BLIND COPY MOVE IMAGE FORGERY DETECTION USING DYADIC UNDECIMATED WAVELET TRANSFORM

Ghulam Muhammad[1, 3], Muhammad Hussain[1], Khalid Khawaji[1], and George Bebis[1, 2]

[1]College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[2]Dept. of Computer Science and Engineering, University of Nevada, Reno, USA
[3]Email: ghulam@ksu.edu.sa

ABSTRACT

In this paper, we propose a blind copy move image forgery detection method using dyadic wavelet transform (DyWT). DyWT is shift invariant and therefore more suitable than discrete wavelet transform (DWT) for data analysis. First we decompose the input image into approximation (LL1) and detail (HH1) subbands. Then we divide LL1 and HH1 subbands into overlapping blocks and measure the similarity between blocks. The key idea is that the similarity between the copied and moved blocks from the LL1 subband should be high, while the one from the HH1 subband should be low due to noise inconsistency in the moved block. We sort pairs of blocks based on high similarity using the LL1 subband and high dissimilarity using the HH1 subband. Using thresholding, we obtain matched pairs from the sorted list as copied and moved blocks. Experimental results show the effectiveness of the proposed method over competitive methods using DWT and the LL1 or HH1 subbands only.

Index Terms— Dyadic wavelets transform, copy-move, image forgery, image forensics

1. INTRODUCTION

Digital imaging has matured to become the dominant technology for creating, processing, and storing pictorial memory and evidence. Though this technology brings many advantages, it can be used as a misleading tool for hiding facts and evidences. This is because today digital images can be manipulated in such perfection that forgery cannot be detected visually. In fact, the security concern of digital content has arisen a long time ago and different techniques for validating the integrity of digital images have been developed. These techniques can be divided into two major groups: intrusive and non intrusive. In intrusive (active) techniques, some sort of signature (watermark, extrinsic fingerprint) is embedded into a digital image, and authenticity is established by verifying if the true signature matches the retrieved signature from the test image [1, 2, and 3]. This approach is limited due to the incapacity of many digital cameras and video recorders available in the market to embed extrinsic fingerprints [4].

The limitations of intrusive techniques have motivated the need for non-intrusive (blind) techniques [5, 6, 7, 8, 9, and 10] to validate the authenticity of digital images. These techniques exploit different kinds of intrinsic fingerprints such as sensor noise of the capturing device or image specific detectable changes for detecting forgery. There are many challenges in blind techniques, for instance, reducing false positive rates (i.e., an authentic image being detected as a forged image), making the system fully automated, localizing the forgery, detecting forgery of any type of image format (compressed or uncompressed), increasing the robustness and reliability, etc.

Existing blind techniques have their limitations. For example, (a) need many prior images to estimate the intrinsic fingerprints, which is a serious bottleneck (i.e., in potential situations only one image is provided) [5] [9], and (b) use one image but the method used for noise estimation is not robust because it is based on the Discrete Wavelet Transform (DWT) [8]. This is mainly because DWT is decimated and is not translation invariant, resulting in many large wavelet coefficients across several scales, creating problems in noise estimation.

In this paper, we propose a blind method for copy move image forgery detection using dyadic wavelets. Copy move is one of the most common techniques used for image forgery. In this type of forgery, one or more objects in an image are hidden by copying a part and moving it to another place of the same image. Some sophisticated image editing tools make this type of forgery undetectable in the naked eye by applying a 'soft' touch at the edges of the moved part. As the color and texture of the moved part is compatible with those of the copied part, it is very difficult to distinguish between these two parts. Also, two or more identical objects in the same original image contribute to the level of difficulty of forgery detection. Most of the existing copy move forgery detection methods either rely on similarity measurements or noise deviation measurements between the parts (blocks of an image). The proposed forgery detection method utilizes two types of information for detecting copy move forgery: (a) similarity between copied and moved

parts in the smoothed version of the image and (b) noise inconsistency between these parts caused by the forgery. Here, we use the dyadic wavelet transform, which is translation invariant. Moreover, we use the scaling coefficients (LL1) and wavelet coefficients (HH1) at scale one to obtain a smoothed version and noise estimation, respectively.

The rest of the paper is organized as follows. Section 2 reviews some of the previous methods in copy move forgery detection. Section 3 describes the proposed method. Experimental results and discussions are provided in Section 4, while Section 5 presents our conclusions.

2. PREVIOUS WORKS ON COPY MOVE FORGERY DETECTION

Quite a few works have been reported on copy move image forgery detection. A bibliography on blind image forgery detection methods can be found in [11]. Bayram et al [12] use a scale and rotation invariant Fourier-Mellin Transform (FMT) and the notion of bloom filters to detect copy-move forgery. Their method is computationally efficient and can detect forgery in highly compressed images. Copy move forgery detection based on blur moment invariants has been proposed in [13]. This method can detect duplicated regions degraded by blurring or corrupted with noise. Huang et al [14] have proposed a copy move forgery detection method based on Scale Invariant Feature Transform (SIFT) descriptors. After extracting the descriptors of different regions, they match them with each other to find possible forgery in images. A sorted neighborhood approach based on DWT and Singular Value Decomposition (SVD) has been proposed in [15]. In this method, first DWT is applied to the image and then SVD is used on low-frequency components to reduce their dimension. SV vectors are then lexicographically sorted, where duplicated blocks will be close in the sorted list. Solario and Nandi [16] use log-polar coordinates to obtain a one dimensional descriptor invariant to reflection, rotation, and scaling for detecting duplicated regions. The Discrete Cosine Transform (DCT) was used in [17]. They use lexicographic sorting after extracting DCT coefficients of each block in an image. A computationally efficient method based on Principal Component Analysis (PCA) was presented in [18]. The DWT and phase correlation based method was proposed in [19]. Their algorithm is based on pixel matching to locate copy move regions. Sutcu et al [20] proposed tamper detection based on the regularity of wavelet coefficients. In their method, they used undecimated DWT. Regularity in sharpness or blurriness is measured in the decay of wavelet coefficients across scales.

Most of the above methods suffer from false positives. Therefore, human interpretation is necessary to obtain the correct result [11].

3. PROPOSED METHOD

We propose a robust blind copy move image forgery detection method using the *dyadic (undecimated) wavelet transform* (DyWT). After extracting low frequency component (approximate) LL1 and high frequency component (detail) HH1 at scale one, a similarity measure is applied between the blocks in LL1 and HH1 separately. A decision is made based on the similarity between blocks in LL1 and dissimilarity between the blocks in HH1.

3.1. Dyadic wavelet transform

Many previous methods on copy move forgery detection use DWT. However due to its lack of shift invariance, the analysis of data is far from optimal. To overcome this drawback of DWT, Mallat and Zhong [21] introduced the DyWT, which is shift invariant. In this case, the wavelet transform does not involve downsampling and the number of wavelet coefficients does not shrink between the scales like in DWT.

Let **I** be the image to be decomposed, and $h[k]$ and $g[k]$ be the scaling (low pass) and wavelet (high pass) filters. The DyWT of an image can be computed using the following *atrous* algorithm.

Start at scale $j = 0$, and take $\mathbf{I}^0 = \mathbf{I}$, and compute the scaling and wavelet coefficients at scales $j = 1, 2, \ldots, J$ using Eqs. (1) and (2):

$$c^{j+1}[n] = \sum_k h[k] c^j[n + 2^j k] \quad (1)$$

$$d^{j+1}[n] = \sum_k g[k] c^j[n + 2^j k]. \quad (2)$$

Let $h^j[k]$ and $g^j[k]$ be the filters obtained by inserting $2^j - 1$ zeros between the terms of $h[k]$ and $g[k]$. Then we can perform DyWT using filtering as follows:

- Start with **I**, which is assumed to be at scale zero, i.e., $\mathbf{I}^0 = \mathbf{I}$.
- To obtain the scaling and wavelet coefficients $\mathbf{I}^j$ and $\mathbf{D}^j$ at scales $j = 1, 2, \ldots, J$
  - filter $\mathbf{I}^{j-1}$ with $h^{j-1}[k]$,
  - filter $\mathbf{I}^{j-1}$ with $g^{j-1}[k]$.

The following diagram (Figure 1) illustrates this algorithm one level decomposition.

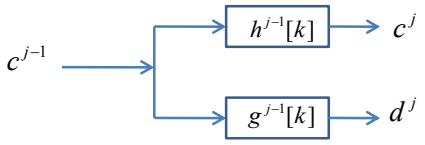

Figure 1. One level decomposition of DyWT.

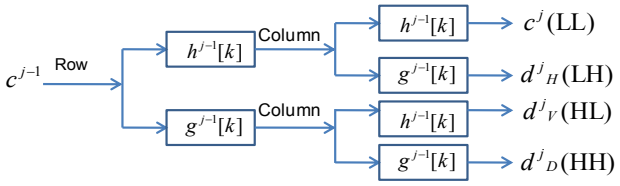

Figure 2. One level decomposition of DyWT of a 2D image.

As mentioned, there is no downsampling involved in DyWT. In the wavelet transform, $\mathbf{I}^{J}$ is called the low pass subband (L) and **Dj** are called the high pass subbands (H). In the case of two dimensional signals like images, we find four subbands LL, LH, HL, and HH at each scale of the decomposition. The size of each of these subbands is the same as the original image. We can decompose a 2D image using DyWT along rows and columns as illustrated in Figure 2.

3.2. Steps of the proposed method

Figure 3 shows the steps involved in the proposed copy move image forgery detection method.

In the proposed method, first, the image in question is decomposed using DyWT up to scale one. We use only LL1 and HH1 for further processing. The LL1 subband is an approximation of the image which is better for duplicate identification. The HH1 subband encodes noise present in the image, which is distorted while performing the forgery. HH1 actually contains high frequency information, which consists of mostly due to noise and sharp edges. In the case of color images, first we convert them to gray-scale before applying DyWT.

The LL1 and HH1 subbands are then divided into 16×16 pixel blocks with 8 pixel overlapping in both row and column. We assume that copy move forgery is performed in at least 16×16 pixel. Copied and moved blocks in LL1 should exhibit similarity between them. However, while performing the image forgery, the noise pattern, which is an intrinsic fingerprint of an image, is distorted. Therefore, copied and moved blocks should exhibit high dissimilarity between them in the HH1 subband. We calculate the similarity using the Euclidean distance:

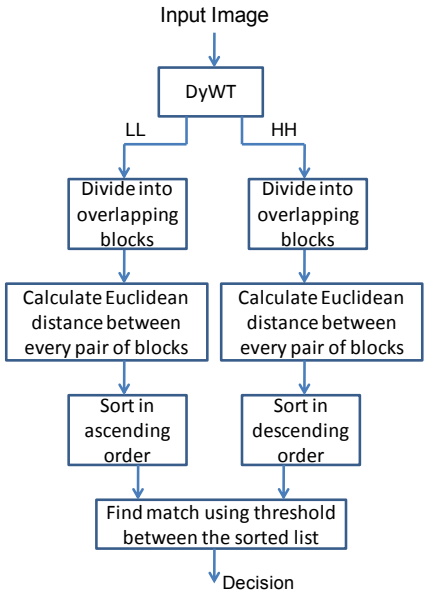

Figure 3. Flowchart of the proposed copy move image forgery detection.

$$d(p,q)=\sqrt{\frac{1}{N}\sum_{i=1}^{n}(p_i-q_i)^2} \qquad (3)$$

where $d(p,q)$ gives the distance between blocks p and q, p_i and q_i are corresponding LL1 or HH1 transform coefficient values and N is the total number of pixels in a block. In our case, N = 256. The distances are normalized by the maximum distance to scale the values between 0 and 1. Before calculating the distance, we arrange the pixels of a block in one dimensional vector.

The distances found using LL1 are then sorted in ascending order (List 1), putting highly similar pairs of blocks at the top of the list. We discard all the pairs of blocks that have distances > 0.7. We refer to this value as threshold 1 (Th_1). On the contrary, the distances calculated using HH1 are sorted in descending order (List 2); this places pairs of blocks with highly inconsistent noise at the top. Again we discard all the pairs of blocks that have distances lower than 0.3. We refer to this value as threshold 2 (Th_2). Now, if a pair of blocks according to its distance appears at the similar location in both of the lists (List 1 and List 2), then the pair is detected as copied and moved block. For example, if block pair (p, q) is located at nth location in List 1, and nth or (n+1)th or (n-1)th location in List 2, then the pair is detected as copy-move blocks. The values of *Th1* and *Th2* were chosen as optimal after several trials.

It should be mentioned that there may be similar objects in an original (not forged) image. In the case of LL1 subband only, similar objects will be identified as copy-moved objects resulting in false positives. On the other hand, in the HH1 subband, these objects will not be

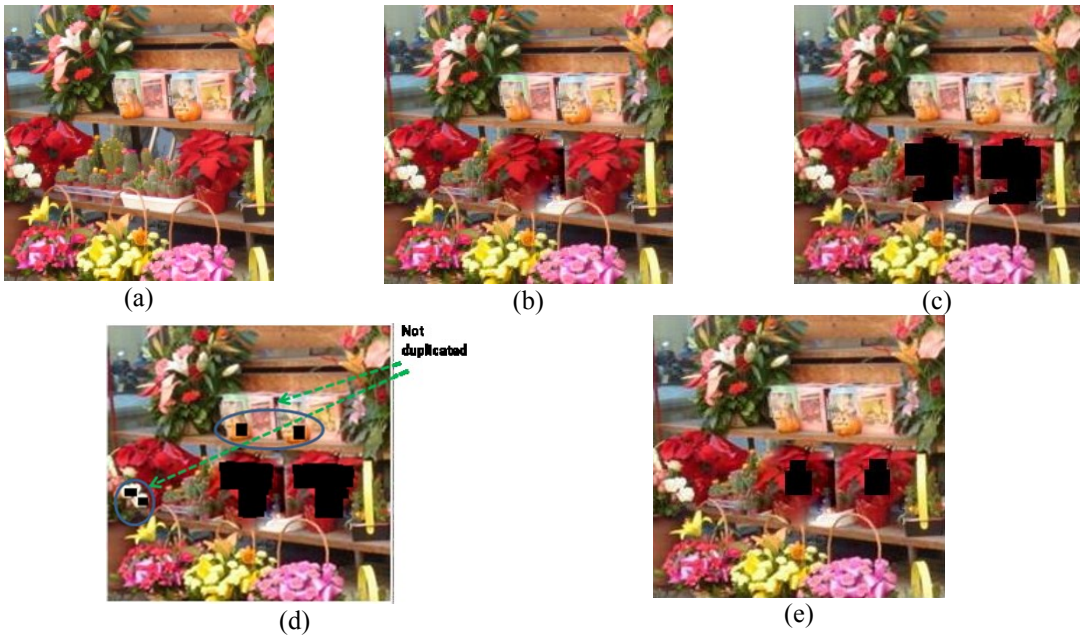

Figure 4. (a) The original image. (b) The forged image where the middle red flower is a copy of the right red flower. (c) The result of the proposed method. (d) The result of the method in [15]; it shows false positives. (e) The result using modified [8]; it shows truncated area of forgery.

identified as copy-moved because of low dissimilarity in noise level. Therefore, we capitalize both on LL1 and HH1 to avoid false positives.

4. EXPERIMENTAL RESULTS

The proposed method was evaluated on several test images that were forged using copy-move operation. There were 10 different image sources and the forgeries on these sources were done using Adobe Photoshop tool. The test images, both original and forged, can be found at http://faculty.ksu.edu.sa/ghulam/Pages/ImageForensics.aspx.

Figure 4 shows an example using the proposed method assuming a color copy-move forged image. The image was forged by copying the right red flower and moving it to the middle position (i.e., middle red flower is a copy of right red flower). Figure 4 (c) shows the output of the proposed method. The black area is identified as copy and move area. We compared our method with that in [15] that uses DWT and LL, and the one in [8] that uses DWT and HH1. We modified the method in [8] in the sense that instead of comparing the median of each block, we used Euclidean distances as described in Eq. (3). Figure 4 (d, e) shows the results produced by the methods in [15] and modified [8], respectively. Figure 4 (d) shows some false positives and Fig. 4 (e) shows some missing area of copy and move blocks.

We tested the proposed method on several test images with different copy-move forgery. There were a total of 574 copied 16×16 blocks (i.e., a total of 574 + 574 = 1148 blocks of copy-move). We considered a block as forged if more than 50% of that block area was copied / moved. The effect

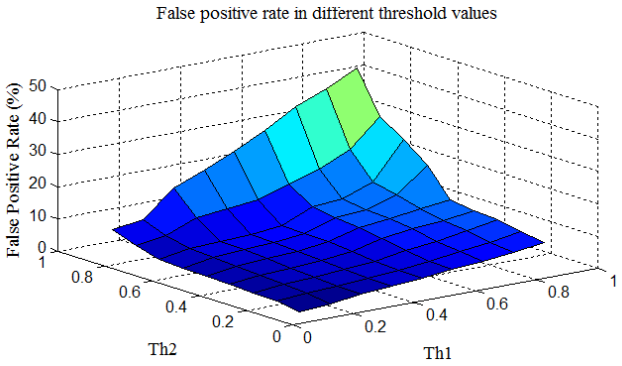

Figure 5. False positive rates (%) for different threshold values of *Th1* and *Th2* with the proposed method.

of thresholds *Th1* and *Th2* on false positive rate is shown in Fig. 5. From the figure, we can see that when the values of *Th1* and *Th2* are increased, false positive rate is also increased. The false positive rate is greater than 10% when the value of the pair <*Th1*, *Th2*> is higher than <0.5, 0.8> and <0.7, 0.7>.

Table 1 gives a comparison between the proposed method and the methods of [15] and modified [8] in terms of detected forged blocks out of 1148. It should be mentioned that although [15] shows comparable performance to the proposed method, it suffers from many false positives. The results shown with the proposed method use $Th1 = 0.7$ and $Th2 = 0.3$ that give the best accuracy.

Example results of two other test images are shown in Fig. 6 and Fig. 7.

Table 1. Number of blocks identified as copy move out of 1148 copy move blocks using different methods.

	Proposed method	Method of [15]	Modified method of [8]
Accuracy	1101 (95.90%)	1045 (91.03%)	932 (81.18%)
False Positive (%)	4.54%	9.65%	10.03%

5. CONCLUSION

We proposed a blind copy move image forgery detection method based on DyWT. We utilized both the LL1 and HH1 subbands to find similarities and dissimilarities between the blocks of an image for robust detection of copy move. The proposed method performed better than some of the previous methods. We are currently extending the proposed method to using color information instead of converting the color images to gray images.

Acknowledgement: This work is supported by the grant 10-INF1140-02 under the National Plan for Science and Technology (NPST), Saudi Arabia.

6. REFERENCES

[1] M. M. Yeung, "Digital watermarking," ACM Commun, vol. 41, no. 7, pp. 30–33, 1998.

[2] C. Rey and J. L. Dugelay, "A survey of watermarking algorithms for image authentication," EURASIP Journal on applied Signal Processing Vol. 2002 N6, pp. 613–621, 2002.

[3] C. Zhang, et al., "Multipurpose Watermarking Based on Multiscale Curvelet Transform," IEEE Trans. Inf. Forensics Security, vol. 3, no. 4, pp. 611–619, December 2008.

[4] H. Farid, "Image forgery detection - a survey," IEEE Signal Processing Magazine, vol. 5, pp. 16–25, March 2009.

[5] M. Chen, et al., "Determining Image Origin and Integrity Using Sensor Noise," IEEE Trans. Inf. Forensics Security, vol. 3, no. 1, pp. 74–90, 2008.

[6] B. Mahdian and S. Saic, "Blind Authentication Using Periodic Properties of Interpolation," IEEE Trans. on Inf. Forensics Security, vol. 3, no. 3, pp. 529–538, September 2008.

[7] H. Farid, "Exposing Digital Forgeries From JPEG Ghosts," IEEE Trans. Inf. Foren. Secu., vol. 4 (1), pp. 154–160, 2009.

[8] B. Mahdian and S. Saic, Using Noise Inconsistencies for Blind Image Forensics, Image and Vision Computing, vol. 27, no. 10, pp. 1497-1503, September 2009.

[9] A. Swaminathan, M. Wu, K. J. R. Liu, "Digital Image Forensics via Intrinsic Fingerprints," IEEE Trans. Inf. Forensics Security, vol. 3, no. 1, 101-117, March 2008.

[10] W.S. Lin, et al., "Digital Image Source Coder Forensics via Intrinsic Fingerprints," IEEE Transactions on Inf. Forensics Security, vol. 4(3), pp. 460–475, Sept. 2009.

[11] B. Mahdian and S. Saic, "A bibliography on blind methods for identifying image forgery," Signal Processing: Image Communication, vol. 25, pp. 389-399, 2010.

[12] S. Bayram, et al., "An efficient and robust method for detecting copy-move forgery," Proc. ICASSP09, pp. 1053-1056, 2009.

[13] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," Forens. Sci. Int., vol. 171, no. 2–3, pp. 180–189, 2007.

[14] H. Huang, et al., "Detection of copy-move forgery in digital images using SIFT algorithm," Proc. Pacific-Asia Workshop on Computational Intell. Industrial App., pp. 272- 276, 2008.

[15] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach detecting duplicated forgeries based on DWT and SVD," Proc. ICME2007, pp. 1750-1753, 2007.

[16] S. B. Solario and A. K. Nandi, "Passive forensic method for detecting duplicated regions affected by reflection, rotation, and scaling," Proc. EUSIPCO09, pp. 824-828, 2009.

[17] J. Fridrich, et al., "Detection of copy-move forgery in digital images," Digital Forensic Research Works., pp. 55–61, 2003.

[18] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Technical report, Dept. of Computer Science, Dartmouth College, 2004.

[19] J. Zhang, H. Wang, and Y. Su, "A new approach of detecting copy-move forgery in digital images," Proc. IEEE Int. Conf. on Communication Systems 2008, pp. 362-366, 2008.

[20] Y. Sutcu, et al., "Tamper Detection Based on Regularity of Wavelet Transform Coefficients," Proc. IEEE ICIP 2007.

[21] S. G. Mallat and S. Zhong, "Characterization of signals from multiscale edges," IEEE Trans. Pattern Anal. Machine Intell., vol. 14, pp. 710–732, July 1992.

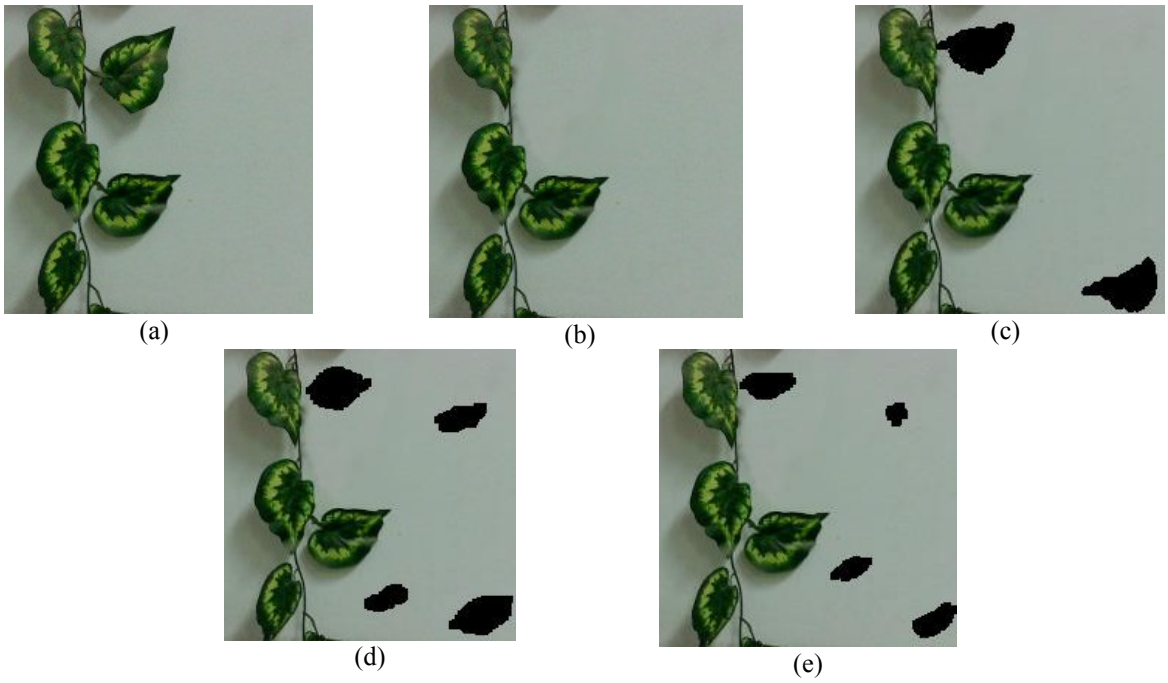

Figure 6. (a) The original image. (b) The upper leaf on right side has been hidden by copying and pasting a portion of the image from lower corner of the image. (c) The result of the proposed method. (d) The result of the method in [15]. (e) The result using modified [8].

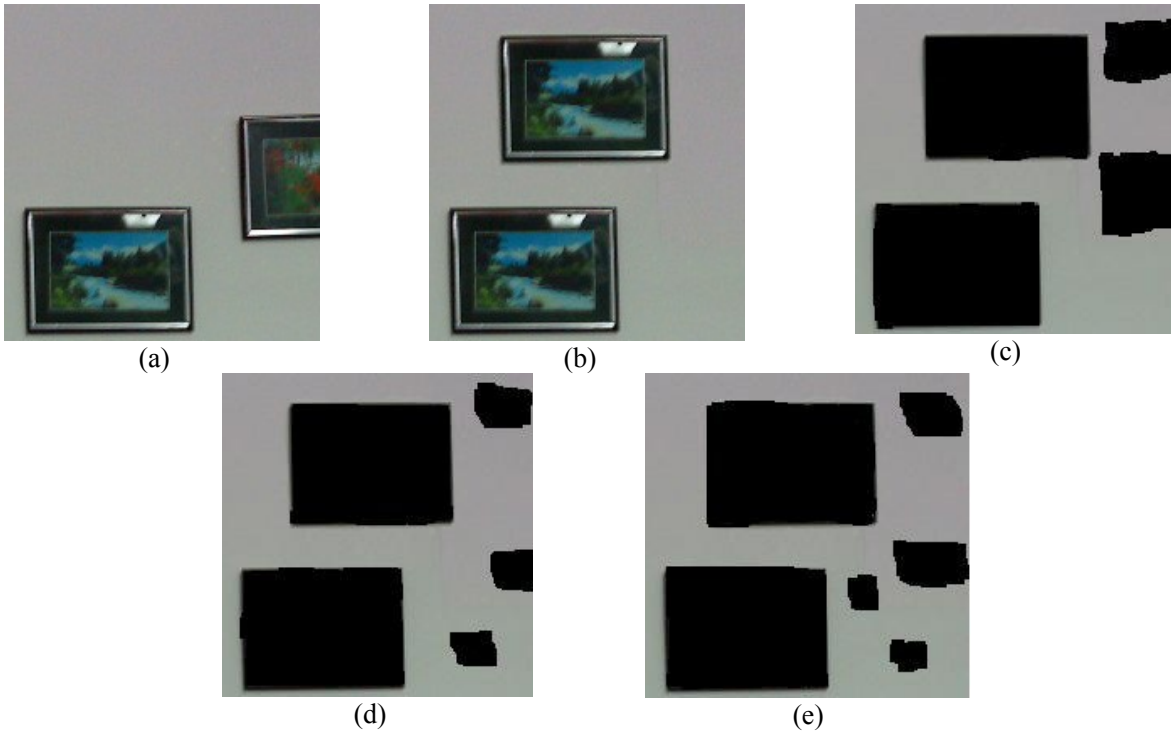

Figure 7. (a) The original image. (b) The picture in the lower part has been copied and pasted in the upper part and the smaller picture has been hidden by copying a portion of the image from the upper part of the image. (c) The result of the proposed method. (d) The result of the method in [15]. (e) The result using modified [8].