

Review of Copy-Move Forgery Detection of Images using Discrete Cosine Transform

Angom Buboo Singh
MTech CSE
NIT Agartala, India

Swapan DebBarma
Asst. Professor, CSE
NIT Agartala, India

Kh.Manglem Singh
Associate Professor, CSE
NIT Imphal, India

ABSTRACT

With the easy availability of cheap digital cameras and powerful image editing software, image forgery is becoming one of the most common cyber-crime. Image forgery is used not only for political, social or financial gain but also for defamation, harassment and also a part of various other cyber-crimes. One of the most common types of image forgery is the copy-move forgery where a part of the image is copied and pasted on another region of the same image. This is mostly done either to highlight an object in the image or to hide or delete an object in the image. There are various techniques to detect these type of image forgery and are commonly referred to as Copy Move Forgery Detection (CMFD). In this paper a review of the basic CMFD algorithm based on Discrete Cosine Transformation (DCT) proposed by Fridrich et. al [1] is given. The algorithm performs robust block matching and exhibit good results and is the basis of many other CMFD algorithms.

Keywords

Copy-Move forgery, Digital Tempering, DCT, Cloning, block matching technique.

1. INTRODUCTION

These days, not only creation of digital images is very easy with the easy availability of cheap digital cameras, but also the distribution and manipulation of it has become a very simple act compared to previous years. Image forgery is not new. Even Stalin concealed his enemies in photographs by tampering the photo in the early 1900s [2]. But compared to today, this was a time-consuming and laborious act. Nowadays, powerful and sophisticated software tools provide the opportunity to even inexperienced users to forge digital images very easily.

Image forgery is done for various reason ranging from political propaganda to scientific fraud, from entertainment to defamation but above all is mainly done for some profit. There are several types of image forgery; however, concealing some objects from natural image is a common form of image forgery, known as Copy Move forgery. An example of CMF is shown in fig 1.



Fig. 1. An example of CMF : (a) the original image with three missiles and (b) the forged image with four missiles.

One of the basic copy move forgery detection algorithm is the one proposed by Fridrich et. al [1]. The algorithm is based on Discrete Cosine Transform and performs a robust block matching which exhibit a good result and is the basis of many other CMFD algorithms. In this paper we give a review of the algorithm and analyse some of the shortcoming of the algorithm.

The paper is organized as follows: Section 2 gives a basic background on DCT, followed by the main CMFD algorithm in section 3. This is followed by experimental result and conclusion in section 4 and 5 respectively.

2. BASIC OF DISCRETE COSINE TRANSFORMATION

In this section a short overview of Discrete Cosine Transform (DCT) is provided. The DCT is a central component of most compression applications for images. There are also many forensic means utilizing the DCT aimed at integrity issues. Basically, the DCT maps discrete values of the spatial of time domain to the frequency domain. Given a 1D input vector x of discrete values the DCT F of x is computed as follows [3]:

1D DCT

$$F_k = \sum_{n=0}^{N-1} 2x_n \cos \frac{\pi k}{2N} (2n + 1)$$

With $k \in [0, N-1]$

As images are 2D objects, the DCT is at first applied row-wise and subsequently applied column-wise to the result of the row-wise DCT. This results in the following definition for the 2D DCT of an $N_1 \times N_2$ matrix [4]:

2D DCT

$$F_{u,v} = \frac{1}{4} C_u C_v \left[\sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f_{x,y} \cos \frac{\pi u(2x+1)}{2N_1} \cos \frac{\pi v(2y+1)}{2N_2} \right]$$

With $(u,v) \in [0, N_1 - 1] \times [0, N_2 - 1]$,

$$C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } u, v = 0 \\ 1, & \text{else} \end{cases}$$

While x and y represent the coordinates of the pixels within the considered block to be encoded in the horizontal and vertical direction, u and v represent the coordinates of the frequencies in horizontal and vertical direction. $f_{x,y}$ describes the pixel value at position (x,y) .

The DCT is very important for many compression applications as it allows us to filter relevant information from information which is perceptually irrelevant for the human auditory and visual system. As the human eye is more sensitive to low frequencies with few and coarse details than to high frequencies with much and fine details, we can reduce the amount of data stored by leaving out the irrelevant high frequencies with the aid of quantization step without perceivable loss of quality. In an 8×8 block the low frequencies are on the upper left and the high frequencies are on the bottom right. For this, we make an illustration; we use an image of Lena with the size of 256 pixels \times 256 pixels.

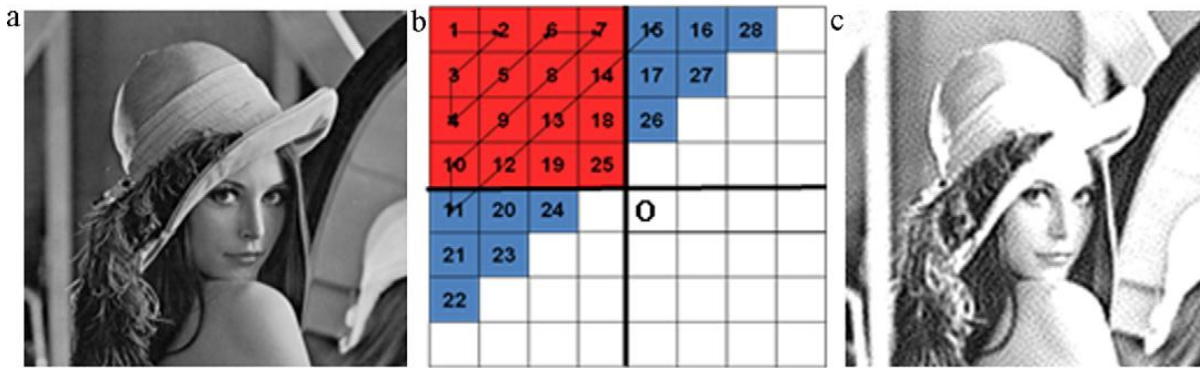


Fig. 2. (a) the Lena image (b) Zigzag order scanning (c) the reconstruction image of Lena by using 1/4 DCT coefficients.

Fig. 2(a) is Lena, then the discrete cosine transform is applied to Fig. 2(a), after that, we extract the low frequency DCT coefficients of Fig. 2(a) in a zigzag order, Fig. 2(b) is an example of zigzag order, the red area is the low frequency part, which occupies the 1/4 energy of the entire DCT coefficients. Fig. 2(c) is the reconstruction image of Lena after extracting the 1/4 DCT coefficients of Fig. 2(a) in a zigzag order. Through the analysis of Fig. 2, if the image block undergoes DCT transform, we can use four-part energy to represent the whole image while without losing any important information.

3. DETECTION ALGORITHM

The detection algorithm begins with the preprocessing of the suspect image. The image is divided in overlapping blocks and DCT is applied to each block. From this the features are extracted and blocks having similar features are detected. In order to reduce false matches some post-processing operation are also performed. The detection algorithm can be depicted with the following architecture:

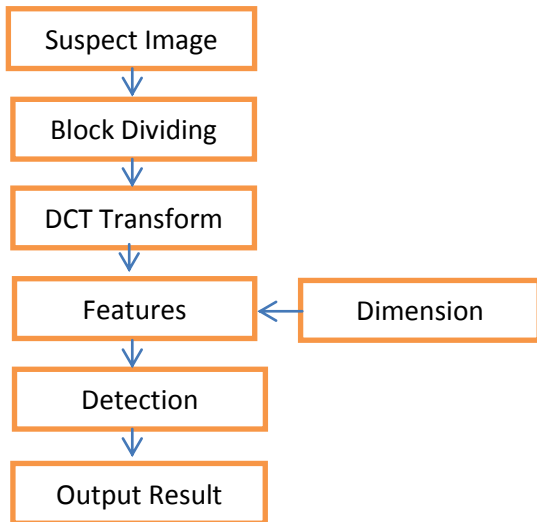


Fig 3.Detection algorithm

The algorithm undergoes several steps as follows:

1. **Grayscale conversion** The color image is converted to a grayscale image with the formula $I = 0.299R + 0.587G + 0.114B$. This is because the human eye perceives the luminance of the colors red, green and blue varyingly strong. Green is the color which is perceived best by the human eye, followed by red and then by blue. Expressed in percentage the luminance of red is 29.9%, that of green is 58.7%, and finally that of blue is 11.4%.

2. **Sliding Block** Let the size of the image is $m \times n$. The second step is scanning the digital image by sliding a $B \times B$ block over the entire image, from the upper left to the lower right, pixel by pixel. By doing this we receive a $B \times B$ extract from the image from every pixel position, a total of $(m-B+1) \times (n-B+1)$ extracts. For each extracted $B \times B$ block the respective start pixel position (e.g. very first pixel of considered block) is memorized.

3. **Block-Based DCT** Next, a $B \times B$ 2D DCT is applied on every extracted $B \times B$ block.

4. **Block Quantization** The quantization of the DCT coefficients of each $B \times B$ block follows with the aid of a $B \times B$ quantization table with a user-specified quality factor Q . A high quality factor Q results in finer quantization, which means that using a high quality factor Q the compared blocks must match more closely to be identified as equal. On the other hand, when using a low quality factor Q the identification of very similar blocks does not require a very close match of the blocks as the quantization is coarser, while on the same time more false matches become possible.

5. **Block Serialization** Each quantized DCT block is serialized to a row and stored into a matrix A .

6. **Row Sorting** The rows (not the elements in a row) in matrix A are sorted lexicographically. This operation results in a matrix where identical and similar rows appear consecutively.

7. **Row Comparing and Shift Vector Calculation** Now the comparison of consecutive block pairs i with the start pixel position (i_1, i_2) and $j = i + 1$ with the start pixel position (j_1, j_2) follows. If a matching block pair is found (i.e. if block i equals block j), the two equal blocks' start positions are stored in a separate list and the shift vector of the two equal blocks is computed as follows:

$$sv = (s_1, s_2) = (|i_1 - j_1|, |i_2 - j_2|)$$

Subsequently, the respective counter of the computed shift vector (s_1, s_2) is incremented by one as follows:

$$C(s_1, s_2) = C(s_1, s_2) + 1$$

while C is initialized to zero before the algorithm is run. At the end of the matching process the counter C represents the occurrence of different shift vectors

8. **Block Coloring** Finally, the coloring of the detected identical blocks is performed. Here it is important that all blocks contributing to the same shift vector are colored with the same color in order to be able to distinguish the various occurrences in the output picture. The amount of identical blocks to be colored can be controlled by a user-specified threshold T . That is to say, all shift vectors $s(1), s(2), \dots, s(k)$ whose occurrence exceed the threshold T :

$$C(s(r)) > T$$

are identified and finally the blocks contributing to the specific shift vectors are colored with the same color representing segments that might have been copied and moved. Hence, same colors correspond to equal shift vectors and different colors correspond to different shift vectors. T and block size B control the size of the smallest segment in the image that can be identified as a possible copied and moved region.

4. EXPERIMENTAL RESULT

In the experiment performed on MATLAB 7.11, copied and moved areas significantly smaller than 16×16 pixels cannot be detected by the algorithm. Whereas, middle-sized and large-sized manipulations of approximately 16×16 to 25×25 (middle-sized) and significantly larger than middle-sized manipulations are detected by the algorithm as well as regions being copied and moved to multiple other regions. False matches of blocks are not avoidable at all, especially for uniform colored regions like the sky or other plain color part of the image. It was also observed that many false matches exhibited a very low shift vector. Increasing the threshold T for the shift vector can also reduce the false matches.

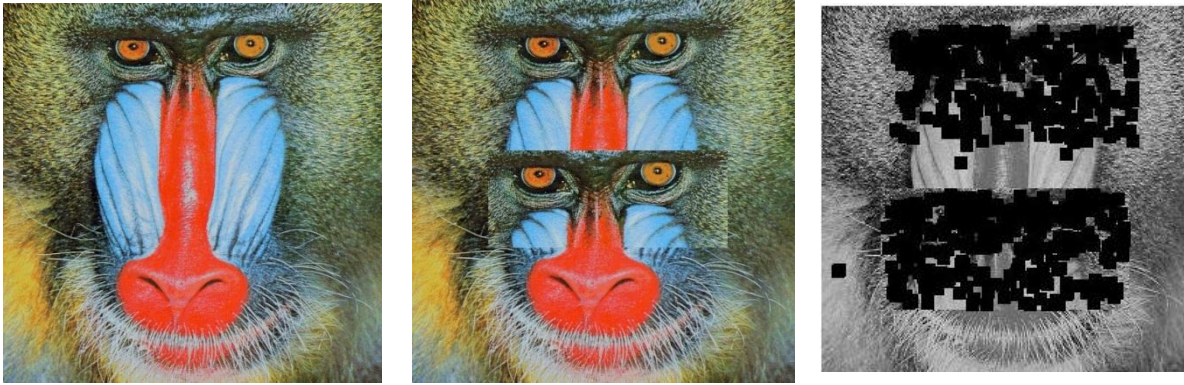


Fig 4. (a) Original Image

(b)Forged Image

(c)Forgery Detected



Fig 5. (a) Original Image

(b)Forged Image

(c)Forgery Detected



Fig 6. (a) Original Image

(b)Forged Image

(c)Forgery Detected

5. CONCLUSION

The CMFD based on DCT algorithm is a good method to detect possible copied and moved areas. One can detect possibly copied and moved areas, while not all of them are necessarily detected. A copied and moved area could be missed if e.g. the copied and moved area is too small or if a flat and uniform area is used for concealing an object in this flat and uniform area. Furthermore, post-processing activities such as rotation, scaling and other transformation of the paste area will make the detection of the copy –move very difficult and sometime make the algorithm fails to detect any match. In the end, we need a human expert to interpret the results of the algorithm to overcome the false matches.

6. REFERENCES

- [1] A. Fridrich, et al., Detection of Copy-move Forgery in Digital Images, 2003.
- [2] H. Farid. Digital doctoring: can we trust photographs? <http://www.ists.dartmouth.edu/library/327.pdf>, 2008.
- [3] J. W. Woods. Multidimensional Signal, Image, and Video Processing and Coding. Elsevier, Academic Press, 2006.
- [4] G. K. Wallace. The jpeg still picture compression standard. Commun. ACM, 34:30–44, 1991