

Detecting Doctored JPEG Images Via DCT Coefficient Analysis

Junfeng He¹, Zhouchen Lin², Lifeng Wang², and Xiaoou Tang²

¹ Tsinghua University, Beijing, China
heroson98@tsinghua.edu.cn

² Microsoft Research Asia, Beijing, China
{zhoulin, lfwang, xitang}@microsoft.com

Abstract. The steady improvement in image/video editing techniques has enabled people to synthesize realistic images/videos conveniently. Some legal issues may occur when a doctored image cannot be distinguished from a real one by visual examination. Realizing that it might be impossible to develop a method that is universal for all kinds of images and JPEG is the most frequently used image format, we propose an approach that can detect doctored JPEG images and further locate the doctored parts, by examining the double quantization effect hidden among the DCT coefficients. Up to date, this approach is the only one that can locate the doctored part automatically. And it has several other advantages: the ability to detect images doctored by different kinds of synthesizing methods (such as alpha matting and inpainting, besides simple image cut/paste), the ability to work without fully decompressing the JPEG images, and the fast speed. Experiments show that our method is effective for JPEG images, especially when the compression quality is high.

1 Introduction

In recent years, numerous image/video editing techniques (e.g. [1]-[12]) have been developed so that realistic synthetic images/videos can be produced conveniently without leaving noticeable visual artifacts (e.g. Figures 1(a) and (d)). Although image/video editing technologies can greatly enrich the user experience and reduce the production cost, realistic synthetic images/videos may also cause problems. The B. Walski event [17] is an example of news report with degraded fidelity. Therefore, developing technologies to judge whether the content of an image/video has been altered is very important.

Watermark [13] has been successful in digital right management (DRM). However, doctored image/video detection is a problem that is different from DRM. Moreover, plenty of images/videos are not protected by watermark. Therefore, watermark-independent technologies for doctored image/video detection are necessary, as pointed out in [14, 19]. Farid *et al.* have done some pioneering work on this problem. They proposed testing some statistics of the images that may be changed after tempering [14] (but did not develop effective algorithms that use these statistics to detect doctored images), including the interpolation relationship among the nearby pixels if resampling happens when synthesis, the double quantization (DQ) effect of two JPEG compression

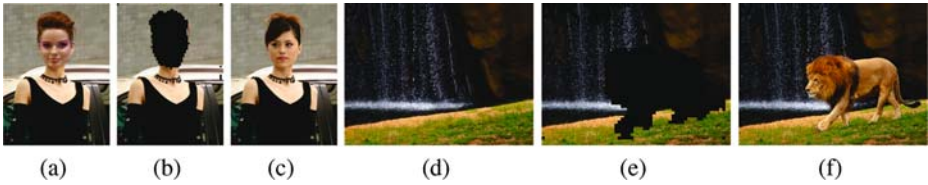


Fig. 1. Examples of image doctored and our detection results. (a) and (d) are two doctored JPEG images, where (a) is synthesized by replacing the face and (b) is by masking the lion and inpainting with structure propagation [9]. (b) and (e) are our detection results, where the doctored parts are shown as the black regions. For comparison, the original images are given in (c) and (f).

steps with different qualities before and after the images are synthesized, the gamma consistency via blind gamma estimation using the bicoherence, the signal to noise ratio (SNR) consistency, and the Color Filter Array (CFA) interpolation relationship among the nearby pixels [15]. Ng [18] improved the bicoherence technique in [14] to detect spliced images. But temporarily they only presented their work on testing whether a given 128×128 patch, rather than a complete image, is a spliced one or not. Lin et al. [19] also proposed an algorithm that checks the normality and consistency of the camera response functions computed from different selections of patches along certain kinds of edges. These approaches may be effective in some aspects, but are by no means always reliable or provide a complete solution.

It is already recognized that doctored image detection, as a *passive* image authentication technique, can easily have counter measures [14] if the detection algorithm is known to the public. For example, resampling test [14] fails when the image is further resampled after synthesis. The SNR test [14] fails if the same noise is added across the whole synthesized image. The blind gamma estimation [14] and camera response function computation [19] do not work if the forger synthesizes in the irradiance domain by converting the graylevel into irradiance using the camera response functions [19] estimated in the component images, and then applying a consistent camera response function to convert the irradiance back into graylevel. And the CFA checking [15] fails if the synthesized image is downsampled into a Bayer pattern and then demosaicked again. That is why Popescu and Farid conclude at the end of [14] that developing image authentication techniques will increase the difficulties in creating convincing image forgeries, rather than solving the problem completely. In the battle between image forgery and forgery detection, the techniques of both sides are expected to improve alternately.

To proceed, we first give some definitions (Figure 2). A “doctored” image (Figure 2(a)) means part of the content of a real image is altered. Note that this concept does not include those wholly synthesized images, e.g. an image completely rendered by computer graphics or by texture synthesis. But if part of the content of a real image is replaced by those synthesized or copied data, then it is viewed as “doctored”. In other words, that an image is doctored implies that it must contain two parts: the undoctored part and the doctored part. A DCT block (Figure 2(b)), or simply called a “block”, is a group of pixels in an 8×8 window. It is the unit of DCT that is used in JPEG. A DCT grid is the horizontal lines and the vertical lines that partition an image into blocks when doing JPEG compression. A doctored block (Figure 2(c)) refers to

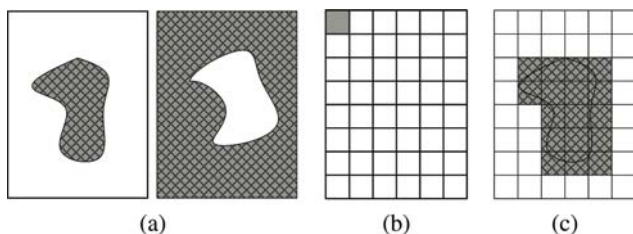


Fig. 2. Illustrations to clarify some terminologies used in the body text. (a) A doctored image must contain the undoctored part (blank area) and the doctored part (shaded area). Note that the undoctored part can either be the background (left figure) or the foreground (right figure). (b) A DCT block is a group of pixels in an 8×8 window on which DCT is operated when compression. A DCT block is also call a block for brevity. The gray block is one of the DCT blocks. The DCT grid is the grid that partition the image into DCT blocks. (c) A doctored block (shaded blocks) is a DCT block that is inside the doctored part or across the synthesis edge. An undoctored block (blank blocks) is a DCT block that is completely inside the undoctored part.

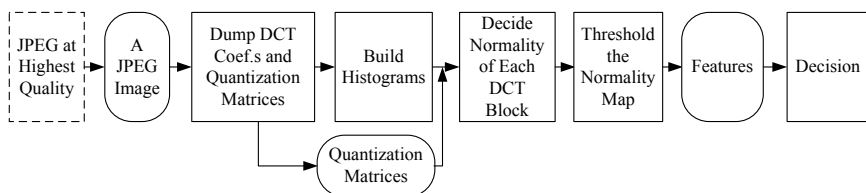


Fig. 3. The work flow of our algorithm

a block in the doctored part or along the synthesis edge and an undoctored block is a block in the undoctored part.

Realizing that it might be impossible to have a universal algorithm that is effective for all kinds of images, in this paper, we focus on detecting doctored JPEG images only, by checking the DQ effects (detailed in Section 2.2) of the double quantized DCT coefficients. Intuitively speaking, the DQ effect is the exhibition of periodic peaks and valleys in the histograms of the DCT coefficients. The reason we target JPEG images is because JPEG is the most widely used image format. Particularly in digital cameras, JPEG may be the most preferred image format due to its efficiency of compression. What is remarkable is that the doctored part can be automatically located using our algorithm. This capability is rarely possessed by the previous methods.

Although DQ effect is already suggested in [14, 20] and the underlying theory is also exposed in [14, 20], those papers actually only *suggested* that DQ effect can be utilized for image authentication: those having DQ effects are possibly doctored. This is not a strong testing as people may simply save the same image with different compression qualities. No workable algorithm was proposed in [14, 20] to tell whether an image is doctored or not. In contrast, our algorithm is more sophisticated. It actually detects the parts that *break* the DQ effect and deems this part as doctored.

Figure 3 shows the work flow of our algorithm. Given a JPEG image, we first dump its DCT coefficients and quantization matrices for YUV channels. If the

image is originally stored in other lossless format, we first convert it to the JPEG format at the highest compression quality. Then we build histograms for each channel and each frequency. Note that the DCT coefficients are of 64 frequencies in total, varying from (0,0) to (7,7). For each frequency, the DCT coefficients of all the blocks can be gathered to build a histogram. Moreover, a color image is always converted into YUV space for JPEG compression. Therefore, we can build at most $64 \times 3 = 192$ histograms of DCT coefficients of different frequencies and different channels. However, as high frequency DCT coefficients are often quantized to zeros, we actually only build the histograms of low frequencies of each channel. For each block in the image, using a histogram we compute one probability of its being a doctored block, by checking the DQ effect of this histogram (more details will be presented in Section 3.2). With these histograms, we can fuse the probabilities to give the normality of that block. Then the normality map is thresholded to differentiate the possibly doctored part and possibly undoctored part. With such a segmentation, a four dimensional feature vector is computed for the image. Finally, a trained SVM is applied to decide whether the image is doctored. If it is doctored, then the segmented doctored part is also output.

Our method has several advantages. First, it is capable of locating the doctored part automatically. This is a feature that is rarely possessed by the existing methods. The duplicated region detection [16] may be the only exception. But copying a part of an image to another position of the image is not a common practice in image forging. Second, most of the existing methods aim at detecting doctored images synthesized by the cut/paste skill. In contrast, our method could deal with images whose doctored part is produced by different kinds of methods such as inpainting, alpha matting, texture synthesis and other editing skills besides image cut/paste. Third, our algorithm directly analyzes the DCT coefficients without fully decompressing the JPEG image. This saves the memory cost and the computation load. Finally, our method is much faster than the bi-coherence based approaches [14, 18], iterative methods [14], and the camera response function based algorithm [19].

However, it is not surprising that there are cases under which our method does not work:

1. The original image to contribute the undoctored part is not a JPEG image. In this case the DQ effect of the undoctored part cannot be detected.
2. Heavy compression after image forgery. Suppose the JPEG compression quality of the real image is Q_1 , and after it is doctored, the new image is saved with compression quality of Q_2 . Generally speaking, the smaller Q_2/Q_1 is, the more invisible the DQ effect of the undoctored part is, hence the more difficult our detection is.

The rest of this paper is organized as follows. We first give the background of our approach in Section 2, then introduce the core part of our algorithm in Section 3. Next we present the experimental results in Section 4. Finally, we conclude our paper with discussions and future work in Section 5.

2 Background

2.1 The Model of Image Forgery and JPEG Compression

We model the image forgery process in three steps:

1. Load a JPEG compressed image I_1 .
2. Replace a region of I_1 by pasting or matting a region from another JPEG compressed image I_2 , or inpainting or synthesizing new content inside the region.
3. Save the forged image in any lossless format or JPEG. When detection, we will re-save the image as JPEG with quantization steps being 1 if it is saved in a lossless format¹.

To explain the DQ effect that results from double JPEG compression, we shall give a brief introduction of JPEG compression. The encoding (compression) of JPEG image involves three basic steps [14]:

1. Discrete cosine transform (DCT): An image is first divided into DCT blocks. Each block is subtracted by 128 and transformed to the YUV color space. Finally DCT is applied to each channel of the block.
2. Quantization: the DCT coefficients are divided by a quantization step and rounded to the nearest integer.
3. Entropy coding: lossless entropy coding of quantized DCT coefficients (e.g. Huffman coding).

The quantization steps for different frequencies are stored in quantization matrices (luminance matrix for Y channel or chroma matrix for U and V channels). The quantization matrices can be retrieved from the JPEG image. Here, two points need to be mentioned:

1. The higher the compression quality is, the smaller the quantization step will be, and vice versa;
2. The quantization step may be different for different frequencies and different channels.

The decoding of a JPEG image involves the inverse of the previous three steps taken in reverse order: entropy decoding, de-quantization, and inverse DCT (IDCT). Unlike the other two operations, the quantization step is not invertible as will be discussed in Section 2.2. The entropy encoding and decoding step will be ignored in the following discussion, since it has nothing to do with our method.

Consequently, when an image is doubly JPEG compressed, it will undergo the following steps and the DCT coefficients will change accordingly:

1. The first compression:
 - (a) DCT (suppose after this step a coefficient value is u).
 - (b) the first quantization with a quantization step q_1 (now the coefficient value becomes $Q_{q_1}(u) = [u/q_1]$, where $[x]$ means rounding x to the nearest integer).

¹ Note that most of the existing image formats other than JPEG and JPEG2000 are lossless.

2. The first decompression:

(a) dequantization with q_1 (now the coefficient value becomes $Q_{q_1}^{-1}(Q_{q_1}(u)) = \lceil u/q_1 \rceil q_1$).

(b) inverse DCT (IDCT).

3. The second compression:

(a) DCT.

(b) the second quantization with a quantization step q_2 (now the coefficient value u becomes $Q_{q_1 q_2}(u) = \lceil \lceil u/q_1 \rceil q_1/q_2 \rceil$).

We will show in the following section that the histograms of double quantized DCT coefficients have some unique properties that can be utilized for forgery detection.

2.2 Double Quantization Effect

The DQ effect has been discussed in [14], but their discussion is based on quantization with the floor function. However, in JPEG compression the rounding function, instead of the floor function, is utilized in the quantization step. So we provide the analysis of DQ effect based on quantization with the rounding function here, which can more accurately explain the DQ effect caused by double JPEG compression.

Denote h_1 and h_2 the histograms of DCT coefficients of a frequency *before* the first quantization and *after* the second quantization, respectively. We will investigate how h_1 changes after double quantization. Suppose a DCT coefficient in the u_1 -th bin of h_1 is relocated in a bin u_2 in h_2 , then

$$Q_{q_1 q_2}(u_1) = \left\lceil \left\lceil \frac{u_1}{q_1} \right\rceil \frac{q_1}{q_2} \right\rceil = u_2.$$

Hence,

$$u_2 - \frac{1}{2} \leq \left\lceil \frac{u_1}{q_1} \right\rceil \frac{q_1}{q_2} < u_2 + \frac{1}{2}.$$

Therefore,

$$\left\lceil \frac{q_2}{q_1} \left(u_2 - \frac{1}{2} \right) \right\rceil - \frac{1}{2} \leq \frac{u_1}{q_1} < \left\lfloor \frac{q_2}{q_1} \left(u_2 + \frac{1}{2} \right) \right\rfloor + \frac{1}{2},$$

where $\lceil x \rceil$ and $\lfloor x \rfloor$ denote the ceiling and floor function, respectively.

If q_1 is even, then

$$q_1 \left(\left\lceil \frac{q_2}{q_1} \left(u_2 - \frac{1}{2} \right) \right\rceil - \frac{1}{2} \right) \leq u_1 < q_1 \left(\left\lfloor \frac{q_2}{q_1} \left(u_2 + \frac{1}{2} \right) \right\rfloor + \frac{1}{2} \right).$$

If q_1 is odd, then

$$q_1 \left(\left\lceil \frac{q_2}{q_1} \left(u_2 - \frac{1}{2} \right) \right\rceil - \frac{1}{2} \right) + \frac{1}{2} \leq u_1 \leq q_1 \left(\left\lfloor \frac{q_2}{q_1} \left(u_2 + \frac{1}{2} \right) \right\rfloor + \frac{1}{2} \right) - \frac{1}{2}.$$

In either cases, the number $n(u_2)$ of the original histogram bins contributing to bin u_2 in the double quantized histogram h_2 depends on u_2 and can be expressed as:

$$n(u_2) = q_1 \left(\left\lfloor \frac{q_2}{q_1} \left(u_2 + \frac{1}{2} \right) \right\rfloor - \left\lceil \frac{q_2}{q_1} \left(u_2 - \frac{1}{2} \right) \right\rceil + 1 \right). \tag{1}$$

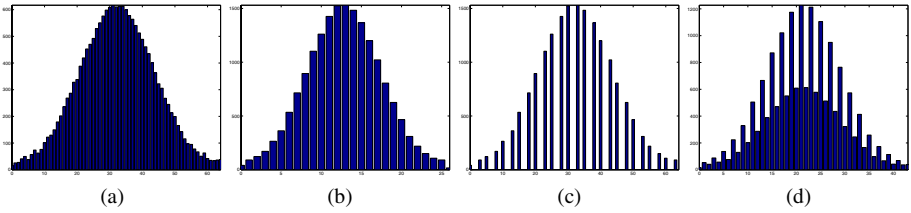


Fig. 4. The left two figures are histograms of single quantized signals with steps 2 (a) and 5 (b). The right two figures are histograms of double quantized signals with steps 5 followed by 2 (c), and 2 followed by 3 (d). Note the periodic artifacts in the histograms of double quantized signals.

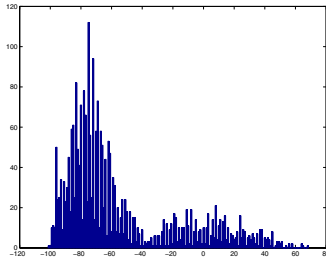


Fig. 5. A typical DCT coefficient histogram of a doctored JPEG image. This histogram can be viewed as the sum of two histograms. One has high peaks and deep valleys and the other has a random distribution. The first “virtual” histogram collects the contribution of undoctored blocks, while the second one collects the contribution of doctored blocks.

Note that $n(u_2)$ is a periodic function, with a period:

$$p = q_1 / \text{gcd}(q_1, q_2),$$

where $\text{gcd}(q_1, q_2)$ is the greatest common divider of q_1 and q_2 . This periodicity is the reason of the periodic pattern in histograms of double quantized signals (Figures 4(c) and (d) and Figure 5).

What is notable is that when $q_2 < q_1$ the histogram after double quantization can have periodically missing values (For example, when $q_1 = 5, q_2 = 2$, then $n(5k + 1) = 0$. Please also refer to Figure 4(c).), while when $q_2 > q_1$ the histogram can exhibit some periodic pattern of peaks and valleys (Figures 4(d) and 5). In both cases, it could be viewed as showing peaks and valleys periodically. This is called the *double quantization (DQ)* effect.

3 Core of Our Algorithm

3.1 DQ Effect Analysis in Doctored JPEG Images

Although DQ effect has been suggested for doctored image detection in [14, 20], by detecting the DQ effect from the spectrum of the histogram and using the DQ effect as the indicator of doctored images, [14, 20] actually did not develop a workable algorithm

for real-world doctored image detection. Since people may simply compress a real image twice with different quality, the presence of DQ effect does not necessary imply the existence of forgery of the image.

However, we have found that if we analyze the DCT coefficients more deeply and thoroughly, it will be possible for us to detect the doctored image, and even locate the doctored part automatically. Our idea is that: as long as a JPEG image contains both the doctored part and the undoctored part, the DCT coefficient histograms of the undoctored part will still have DQ effect, because this part of the doctored image is the same as that of the double compressed original JPEG image. But the histograms of doctored part will not have DQ effects. There are several reasons:

1. Absence of the first JPEG compression in the doctored part. Suppose the doctored part is cut from a BMP image or other kind of images rather than JPEG ones, then the doctored part will not undergo the first JPEG compression, and of course does not have DQ effect. Similarly, when the doctored part is synthesized by alpha matting or inpainting, or other similar skills, then the doctored part will not have DQ effect either.
2. Mismatch of the DCT grid of the doctored part with that of the undoctored part. Suppose the doctored part is cut from a JPEG image, or even the original JPEG image itself, the doctored part is still of little possibility to have DQ effect. Recall the description in Section 2.1, one assumption to assure the existence of DQ effect is that the DCT in the second compression should be just the inverse operation of IDCT in the first decompression. But if there is mismatch of the DCT grids, then the assumption is violated. For example, if the first block of a JPEG image, i.e. the block from pixel (0,0) to pixel (7,7), is pasted to another position of the same image, say to the position from pixel (18,18) to (25,25), then in the second compression step, the doctored part will be divided into four sub-blocks: block (18,18)-(23,23), block (24,18)-(25,23), block (18,24)-(23,25), and block (24,24)-(25,25). None of these sub-blocks can recover the DCT coefficients of the original block.
3. Composition of DCT blocks along the boundary of the doctored part. There is little possibility that the doctored part exactly consists of 8×8 blocks, so blocks along the boundary of the doctored part will consist of pixels in the doctored part and also pixels in the undoctored part. These blocks also do not follow the rules of DQ effect. Moreover, some post-processing, such as smoothing or alpha matting, along the boundary of the doctored part can also cause those blocks break the rules of DQ effect.

In summary, when the doctored part is synthesized or edited by different skills, such as image cut/past, matting, texture synthesis, inpainting, and computer graphics rendering, there might always exist one or more reasons, especially the last two, that cause the absence of DQ effect in the doctored part. Therefore, the histogram of the whole doctored JPEG image could be regarded as the superposition of two histograms: one has periodical peaks and valleys, and the other has random bin values in the same period. They are contributed by the undoctored part and the doctored part, respectively. Figure 5 shows a typical histogram of a doctored JPEG image.

3.2 Bayesian Approach of Detecting Doctored Blocks

From the analysis in Section 3.1, we know that doctored blocks and undoctored blocks will have different possibility to contribute to the same bin in one period of a histogram h . Suppose a period starts from the s_0 -bin and ends at the $(s_0 + p - 1)$ -th bin, then the possibility of an undoctored block which contributes to that period appearing in the $(s_0 + i)$ -bin can be estimated as:

$$P_u(s_0 + i) = h(s_0 + i) / \sum_{k=0}^{p-1} h(s_0 + k), \tag{2}$$

because it tends to appear in the high peaks and the above formula indeed gives high values at high peaks. Here, $h(k)$ denotes the value of the k -th bin of the DCT coefficient histogram h . On the other hand, the possibility of a doctored block which contributes to that period appearing in the bin $(s_0 + i)$ can be estimated as:

$$P_d(s_0 + i) = 1/p, \tag{3}$$

because its distribution in one period should be random. From the naive Bayesian approach, if a block contributes to the $(s_0 + i)$ -th bin, then the posteriori probability of it being a doctored block or an undoctored block is:

$$P(\text{doctored} | s_0 + i) = P_d / (P_d + P_u), \text{ and} \tag{4}$$

$$P(\text{undoctored} | s_0 + i) = P_u / (P_d + P_u), \tag{5}$$

respectively.

In the discussion above, we need to know the period p in order to compute P_u or P_d . It can be estimated as follows. Suppose s_0 is the index of the bin that has the largest value. For each p between 1 and $s_{max}/20$, we compute the following quantity:

$$H(p) = \frac{1}{i_{max} - i_{min} + 1} \sum_{i=i_{min}}^{i_{max}} [h(i \cdot p + s_0)]^\alpha,$$

where $i_{max} = \lfloor (s_{max} - s_0) / p \rfloor$, $i_{min} = \lceil (s_{min} - s_0) / p \rceil$, s_{max} and s_{min} are the maximum and minimum index of the bins in the histogram, respectively, and α is a parameter (can be simply chosen as 1). $H(p)$ evaluates how well the supposed period p gathers the high-valued bins. The period p is finally estimated as: $p = \arg \max_p H(p)$. If $p = 1$, then this histogram suggests that the JPEG image is single compressed. Therefore, it cannot tell whether a block is doctored or not and we should turn to the next histogram.

If $p > 1$, then each period of the histogram assigns a probability to every block that contributes to the bins in that period, using equation (4). And this is done for every histogram with estimated period $p > 1$. Consequently, we obtain a normality map of blocks of the image under examination, each pixel value of which being the accumulated posterior probabilities.

3.3 Feature Extraction

If the image is doctored, we expect that low normality blocks cluster. Any image segmentation algorithm can be applied to do this task. However, to save computation, we simply threshold the normality map by choosing a threshold:

$$T_{opt} = \arg \max_T (\sigma / (\sigma_0 + \sigma_1)), \quad (6)$$

where given a T the blocks are classified into two classes C_0 and C_1 , σ_0 and σ_1 are the variances of the normalities in each class, respectively, and σ is the squared difference between the mean normalities of the classes. The formulation of (6) is similar to the Fisher discriminator in pattern recognition.

With the optimal threshold, we expect that those blocks in class C_0 (i.e. those having normalities below T_{opt}) are doctored blocks. However, this is still insufficient for confident decision because any normality map can be segmented in the above manner. However, based on the segmentation, we can extract four features: T_{opt} , σ , $\sigma_0 + \sigma_1$, and the connectivity K_0 of C_0 . Again, there are many methods to define the connectivity K_0 . Considering the computation load, we choose to compute the connectivity as follows. First the normality map is median filtered. Then for each block i in C_0 , find the number e_i of blocks in class C_1 in its 4-neighborhood. Then $K_0 = \sum_i \max(e_i - 2, 0) / N_0$,

where N_0 is the number of blocks in C_0 . As we can see, the more connected C_0 is, the smaller K_0 is. We use $\max(e_i - 2, 0)$ instead of e_i directly because we also allow narrowly shaped C_0 : if e_i is used, round shaped C_0 will be preferred.

With the four-dimensional feature vector, i.e. T_{opt} , σ , $\sigma_0 + \sigma_1$, and K_0 , we can safely decide whether the image is doctored by feeding the feature vector into a trained SVM. If the output is positive, then C_0 is decided as the doctored part of the image.

4 Experiments

The training and evaluation of a doctored image detection algorithm is actually quite embarrassing. If the images are donated by others or downloaded from the web, then we cannot be completely sure about whether they are doctored or original because usually we cannot tell them by visual inspection. Even the donator claims that s/he does not make any change to the image, as long as the image is not produced by him or her, it is still unsafe. To have a large database, maybe the only way is to synthesize by ourselves, using the images that are also captured by ourselves. However, people may still challenge us with the diversity of the doctored techniques and the doctored images. Therefore, temporarily maybe the best way is to present many detection results that we are sure about the ground truth.

We synthesized 20 images using the Lazy Snapping tool [11], the Poisson Matting tool [8], the image completion tool [9], and the image inpainting tool (it is a part of the image completion tool), and trained an SVM using these images. Then we apply our algorithm and the SVM to detect the images that are contributed by authors of some Siggraph papers. As we believe in their claims that they are the owner of the images, we take their labelling of doctored or undoctored as the ground truth.

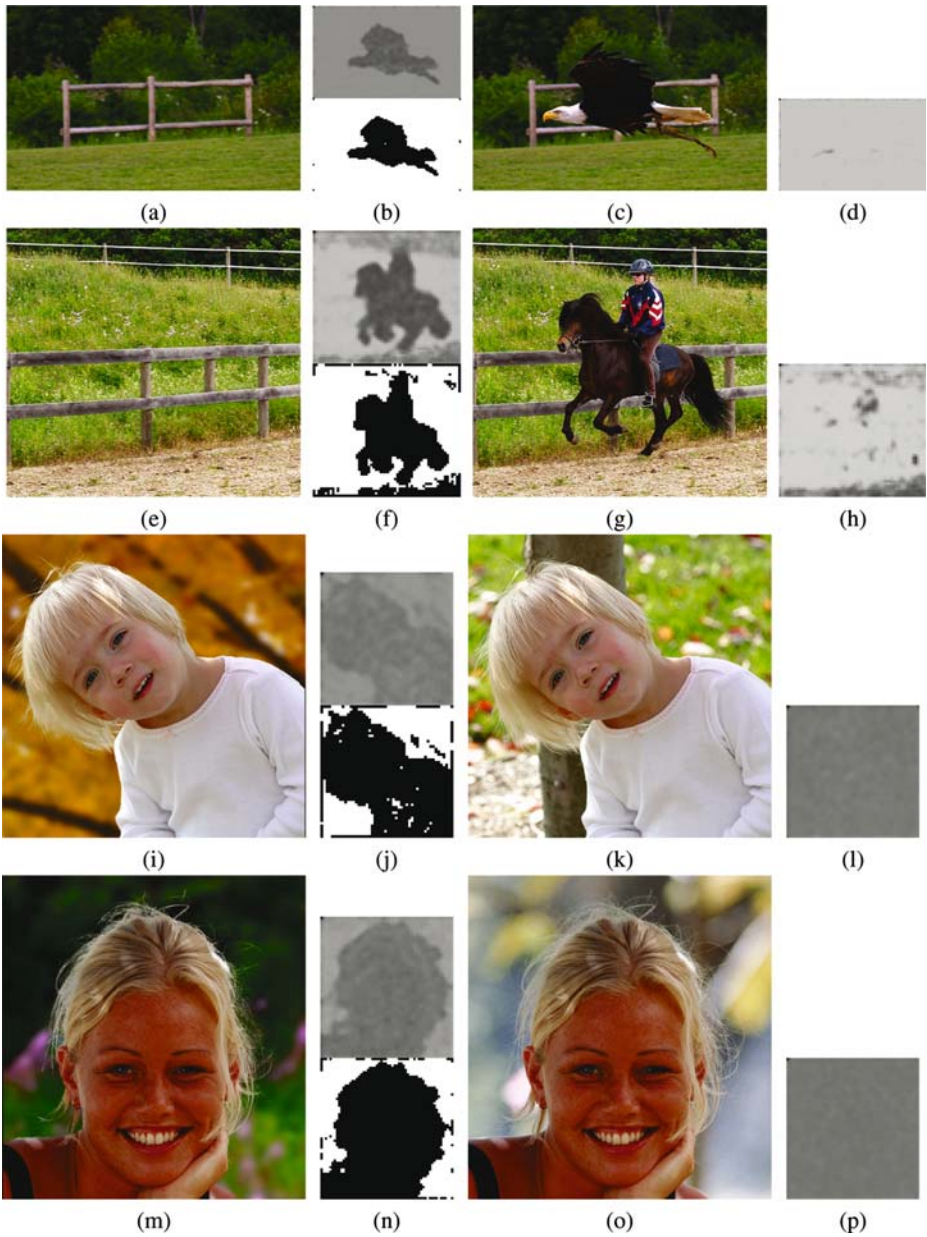


Fig. 6. Some detection results of our algorithm. The images are all taken from Siggaph papers. The first two images are doctored by inpainting. The last two images are doctored by matting. The left columns are the doctored images. The third column are the original images. The normality maps and the masks of doctored parts are shown in the middle column. For comparison, the normality maps of original images are also shown on the right-most column. Visual examination may fail for these images.

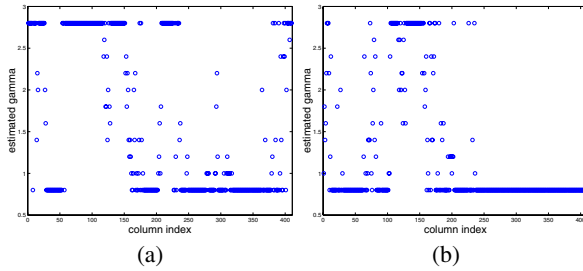


Fig. 7. The estimated column-wise gammas using the blind gamma estimation algorithm in [14]. (a) and (b) correspond to Figures 6(i) and (k), respectively. The horizontal axis is the column index and the vertical axis is the gamma value. The gamma is searched from 0.8 to 2.8 with a step size 0.2. By the methodology in [14], Figure 6(k) is more likely to be classified as doctored than Figure 6(i) is because the gamma distribution in (b) is more abnormal than that in (a).

Figure 6 shows some examples of successful detection. Given the doctored images shown in the first column, human inspection may fail. However, our algorithm can detect the doctored parts almost correctly. In comparison, the normalities of the original images do not show much variance.

Our algorithm is fast. Analyzing an image of a size 500×500 only requires about 4 seconds on our Pentium 1.9GHz PC, with unoptimized codes. For comparison, Figures 7 (a) and (b) show the estimated gammas for each column of Figures 6(i) and (k), respectively, using the blind gamma estimation algorithm proposed in [14]. Our algorithm only took 4.1 seconds to analyze Figure 6(i) or (k) and gave the correct results, while the blind gamma estimation algorithm [14] took 610 seconds but the detection was still erroneous.

5 Discussions and Future Work

With the improvement of image/video editing technologies, realistic images can be synthesized easily. Such eye-fooling images have caused some problems. Thus it is necessary to develop technologies that detect or help us detect those doctored images. Observing that JPEG is the most frequently used image format, especially in digital cameras, we have proposed an algorithm for doctored JPEG image detection by analyzing the DQ effects hidden among the histograms of the DCT coefficients. The four advantages possessed by our algorithm, namely automatic doctored part determination, resistant to different kinds of forgery techniques in the doctored part, ability to work without full decompression, and fast detection speed, make our algorithm very attractive.

However, more investigations are still needed to improve our approach. For example, a more accurate definition of (2) should be:

$$P_u(s_0 + i) = n(s_0 + i) / \sum_{k=0}^{p-1} n(s_0 + k).$$

But we need to know q_1 and q_2 in order to compute $n(k)$ according to (1). Actually q_2 can be dumped from the JPEG image. Unfortunately, q_1 is lost after the first

decompression and hence has to be estimated. Although Lukas and Fridrich [20] have proposed an algorithm to estimate the first quantization matrix, the algorithm is too restrictive and may not be reliable. Hence we are exploring a simple yet practical method to estimate q_1 . Moreover, since counter measures can be easily designed to break our detection (e.g. resizing the doctored JPEG image or compressing the doctored image heavily after synthesis), we still have to improve our algorithm by finding more robust low-level cues.

Acknowledgment. The authors would like to thank Dr. Yin Li, Dr. Jian Sun, and Dr. Lu Yuan for sharing us test images, Mr. Lincan Zou for collecting the training samples, and Dr. Yuwen He and Dr. Debing Liu for providing us the code to dump the DCT coefficients and the quantization matrices in the JPEG images.

References

1. A. Agarwala *et al.* Interactive Digital Photomontage. *ACM Siggraph 2004*, pp. 294-301.
2. W.A. Barrett and A.S. Cheney. Object-Based Image Editing. *ACM Siggraph 2002*, pp. 777-784.
3. Y.-Y. Chuang *et al.* A Bayesian Approach to Digital Matting. *CVPR 2001*, pp.II: 264-271.
4. V. Kwatra *et al.* Graphcut Textures: Image and Video Synthesis Using Graph Cuts. *ACM Siggraph 2003*, pp. 277-286.
5. C. Rother, A. Blake, and V. Kolmogorov. Grabcut - Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Siggraph 2004*, pp. 309-314.
6. Y.-Y. Chuang *et al.* Video Matting of Complex Scenes. *ACM Siggraph 2002*, pp. 243-248.
7. P. Pérez, M. Gangnet, and A. Blake. Poisson Image Editing. *ACM Siggraph 2003*, pp. 313-318.
8. J. Sun *et al.* Poisson Matting. *ACM Siggraph 2004*, pp. 315-321.
9. J. Sun, L. Yuan, J. Jia, H.-Y. Shum. Image Completion with Structure Propagation. *ACM Siggraph 2005*, pp. 861-868.
10. Y. Li, J. Sun, H.-Y. Shum. Video Object Cut and Paste. *ACM Siggraph 2005*, pp. 595-600.
11. Y. Li *et al.* Lazy Snapping. *ACM Siggraph 2004*, pp. 303-308.
12. J. Wang *et al.* Interactive Video Cutout. *ACM Siggraph 2005*, pp. 585-594.
13. S.-J. Lee and S.-H. Jung. A Survey of Watermarking Techniques Applied to Multimedia. *Proc. 2001 IEEE Int'l Symp. Industrial Electronics (ISIE2001)*, Vol. 1, pp. 272-277.
14. A.C. Popescu and H. Farid. Statistical Tools for Digital Forensics. *6th Int'l Workshop on Information Hiding*, Toronto, Canada, 2004.
15. A.C. Popescu and H. Farid. Exposing Digital Forgeries in Color Filter Array Interpolated Images. *IEEE Trans. Signal Processing*, Vol. 53, No. 10, pp. 3948-3959, 2005.
16. A.C. Popescu and H. Farid. Exposing Digital Forgeries by Detecting Duplicated Image Regions. Technical Report, TR2004-515, Dartmouth College, Computer Science.
17. D.L. Ward. Photostop. Available at: <http://angelingo.usc.edu/issue01/politics/ward.html>
18. T.-T. Ng, S.-F. Chang, and Q. Sun. Blind Detection of Photomontage Using Higher Order Statistics. *IEEE Int'l Symp. Circuits and Systems (ISCAS)*, Vancouver, Canada, May 2004, pp. 688-691.
19. Z. Lin, R. Wang, X. Tang, and H.-Y. Shum. Detecting Doctored Images Using Camera Response Normality and Consistency, *CVPR 2005*, pp.1087-1092.
20. J. Lukas and J. Fridrich. Estimation of Primary Quantization Matrix in Double Compressed JPEG Images, *Proc. Digital Forensic Research Workshop 2003*.