at Pepper's image**

| Name image       | Size image | Operation | Percentage change |
|------------------|------------|-----------|-------------------|
| Source image     | 128*128    | No operation | 0%                |
| Attack image-1   | 128*128    | Brightness | 100%              |
| Attack image-2   | 128*128    | Darkness   | 100%              |
| Attack image-3   | 128*128    | Rotation   | 100%              |

![Source image](image1)

![Attack image -1](image2)

![Attack image -2](image3)

![Attack image -3](image4)

**Figure 4. Baboon's attack images in RGB**
Table 3. Attack types at Baboon's image

| Name image  | Size image | Operation  | Percentage change |
|-------------|------------|------------|-------------------|
| Source image| 128*128    | No operation | 0%                |
| Attack image-1 | 128*128  | Brightness | 54%               |
| Attack image-2 | 128*128  | Darkness   | 85%               |
| Attack image -3 | 128*128 | Rotation   | 100%              |

This stage represents the similarity measurements between the source and attack images. Euclidean distance metric is used to amount the likeness between the source and attack images. The similarity depends on the threshold value that depends on the user’s desire. The selected threshold is 3, 5 or 7, this different value achieves a reasonable match between an image and their relative attack image. Table (4), Table (5) and Table (6) present the output results to measure the similarity using.

Table 4. Similarity Measurement (Distance) of Lina's image for $T = 5$

| Name Image   | Similarity Measurement (D) | Status |
|--------------|----------------------------|--------|
| Source image | 0                          | Accept |
| Attack image-1 | 2.0998                | Accept |
| Attack image-2 | 1.5637                   | Accept |
| Attack image-3 | 2.8834                   | Accept |

Table 5. Similarity measurement (Distance) of Pepper's image $T = 5$

| Name Image   | Similarity measurement(d) | Status |
|--------------|---------------------------|--------|
| Source image | 44.1404                    | Refusal|
| Attack image-1 | 52.8972                | Refusal|
| Attack image-2 | 2.8683                  | Accept |

Table 6. Similarity measurement (Distance) of Baboon's image with $T = 5$

| Name Image   | Similarity measurement(d) | Status |
|--------------|---------------------------|--------|
| Source image | 3.9794                     | Accept |
| Attack image-1 | 3.9794                 | Accept |
| Attack image-2 | 4.9590                  | Accept |
| Attack image-3 | 2.9905                  | Accept |

Consequently, the attack images are considered the same as the original image as demonstrated graphically in Figures (1), Figures (2) and Figures (3), respectively.

Figures 5. Similarity distance to Lina image with $T = 5$. 
Figure 6. Similarity distance to Pepper image with $T = 5$

Figure 7. Similarity distance to Baboon image with $T=5$

8- MSE and PSNR Performance Metrics

The MSE and PSNR metrics presents a good indicator for the implementing of perceptual image hash algorithm for identifying the image similarity. Although, a line between the two images can be extracted, it can be used (MSE) and (PSNR) for the Lina, Pepper, Baboon images. The threshold value is used for PSNR is 40. If the PNSR value is smaller than 40, the percentages are refusal and if they are greater than 40, the percentages extracted are acceptable as shown in Table (7), Table (8) and Table (9), respectively.
Table 7. MSE and PSNR for Lina’s image’s image

| Image Name      | Operation | (MSE)      | (PSNR)     | Stats  |
|-----------------|-----------|------------|------------|--------|
| Source Image    | No operation | 0         | $\infty$   | Refusal|
| Attack Image-1  | Brightness      | 3.0860 e + 03 | 7.2502    | Refusal|
| Attack Image-2  | Darkness        | 6.1986 e + 03 | 4.2213    | Refusal|
| Attack Image-3  | Rotation        | 4.3739 e + 03 | 5.7356    | Refusal|

Table 8. MSE and PSNR for Pepper's image

| Image Name      | Operation | (MSE)      | (PSNR)     | Stats  |
|-----------------|-----------|------------|------------|--------|
| Source Image    | No operation | 0         | $\infty$   | Refusal|
| Attack Image-1  | Brightness      | 537.2688   | 14.8423    | Refusal|
| Attack Image-2  | Darkness        | 1.8003 e + 03 | 9.5908    | Refusal|
| Attack Image-3  | Rotation        | 4.9078 e + 03 | 5.2353    | Refusal|

Table 9. MSE and PSNR for Baboon 's image

| Image Name      | Operation | (MSE)      | (PSNR)     | Stats  |
|-----------------|-----------|------------|------------|--------|
| Source Image    | No operation | 0         | $\infty$   | Refusal|
| Attack Image-1  | Brightness      | 846.6935  | 12.8669    | Refusal|
| Attack Image-2  | Darkness        | 4.7352 e + 03 | 5.3908    | Refusal|
| Attack Image-3  | Rotation        | 4.7353 e + 03 | 5.3908    | Refusal|

9- Conclusions

The subsequent observations for implementing the perceptual image hash algorithm conclude that the feature extraction, the hue, Chroma and Intensity are all considered of the block to the image. To make discriminating to the image hash function ensures that it fully indicates the color characteristic of the extracted features from the image. The earlier experiment operations show a robustness of the algorithm by using different content-preserving manipulations. The type of attack mainly influences the similarity performance, especially if the attack changes the color frequency distribution over the whole image as shown in Pepper image.

10- Acknowledgement

I would like to express my deepest thanks to the person who helped me in this project and support me and he has been with me in every minute many thanks to the electronic and communications engineer. (Eng. Karar Ali Kareem).

References

[1] Nada Hussein M Ali, Abdul Monem S Rahma and Abeer Salim Jamil 2015 Text hiding in Color Images using the Secret Key Transformation Function in GF(2^n) *Iraqi Journal of Science*, Vol 56, No.4B, pp 3240-3245
[2] Rawat S and Raman B 2011 A chaotic system based fragile watermarking scheme for image tamper detection AEU-International Journal of Electronics and Communications, 65(10), pp 840-847

[3] Qin C, Chang C C and Chen P Y 2012 Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism. Signal Processing, 92(4) pp1137-1150

[4] Tang Z J, Zhang X Q, Dai Y M and Lan W W 2013 Perceptual image hashing using local entropies and DWT The Imaging Science Journal, 61(2), pp 241-251

[5] Ground Truth Database. Available from: http://www.cs.washington.edu/research/imagedatabase/groundtruth/ [accessed 08.05.08].

[6] Yussra Majid Hameed, Nada Hussein M Ali 2019 Enhanced RC5 Key Schedule Using One-Dimensional Cellular Automata for Audio File Encryption Iraqi Journal of Science, Volume 60, No.2, pp 388-401

[7] Sumalatha L, Krishna V V, Kumar V V and Rajahmundry A P 2012 Local content based image authentication for tamper localization. International Journal of Image, Graphics and Signal Processing 4(9), pp 30-36

[8] Qin C, Chang C C and Tsou P L 2012 Perceptual image hashing based on the error diffusion halftone mechanism. International Journal of Innovative Computing, Information and Control, 8(9), pp 6161-6172

[9] Zhao Y, Wang S, Zhang X and Yao H 2013 Robust hashing for image authentication using Zernike moments and local features. IEEE transactions on information forensics and security, 8(1), pp 55-63

[10] Wang X, Pang K, Zhou X, Zhou Y, Li L and Xue J 2015 A visual model-based perceptual image hash for content authentication. IEEE Transactions on Information Forensics and Security, 10(7) pp 1336-1349

[11] Freitas P G, Rigoni R and Farias M C 2016 Secure self-recovery watermarking scheme for error concealment and tampering detection Journal of the Brazilian Computer Society, 22(1), p 5

[12] Ur-Rehman O and Zivic N 2018 Authentication with Block Level Error Localization. In Noise Tolerant Data Authentication for Wireless Communication pp 61-70 Springer, Cham.

[13] Plataniotis K N and Venetsanopoulos A N 2013 Color image processing and applications. Springer Science & Business Media

[14] Shih T Y 1995 The reversibility of six geometric color spaces Photogrammetric Engineering and Remote Sensing, 61(10), pp 1223-1232

[15] Patil N K, Yadahalli R M and Pujari J 2011 Comparison between HSV and YCbCr color model color-texture based classification of the food grains. International Journal of Computer Applications, 34(4), pp 51-57

[16] Wen C Y and Chou C M 2004 Color image models and its applications to document examination. Forensic Science Journal, 3(1), pp 23-32

[17] Rawat S and Raman B 2011 A chaotic system based fragile watermarking scheme for image tamper detection AEU-International Journal of Electronics and Communications, 65(10) pp 840-847

[18] Khelifi F and Jiang J 2010 Perceptual image hashing based on virtual watermark detection. IEEE Transactions on Image Processing, 19(4) pp 981-994

[19] Tang Z, Zhang X, Dai X, Yang J and Wu T 2013 Robust image hash function using local color features. AEU-International Journal of Electronics and Communications, 67(8) pp 717-722