PROPOSED IMAGE HASHING SCHEME AND AN IMAGE AUTHENTICATION METHOD USING A HASH CONSTRUCTION

Eva Edward[*]

J.A.Jevin[**]

Abstract—Image authentication verifies the originality of an image by detecting malicious manipulations. Existing methods for image authentication treat all types of manipulation equally However, some applications demand techniques that can distinguish acceptable manipulations from malicious ones. In this paper, we describe an effective technique for image authentication which can prevent malicious manipulations. We use a combination of both the global and local features in order to form a hash sequence. The global features depend on Zernike moments to represent the coarseness and contrast of an image, whereas the local features depict the position and texture information of the image. A hash is constructed and the features are extracted by applying secret keys . The hash is sensitive to malicious tampering and, therefore, applicable to image authentication. The test image is compared with that of a trusted image to judge if the image is fake or tampered.Theoretical and experimental results indicate that this technique is effective for image authentication.

Index Terms—**Image authentication, malicious manipulations, global features, local features, secret keys.**

[*] PG Scholar, SCAD Engineering College, Tirunelveli.

[**] Asst.Professor, FX Engineering College, Tirunelveli.

I. INTRODUCTION

In this work, we focus on image authentication. Image authentication techniques protect images from malicious manipulation at every stage of transmission and storage. Reliable image authentication technology must be able to protect an image from the time it was first produced until the final stage of use. Image hashing is a technique that extracts a short sequence from the image to represent its contents, and therefore can be used

for image authentication. If the image is maliciously modified, the hash must be changed significantly. The purpose of manipulations may represent transformations and attacks. The former are usually acceptable, and the latter, unacceptable. There are two kinds of representation transformations:

1)*Format transformation and Lossless Compression*. Disregarding the noise caused by the precision limitation during computation, pixel values in an image are not changed after these manipulations.

2)*An Application specific transformation*. Some applications may require the lossy compression in order to satisfy the resource constraints on bandwidth or storage. Some applications may also need to enhance the image quality, crop the image, change the size, or perform some other operations.

A common aspect of these manipulations is that they change the pixel values, which results in different levels of visual distortion on the image. Among all the application specific transformations, the lossy compression is one of the operations that tries to minimize the visual distortion. Attacks, or malicious manipulations, change the image to a new one which carries a different visual meaning to the image observer .

It should also be unique in the sense that different images have significantly different hash values and secure so that any unauthorized party cannot break the key and coin the hash. To meet all the requirements simultaneously, especially perceptual robustness and sensitivity to tampering, is a challenging task.

In practice, when an image is sent to a user, a possible solution to prove the authenticity is to generate a *hash value* and send it securely to the user. The hash value is a compact string – an abstract of the content. A user can re-generate a hash value from the received image, and compare it with the original hash value. If they match, the content is considered as authentic. In order to allow incidental distortion, the hash value must possess some *robustness*. Therefore, a new generation of hash algorithms has emerged, called robust or perceptual hash (PH) algorithms. A perceptual hash value is computed from robust and distinctive image features. It must fairly represent the corresponding content. A PH algorithm typically possesses the following properties:

1)Compactness – the hash value is compact

2)Robustness – hash computation is insensitive to a certain range of distortion to the input image

Content authentication can be achieved on different levels. On a low security level, we only need to know if the majority of the content has been tampered; on a high security level, we need to know if any local area has been tampered. The accuracy depends not only on the extracted features, but also the way of watermark embedding, both of which can be categorized as *global* or *local*. Global features are extracted from a global perspective. They are usually very compact, but only represent the content on a macro level. Local features are extracted from local regions. They help with tamper location, but contain much more information. In global embedding, embedding regions do not correspond to image regions. This allows a large payload, but tamper location does not work. In local embedding, there is a correspondence between embedding regions and image regions. This facilitates tamper location, but suffers from a limited payload size.

In the present paper, we propose a method combining advantages of both global and local features. The objective is to provide a reasonably short image hash with good performance, i.e., being perceptually robust while capable of detecting and locating content forgery. We use Zernike moments of the luminance/ chrominance components to reflect the image's global characteristics, and extract local texture features from salient regions in the image to represent contents in the corresponding areas. Distance metrics indicating the degree of similarity between two hashes are defined to measure the hash performance. The method can be used to locate tampered areas and tell the nature of tampering, e.g., replacement of objects or abnormal modification of colors.

Compared with some other methods using global features or local features alone, the proposed method has better overall performance in major specifications, especially the ability of distinguishing regional tampering from content-preserving processing. Basically, we divide an image into blocks, compute a hash value from each block and embed it into the block. This way, we can tell whether the image has been tampered by verifying the block hash values. The proposed system exhibits good performance under extensive tests, and significantly outperforms the state-of-the-art algorithm . The results justify the validity of the design and serve as a reference for research and practice in this field. The rest of the work is organized as follows. Section **II** describes the details of the existing system, Section **III** describes the details of the proposed image construction method using a hash sequence, Section **IV** shows the experimental results and analysis. Section **V** concludes the work.

II.EXISTING SYSTEM

Xiang *et al* [2] proposed a robust image hash algorithm using the invariance of the image histogram shape to geometric deformations. Robustness and uniqueness of the proposed hash function are investigated in detail by representing the histogram shape as the relative relations in the number of pixels among groups of two different bins. It is found from extensive testing that the histogram-based hash function is robust to geometric attacks, but cannot distinguish images with similar histograms but different contents

Tang *et al.* [3] developed a global method using nonnegative matrix factorization (NMF). The image is first converted into a fixed-sized pixel array. A secondary image is obtained by rearranging pixels and applying NMF to produce a feature-bearing coefficient matrix, which is then coarsely quantized. The obtained binary string is scrambled to generate the image hash.

In [10], a wavelet-based image hashing method is developed. The input image is partitioned into non overlapping blocks, and the pixels of each block are modulated using a permutation sequence. The image undergoes pixel shuffling and then wavelet transform. The sub-band wavelet coefficients are used to form an intermediate hash, which is permuted again to generate the hash sequence. This method is robust to most content-preserving operations and can detect tampered areas.

Besides the fore mentioned methods, a concept of forensic hash for information assurance is proposed in [12]. A compact forensic hash is based on Radon transform and the scale space theory, and used as side information in image authentication. It is aimed to address a wider range of issues than simply deciding whether an image is a fake. These include telling the history of image manipulations and estimating parameters of geometrical transformation. The geometrical transformation parameters allow image registration to be achieved without resorting to the original image so that the forged areas can be located.

Lv *et al.* [11] propose a SIFT-Harris detector to identify the most stable SIFT key points under various content-preserving operations. The extracted local features are embedded into shape-context-based descriptors to generate an image hash. The method is robust against geometric attacks and can be used to detect image tampering. The performance is degraded when the detected key points from the test image do not coincide with that of the original.

Previous schemes are either based on global [2]–[5] or local [6]–[11] features. Global features are generally short but insensitive to changes of small areas in the image, while local features can reflect regional modifications but usually produce longer hashes.

A. Saliency Region Detection

A salient region in an image is one that attracts visual attention. Information in an image can be viewed as a sum of two parts: that of innovation and that of prior knowledge. The former is new and the latter redundant. The information of saliency is obtained when the redundant part is removed.

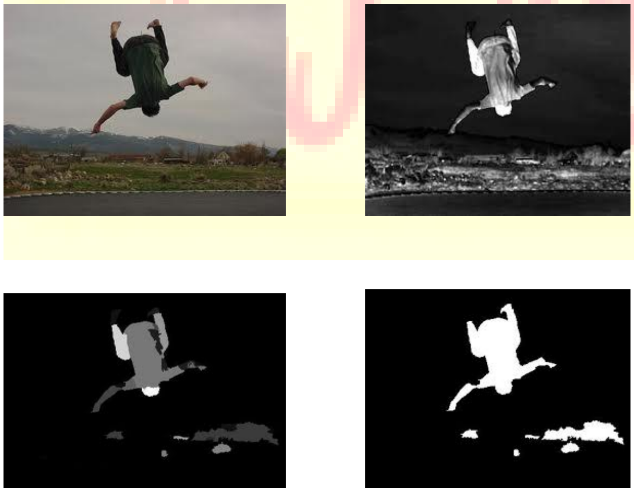


Fig.1. Saliency region detection

Log spectrum of an image, L(f) , is used to represent general information of the image. Because log spectra of different images are similar, there exists redundant information in L(f) . Let A(f) denote the redundant information defined as convolution between L(f)and l x l an low-pass kernel h_l:

$$A(f) = h_l * L(f)$$

B. Texture Features

Texture is an important feature to human visual perception. Texture is defined as an attribute of a field having no components that appear enumerable. The phase relations between the components are thus not apparent.The six texture features relating to visual perception are coarseness, contrast, directionality, likeness, regularity and roughness. In this work, we use coarseness C1 and contrast C2 as defined below, to describe the texture properties. Contrast describes the degree of image brightness variation, calculated from variance σ^2 and the fourth-order moment μ_4 of the gray values within the region:

$$C_2 = \sigma^2 \mu_4^{-4}$$

III.PROPOSED SYSTEM

Fig 2 defines a framework for an image hashing scheme and the procedure of image authentication using the hash. The hash is formed from Zernike moments to represent global properties of the image, and the texture features in salient regions to reflect local properties.

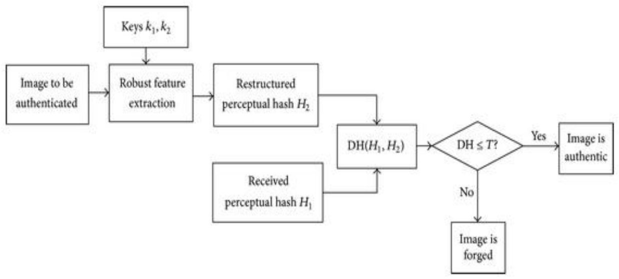
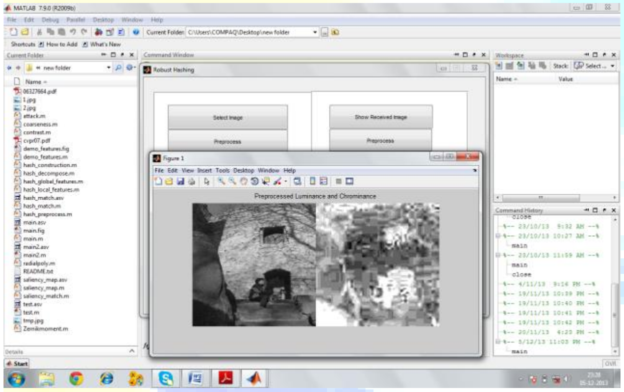

Keys k_1, k_2
Image to be authenticated
Robust feature extraction
Restructured perceptual hash H_2
DH(H_1, H_2)
DH ≤ T?
Yes
Image is authentic
No
Image is forged
Received perceptual hash H_1

Fig.2. Framework of the proposed Image hash

A. Image Hash Construction

The proposed image hash generation procedure includes the following steps,

1) Preprocessing: The image is first rescaled to a fixed size F x F with bilinear interpolation, and converted from RGB to the YCbCr representation as shown in Fig.3. Y and |Cb-Cr| are used as luminance and chrominance components of the image to generate the hash. The aim of preprocessing is an improvement of the image data that suppresses unwanted distortions and enhances some image features important for further processing.



```
close
%-- 23/10/13  9:32 AM --%
%-- 23/10/13 10:27 AM --%
  main
%-- 23/10/13 11:59 AM --%
  main
  close
%-- 4/11/13  9:16 PM --%
%-- 19/11/13 10:39 PM --%
%-- 19/11/13 10:40 PM --%
%-- 19/11/13 10:41 PM --%
%-- 19/11/13 10:42 PM --%
%-- 20/11/13  4:23 PM --%
%-- 5/12/13 11:03 PM --%
  main
```

Fig.3.Preprocessing an image

2) Global Feature Extraction: Zernike moments of Y and |Cb-Cr| are calculated. Because shape features can be obtained from a small number of low frequency coefficients, the order n does not need to be large. We choose n=5. Further, since $Z_{n,-m} = Z_{n,m}$, only $Z_{n,m}(m \geq 0)$ is needed.

Order *n*	Zernike moments	Number of moments
1	$Z_{1,1}$	1
2	$Z_{2,0}$, $Z_{2,2}$	2
3	$Z_{3,1}$, $Z_{3,3}$	2
4	$Z_{4,0}$, $Z_{4,2}$, $Z_{4,4}$	3
5	$Z_{5,1}$, $Z_{5,3}$, $Z_{5,5}$	3

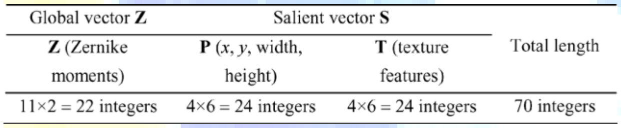
Table I. Zernike moments of order 5

Table I lists the Zernike moment features from order 1 to order 5. Thus we have 11 x 2 = 22 Zernike moments in total. Magnitudes of the Zernike moments are rounded and used to form a global vector, . Each element in is no more than 255.

A secret Key K1 is used to randomly generate a row vector X1 with 22 random integers in [0, 255]. The encrypted global vector Z is obtained a $\mathbf{Z} = [(\mathbf{Z}' + \mathbf{X}_1) \bmod 256]$.

3) Local Feature Extraction: K largest salient regions are detected from the luminance image Y. The coordinates of top left corner, and width/height of each circumscribed rectangle are used to form a K-element vector $\mathbf{p}^{(k)}$(k=1,.....,K) , representing the position and size of each salient region.

4) Hash Construction: The global and salient local vectors are concatenated to form an intermediate hash, namely $\mathrm{H}' =$ [Z S] , which is then pseudo-randomly scrambled based on a secret key K3 to produce the final hash sequence H.



Global vector **Z**	Salient vector **S**		Total length
Z (Zernike moments)	**P** (*x*, *y*, width, height)	**T** (texture features)	
11×2 = 22 integers	4×6 = 24 integers	4×6 = 24 integers	70 integers

Table II. Constitution of image hash

Table II gives the constitution of an image hash of 70 integers. Since all integers are in the range of [0, 255], the hash is 70 x 8 = 560 bits long.

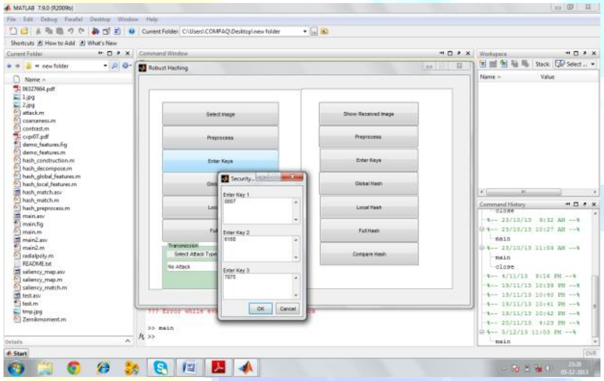
B. Image Authentication

In image authentication, the hash of a trusted image, H0, is available and called the reference hash. The hash of a received image to be tested, H1, is extracted using the above method. These two hashes are compared to determine whether the test image has the same contents as the trusted one or has been maliciously tampered, or is simply a different image. Here, two images having the same contents (visual appearance) do not need to have identical pixel values. One of them, or both, may have been modified in normal image processing such as contrast enhancement and lossy compression.

In this case, we say the two images are perceptually the same, or *similar*. The image authentication process is performed in the following way.

1) Feature Extraction: Pass the test image through the steps as described in Section A to obtain the intermediate hash without encryption, namely $\mathbf{H}'_1 = [\mathbf{Z}_1\ \mathbf{P}_1\ \mathbf{T}_1]$.

2) Hash Decomposition: With the secret keys K1, K2 and K3 , restore the intermediate hash from the reference hash to obtain $\mathbf{H}'_0 = [\mathbf{Z}_0\ \mathbf{P}_0\ \mathbf{T}_0]$, , which is a concatenated feature sequence of the trusted image. Decompose it into global and local features.



Enter Key 1
8807

Enter Key 2
6168

Enter Key 3
7875

```
??? Error while ev
>> main
>>
```

Fig.4.Key generation for authentication

3) Salient Region Matching: Check if the salient regions

found in P1 of the test image match those in P0 of the trusted image. If the matched areas of a pair of regions are large enough, the two regions are considered as being matched. Reshuffle the texture vectors by moving the matched components in each of the texture vector pair to the left-most call them T0 and T1 .

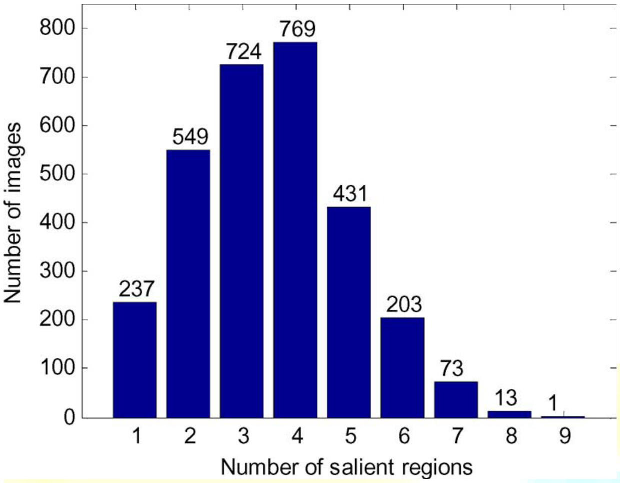


Fig.5. Statistics of salient regions based on 1000 images

4) Distance Calculation and Judgment: We use a distance between hashes of an image pair as a metric to judge similarity/ dissimilarity of the two images. To define the hash distance, a feature vector V is formed by concatenating the global feature vector Z and the reshuffled texture feature vector T , namely . The vector P does not contribute to the distance calculation but will be used to locate forged regions. The hash distance between the test image and the reference is the Euclidean distance between V0 and V1 :

$$D = \| V1 – V0 \|$$

For a pair of similar images, texture features in the corresponding salient regions are close to each other. However, since no currently available method of saliency detection is perfect, the salient regions obtained from an image after content-preserving processing may not always precisely match that of the original. If this happens, difference between the test image and the original will be exaggerated. In practice, the global structure of an image represented by Zernike moments is sufficient to distinguish similar from dissimilar. To minimize the adverse influence of saliency detection inaccuracy, we omit T in calculating the hash distance for similar images:

$$D \approx \|\mathbf{Z}_1 - \mathbf{Z}_0\| \triangleq D_G$$

C. Determination of Thresholds

To determine a threshold for differentiating two sets of data, A and B, we need to know the probability distribution functions (PDF) of samples taken from these data sets. The chi-square test [20] is used for the purpose. Assume that the data satisfy one of several common distributions: Poisson, lognormal, and normal, and apply the chi-square test to find which is the closest. The

Statistic χ^2 is calculated as

$$\chi^2 = \sum_{i=0}^{L} \frac{(v_i - vp_i - 0.5)^2}{vp_i}$$

D. Forgery Classification and Localization

Having found that a test image is a fake, the next job is to locate the forged region and tell the nature of forgery. Four types of image forgery can be identified: removal, insertion and replacement of objects, and unusual color changes. Forgery classification and localization are performed as follows, and schematically illustrated.

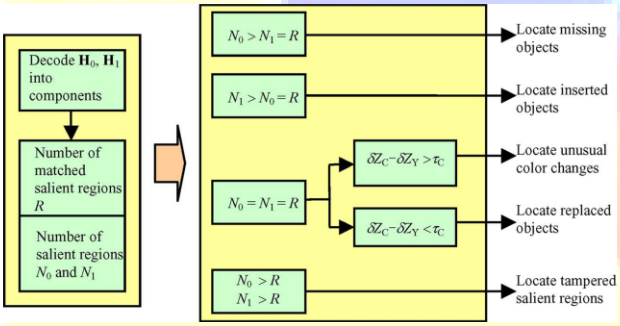

Decode $\mathbf{H}_0$, $\mathbf{H}_1$ into components
Number of matched salient regions R
Number of salient regions N_0 and N_1
$N_0 > N_1 = R$
$N_1 > N_0 = R$
$N_0 = N_1 = R$
$\delta Z_C - \delta Z_Y > \tau_C$
$\delta Z_C - \delta Z_Y < \tau_C$
$N_0 > R$
$N_1 > R$
Locate missing objects
Locate inserted objects
Locate unusual color changes
Locate replaced objects
Locate tampered salient regions

Fig.6.Forgery classification and Localization

Decode H0 and H1 into components representing global and local features, and find the number of matched salient regions R and the numbers of salient regions in the reference and test images, N0 and N1 .

1) If N0 > N1 = R , some objects have been removed from the received test image. Positions of the missing objects are located by comparing the saliency indices.

2) If N1 > N0 = R , the test image contains some additional objects whose positions are located by comparing the saliency indices.

3) If N1 = N0 = R , check the luminance and chrominance components in the Zernike moments and calculate the following distances:

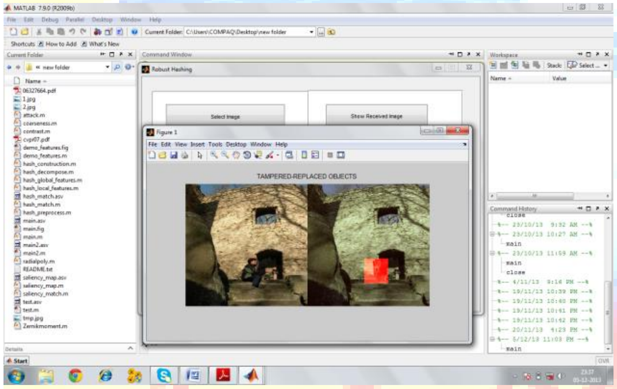
$$\delta Z_C = \|\mathbf{Z}_{C1} - \mathbf{Z}_{C0}\|, \quad \delta Z_Y = \|\mathbf{Z}_{Y1} - \mathbf{Z}_{Y0}\|$$

4) If N1 = N0 = R and $(\delta Z_C - \delta Z_Y)$ is less than , the test image contains replaced objects because in this case luminance changes are dominant.

5) If N0 > R and N1 > R , some of the salient regions are not matched. Mark the mismatched salient regions in the test image as being tampered.

IV.EXPERIMENTAL RESULTS AND ANALYSIS

A. Forgery Detection Capability



```
close
%-- 23/10/13  9:32 AM --%
%-- 23/10/13 10:27 AM --%
main
%-- 23/10/13 11:59 AM --%
main
close
%-- 4/11/13  9:16 PM --%
%-- 19/11/13 10:39 PM --%
%-- 19/11/13 10:40 PM --%
%-- 19/11/13 10:41 PM --%
%-- 19/11/13 10:42 PM --%
%-- 20/11/13  4:23 PM --%
%-- 5/12/13 11:03 PM --%
main
```

Fig.7.Detecting a forged image

As shown in Fig. 7, the proposed method can detect a tampered image. A qualitative comparison between the proposed method and [3], [6], [7], and [10] shows good overall performance of the proposed method, which generates hashes with the third shortest length.Cropping that changes the image's geometrical centre will cause significant differences in the Zernike moments, and large-angle rotation will affect the saliency-related

local features. These are the limitations of the method. Nonetheless, removal of a few lines (no more than 2% of the image width/height) and small-angle rotation are tolerable as will be shown in the ROC performance below. As far as image rotation is concerned, a picture will clearly appear abnormal from the photographical point of view if it is rotated by, say, more than 5 . Such rotation may be viewed as a destructive modification,

viz., tampering.

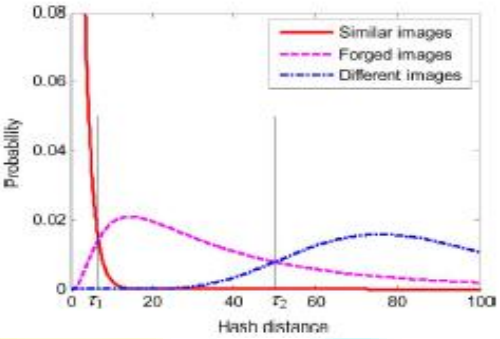
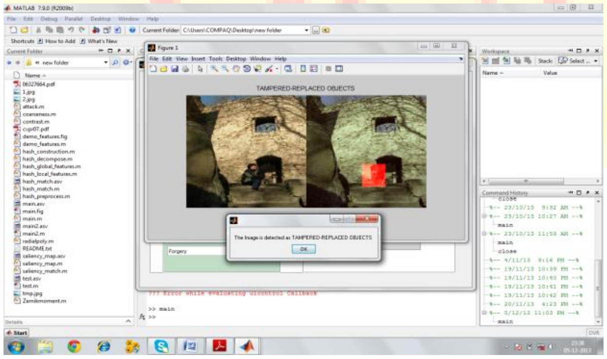


Fig.8.Distribution of test images and trusted images

B. Forgery Localization

We have tested 150 image pairs, with the original images

downloaded from the Internet and the forged ones produced manually using Photoshop. The forged images are all correctly detected. Without considering forgery classification, the success rate of forgery localization is 96%. The success rate drops to 87% when both localization and classification are considered. The latter is lower because saliency detection is imperfect so that accuracy of forgery classification may be affected.



The Image is detected as TAMPERED-REPLACED OBJECTS

```
??? Error while evaluating uicontrol Callback

>> main
>>
```

```
close
%-- 23/10/13  9:32 AM --%
%-- 23/10/13 10:27 AM --%
main
%-- 23/10/13 11:58 AM --%
main
close
%-- 4/11/13  9:16 PM --%
%-- 19/11/13 10:39 PM --%
%-- 19/11/13 10:40 PM --%
%-- 19/11/13 10:41 PM --%
%-- 19/11/13 10:42 PM --%
%-- 20/11/13  4:23 PM --%
%-- 5/12/13 11:03 PM --%
main
```

Fig.9.Locating a forged image

C. Computation Complexity

To compare computation complexity of the different methods, we consider average time consumed in calculating image hashes on a desktop computer with Dual Core 2.8-GHz CPU and 2 GB RAM, running Matlab.

$$P_{\mathrm{FN}} = \frac{\text{Number of forged images judged as natural images}}{\text{Total number of forged images}}$$

$$P_{\mathrm{FP}} = \frac{\text{Number of natural images judged as forged images}}{\text{Total number of natural images}}$$

The average time of the proposed method is 2.7 s, and those of [3] and [7] are 2.98 s and 1.43 s respectively. These are not significantly different.We may conjecture that the proposed method does not take much time,and the major computation load is in determining the distance threshold by considering various image processing operations.

V. CONCLUSION

In this paper, an image hashing method is developed using both the global and local features. The global features are based on Zernike moments taking into account the coarseness and contrast of an image and the local features include position and texture information of salient regions in the image. The global and salient local vectors are concatenated to form an intermediate hash. We use the distance between hashes of an image pair as a metric to judge similarity and dissimilarity of the two images. The proposed scheme has a reasonably short hash length and good overall performance.

The method described in this paper is aimed at image authentication. The hash can be used to differentiate similar, forged, and different images. At the same time, it can also identify the type of forgery and locate fake regions containing salient contents. In the near future one can intend to find the features that better represent the image contents, so as to enhance the detection of image forgery even in relatively smaller areas. A better saliency detection procedure can also be applied into the algorithm for improved performance.

REFERENCES

[1] V. Monga, A. Banerjee, and B. L. Evans, "A clustering based approach to perceptual image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 68–79, Mar. 2006.

[2] S. Xiang, H. J. Kim, and J. Huang, "Histogram-based image hashing scheme robust against geometric deformations," in *Proc. ACM Multimedia and Security Workshop*, New York, 2007, pp. 121–128.

[3] Z. Tang, S.Wang,X. Zhang, W.Wei, and S. Su, "Robust image hashing for tamper detection using non-negative matrix factorization," *J. Ubiquitous Convergence Technol.*, vol. 2, no. 1, pp. 18–26, May 2008.

[4] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230, Jun. 2006.

[5] Y. Lei, Y.Wang, and J. Huang, "Robust image hash in Radon transform domain for authentication," *Signal Process.: Image Commun.*, vol. 26, no. 6, pp. 280–288, 2011.

[6] F. Khelifi and J. Jiang, "Perceptual image hashing based on virtual watermark detection," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp.
981–994, Apr. 2010.

[7] V. Monga and 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.

[8] K. Fouad and J. Jianmin, "Analysis of the security of perceptual image hashing based on non-negative matrix factorization," *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 43–46, Jan. 2010.

[9] Z. Tang, S. Wang, X. Zhang, W. Wei, and Y. Zhao, "Lexicographical framework for image hashing with implementation based on DCT and NMF," *Multimedia Tools Applicat.*, vol. 52, no. 2–3, pp. 325–345, 2011.

[10] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A secure and robust hashbased scheme for image authentication," *Signal Process.*, vol. 90, no. 5, pp. 1456–1470, 2010.

[11] X. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1081–1093, Jun. 2012.

[12] W. Lu, A. L. Varna, and M. Wu, "Forensic hash for multimedia information," in *Proc. SPIE,Media Forensics and Security II*, San Jose, CA, Jan. 2010, 7541.

[13] W. Lu and M.Wu, "Multimedia forensic hash based on visual words," in *Proc. IEEE Conf. on Image Processing*, Hong Kong, 2010, pp. 989–992.

[14] H. Lin, J. Si, and G. P. Abousleman, "Orthogonal rotation-invariant moments for digital image processing," *IEEE Trans. Image Process.*, vol. 17, no. 3, pp. 272–282, Jan. 2008.

[15] S. Li, M. C. Lee, and C. M. Pun, "Complex Zernike moments features for shape-based image retrieval," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 1, pp. 227–237, Jan. 2009.

[16] Z. Chen and S. K. Sun, "A Zernike moment phase based descriptor for local image representation and matching," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 205–219, Jan. 2010.

[17] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*,
Minneapolis, MN, 2007, pp. 1–8.

Author's Profile:

Mrs.Eva Edward completed her Bachelor of Technology(Information Technology) in Jeya Engineering college,Chennai in 2012 and pursuing her Masters in Engineering(Computer science) in SCAD Engineering College, Tirunelveli and would be awarded the degree in 2014.

Her interests in research includes Image processing and medical imaging.

Mr.Jevin completed his Bachelor of Engineering(Computer science) in KSR college of Engineering and Technology,Tiruchengode in 2006 and completed his Masters in Engineering(Computer science) in University Department, Anna university, Tirunelveli in 2012

He is working as an Assistant Professor in Francis Xavier Engineering College, Tirunelveli from 2013.

His research interests includes Medical imaging and Cyber security.