A Natural Image Model Approach to Splicing Detection

Yun Q. Shi
New Jersey Institute of Technology
University Heights
Newark, NJ, USA 07102
+1-973-596-3501
Shi@njit.edu

Chunhua Chen
New Jersey Institute of Technology
University Heights
Newark, NJ, USA 07102
+1-973-642-7994
cc86@njit.edu

Wen Chen
New Jersey Institute of Technology
University Heights
Newark, NJ, USA 07102
+1-973-642-7994
wc47@njit.edu

ABSTRACT

Image splicing detection is of fundamental importance in digital forensics and therefore has attracted increasing attention recently. In this paper, we propose a blind, passive, yet effective splicing detection approach based on a natural image model. This natural image model consists of statistical features extracted from the given test image as well as 2-D arrays generated by applying to the test images multi-size block discrete cosine transform (MBDCT). The statistical features include moments of characteristic functions of wavelet subbands and Markov transition probabilities of difference 2-D arrays. To evaluate the performance of our proposed model, we further present a concrete implementation of this model that has been designed for and applied to the Columbia Image Splicing Detection Evaluation Dataset. Our experimental works have demonstrated that this new splicing detection scheme outperforms the state of the art by a significant margin when applied to the above-mentioned dataset, indicating that the proposed approach possesses promising capability in splicing detection.

Categories and Subject Descriptors

I.4 [Image Processing]: Miscellaneous

General Terms

Security, Algorithms.

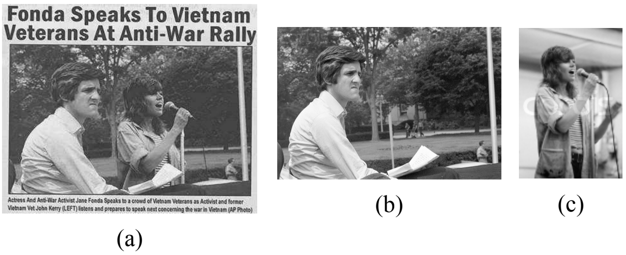
Keywords

Splicing detection, digital forensics, block discrete cosine transform, statistical moment, Markov process.

1. INTRODUCTION

Evidently, we are now living in a digital epoch. Look at things around us: computers, digital cameras, digital camcorders, digital audio players, the Internet, and so on. Many people are woken up by their personal digital assistants (PDA's), which are the tools at the first sight of their daily life.

Digital cameras, computers, image processing and graphic software have built up an environment in which manipulation of digital images becomes easy. That is, the conventional darkroom is not necessary any more. Fake images are so many nowadays that one cannot ignore their existence. In 2004, a picture of John Kerry and Jane Fonda at an anti-war rally during the early 1970's surfaced on the Internet for some political motivations, which is shown in Figure 1(a). It was reported later that this picture was created by merging two different authentic photographs as shown in Figure 1(b) and Figure 1(c). Another example is an image about Israel air striking Beirut, Lebanon in August 2006, as shown in Figure 2(a). This image was later found altered by the photographer and the authentic picture is shown in Figure 2(b). Compared with the authentic image, the altered image has made smoke darker by using some image processing software. This forged picture caused Reuters to withdraw 920 pictures taken by the photographer from sale.

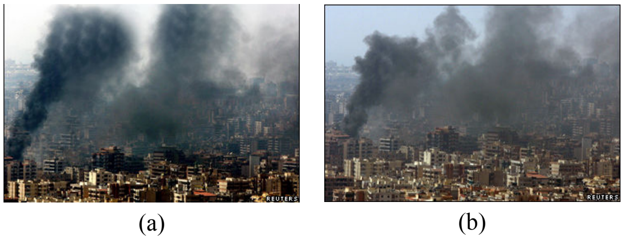


Fonda Speaks To Vietnam Veterans At Anti-War Rally

Actress And Anti-War Activist Jane Fonda Speaks to a crowd of Vietnam Veterans as Activist and former Vietnam Vet John Kerry (LEFT) listens and prepares to speak next concerning the war in Vietnam (AP Photo)

(a) (b) (c)

Figure 1. A spliced image and two original authentic images [1].



(a) (b)

Figure 2. An altered image and its original authentic image [2].

Replacing one or more parts of a host picture with fragment(s) from the same host picture or other pictures is called as splicing, which may belong to malicious manipulation to forge a scene that actually never exists to mislead the observers on purpose.

Image splicing, as its name implies, is a simple process of cropping and pasting regions from the same or different images to form another image without post-processing such as edge

Conference'07, September 20–21, 2007, Dallas, Texas, USA.

smoothing. Image splicing is one of the simple and commonly used image tampering schemes. Since splicing is often used for image tampering as an initial step, and splicing itself, with modern image processing techniques, can often hardly be caught by the human visual system, image splicing detection is of fundamental importance in image tampering detection.

Image splicing detection is hence urgently called for digital data forensics and information assurance. People need to tell if a given image is spliced or not without any a priori knowledge. In other words, the splicing detection should be blind in nature.

Researchers have recently made efforts on image splicing detection and different methods have been proposed. In [11] and [20], two algorithms to detect duplicate regions in images are proposed. Image lighting inconsistency [14], camera parameters [9, 10, 13, 15], bicoherence statistics [18, 19], Hilbert-Huang transform [12], and statistics of 2-D phase congruency [6] have been used to detect spliced images.

Splicing detection is challenging. The blind splicing detection methods [19, 12, 6] have achieved success detection rates of 72%, 80%, and 82%, respectively, over the Columbia Image Splicing Detection Evaluation Dataset [4]. Further research is hence necessary.

In this paper, we propose a blind and effective splicing detection approach based on a natural image model. Combining features extracted from the given test image as well as 2-D arrays generated by applying to the test images multi-size block discrete cosine transform (MBDCT), the proposed scheme provides promising splicing detection capability. Our experimental works with a concrete implementation of the proposed approach show that the new approach outperforms the prior arts [19, 12, 6] when applied to the same image database [4] with a success detection rate of 92%. It is expected that this new approach can play an effective role in tampering detection as well.

The rest of this paper is organized as follows. The proposed natural image model, i.e., the feature construction framework, is described in Section 2. In Section 3, we present a concrete implementation of this proposed framework specifically designed for the publicly available database — the Columbia Image Splicing Detection Evaluation Dataset [4]. Experimental works and results are given in Section 4. The effect of multi-size block discrete cosine transform is then analyzed in Section 5. Next, some issues in implementation are addressed in Section 6. Finally, conclusion and discussion are provided in Section 7.

2. CAPTURING SPLICING ARTIFACTS BY USING A NATURAL IMAGE MODEL

We are facing a binary decision problem in splicing detection at this stage. In other words, we need to tell if a given test image is a spliced image or a non-spliced (authentic) image. Therefore, splicing detection can be treated as a two-class pattern recognition problem. Although the splicing operation is simple, the introduced artifacts are often not easy to capture. As a key step, feature extraction is to capture the splicing artifacts thus distinguishing spliced images from non-spliced images in the feature space.

As described in Section 1, image splicing operation can be carried out simply by cropping and pasting. For a host image, the cut-and-pasted image fragments are a kind of "strangers". These strangers may cause disturbances in the smoothness, consistency, continuity, regularity, and/or periodicity of the original host images, and hence change the correlation between image pixels.

In general, we may treat the image splicing operation as additive, i.e., model the error between the spliced image and original host image as additive noise. Although this assumption does not always hold, it would simplify problems in splicing detection scenarios and thus benefit our analysis.

In this paper, we propose a natural image model to separate spliced images from natural images. This natural image model is represented by features extracted from a given test image and 2-D arrays produced by applying multi-size block discrete cosine transform to the given test image. That is, we consider the spatial representation of the given test image (i.e., the image pixel 2-D array, or referred to as image 2-D array for short in this paper) and extract statistical moments of characteristic functions and Markov transition probabilities from this image pixel 2-D array. Furthermore, we propose to apply block discrete cosine transform (BDCT) with a set of block sizes to the test image, resulting in a set of BDCT 2-D arrays (i.e., MBDCT coefficient 2-D arrays, or MBDCT 2-D arrays for short in this paper). From these MBDCT 2-D arrays, we extract statistical moments of characteristic functions and Markov transition probabilities as features.

The main merit of this natural image model lies in the following two *combinations*: 1) the combination of features derived from the image pixel 2-D array and those derived from the MBDCT coefficient 2-D arrays; 2) the combination of moments of characteristic functions based features and Markov process based features. Features generated using different methodologies can make up each other if properly applied. It will be shown later in this paper that the features generated from the MBDCT 2-D arrays can greatly improve the splicing detection performance of our proposed scheme.

In this section, we present the general framework of our proposed natural image model, which is shown in Figure 3.

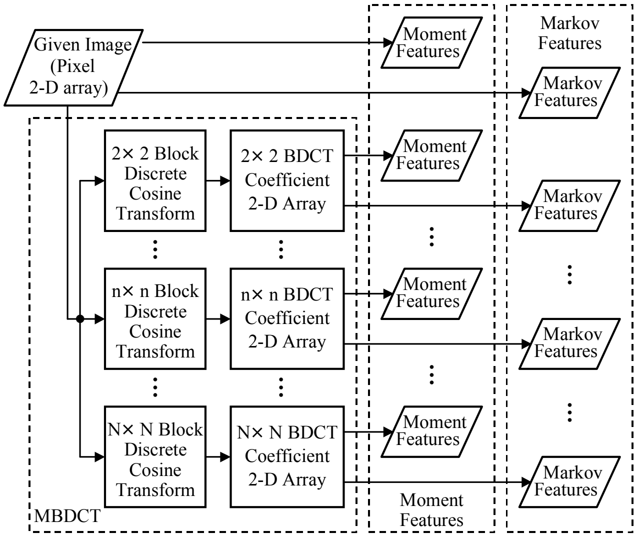

Given Image (Pixel 2-D array)
Moment Features
Markov Features
Markov Features
2× 2 Block Discrete Cosine Transform
2× 2 BDCT Coefficient 2-D Array
n× n Block Discrete Cosine Transform
n× n BDCT Coefficient 2-D Array
N× N Block Discrete Cosine Transform
N× N BDCT Coefficient 2-D Array
Moment Features
Markov Features
Moment Features
Markov Features
Moment Features
Markov Features
MBDCT
Moment Features

Figure 3. A general natural image model.

2.1 Multi-size Block Discrete Cosine Transform (MBDCT) 2-D Arrays

The block discrete cosine transform (BDCT) has been widely used in the international image and video compression standards due to its superior capability in decorrelation and energy compaction. For example, 8×8 BDCT has been adopted in JPEG and MPEG-2 (Moving Picture Experts Group) standards.

We propose to use BDCT with a set of different block sizes in this novel natural image model for splicing detection. This is to utilize the complementary decorrelation capabilities contributed by BDCT's with various block sizes. The splicing procedure changes the local frequency distribution of the host images. Coefficients of the BDCT's can reflect these changes. It is noted that the pattern in which the correlation changes is various and complicated due to different possible splicing operations, different host images, and different pasted image fragments. Therefore, we cannot expect to catch this change effectively by one single-block-size BDCT. With various block sizes, the MBDCT coefficients can perceive the change of frequency distribution in a variety of ways and hence the spliced images can be distinguished from natural images with features extracted from these MBDCT 2-D arrays. This point has been supported by our extensive experimental results shown in Sections 4 and 5. There, it is shown that from 2×2, 4×4, to 8×8, each BDCT 2-D array contributes. That is, if we eliminate any one of these BDCT 2-D arrays, the splicing detection rate reduces.

The application of an n×n BDCT is described as follows. Firstly, the given image is divided into non-overlapping n×n blocks. Then, 2-D discrete cosine transform (DCT) is applied to each block independently. Finally, we obtain a 2-D array consisting of all the BDCT coefficients of all these blocks, as shown in Figure 4, where n is equal to 4, i.e., a 4×4 BDCT coefficient 2-D array is shown, and S_u, S_v denote the image dimension in the horizontal direction and vertical direction, respectively. Denoting an n×n image block by $f(x,y), x,y = 0,1,\cdots,n-1$, the 2-D block DCT coefficients are given by

$$F(s,t) = \frac{2}{n}\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}\Lambda(x)\Lambda(y)\cos\frac{\pi s(2x+1)}{2n}\cos\frac{\pi t(2y+1)}{2n}f(x,y), \quad (1)$$

where

$$\Lambda(x) = \begin{cases}\frac{1}{\sqrt{2}}, x=0\\ 1, otherwise\end{cases}, \quad (2)$$

and $s,t \in \{0,1,\cdots,n-1\}$.

With each individual block size, we can obtain one BDCT 2-D array, from which we can generate one subset of features. As shown in Sections 4 and 5, each feature subset associated with a specific block size does contribute to the proposed splicing analyzer.

The choice of N, the largest block size, should be considered carefully in reality. Image size, feature dimensionality, correlation among pixels in an image, and computational complexity are influencing the choice. Generally speaking, this N should not be too large. That is, when N becomes too large, the corresponding N×N BDCT 2-D array will not bring more benefit in decorrelation because the correlation between image pixels separated by a large distance (say, larger than 50) becomes rather weak. On the other hand, for a large N, it implies more BDCT 2-D arrays need to be calculated, hence more computation. This is because, as said in the second paragraph of this section and will be shown in Sections 4 and 5, each of 2×2, 4×4, and 8×8 BDCT 2-D arrays contributes to splicing detection. That is, if we eliminate one of the three BDCT 2-D arrays while keeping the rest two BDCT 2-D arrays, the detection rate will go down. Hence, as N is large, it means more BDCT 2-D arrays to calculate, resulting in more computation. In addition, for a large N, more BDCT 2-D arrays, hence more features will be generated, which also leads to more computation. Furthermore, the relation between the number of features (feature dimensionality) and the size of the available image dataset, discussed in Sections 3 and 4, indicates one more constraint in choosing a large N.

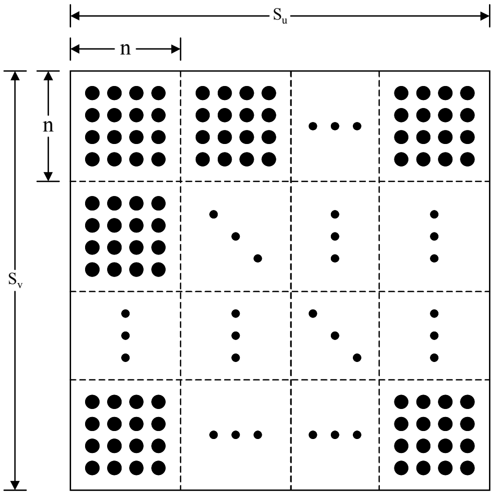


Figure 4. An n×n BDCT 2-D array.

2.2 Moment Based Features

The moment features are derived from the 1-D characteristic functions (discrete Fourier transform (DFT) of the first-order histograms), as well as from the 2-D characteristic functions (2-D DFT of the second-order histograms). Especially, since the second-order histogram involves two pixels or two coefficients at a time and hence brings out the second-order statistics, it is more capable in capturing splicing artifacts.

The block diagram of moment extraction is shown in Figure 5.

2.2.1 Prediction-error 2-D Array

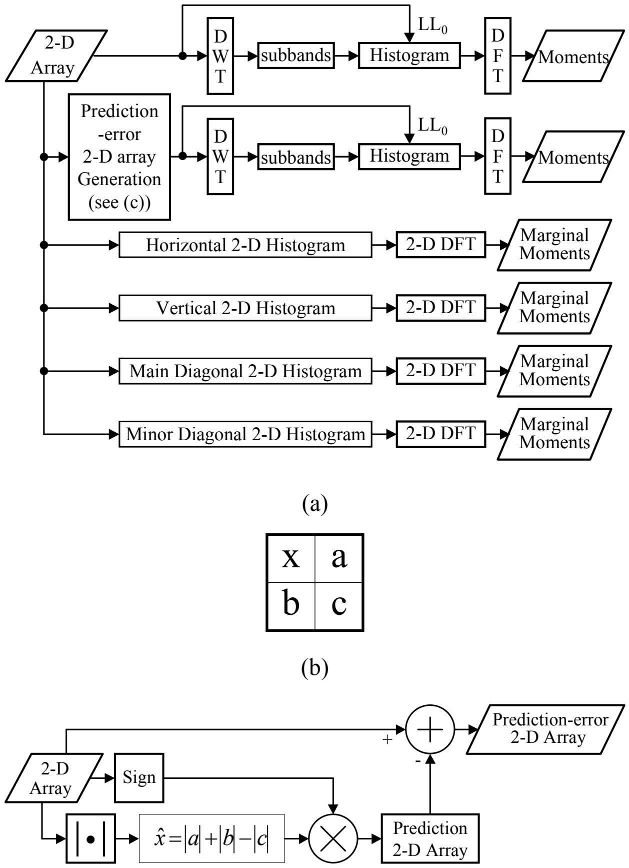
A prediction-error 2-D array is used to reduce the influence caused by diversity of the image content and to simultaneously enhance the statistical artifacts introduced by splicing. In other words, the prediction-error is used to eliminate the "side" effect caused by diversity of image content on splicing detection.

The prediction context is shown in Figure 5 (b), i.e., we need to predict the value of *x* using the values of its neighbors *a*, *b*, and *c*. The generation of prediction-error 2-D array is shown in Figure 5 (c). The prediction 2-D array can be given by

$$\bar{x} = sign(x)\cdot\{|a|+|b|-|c|\}, \quad (3)$$

and prediction-error 2-D array can be expressed by:

$$\triangle x = x - \bar{x} = x - sign(x)\cdot\{|a|+|b|-|c|\}. \qquad (4)$$



(a)

(b)

(c)

Figure 5. Moment extraction procedure. (a): general block diagram; (b): prediction context; (c): prediction-error 2-D array generation.

2.2.2 Discrete Wavelet Transform (DWT)

Wavelet analysis has been widely used in digital image processing applications owing to its superior multi-resolution and space-frequency analytical capability. It is well known that the DWT is suitable to catch transient or localized changes in spatial and frequency domains. Hence, DWT is a good tool for splicing detection.

Wavelet decomposition has demonstrated its efficiency in digital forensics applications. For instance, in [7], higher-order wavelet statistics have been applied to digital forensics. The application of three-level Haar wavelet transform to splicing detection has been reported and justified in [12, 6].

We therefore introduce discrete wavelet transform in our natural image model, which is applied to the image pixel 2-D array, MBDCT coefficient 2-D arrays, and prediction-error 2-D arrays, resulting in wavelet subbands. From these subbands, we calculate statistical moments as follows.

2.2.3 Moments and Marginal Moments

As mentioned in the beginning of Subsection 2.2, the 1-D characteristic function (CF) is the DFT of the first-order histogram of each wavelet subband (we treat the image and coefficient 2-D array as the low-low subbands at level 0 and denote them LL$_0$'s). Given *H(x$_i$),* which is the CF component at frequency *x$_i$*, and *K,* which is the total number of different values assumed by all of coefficients in a subband under consideration, i.e., the tap length of DFT, the absolute moments of the 1-D CF are defined as follows:

$$M_l = \frac{\sum_{i=1}^{K/2} x_i^l |H(x_i)|}{\sum_{i=1}^{K/2} |H(x_i)|}, \qquad (5)$$

where l is an integer representing the order of moment.

Under the assumption that the pasted image fragment is additive to the host image, the additive noise (i.e., the difference between the spliced image and the original host image) is independent to the host image, and the magnitude of the characteristic function of the noise is non-increasing, it can be shown that the above-defined moments will not increase after splicing by using the discrete Chebyshev inequality [17]. Therefore, it is possible to separate spliced images from natural images using these statistical moments. This theoretical analysis together with experimental works reported in [12, 6] and in Section 4 of this paper justify the use of moments of characteristic functions instead of moments of histograms.

To further enhance splicing detection capability, we propose to include the second-order statistics into our natural image model. By measuring the intensity change of pixels with respect to their neighbors, 2-D histograms can reflect the statistical effects of splicing artifacts more efficiently than 1-D histograms, which consider one pixel at a time and do not reflect the intensity/position correlation among neighboring pixels. Therefore, the second-order statistics are expected to be able to catch statistical artifacts caused by splicing when the splicing operates in the spatial domain of a host image and it keeps the histogram unchanged during the splicing operation.

It is known that the second-order histogram [21] is a measure of the joint occurrence of pairs of pixels separated by a specified distance and orientation. Denote the separation by the distance between two pixels, ρ , at the angle of the line linking these two pixels with respect to the horizontal axis, θ . The second-order histogram is defined as

$$h_d(j_1, j_2; \rho, \theta) = \frac{N(j_1, j_2; \rho, \theta)}{N_T(\rho, \theta)}, \qquad (6)$$

where $N(j_1, j_2; \rho, \theta)$ is the number of pixel pairs for which the first pixel value is j_1 while the second pixel value is j_2 , and $N_T(\rho, \theta)$ is the total number of pixel pairs in the image with separation (ρ, θ) . The second-order histogram is also called dependency matrix or co-occurrence matrix.

After applying 2-D DFT to the second-order histogram, we obtain a 2-D CF. Two marginal moments of the 2-D CF are thereafter calculated by

$$\begin{aligned} M_{u,l} &= \frac{\sum_{j=1}^{K/2}\sum_{i=1}^{K/2} u_i^l \left|H(u_i,v_j)\right|}{\sum_{j=1}^{K/2}\sum_{i=1}^{K/2} \left|H(u_i,v_j)\right|}, \\ M_{v,l} &= \frac{\sum_{j=1}^{K/2}\sum_{i=1}^{K/2} v_j^l \left|H(u_i,v_j)\right|}{\sum_{j=1}^{K/2}\sum_{i=1}^{K/2} \left|H(u_i,v_j)\right|}, \end{aligned} \tag{7}$$

where $H(u_i,v_j)$ is the 2-D CF component at DFT frequency (u_i,v_j) and l is an integer representing the order of moments.

2.3 Markov Based Features

Moment feature is a kind of measure which reflects the statistical changes caused by image splicing. Markov feature, which will be introduced in this subsection, is another kind of measure which can also reflect the statistical changes caused by image splicing. Combining these two kinds of features, our proposed natural image model can be enhanced to detect spliced images more effectively.

In this subsection, we describe the Markov feature extraction procedure. At first, we form difference 2-D arrays from the given image and/or coefficient 2-D array. These difference 2-D arrays are modeled by Markov process and then the transition probability matrix is calculated for each difference array. The entries of all the transition probability matrices are utilized as features to build up another part of the natural image model. In addition, a thresholding technique is developed to greatly reduce the dimensionality of the transition probability matrices, and hence the dimensionality of feature vectors, thus making the computational complexity manageable.

The general block diagram of Markov feature extraction is shown in Figure 6.

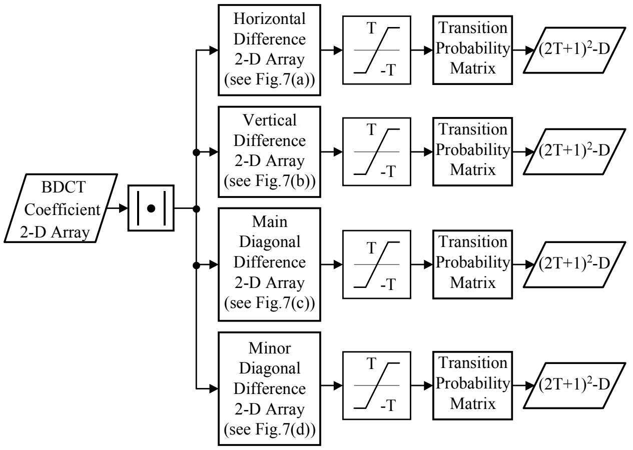

BDCT Coefficient 2-D Array
|•|
Horizontal Difference 2-D Array (see Fig.7(a))
Vertical Difference 2-D Array (see Fig.7(b))
Main Diagonal Difference 2-D Array (see Fig.7(c))
Minor Diagonal Difference 2-D Array (see Fig.7(d))
T
-T
Transition Probability Matrix
(2T+1)^2-D

Figure 6. General block diagram of Markov feature extraction procedure.

2.3.1 Difference 2-D Array

As indicated in Subsection 2.2.1, a prediction-error 2-D array is used to reduce the effect caused by the diversity of image content on splicing detection while popping out the splicing artifacts. Hence, by simply predicting an image pixel or a BDCT coefficient using its neighboring pixel or coefficient, it is expected that the disturbance caused by splicing operation can be emphasized by observing the prediction-error, i.e., the difference between an element and its neighbor in an image or BDCT 2-D array. Therefore, we introduce the difference 2-D array in this subsection.

Note that in Equations (8) to (11), u,v denote coordinates in either an image pixel 2-D array or a BDCT coefficient 2-D array; $F_h(u,v)$ stands for an element in the 2-D array with its subscript for one of four directions (horizontal (*h*), vertical (*v*), diagonal (*d*), and minor-diagonal (*m*)). For a given image, we can form the difference arrays by directly using Equations (8) to (11). For a BDCT coefficient 2-D array, we first round the coefficients to the nearest integers, then take their absolute values, and finally obtain the horizontal, vertical, main diagonal, and minor diagonal difference 2-D array $F_h(u,v)$, $F_v(u,v)$, $F_d(u,v)$, and $F_m(u,v)$ by applying Equations (8) to (11).

$$F_h(u,v) = F(u,v) - F(u+1,v)\,, \tag{8}$$

$$F_v(u,v) = F(u,v) - F(u,v+1)\,, \tag{9}$$

$$F_d(u,v) = F(u,v) - F(u+1,v+1)\,, \tag{10}$$

$$F_m(u,v) = F(u+1,v) - F(u,v+1)\,, \tag{11}$$

where $u \in [0, S_u - 2], v \in [0, S_v - 2]$, S_u, S_v denote the 2-D array's dimensions in the horizontal direction and vertical direction, respectively, and $F(u,v)$ is either the image pixel or the absolute value of rounded BDCT coefficient. The block diagrams of formation of these difference 2-D arrays (difference arrays for short) are shown in Figure 7.

As discussed at the second paragraph of Section 2, splicing operation changes the correlation between image pixels. According to random process theory, Markov random process (Markov process for short) is a tool to characterize the correlation. Instead of applying Markov process directly to image/coefficient 2-D array, we apply Markov process to the difference array introduced above. The reason of doing so is justified as follows.

2.3.2 Transition Probability Matrix Derived from Difference 2-D Arrays

As discussed at the second paragraph of Section 2, splicing operation changes the correlation between image pixels. According to random process theory, Markov random process is a tool to characterize the correlation. Instead of applying Markov process directly to image/coefficient 2-D array, we apply Markov process to the difference array introduced above. The reason of doing so is justified as follows.

Proverbially, there exists correlation between pixels/coefficients in an image/coefficient 2-D array. Therefore, the distribution of the elements in the difference array is somehow surrounding zero. The extent to which the distribution of the elements is concentrated on zero reflects the strength of the correlation among pixels/coefficients. According to the theory of random process, a transition probability matrix can be used to characterize a Markov process. In our model, we use the so called one-step transition probability matrix to characterize those difference arrays [16]. Hence, applying Markov process to the difference array leads to reduction of dimensionality of Markov transition probability matrix. Motivated by this observation, we apply Markov process to model the above-defined difference array in our proposed model.

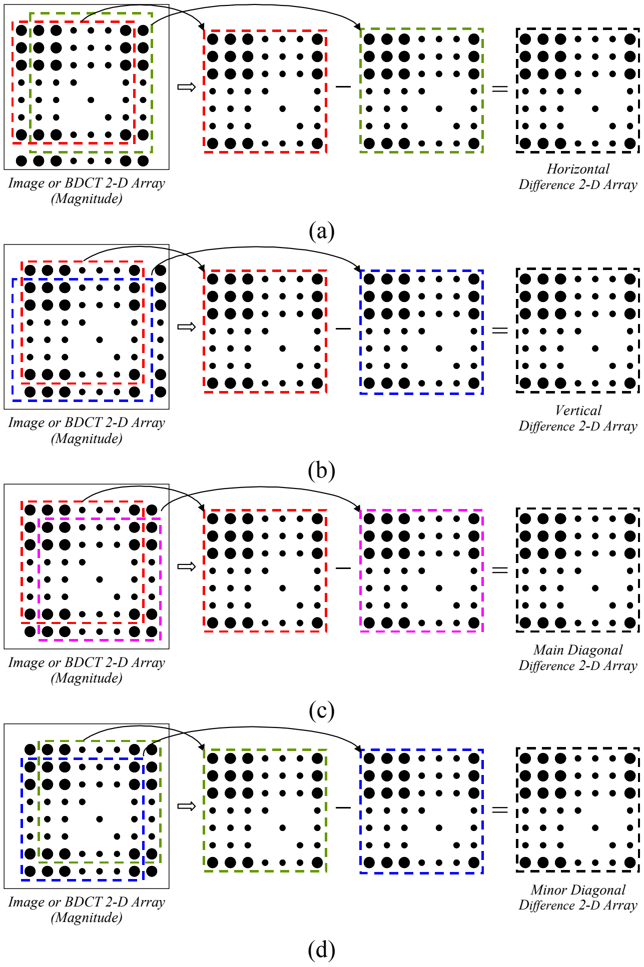


Image or BDCT 2-D Array (Magnitude)
Horizontal Difference 2-D Array
(a)
Image or BDCT 2-D Array (Magnitude)
Vertical Difference 2-D Array
(b)
Image or BDCT 2-D Array (Magnitude)
Main Diagonal Difference 2-D Array
(c)
Image or BDCT 2-D Array (Magnitude)
Minor Diagonal Difference 2-D Array
(d)

Figure 7. Difference 2-D array formation block diagram. (a): horizontal difference 2-D array formation; (b): vertical difference 2-D array formation; (c): main diagonal difference 2-D array formation; (d): minor diagonal difference 2-D array formation.

According to the probability theory [e.g., 16], it is well known that under the assumption that the pasted image fragment is additive to the host image, and the additive noise (i.e., the error between the spliced image and the original host image) is independent to the host image, the distribution of the spliced image is the convolution of the distribution of the host image and that of the additive (splicing) noise. If the distribution of the additive splicing noise is Gaussian-like, then the splicing on the image causes the concentration along the main diagonal of Markov transition probability matrix of the difference array to spread from the main diagonal towards the rest of the matrix. This statistical artifact can be utilized to separate spliced images from natural images.

In order to further reduce computational complexity, we resort to a thresholding technique. That is, if the value of an element in a difference array is either larger than T or smaller than -T, it will be represented by T or -T, respectively. This procedure results in a transition probability matrix of dimensionality (2T+1) × (2T+1). The elements of the matrix associated with the Markov process applied to the difference arrays are given by

$$p\{F_h(u+1,v)=n\,|\,F_h(u,v)=m\}=\frac{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_h(u,v)=m,F_h(u+1,v)=n)}{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_h(u,v)=m)}, \quad (12)$$

$$p\{F_v(u,v+1)=n\,|\,F_v(u,v)=m\}=\frac{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_v(u,v)=m,F_v(u,v+1)=n)}{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_v(u,v)=m)}, \quad (13)$$

$$p\{F_d(u+1,v+1)=n|F_d(u,v)=m\}=\frac{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_d(u,v)=m,F_d(u+1,v+1)=n)}{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_d(u,v)=m)}, \quad (14)$$

$$p\{F_m(u,v+1)=n|F_m(u+1,v)=m\}=\frac{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_m(u+1,v)=m,F_m(u,v+1)=n)}{\sum_{v=0}^{S_v-2}\sum_{u=0}^{S_u-2}\delta(F_m(u+1,v)=m)}, \quad (15)$$

where $m,n\in\{-T,-T+1,\cdots,0,\cdots,T\}$, and

$$\delta(A=m,B=n)=\begin{cases}1, & if\ A=m\ and\ B=n\\ 0, & Otherwise\end{cases}. \quad (16)$$

Note that all the elements of the transition probability matrix are used as features to form our natural image model.

2.4 Summary of Procedures in Generating Proposed Natural Image Model

We summarize how to generate our features used to characterize our proposed natural image model as follows:

1) Given a test image:
   a) Apply wavelet transform to this image and obtain all the subbands (we include the test image itself, which is denoted as LL_0 subband).
   b) Obtain histogram for each subband.
   c) Apply discrete Fourier transform to the histogram of each subband to obtain its characteristic function.
   d) Apply Equation (5) to calculate moments.
   e) Obtain prediction-error 2-D array from the given image.
   f) Repeat a) to d) to the prediction-error 2-D array.
   g) Obtain 2-D histograms for the given image.
   h) Apply 2-D discrete Fourier transform to each 2-D histogram to obtain the 2-D characteristic function.
   i) Apply Equation (7) to calculate marginal moments.
   j) Apply Equations (8) to (11) to the image to obtain difference 2-D arrays.
   k) Apply Equations (12) to (15) to obtain the elements of the Markov transition probability matrices.
2) Apply 2×2 BDCT to the given image, round those BDCT coefficients to the nearest integers, and then obtain the 2×2 BDCT coefficient 2-D array:
   a) Apply a) to i) in 1) to the BDCT 2-D array to obtain moment features.
   b) Take absolute value of the BDCT 2-D array.
   c) Apply j) to k) in 1) to the BDCT 2-D array (magnitude) to obtain the elements of the Markov transition probability matrices.

3) Increase the block size to the next larger block size, apply BDCT to the given image, round those BDCT coefficients to the nearest integers, and then obtain the BDCT coefficient 2-D array. Apply a) to c) in 2) to the BDCT 2-D array to obtain moment features and Markov features.

4) Repeat 3) until apply N×N BDCT to the given image, round those BDCT coefficients to the nearest integers, and then obtain the N×N BDCT coefficient 2-D array. Apply a) to c) in 2) to the N×N BDCT 2-D array to obtain moment features and Markov features.

Arrange the above-obtained features as a vector, which formulates our natural image model to represent the given image in our splicing detection scheme.

3. FEATURE EXTRACTION IN A CONCRETE IMPLEMENTATION

We have presented the general framework of our natural image model based approach to splicing detection in Section 2. In implementation, however, we have many choices according to the conditions and requirements of specific application scenario. In this section, we provide a concrete implementation of proposed approach, which is actually used in our experimental investigation reported in Section 4, where the Columbia Image Splicing Detection Evaluation Dataset [4] is utilized. This image dataset [4] has only 933 authentic images and 912 spliced images, each of which has a dimension of 128×128. These facts have determined the implementation of our proposed scheme reported in this section.

The most obvious constrain is that the limited number of total images available in the dataset [4] requires that the dimensionality of feature vectors generated according to our proposed model be bonded by the size of the dataset, i.e., the number of images in the dataset according to the theory of pattern recognition [22]. Based on this consideration, our concrete implementation using the dataset [4] is described below

3.1 Multi-size BDCT 2-D Arrays

In this implementation, we choose a set of different block sizes: 2×2, 4×4, and 8×8, partially because this choice (power of 2) is of computational benefits in implementing DCT. Furthermore, our experimental investigation on the image database [4] has shown that, when we choose block size as 2×2, 3×3, up to 8×8, i.e., seven BDCT 2-D arrays in total, the performance of the splicing analyzer does not improve much but evidently the feature dimensionality and computational cost rise dramatically. Our experimental investigation has also shown that, when we include 16×16 BDCT 2-D array, the performance of the splicing analyzer is almost the same as that without 16×16 BDCT. Some discussion in this regard, i.e., the selection of N, has been made at the end of Subsection 2.1. Further discussion will be made in Subsection 6.1.

As a result, we have in this concrete implementation the image pixel 2-D array, and three BDCT 2-D arrays, i.e., 2×2, 4×4, and 8×8 BDCT 2-D arrays, for each given test image.

3.2 Moment Based Features

3.2.1 Discrete Wavelet Transform (DWT)

Haar wavelet is used in our implementation due to its simplicity in implementation. Furthermore, to balance splicing detection capability and computational complexity, we only conduct one-level wavelet decomposition in this scheme and thus have five subbands (including the given image 2-D array, denoted as LL_0 subband). Compared to three-level DWT decomposition, the number of subbands reduces to 38%.

3.2.2 Moments and Marginal Moments

For each given image 2-D array, each of its MBDCT 2-D arrays, and their prediction-error 2-D arrays, after applying 1-level DWT, we compute the three lowest order moments using Equation (5). Hence, there are 15 resultant moments for each image 2-D array, each MBDCT 2-D array, and each prediction-error 2-D array.

When calculating the second-order histograms, we only use horizontal and vertical 2-D histograms (refer to Figure 5(a)). For each 2-D characteristic functions (the DFT of the 2-D histogram), we obtain the three lowest order marginal moments using Equation (7). As a result, there are 12 resultant marginal moments, i.e., 2 (two 2-D histograms) × 3 (three lowest order marginal moments) × 2 (two marginal moments each associated with one frequency variables in Equation (7)) = 12.

In summary, 42 moment features are obtained from each given image and each BDCT 2-D array. Since we have one image pixel 2-D array and three derived BDCT 2-D arrays, we have 42 × 4 = 168 moment features for each given image in this specific implementation.

3.3 Makov Based Features

To limit the dimensionality of feature vectors, in this specific implementation, we only apply the Markov feature extraction procedure to the 8×8 BDCT 2-D array of the given image to reduce the feature dimensionality. Experimental results show that Markov features extracted from the 8×8 BDCT 2-D array give better performance than that from the image 2-D array, the 2×2, or 4×4 BDCT 2-D array. Furthermore, we only use two directions in difference array formation, i.e., horizontal difference array and vertical difference 2-D array.

In order to reduce computational cost further, we set the threshold $T = 3$ in this implementation. That is, if the value of an element in the difference array is either larger than 3 or smaller than -3, it will be represented by 3 or -3, respectively. This procedure results in a transition probability matrix of dimensionality 7×7 = 49. Since we only use two difference 2-D arrays, the derived Markov features are of dimensionality 98. Further discussion on threshold selection is made in Subsection 6.2.

3.4 Summary of This Concrete Implementation

As mentioned in the beginning of this section, what is presented in this section is just ONE concrete implementation of our proposed scheme and for demonstration purpose. The principal idea of our proposed natural image model has been described in detail in Section 2.

The main concern of this concrete implementation is the limited number of images in the test image database [4]. Since this image database has only 933 authentic images and 912 spliced images, *we make many efforts to reduce our feature dimensionality to avoid trustless experimental results caused by too small ratio of training sample size to feature dimensionality* [22]. What we have done in this implementation to reduce the feature dimensionality is summarized as follows.

1) When generating the multi-size BDCT coefficient 2-D arrays, we only use the block size as 2×2, 4×4, and 8×8, instead of 2×2, 3×3, 4×4, 5×5, 6×6, 7×7, 8×8. We choose to use integer power of two. Furthermore, we use 2×2, 4×4, and 8×8 only, instead of using 2×2, 4×4, 8×8, 16×16, 32×32, and more.
2) When applying discrete wavelet transform, we conduct one-level Haar wavelet decomposition instead of more level decomposition and more complicated wavelet transforms.
3) When calculating 2-D histograms, we only compute horizontal 2-D histograms and vertical 2-D histograms instead of including diagonal and minor diagonal 2-D histograms.
4) When computing Markov features, we only apply Markov process to 8×8 BDCT coefficient 2-D array instead of including Markov features extracted from the given image pixel 2-D array, and its 2×2 and 4×4 BDCT coefficient 2-D arrays.
5) When forming difference 2-D array, we only use horizontal and vertical difference 2-D array instead of including the main diagonal and minor diagonal difference 2-D arrays.
6) When forming transition probability matrices, we set up a threshold T = 3.

In summary, we have 168 moment features and 98 Markov features, or 266 features in total in this concrete implementation, which are used in our experimental works reported in Section 4.

This implementation is illustrated in Figure 8.

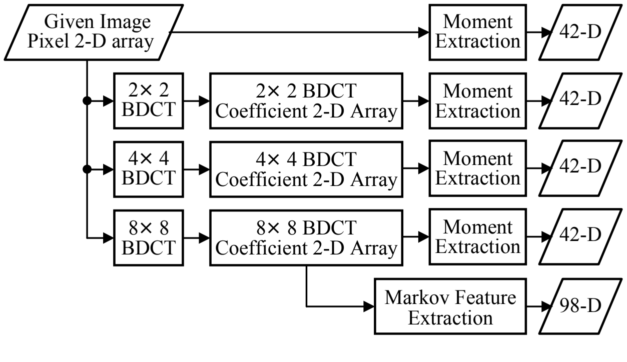


Figure 8. Feature extraction procedure in this concrete implementation.

4. EXPERIMENTS AND RESULTS

4.1 Image Dataset

The Columbia Image Splicing Detection Evaluation Dataset [4] is used in our experimental work by courtesy of DVMM, Columbia University. This dataset is created by DVMM, Columbia University for benchmarking the blind passive image splicing detection algorithms. Content diversity, source diversity, balanced distribution, and realistic operation are emphasized while this image data set is created. There are five image block types for the authentic and the spliced classes in this data set, i.e., image with an entirely homogeneous textured region, image with an entirely homogeneous smooth region, image with an object boundary between a textured region and a smooth region, image with an object boundary between two textured regions, and image with an object boundary between two smooth regions, respectively. Two kinds of splicing techniques are used: arbitrary-object-shaped splicing and straight line splicing. It is a data set open for downloading.

There are 933 authentic and 912 spliced images in this data set. Some authentic images and spliced images are shown in Figure 9 and Figure 10, respectively. For more information about the image set, readers are referred to the technical report [3].

Figure 9. Some sample authentic images used in this experimental work [4].

Figure 10. Some sample spliced images used in this experimental work [4].

4.2 Feature Generation

To evaluate the effectiveness of our proposed approach, we generate a 266-D feature vector using the implementation of feature extraction described in Section 3 for each authentic and each spliced image.

4.3 Classification

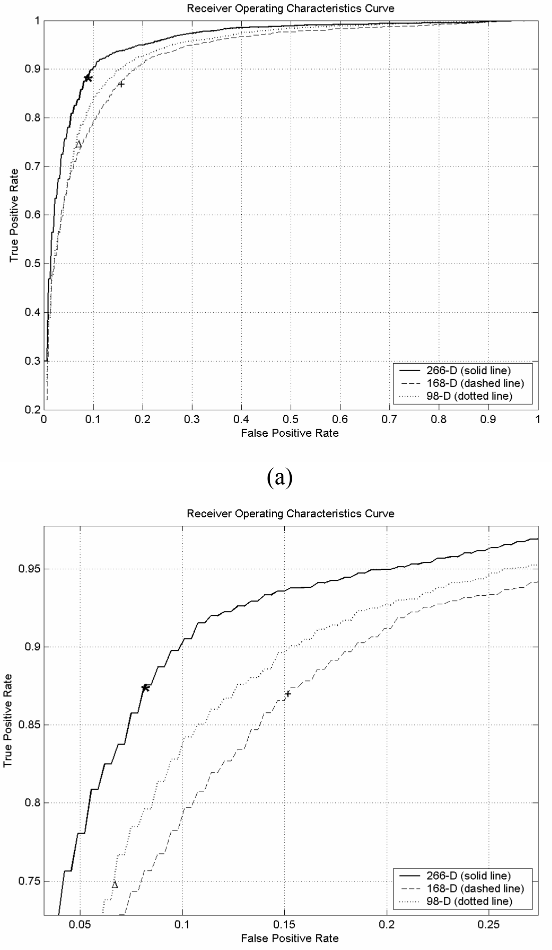
The support vector machine (SVM) is a kind of supervised machine learning method, which is widely used in pattern recognition applications. While simultaneously maximizing the geometric margin between two different classes, SVM can minimize the empirical classification error. SVM codes in Matlab can be downloaded from [5], which provides four basic kernels: linear, polynomial, radial basis function (RBF), and sigmoid. We use the RBF kernel in our reported experimental works.

In each experiment, randomly selected 5/6 of the authentic images and 5/6 of the spliced images are used to train a SVM classifier. Then the remaining 1/6 of the authentic images and 1/6 of the spliced images are used to test the trained classifier. The receiver operating characteristics (ROC) curve is obtained to demonstrate the performance of a trained classifier. Two numerical methods can be used to show the classifier's performance. One method is to calculate the area under the ROC curve, referred to as AUC. Readers are referred to [8] for more information of ROC and

AUC. Another method is to obtain detection rates, i.e., true negative (TN) rate, true positive (TP) rate, and accuracy of the trained classifier, which is the arithmetic average of TN and TP. To eliminate the effect of randomness incurred by image selection for training and testing, for each reported experimental result, we conduct the experiment 20 times by randomly select 5/6 images for training and the rest 1/6 for testing. The arithmetic average of these 20 independent random tests is then reported.

4.4 Experimental Results with Concrete Implementation

The averaged ROC curve of experiments using our proposed scheme is given in Figure 11 (the curve of 266-D in the figure, which is marked with an asterisk). The averaged detection rates and AUC are also given in Table 1. Compared to the prior arts [19, 12, 6], which achieve a detection accuracy of 72%, 80%, and 82%, respectively, the concrete implementation of the proposed approach has achieved an accuracy of 92%, making a significant advancement in splicing detection.



(a)

(b)

Figure 11. The ROC curves of the proposed splicing detection scheme. (a): the whole ROC's; (b): zoomed-in of the upper left part of (a). In this figure, 266-D means the concrete implementation including the moment features and Markov features, 168-D means only the moment features are used, and 98-D means only the Markov features are used.

Table 1. Experimental results using a concrete implementation (standard deviation among 20 random tests in parentheses)

Feature set	Proposed scheme
TN rate	91.00% (3.14%)
TP rate	92.76% (2.07%)
Accuracy	91.87% (1.80%)
AUC	0.9523 (0.0124)

4.5 Detecting Real Images

In Section 1, we have given two altered images and their associated originals (five images in total). We used the trained classifier resulted in the 20 random experiments mentioned in subsection 4.4 to test these five images (two altered and three authentic). The 20 test results are shown in Table 2. That is, among these 100 image-tests, 97 tests provided correct classification. These results are rather encouraging. No doubt, however, much more efforts need to be made for image splicing/tampering detection research in the future.

Table 2. Test results on real images (√— correct, × — wrong)

Test Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Figure 1(a)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Figure 1(b)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Figure 1(c)	√	×	√	√	√	√	√	×	√	√	√	√	√	√	√	√	√	√	×	√
Figure 2(a)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Figure 2(b)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

4.6 Applying Moment Features and Markov Features Individually

We also implement experiment with reduced dimensionality of feature vectors in order to examine the contributions made by moment features and Markov features alone in this implementation. The results are shown in Figure 11 and Table 3. In Figure 11, the 168-D curve, which is marked with a '+', corresponds to moment features and the 98-D curve, which is marked with a '□', corresponds to Markov features.

Table 3. Detection rate with moment and Markov feature sets separately (standard deviation in parentheses)

Feature sets	168-D moment features	98-D Markov features

TN rate	86.23% (2.97%)	86.61% (3.00%)
TP rate	87.43% (3.28%)	90.03% (2.63%)
Accuracy	86.82% (2.46%)	88.31% (1.85%)
AUC	0.9265 (0.0176)	0.9350 (0.0128)

It is seen that the contribution from the moment features is comparable to that from the Markov features. Independently applied, each of these two feature sets has outperformed the prior arts [19, 12, 6]. Comparing Table 1 and Table 3, we can observe that combining these two feature sets has further enhanced the splicing detection rate.

5. CONTRIBUTIONS OF MULTI-SIZE BDCT 2-D ARRAY

In this implementation, the features derived from statistical moments are of dimensionality 168. The first group of 42 feature components is derived from the image spatial 2-D array. The second, third, and fourth groups of 42 features are derived from BDCT coefficient 2-D arrays with block size 2×2, 4×4, and 8×8, respectively.

In this section, we demonstrate the effect of the multi-size BDCT coefficient 2-D array. As will be seen next, each BDCT 2-D array does contribute to the splicing analyzer and plays an indispensable role in splicing detection.

5.1 Using Moment Features from a Given Image and from MBDCT 2-D Array Individually

With the Columbia Image Splicing Detection Evaluation Dataset [4], we apply each of four groups of features independently. The results are given in Table 4.

From Table 4 one can see that each group of features, i.e., the moment features derived from a given image, and those from each BDCT 2-D array do contribute to splicing detection. Among the four groups, however, moment features derived from 8×8 BDCT 2-D array contributes most.

Table 4. Detection rate by applying four groups of moment features individually (standard deviation in parentheses)

Feature sets	From given image	From 2×2 BDCT	From 4×4 BDCT	From 8×8 BDCT
TN rate	74.48% (3.61%)	70.84% (3.56%)	77.10% (3.76%)	84.42% (3.27%)
TP rate	72.66% (3.90%)	76.05% (4.32%)	78.68% (6.47%)	80.26% (2.40%)
Accuracy	73.58% (2.32%)	73.42% (2.43%)	77.88% (3.87%)	82.36% (1.93%)
AUC	0.7802 (0.0260)	0.7893 (0.0276)	0.8110 (0.0347)	0.8727 (0.0216)

5.2 Using Moment Features Derived From Image and BDCT 2-D Arrays Incrementally

In Table 5, we give experimental results with incrementally adding a group of moment features at a time. In this table, from left to right, more and more groups of moment features are added. In table 5, “42-D” means only the first group of 42 features (derived from the given image 2-D array) is used, “84-D” means the first and the second groups of 42 feature components (from the 2×2 BDCT) are used, and so on.

It is observed that, the more the groups of features derived from BDCT 2-D arrays (up to 8×8 in this implementation) are included, the better the detection performance.

Table 5. Detection rate with gradually added moment feature sets (standard deviation in parentheses)

Feature sets	42-D	84-D	126-D	168-D
TN rate	74.48% (3.61%)	76.42% (3.47%)	81.13% (3.48%)	86.23% (2.97%)
TP rate	72.66% (3.90%)	76.88% (4.37%)	82.86% (3.62%)	87.43% (3.28%)
Accuracy	73.58% (2.32%)	76.65% (2.78%)	81.99% (2.27%)	86.82% (2.46%)
AUC	0.7802 (0.0260)	0.8336 (0.0260)	0.8738 (0.0225)	0.9265 (0.0176)

5.3 Experimental Results by Excluding Moment Features Derived from Three MBDCT 2-D Arrays Individually

We also conduct experiments where some group of moment features is not included. These results are given in Table 6. In Column 2, only the group of moment features derived from the 2×2 BDCT 2-D array is excluded. In Column 3, only the group of moment features from the 4×4 BDCT 2-D array is excluded. In Column 4, only the group of moment features from the 8×8 BDCT 2-D array is excluded.

It is observed that, when we exclude any group of moment features derived a BDCT 2-D array in this implementation, the detection performance deteriorates.

Table 6. Detection rate with some reduced moment feature set (standard deviation in parentheses)

Feature sets	No 2×2 BDCT	No 4×4 BDCT	No 8×8 BDCT
TN rate	86.39% (3.25%)	85.26% (3.50%)	81.13% (3.48%)
TP rate	83.91% (3.28%)	85.82% (3.44%)	82.86% (3.62%)
Accuracy	85.16% (2.56%)	85.54% (2.70%)	81.99% (2.27%)
AUC	0.9119 (0.0172)	0.9191 (0.0200)	0.8738 (0.0225)

In summary, as shown in this subsection, not only the moment features from each BDCT coefficient 2-D array do contribute to the success of the proposed splicing analyzer independently, but also the combined utilization of these features derived MBDCT 2-D arrays does enhance the splicing capability of the proposed scheme. In other words, each BDCT coefficient 2-D array in this implementation is indispensable.

6. SOME ISSUES IN IMPLEMENTATION

The implementation presented in Section 3 is a choice of trade-off between detection capability and computational complexity designed for the available image database [4]. In this section, we make some further discussions in this regard.

6.1 Choice of Block Size in Multi-size BDCT

In this implementation, the features of statistical moments are derived from the image 2-D array as well as the MBDCT 2-D arrays. Specifically, in the implementation of MBDCT, we only use block sizes: 2×2, 4×4, and 8×8. Given in Table 7 is a performance comparison of current implementation (i.e., 266-D features) to an implementation including 16×16 BDCT (thus, 308-D features). We can see that the performance of the splicing analyzer is not enhanced with addition of features derived from 16×16 BDCT 2-D array, though the feature size has increased obviously. This can be explained as the correlation between image pixels has become rather weak as the distance between pixels is rather large (given that the image sizes in dataset [4] are all of 128×128).

Table 7. Performance comparison: with vs without 16×16 BDCT (standard deviation in parentheses)

Feature sets	266-D features	308-D features
TN rate	91.00% (3.14%)	90.77% (3.03%)
TP rate	92.76% (2.07%)	92.04% (2.37%)
Accuracy	91.87% (1.80%)	91.40% (1.87%)
AUC	0.9523 (0.0124)	0.9530 (0.0123)

6.2 Choice of Threshold T

Another issue is the choice of the threshold T, which is used to reduce the Markov features' dimensionality. To select an appropriate T, the following points should be taken into consideration. The T cannot be too small. With a too small T, the elements of the transition probability matrix will not be able to sensitively catch the artifacts caused by splicing. On the other hand, this T cannot be too large. With a too large T, the dimensionality of the transition probability matrix will be too large, making computational complexity non-manageable, hence losing the meaning of using thresholding technique.

Table 8. Performance comparison: different choices of T (standard deviation in parentheses)

Feature sets	50-D features (T=2)	98-D features (T=3)	162-D features (T=4)
TN rate	87.58% (2.74%)	86.61% (3.00%)	87.61% (2.72%)
TP rate	79.70% (2.62%)	90.03% (2.63%)	90.69% (2.75%)
Accuracy	83.68% (2.06%)	88.31% (1.85%)	89.14% (1.50%)
AUC	0.8975 (0.0222)	0.9350 (0.0128)	0.9400 (0.0116)

In Table 8, we provide the performance of Markov features with three different T's, i.e., T = 2, 3, and 4, respectively. From this table, we can observe that the performances of T = 3 and T = 4 are comparable. In fact, when T is large enough ($T \geq 3$ in this case), most information related to image splicing operation that can be caught by Markov process is kept in the transition probability matrix. Therefore T = 3 is the best choice which balances the computational cost and detection rates.

7. CONCLUSION AND DISCUSSION

In this paper, we have proposed a natural image model to classify spliced images from authentic images. One concrete implementation of this splicing detection approach specifically designed for a publicly available splicing detection evaluation dataset has demonstrated its superior performance over the prior arts [19, 12, 6] on the same dataset by a significant margin. The proposed splicing detection approach can be summarized as follows:

1) The splicing procedure may cause changes in the smoothness, consistency, continuity, regularity, and periodicity of the authentic images, and therefore cause changes in the correlation between image pixels. Hence, an advanced natural image model will be able to catch the changes caused by splicing.

2) The proposed natural image model consists of the following aspects:

    a) It relies on image pixel 2-D array as well as multi-size block discrete cosine transform (MBDCT) coefficient 2-D arrays.

    b) For each of these 2-D arrays, its prediction-error 2-D array, and all of their wavelet subbands, statistical moments of the corresponding 1-D and 2-D characteristic functions are extracted as moment features.

    c) For each of these 2-D arrays, we round each element to integer, take absolute value (for an image pixel 2-D array, these operations result in the image pixel 2-D array itself), and then form difference 2-D arrays. From these difference 2-D arrays, elements of Markov transition probability matrix are calculated as features.

    d) These two parts of features form the feature vector for a given image.

    e) Support vector machine (SVM) with radial basis function (RBF) kernel can be used as classifier.

    f) The classifier is trained with a dataset of sufficiently large number of images before used as a classifier for splicing detection.

3) The multi-size block discrete cosine transform (MBDCT) is powerful in modeling images. It functions like the rake receivers, which are widely used in wireless communications, where each reflected signal contributes to the rake receiver to improve the SNR (signal to noise ratio) of the received signal. Here, each BDCT coefficient 2-D array with a specific block size contributes to the proposed splicing analyzer. Collectively, the features extracted from MBDCT's complement each other and thereby the

utilization of the MBDCT raises splicing detection capability significantly.

While the concrete implementation of our proposed new image model has obtained a high (higher than 90%) detection rate on the Columbia Image Splicing Detection Evaluation Dataset [4], and has demonstrated its promising in detection of two well-known actual splicing/tampering cases (refer to Figures 1 and 2), further research on effective features is necessary. Combining various effective features reported in the literature is another subject for future research. The goal is a practically powerful and blind image tampering detection system.

The Columbia Image Splicing Detection Evaluation Dataset [4] is scientifically designed for splicing detection and kindly made for public usage, hence, playing an important and active role in advancing image splicing detection research. At the initial stage, however, the size of the database and the size of the images in the database are limited. As the reported success detection rate on the database now reaches 92%, a larger, more diverse, and more realistic image database is called for splicing and tampering detection research.

8. ACKNOWLEDGMENTS

Credits for the use of the Columbia Image Splicing Detection Evaluation Dataset are given to the DVMM Laboratory of Columbia University, CalPhotos Digital Library, and the photographers, whose names are listed in [3].

9. REFERENCES

[1] http://www.snopes.com/photos/politics/kerry2.asp.

[2] http://news.bbc.co.uk/2/hi/middle_east/5254838.stm.

[3] http://www.ee.columbia.edu/dvmm/publications/04/TR_splicingDataSet_ttng.pdf.

[4] Columbia DVMM Research Lab. (2004) Columbia Image Splicing Detection Evaluation Dataset, http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSpliced DataSet.htm.

[5] Chang. C. C. and Lin, C. J. LIBSVM: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/cjlin/libsvm.

[6] Chen, W., Shi, Y. Q., and Su, W. Image Splicing Detection Using 2-D Phase Congruency and Statistical Moments of Characteristic Function. In: Delp, E. J., Wong, P.W. (eds.): *Security, Steganography and Watermarking of Multimedia Contents IX*, Proceeding. *of SPIE*, San Jose, CA, USA, 2007.

[7] Farid, H. and Lyu S. Higher-order wavelet statistics and their application to digital forensics. *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, WI, USA, 2003.

[8] Fawcett, T. *Roc Graphs: Notes and Practical Considerations for Researchers*. http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf.

[9] Fridrich, J. and Lukas, J. Digital Bullet Scratches for Images. In: Proceeding of *IEEE International Conference on Image Processing*, Genova, Italy, 2005.

[10] Fridrich., J., Lukas, J., and Goljan, M. Detecting Digital Image Forgeries Using Sensor Pattern Noise. In: Proceeding of *SPIE Electronic Imaging*, Photonics West, San Jose, CA, USA, 2006.

[11] Fridrich, J., Soukal D., and Lukas, J. Detection of Copy-Move Forgery in Digital Images. In: *Digital Forensic Research Workshop*, Cleveland, OH, USA, 2003.

[12] Fu, D., Shi, Y. Q., and Su, W. Detection of Image Splicing Based on Hilbert-Huang Transform and Moments of Characteristic Functions with Wavelet Decomposition. In: Shi, Y. Q., Jeon, B. (eds.): Digital Watermarking, Proceeding of 5th *International Workshop on Digital Watermarking*, Jeju Island, Korea, 2006.

[13] Huang, Y. Can Digital Image Forgery Detection Be Unevadable? A Case Study: Color Filter Array Interpolation Statistical Feature Recovery. In: Proceedings of *SPIE* – Volume 5960. *Visual Communications and Image Processing 2005*, Eds: Li., S., Pereira, F., Shum, H.-Y., Tescher, A. G., 59602W, San Jose, CA, USA, 2006.

[14] Johnson, M. K. and Farid, H. Exposing Digital Forgeries by Detecting Inconsistencies in Lighting. In: Proceeding of *ACM Multimedia and Security Workshop*, New York, NY, USA, 2005, pp. 1–9.

[15] Johnson, M. K. and Farid, H. Exposing Digital Forgeries Through Chromatic Aberration, *ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006.

[16] Leon-Garcia, A.: *Probability and Random Processes for Electrical Engineering*, 2nd Edition, Addison-Wesley Publishing Company, 1994.

[17] Mitrinovic, D. S., Pecaric, J. E., and Fink, A. M. *Classical and New Inequalities in Analysis*, Dordrecht, The Netherlands: Kluwer Academic Publishers, 1993.

[18] Ng, T.-T. and Chang, S.-F. A Model for Image Splicing. In *IEEE International Conference on Image Processing*, Singapore, 2004.

[19] Ng, T.-T., Chang, S.-F., and Sun, Q. Blind Detection of Photomontage Using Higher Order Statistics. *IEEE International Symposium on Circuits and Systems*, Vancouver, BC, Canada, 2004.

[20] Popescu, A. C. and Farid, H. *Exposing Digital Forgeries by Detecting Duplicated Image Regions*. Technical Report TR2004-515, Computer Science, Dartmouth College, 2004.

[21] Pratt, W. K. *Digital Image Processing*, 3rd Edition, John Wiley & Sons, Inc., 2001.

[22] Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd Edition, Springer, 1999.