Chapter 20
Image Forgery Detection

Learning Objectives
The objectives of this chapter are to:

- Understand digital image processing fundamental
- Explore about image forgery detection techniques
- Perform passive-blind image forgery detection techniques

Digital images have been around for decades, stemming from capturing events, people, and places on photos to steaming HD picture feeds on our computers. It's a technology that becomes heavily used to communicate online like sharing special moments with friends who are far away from the place that they live. As a result, digital media especially digital images, are now the primary sources of the news, entertainment, and information. In fact, digital images are also increasingly used as evidences for crime or proves for malignant action in a court of law, as part of crime or medical records. Yet despite how attractive the use of digital image can be, it still faces with question of its credibility, since nowadays even a novice person can digitally manipulate an image to alter the message contained in the visual media with widely available software. There is an old saying: "Seeing is believing". However, it is not true anymore in today's digital era. Images come up all the time in our daily lives, but we have no way of knowing if they are portraying the truth or not. Consequently, image forgery detection technologies for verifying the integrity of images and detecting traces of tampering in many fields, such as maintaining judicial notarization and news integrity, turns to be an important research field.

Image forgery detection is based on digital image processing from the statistical characteristics and formation mechanism. Generally, digital image processing focuses on two major tasks: Improvement of image quality for human interpretation; processing of image data for storage, transmission, display and representation for automatic machine perception. In this chapter, we first discuss fundamentals of

X. Lin, *Introductory Computer Forensics*,
https://doi.org/10.1007/978-3-030-00581-8_20

digital image processing. Then, various image tampering techniques are classified. Finally, we learn Image forgery detection techniques including both active and passive detection.

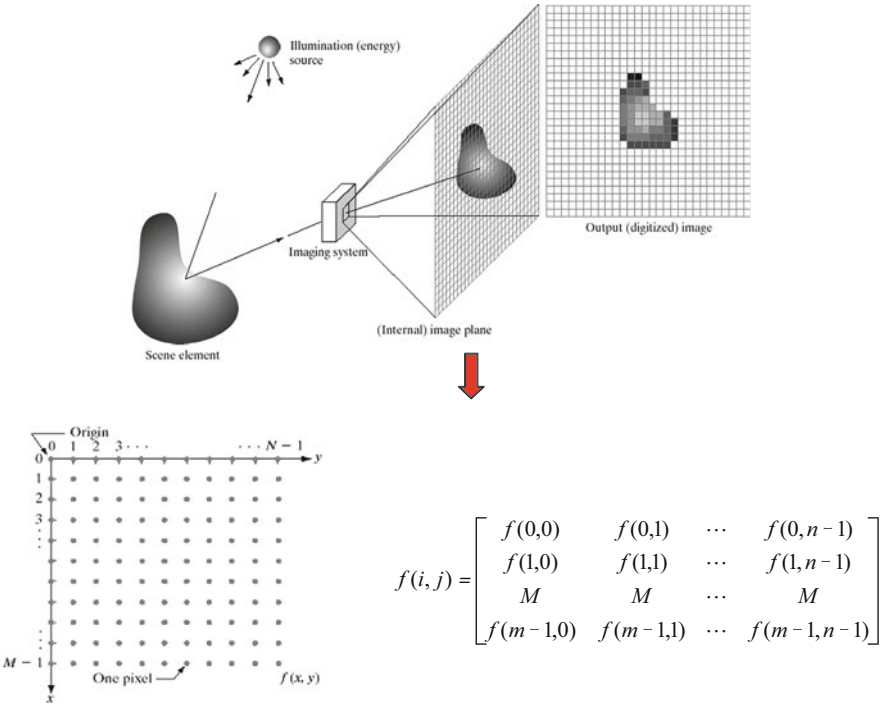
20.1 Digital Image Processing Fundamentals

In this section, digital image basis including basic concept, basic operation and transform are presented.

20.1.1 Digital Image Basis

20.1.1.1 Image and Pixel

An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point, as shown in Fig. 20.1. Each



$$f(i,j) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,n-1) \\ f(1,0) & f(1,1) & \cdots & f(1,n-1) \\ M & M & \cdots & M \\ f(m-1,0) & f(m-1,1) & \cdots & f(m-1,n-1) \end{bmatrix}$$

Fig. 20.1 The presentation of image

element, having a particular location and value, is called pixel. Pixel values typically represent gray levels, colors, opacities, etc..

20.1.1.2 Spatial Resolution: $M * N$

Intuitively, the spatial resolution of an image is a measure of the smallest discernible detail that an image holds. There are a lot of measures of explaining the image resolution in quantity, such as line pairs per unit distance and dots per inch. Spatial resolution determines the storage size of an image (bytes) and image sharpness. Imaging that we can describe an image with alternate dark and light lines, the image resolution quantifies how much the two kinds of lines can be close to each other and still be visibly resolved. The resolution can be specified as number of line pairs per unit distance, say ten line pairs per mm or five line pairs per mm. Another measure of image resolution is dots per inch, i.e., the number of discernible dots per inch.

Notably, the spatial resolution makes sense only when it is related with the spatial distance. If the numbers of pixels are increased for a fixed size of the physical display area, then the spatial resolution improves. Normally, if there is no need to measure the physical resolution of pixels, we usually call an M*N digital image with a spatial resolution of M*N pixels.

20.1.1.3 Gray Intensity Level Resolution: L

Gray intensity level resolution of an image refers to the smallest possible intensity which can be distinguished in an image grayscale. With the gradual decrease of the gray level resolution of an image, the number of colors in the image becomes less, which results in the loss of the image color information and the expression of the details of the image. The number of gray intensity level is commonly chosen as an integer power of 2 (commonly chosen as $L = 256$). Gray intensity level means the number of bits used to quantify gray levels, for example, normally, a display capable of an 8-bit image has the capability to quantize the gray intensity or color intensity in fixed increments of 1/256 units of intensity value.

Usually, the M and N are positive integers, and the number of gray levels is an integer power of 2 (Table 20.1):

$$L = 2^{k} \tag{20.1}$$

Table 20.1 L-level digital image of size $M \times N$

Parameter	Symbol	Typical values
Rows	N	256,512,525,625,1024,1035
Columns	M	256,512,768,1024,1320
Gray Levels	L	2,64,256,1024,4096,16384

$$b = M \times N \times k(Bit) \tag{20.2}$$

Both spatial and gray level resolutions determine the storage size of an image (bytes).

20.1.1.4 Image Sampling and Quantization

To create a digital image, we need to convert the continuous sensed data into digital form. This involves two steps: **Sampling** and **quantization**. Digitizing the coordinate values is called sampling, and digitizing the amplitude values is called quantization.

The spatial resolution of an image is determined by how sampling was carried out. The more intensity levels are used, the finer detail discernable level of an image will be. Intensity level resolution is usually given in terms of the number of bits used to store each intensity level.

Quantization is the process of converting a continuous analogue signal into a digital representation. It involves representing the sampled data by a finite number of levels based on some criteria such as minimization of the quantize distortion. Quantize design includes input (decision) levels and output (reconstruction) levels as well as number of levels. The decision can be enhanced by psycho visual or psychoacoustic perception. Quantizes can be classified as memory less (Each sample is quantized independently) or with memory (It takes into account previous sample).

The quality of digital images largely depends on the number of samples and gray levels used in sampling and quantization. Generally, when limiting the size of a digital image, the following principles may be used in order to obtain a better quality images:

- For the images with slow changes, we should adopt coarse sampling and fine quantization to avoid false contours.
- For the images with rich details, fine sampling and coarse quantization are better choice to avoid aliasing.

20.1.2 Image Types

20.1.2.1 Binary Image

Binary image also called black and white image, that is to say the pixel values are just 0 and 1 in the image. Usually the value 0 represents black and 1 represents white. Figure 20.2 is one binary image.

Fig. 20.2 An example of a binary image

Fig. 20.3 An example of a grayscale image

20.1.2.2 Grayscale Image

Compared with the binary image, there are more gray scales in the gray scale image. For example, when the gray scales contain 8 bits and the gray scale image' gray scales are 256(2^8). So, each gray value ranges from 0 to 255 in the gray scale image. Also the 0 represents black and 1 represents white, and other values represent different grays.

Figure 20.3a is the gray scales bar, and Fig. 20.3b is one gray scale image.

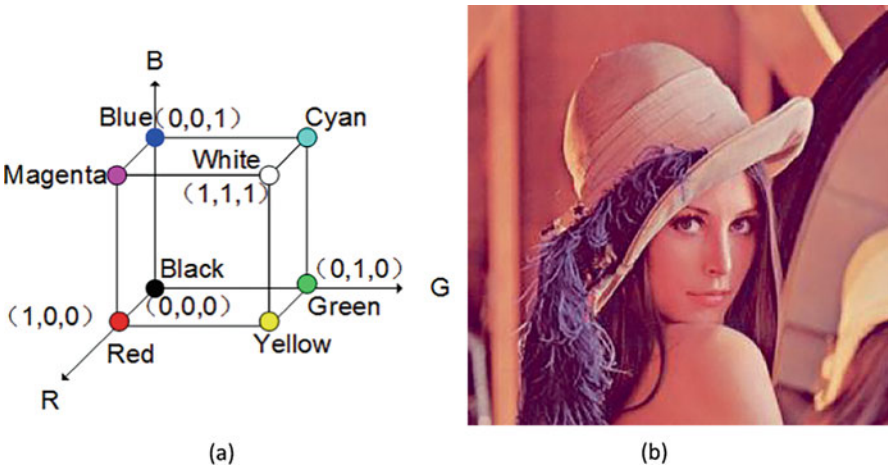


Fig. 20.4 RGB image. (**a**) The model of RGB system. (**b**) An example of a chromatic image

20.1.2.3 RGB Image

The CIE (Commission Internationale de L'Eclairage) developed the CIE color system (CIE Color System) as the basis for other color system, which choses three colors, red (wavelength $\lambda = 700.0$ nm), green ($\lambda = 546.1$ nm), and blue ($\lambda = 438.8$ nm), as the primary colors. The other colors can be represented by superimposing different proportion of the basic colors, which is expressed as:

$$C = R(R) + G(G) + B(B). \tag{20.3}$$

Figure 20.4a is the model of RGB system. In the RGB image, the intensity value of each pixel is superimposed in a fixed proportion of the basic colors. For example, one chromatic image is 8 bits per channel so that the intensity value of every basic color ranges from 0 to 255. In a red picture, the red color intensity value of each pixel is 255 and the intensity values of green and blue are 0. Figure 20.4b is a chromatic image.

20.1.3 Basic Operation and Transform

First, we introduce an example for a known waveform $\sin(3\pi x) + \sin(5\pi x)$. Notably, if we need remove the waveform $\sin(5\pi x)$ from the original waveform and get the waveform $\sin(3\pi x)$ without giving exact expression, it is very difficult to achieve in the time domain as shown in Fig. 20.5. But when $\sin(3\pi x) + \sin(5\pi x)$ is transformed into to the frequency domain by using Fourier Transformation (FT), it is a simple task to reserve any wave the expression, as illustrated in Fig. 20.6.

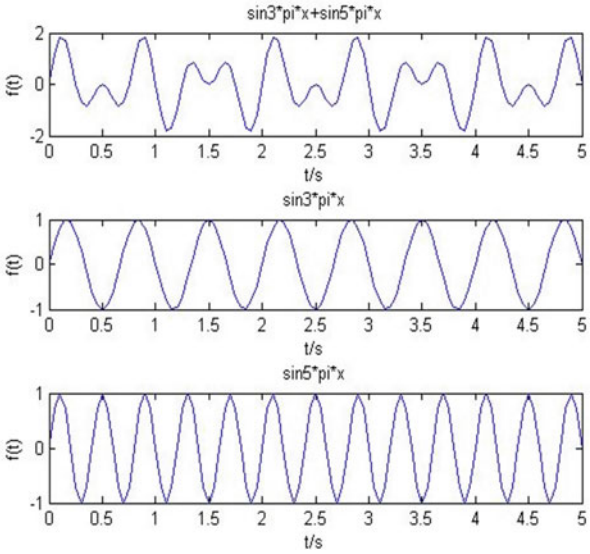


Fig. 20.5 Time domain of $\sin(3\pi x) + \sin(5\pi x)$

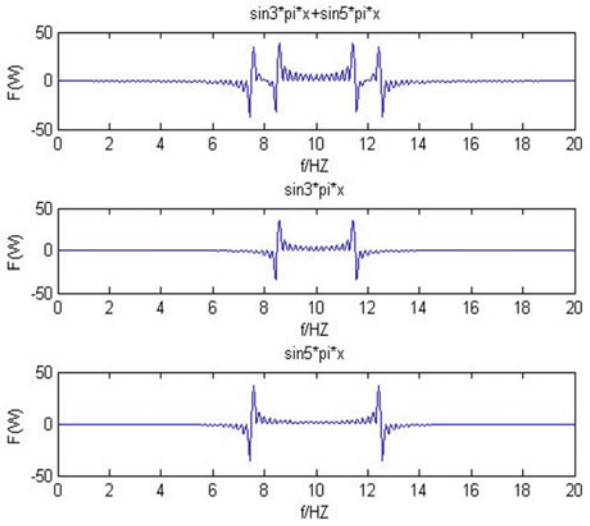


Fig. 20.6 Frequency domain of $\sin(3\pi x) + \sin(5\pi x)$

Figure 20.6 shows the signal $\sin(5\pi x)$ and the signal$\sin(3\pi x)$ located at different frequency band after FT. As a result, the signal $\sin(5\pi x)$ can be easily removed at the frequency domain. The Fourier basis is sinusoid, which is periodic. Thus the Fourier representation is particularly useful in discovering periodic patterns in a signal that might not otherwise be obvious when the signal is represented with respect to a canonical basis. Consequently, Fourier Transform and its various transformations are important tools for the processing of digital images. This section will give a brief introduction to these tools without any specific motivation.

20.1.3.1 Fourier Transforms

Fourier Series

For the periodic function $f(x)(f(x + T) = f(x))$, it can be written as a linear combination of the sine and cosine, which is expressed as:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{2n\pi x}{T} + b_n \sin \frac{2n\pi x}{T} \right) \tag{20.4}$$

where

$$a_0 = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)dx,$$

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \cos \left(\frac{2n\pi x}{T} \right) dx \qquad n = 1, 2, 3, \ldots,$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \sin \left(\frac{2n\pi x}{T} \right) dx \qquad n = 0, 1, 2, 3, \ldots.$$

If the periodic function $f(x)$ satisfies the Dirichlet condition, the Fourier series are limited. It can be expressed as:

$$f(x) = a_0 + \sum_{n=1}^{\infty} A_n \cos (n\omega_0 x + \phi_n) \tag{20.5}$$

where $\omega_0 = \frac{2\pi}{T}$, $A_n = \sqrt{a_n^2 + b_n^2}$,$\phi_n = -\arctan\left(\frac{b_n}{a_n}\right)$.

One-Dimensional Continuous Fourier Transformation

For non-periodic function $f(x)$, the one-dimensional Fourier Transformation is expressed as:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx = R_e(u) + jI_m(u). \tag{20.6}$$

The inverse transformation expression is:

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du, \tag{20.7}$$

where

$$\text{Amplitude} : |F(u)| = \sqrt{R_e^2(u) + I_m^2(u)} \tag{20.8}$$

$$\text{Phase} : |\Phi(u)| = \tan^{-1}\left({}^{I_m(u)}\!/_{R_e(u)}\right) \tag{20.9}$$

The magnitude describes the overall contribution of a frequency in constructing a signal, and the phase describes the relative position of each frequency.

Two-Dimensional Continuous Fourier Transformation

For non-periodic function $f(x, y)$, the two-dimensional Fourier transformation is expressed as:

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy = R_e(u,v) + jI_m(u,v) \tag{20.10}$$

The inverse transformation expression is as following:

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{j2\pi(ux+vy)}dudv \tag{20.11}$$

$$\text{Amplitude} : |F(u)| = \sqrt{R_e^2(u) + I_m^2(u)} \tag{20.12}$$

$$\text{Phase} : |\Phi(u,v)| = \tan^{-1}\left({}^{I_m(u,v)}\!/_{R_e(u,v)}\right) \tag{20.13}$$

One-Dimensional Discrete Fourier Transformation

By sampling the continuous function, the discrete function $f(x)$ is expressed as:

$$f(x) = f(x_0 + x\Delta x), \qquad x = 0, 1, 2 \ldots, N-1. \tag{20.14}$$

The discrete Fourier transformation (DFT) of $f(x)$ is expressed as:

$$F(u) = \sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N} \qquad u = 0, 1, 2 \ldots, N-1 \tag{20.15}$$

The inverse transformation is expressed as:

$$f(x) = \frac{1}{N}\sum_{x=0}^{N-1} F(u)e^{j2\pi ux/N} \qquad x = 0, 1, 2, \ldots N-1 \tag{20.16}$$

According to the Euler formula $e^{j\theta} = \cos\theta + j\sin\theta$, (20.13) and (20.14) can also be expressed as:

$$F(u) = \sum_{x=0}^{N-1} f(x)\left(\cos\frac{2\pi ux}{N} - j\sin\frac{2\pi ux}{N}\right) \tag{20.17}$$

$$f(x) = \frac{1}{N}\sum_{x=0}^{N-1} F(u)\left(\cos\frac{2\pi ux}{N} + j\sin\frac{2\pi ux}{N}\right) \tag{20.18}$$

Two-Dimensional Discrete Fourier Transformation

From the one-dimensional discrete Fourier transformation, the two-dimensional discrete Fourier transformation can be expressed as:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)} \qquad u = 0, 1, 2, \ldots M-1; v = 0, 1, 2, \ldots N-1 \tag{20.19}$$

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)} \qquad x = 0, 1, 2, \ldots M-1; y = 0, 1, 2, \ldots N-1 \tag{20.20}$$

20.1.3.2 Discrete Cosine Transformation

DFT is used for discrete signals and spectra, since the computer works in a digital environment dealing with only discrete calculating as close as possible to the continuous signal in reality. Discrete Cosine Transformation (DCT) is a form of DFT for filtering using a slightly different form of convolution, called symmetric convolution. It is widely used in image and video compression applications, e.g. JPEG and MPEG.

The Definition of DCT

Actually, the DCT is the real part of DFT, i.e., the cosine items. So the one-dimensional DCT is expressed as follows:

$$F(\mathrm{u}) = a_0 c(u) \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N} \qquad u = 0, 1, 2, \ldots N-1 \tag{20.21}$$

where $\mathrm{a}_0 = \frac{2}{\sqrt{N}}$ $\mathrm{c}(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & u \neq 0 \end{cases}$.

The inverse transformation is:

$$f(x) = a_1 \sum_{u=0}^{N-1} c(u) F(u) \cos \frac{(2x+1)u\pi}{2N} \qquad x = 0, 1, 2, \ldots N-1 \tag{20.22}$$

where $\mathrm{a}_1 = \frac{2}{\sqrt{N}}$ $\mathrm{c}(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & u \neq 0 \end{cases}$.

Two-Dimensional DCT

From the one-dimensional DCT, we also can get the two-dimensional discrete DCT and the expression as followings:

$$\begin{aligned} F(u,v) &= a_0 c(u,v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \qquad u \\ &= 0, 1 \ldots N-1; v = 0, 1 \ldots N-1 \end{aligned} \tag{20.23}$$

where $\mathrm{a}_0 = \frac{2}{\sqrt{N}}$, $\mathrm{c}(u,v) = \begin{cases} 1/2 & u = v = 0 \\ 1/\sqrt{2} & uv = 0, \mathrm{u} \neq v \\ 1 & uv > 0 \end{cases}$.

The inverse transformation is:

$$\begin{aligned} \mathrm{f}(x,y) &= a_1 \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u,v) F(u,v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \qquad x \\ &= 0, 1 \ldots N-1; y = 0, 1 \ldots N-1 \end{aligned} \tag{20.24}$$

where $a_1 = \frac{2}{\sqrt{N}}\, c(u,v) = \begin{cases} 1/2 & u = v = 0 \\ 1/\sqrt{2} & uv = 0, u \neq v \\ 1 & uv > 0 \end{cases}$.

20.1.3.3 Windowed Fourier Transform

The Windowed Fourier Transform (WFT), also called Short Time Fourier Transform (STFT), is transformed from FT by multiplying the initial function $f(x)$ with a window function $g(x - t)$. The expression of WFT is:

$$F(w,t) = \int_{-\infty}^{\infty} g(x-t) f(x) e^{-jwt} dx \tag{20.25}$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(w,t) g(x-t) e^{jwt} dw dt \tag{20.26}$$

Notably, the WFT $F(w, t)$ is not only with frequency information w but also with time-domain information t.

20.2 Image Forgery Detection

The authenticity of digital images has an essential role as these images are popularly used as supporting evidences and historical records in growing number and wide range of applications from forensic investigation, journalistic photography, criminal investigation, law enforcement, insurance claims, and medical imaging. Unfortunately, digital image forgery has a long history, and digital image crimes (e.g., posting fake photos to deliberately ruin people's reputations) have been growing at a speed that exceeds the defensive measures. Furthermore, the advanced image processing software and the powerful digital cameras have given rise to large amounts of tampered images that leaves no obvious traces of it being manipulated. Marketing and advertising companies are known to modify images of women such that they appear more attractive. The media often strives to show groundbreaking pictures of global events, and this can sometimes lead to this pressure causing the photographer to alter the picture.

For example, photo manipulation has been regularly used to deceive or persuade viewers, or for improved storytelling. Often even subtle and discreet changes can have profound impacts on how we interpret or judge a photograph. The "Missing umbrella" news published in March 2015, for example, had what appears to be a normal photo of deputy minister Su Su Hlaing and her team visiting Kawthaung on a sunny day. However, many social media outlets notice an unusual shadow beneath Su Su Hlaing's feet and the man beside her was leaning into her left shoulder, in a

Fig. 20.7 The mystery of Myanmar's missing umbrella [1]

Fig. 20.8 California attorney facing suspension for fake photos with celebs

pose that would suggest he was holding the invisible parasol. It was later revealed that the umbrella was airbrush out for decency and respect towards Kawthaung culture as a man holding an umbrella for woman is considered embarrassing. An image of the famous Missing Umber photo is shown in Fig. 20.7.

Also, an individual may alter image to improve one's self-expression. A notable case of such controversial photo manipulation is Svitlana Sangary's story [2]. She was a lawyer with reputational background of defending well known politicians and celebrities. In fact, she made a website dedicated to advertising her work. However, the pictures on her site of these supposes proof of her "friendly" relationships with her clients and court achievement was later identified as being edited through a technique known as splicing (will be detailed later) in which the image (Fig. 20.8) and the politician's face were mashed together. This incident resulting in Svitlana losing her reputation as a respectable lawyer and licenses.

This issue reveals authentic weaknesses and reduces the authenticity of digital images. Because of the fact that digital images are possible to be presented in a court or in the news, discovering methods for verifying the integrity and the authenticity of these images has now become very imperative. There's a great need for an approach to verify the authenticity of photographs. Due to the technological advancement in the recent years, law enforcement has needed to stay abreast of emerging technological advances and use these in the investigation of crime. The Scientific Working Group on Imaging Technology (SWGIT) provides recommendations and guidelines

to law enforcement agencies and others in the criminal justice system regarding the best practices for photography, videography, and video and image analysis. SWGIT provides information on the appropriate use of various imaging technologies for use by personnel in the criminal justice system through the release of documents such as the SWGIT best practices documents.

As an important aspect of forensic image analysis, image forgery detection aims to verify the authenticity of a digital image. Image authentication solution can be classified into two types: Active image forgery detection and passive-blind image forgery detection. An active forgery detection technique uses a known authentication code embedded into the image content before the images are sent through an unreliable public channel. By verifying the presence of such authentication code authentication may be proved by comparing with the original inserted code. Digital signature and watermarking are two active methods used to verify the contents and the authenticity of digital images. Since some specific procedures are required for watermarking and signature techniques, their practical applications are limited. For example, when an image is created using the signature-based method, a digital signature is needed to be generated for this image whereas, the watermarking based method requires embedding watermark onto the image. However, these active methods face challenges especially when used on a large scale basis or adopted widely in todays' digital imaging devices (e.g., digital camera).

In order to combat this issue, a new technique for checking the contents of digital images has been developed called passive-blind forgery detection. Passive-blind forgery detection techniques use the received image only for assessing its authenticity or integrity, without any signature or watermark of the original image from the sender. It is based on the assumption that although digital forgeries may leave no visual clues of having been tampered with, they may highly likely disturb the underlying statistics property or image consistency of a natural scene image which introduces new artifacts resulting in various forms of inconsistencies. These inconsistencies can be used to detect the forgery. This technique is popular as it does not need any prior information about the image. Existing techniques identify various traces of tampering and detect them separately with localization of tampered region.

In this section, image tampering techniques, including copy-move forgery and image-splicing forgery, are firstly described. Then, image forgery detection methods are introduced from the aspects of active image forgery detection and passive-blind image forgery detection techniques.

20.2.1 Image Tampering Techniques

Digital image tampering is a technique used to change or alter content on a digital image in a way that looks authentic, but not. It is often used for negative purposes. With recent advances in technology and variety of photo editing tools out there in the industry, tampering of digital image has become easier to make and unfortunately, harder to detect. In this section, we will discuss the basic image tampering

techniques. There are two common techniques used for altering semantic content on digital images. The first technique is known as Copy-Move Forgery. It copes an object or part of image and pastes it onto the same image source. The second technique is known as Image-Splicing (or Compositing) Forgery, which is coping an object or part of another image and pasting it onto the image source, adding content that doesn't belong to the original image.

20.2.1.1 Copy-Move Forgery

The technique "Copy-Move Forgery" is the most common image tampering technique used due to its simplicity and effectiveness, in which parts of the original image is copied, moved to a desired location and pasted. This is usually done in order to hide certain details or to duplicate certain aspects of an image. Textured regions are used as ideal parts for copy-move forgery, since textured areas have similar color and noise variation properties to that of the image which are unperceivable for human eye looking for inconsistencies in image statistical properties. Blurring is usually used along the border of the modified region to lessen the effect of irregularities between the original and pasted region.

An example of such technique is shown in Fig. 20.9, where the tree in the first image shot is used to cover the moon in the second image shot. Thus, changing the user's perspective (in terms of users' sense of time in this case) of the image as showing a house with two trees on an evening of a new moon to a house with one tree on an evening with a crescent moon. Because the cropped image uses the same sources image to overlay a particular target area, we can detect the subtle alteration by looking at the image property, in particular at how much "noise" there is. This reason stem from the fact that the temperature of color and illumination conditions are likely very coordinated compared with the altered area of the image. However, this process can be concealed. The malicious attackers could perform geometric transforms such as rotation or scaling on the copied region. Also, matting and blending techniques are exploited in order to create a smoother transition between the surrounded area of the original image and the object being pasted [3]. Using sophisticated technologies such as photo editing software, makes it easy to create image composites in which the produced results are not easily detected by the human naked eyes.

Fig. 20.9 Copy-move Forgery. The image indicates that forger attempted to copy another tree and post front of crescent moon in terms of changing picture's meaning [10]

Fig. 20.10 The photomontage is very famous for John Kerry and Jane Fonda. The estimate direction for Kerry is 123°, and the direction for Fonda is 86° [4]

20.2.1.2 Image-Splicing Forgery

The technique "Image-Splicing (or Compositing) Forgery" uses multiple different image sources to hid a particular object or alter the original image, which is different from the "Copy-Move Forgery" as the previous method uses the same image source to alter itself. An example of such technique is shown in Fig. 20.10, where the second image showing an influential lady Jane Fonda was cropped out and pasted along with a man named John Kerry who was sitting-in for an Anti-War Rally. Thus, the image would change the user's perspective that Jane Fonda supported the rally, and pursued those admirers of her should also support this rally. The photo generated Buzz for John Kerry when originally released, but was later found fake, causing John Kerry to lose his 2004 presidential election. However, the damage had already been done by this tampered image.

The technique of creating composite images may sometimes need geometric transformation to make sure that the merged object follows the required original image's perspective and scale, rotation, scaling and translation. To make it even more convincing, many malicious users would use image blending and mating techniques to hide the edges of the spliced regions. This also gives more uniform aspects to the image, making it difficult to detect image splicing [3].

20.2.2 Active Image Forgery Detection

The important aspect of digital image forensics is evaluating the authenticity and the credibility of the images. In the past, a photograph was always recognized as a true image, however, due to the rapid development of modern editing tools and software technologies, this cannot always be certain today. Nowadays, digital imagery has become the main source for the news and information. Since people understand the social events in a visual way rather than through words alone, we are in dire need of a

way to enable them to verify the authenticity of the digital images to maintain their credibility, which is also known as digital image forgery detection.

Digital image forgery detection technologies can be classified into two types: active defense and passive blind detection. The active defense were created in order to verify the contents and the authenticity of the digital images through insertion of identifiable information into the image. There are two active methods that have been suggested in order to verify and protect the truthfulness of the digital images: Watermarking and digital signature.

20.2.2.1 Digital Watermarking

The origin of the digital watermarking was to hide a message inside a digital signal (e.g., an audio, image, and video) for various purposes, particularly, in order to protect it for copyright reasons [6]. In doing so, information used to identify the owner of an image or photo could be embedded in the image itself using watermarking. Many watermarking algorithms have been proposed in the past, and are implemented either directly in the spatial domain or in the frequency domain by using discrete image transform such as DFT or DCT, or in the wavelet domain by using discrete wavelet transform (DWT). It is also worth noting digital watermarking is closely related to steganography, which intends to prevent unauthorized party from knowing existence of secret message hidden within a digital signal. The watermark may be visible or invisible to the naked eye. Figure 20.11 shows an example of a photo with a visible watermark, which is created by using JACo

Fig. 20.11 A photo with visible watermark

Watermark, an open source tool for adding watermark to an image or photo [25]. This visible watermark displays it is copyrighted by Xiaodong Lin.

In fact, watermarking system is designed from two main types, of which the first type is the source side. This side is used in order to produce the signal for watermark W and embed this watermark signal W with the original image X to acquire the watermarked image Y. It is also referred to as visible watermarking. The most common example of visible watermarking is the identification of commercial advertisements, of which the main purpose is to clearly identify copyright to prevent illegal use. A good example of it is Fig. 20.11.

In contrast, the second side is for extracting the watermark W, and gives the confidence measure for the detected image. Hence, if attackers try to modify any image embedded with watermarking, the watermark signal inside the original image will be disrupted. Thus, the digital image can be detected as forgery based on the watermark that is inside the original image, but is damaged because of modification made onto the image.

20.2.2.2 Digital Signature

A digital signature is designed as an electronic signature. The main goal of this method is to verify the authenticity and integrity of a document (e.g., an image) that has been sent in a correct way by a trustworthy sender without any change. Digital signature is usually implemented based on public key cryptosystem, such as RSA and ElGamal [7]. As shown in Fig. 20.12, the original image can be firstly hashed into a short, fixed-length message (or hash value) (for example using MD5 or SHA-1 hashing [7]), and then the person of creating images ("the creator") uses his/her private key to digitally sign the hash value of the image. Afterwards, the signature and image are saved individually. Upon receiving the image and signature, the recipients would decrypt this signature in order to match the hash value of this image to the values that exist in the original signature. If they match, this image can

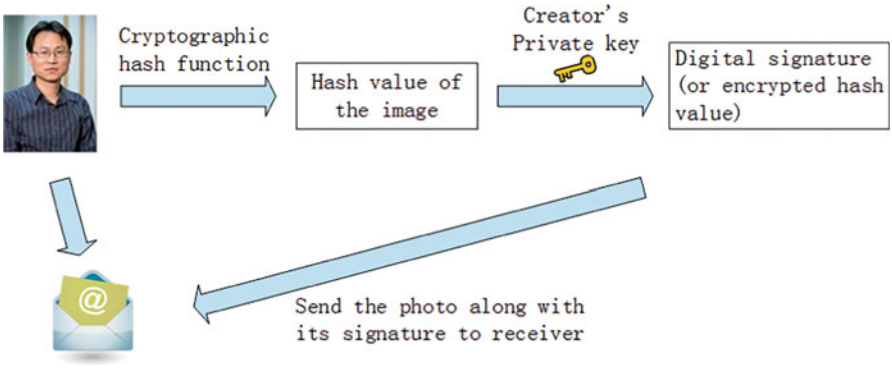


Fig. 20.12 Digital signature

be claimed to be "authentic", and the image will be verified as the original image. Please note that besides using cryptographic hash value, the digital signature of an image can also use invariance properties in JPEG lossy compression as robust feature codes to uniquely represent the image, and these feature codes are encrypted as the signature using the creator's private key [8].

As mentioned above, the watermark-based and the digital signature technologies protect the credibility of the digital image, but they have faced several challenges to identify whether the image has been changed or not. For example, for most of the conventional digital signature schemes, the creator has to send his/her public key to the receivers by which the receivers can verify the authenticity of the images received. When authentication is performed in a large-scale, a Certificate Authority (CA) is usually created as a trusted third party to bind together a public key with identity information such as user identity, also known as public key certificate. It allows the receiver to verify ownership of a public key. Once public key ownership is confirmed, the public key can be used to validate digital signatures on the images originated from the creator. Nevertheless, a public key certificate may have to be revoked due to various reasons. For example, its corresponding private key is compromised, and so it is no longer valid but still within its validity period. This has been accomplished through a certificate revocation list (CRL), which maintains a list of certificates that have been revoked. Note that in the traditional Public Key Infrastructure (PKI), CRL can be maintained in a centralized server that keeps all the revoked certificates, and each user only needs to access to CRL during the certification validation procedure. The above solution, unfortunately, may not be feasible in the highly distributed environment where no any central intelligence can be available to claim the compromised public key certificate. It is challenging to manage a large scale PKI.

However, in these methods, the attackers can manipulate the image by changing its properties before the watermarking or even digital signature creating phase. Additionally, the main disadvantage in these methods is that they take time to generate the signature or embedding the watermark on the digital image in order to protect it, which makes it very limited for practical uses. Furthermore, the embedded watermark in active forgery detection must have strong robustness, that is, watermarks can resist attacks and cannot be easily removed or re-embedded, and this robustness cannot be prove to be fully achieved currently in theory. To deal with these issues, passive-blind method has been developed to examine the content of digital images. This method formulates an assessment on a digital document by resorting only to the digital asset itself.

20.2.3 Passive-Blind Image Forgery Detection

As mentioned above, an active approach in the digital image forensics requires some certain procedures, which reduce their applicability in practice. In fact, people always upload their images to the Internet without making any digital signature or

Fig. 20.13 The process of passive-blind image forgery detection

watermark on those images due to the time factor. A new method known as the passive-blind approach can solve the problem, which does not need any procedures in protecting digital images from forgery. The passive-blind detection method identifies the copied region from the image's pixels. In fact, this method is able to determine if the image has any manipulation from attackers through two main principals, first by trying to expose semantic manipulation (forgery) through studying the inconsistencies in the statistics of natural images. The second group of techniques answers the questions such as which device was used to capture this image [5].

Mostly existing blind image forgery detection approaches extract features from images firstly, then select a classifier and train the classifier using the features extracted from training image sets, and finally classify the features [3]. A generalized framework of blind image forgery detection approach consists of the following major steps, as shown in Fig. 20.13.

1. Image preprocessing: Before feature extraction process some operations are performed over the images under consideration, such as cropping, transforming RGB image into grayscale, DCT or DWT transformation to improve the classification performance.
2. Feature extraction: A set of features are extracted for each class that helps distinguish it from other classes, while remaining invariant to characteristic differences within the class from the input forged data. In particular, extracting informative features and selecting feature must be sensitive to image manipulation. One of the desirable characteristics of selected features and constructed feature vector should be with low dimension, which will reduce the computational complexity of training and classification.
3. Classifier selection and feature preprocessing: Based on the extracted set of features, select or design appropriate classifiers and choose a large set of images to train classifiers, obtain some important parameters of classifiers, which can be utilized for the classification. Feature preprocessing is used to reduce the dimensionality of features without decreasing the machine learning based classification performance and achieve reduction in computational complexity at the same time.
4. Classification: The purpose of classifier is to discriminate the given images and classify them into two categories: original and forged images.
5. Post-processing: Post-processing operation usually involves localization of forged region. It is optional.

In general, from the aspects of forgery detection objects, the current passive-blind image forgery detection methods can be classified into three categories: Image processing operation detection, device-based image forgery detection and format-based image forgery detection.

20.2.3.1 Image Processing Operation Detection

Various image processing operations are often applied to conceal traces of tampering the images when altering an image. These image processing operations typically involve copy-move, re-sampling, and blurring. Detection of these operations helps to identify the forgeries.

Copy-Move Forgery Detection (CMFD)

Copy-move is the most common image tampering technique due to its simplicity and effectiveness, in which parts of the original image are copied, moved to a desired location and pasted. This is usually done in order to hide certain details or to duplicate certain aspects of an image. The copied regions may range from background, object, creature to letter. CMFD techniques can be further organized into two approaches: block-based and keypoint-based.

Block-Based Approach

The block-based approach is widely used due to its compatibility with various feature extraction techniques and increased matching performance. It is quite effective due to the fact that there must be at least two similar regions in the tampered image by copy-move forgery. Also, these similar regions are usually small in order to avoid being spotted. The approach is composed of three stages: Block division, Feature extraction and Matching. Firstly, block division splits an image into overlap or non-overlap blocks for analysis. It can reduce the computational time for matching process to find the similar feature vector in an image compared to exhaustive searching approach.

The feature extraction techniques extract the features from these blocks. They can be in the form of frequency transform, texture and intensity, moment invariant, log polar transform, and dimension reduction [9]. Frequency transform is the most popular feature extraction technique due to its robustness to noise, blurring and JPEG compression. Among the transform functions, DCT is widely used for its robustness against noise addition and JPEG compression. Several enhancements of DCT, such as fast Walsh-Hadamard Transform (FWHT), Discrete Wavelet Transform (DWT), Dyadic Wavelet Transform (DyWT) and Wiener Filter Wavelet, have been proposed to reduce feature dimensions for low computational complexity. Texture and intensity exist in natural scenes such as grass, cloud, tree, and ground. They are usually measured and characterized through intensity, pattern or color

information. Moments invariant is a set of features that are invariant to translation, rotation and scale, which can be used to classify shape and recognize object in binary images. Log polar transform works by mapping from the points on the Cartesian plane (x, y) to points in the log-polar (x, h). It is invariant to rotation, scaling and translation. Dimension reduction techniques, including Singular Value Decomposition (SVD) and Locally Linear Embedding (LLE), are usually used to reduce the dimensionality of the image and improve the complexity. The SVD is generally stable, scales, and achieves rotation invariance for both algebraic and geometric properties while resulting in loss of image details. Alternatively, LLE can be implement to reduce dimensionality in high-dimensional dataset. LLE is able to find the fused edge that hides the traces in forged image without changing the relative locations at the cost of long processing time.

Finally, matching technique compares the features against each other to determine the similarity between blocks within the image to define the manipulated area. The matching techniques can be divided into *sorting, hash, correlation and euclidean distance*. Sorting technique orders the features in a certain arrangement for quickly finding the duplicated area. Lexicographical is the most widely employed sorting technique, which detects potentially tampered region through the adjacent identical pairs of blocks. The accuracy of lexicographical techniques can also be improved using kd-tree, a nearest neighbor searching technique. *Hash* is usually used to ensure that any modification to the data can be detected, which can be applied to find the duplicated features. Counting Bloom Filters (CBF) and Locality-Sensitive Hashing (LSH), two popular techniques employing hash functions for duplication detection. In CBF, the identical feature has the same hash value, while the element only increases for different hash values. Thus, the element with value larger than two is expected to be duplicated pairs in CMFD. LSH searches the approximate nearest neighbor through hashing the feature vectors and selecting the identical hash value. *Correlation* is a statistical measurement of two or more variables to indicate the level of change. In CMFD, the region is suspected of being tampered if the value of the correlation peak exceeds the predefined threshold. *Euclidean distance* is a measurement of distances between two vectors in Euclidean space. An image is identified as potentially tampered if two blocks are near to each other with a similar neighborhood.

Notably, in order to display and localize the tampered regions in the forged image, visualization process is optional by coloring or mapping the region of the matching blocks.

Keypoint-Based Approach

Keypoint-based approach extracts the distinctive local features such as corners, blobs, and edge from the image. Each feature is presented with a set of descriptor, which helps to increase the reliability of the features. Then, both features and descriptors in the image are classified and matched to each other to find the duplicated regions in the copy-move forgery [9].

The feature extraction techniques of keypoint-based approach can be divided into three types: Scale Invariant Feature Transform (SIFT), Harris Corner Detector, and Speed Up Robust Features (SURF). SIFT detects salient points at different scales from Difference of Gaussian (DoG) pyramid in scale-space representation. It is the most popular keypoint feature extraction technique in CMFD due to its high stability for both intermediate and post-processing operations. However, it is unable to detect the duplicate regions in flat areas or little visual structure and unable to define a shape or a single patch due to their non-uniform distribution. Additionally, it is incapable of differentiating between regions that are intentionally inserted or naturally similar. Therefore, Harris Corner Detector and SURF are proposed to improve SIFT-based technique. Harris Corner Detector extracts corners and edges from the regions based on the local auto-correlation function, resulting in consistencies in natural imagery, which is found to be robust to rotation, scale, JPEG compression, noise and blurring. SURF improves the processing time, and is robust to certain transformation and post processing operations at the cost of reducing the accuracy.

In keypoint-based approach, the nearest neighbor and clustering are the main matching techniques. Nearest neighbor examines the similarity between points by calculating the distance of each point in vector space. The points are considered similar if the distances satisfy the designated threshold. Clustering technique groups a set of objects that are similar to each other and the object with similar shape and texture can be considered as the real copy.

Resampling Detection

When creating image composites, geometric transformations are needed to give the image a more uniform aspect. These geometric transformations typically involve re-sampling. This re-sampling is often imperceptible, but it introduces specific correlations into the image. Thus, these correlations can be detected as evidence of digital tampering [12].

$A_{p/q}$ resampling of a 1-D discrete sequence $f(k)$ with m samples involves the following three steps [3]:

(a) Up-sample: Create a new signal $f_u(k)$ with pm samples, where $f_u(pk) = f(k)$, $k = 1,2,\ldots,m$, and $f_u(k) = 0$ otherwise.
(b) Interpolate: Convolve $f_u(k)$with a low-pass filter: $f_i(k) = f_u(k) * h(k)$.
(c) Down-sample: Create a new signal $f_d(k)$ with m samples, where $f_d(k) = f_i(qk)$. Denote the re-sampled signal as $g(k) \equiv f_d(k)$.

Different types of re-sampling algorithms (e.g., linear, cubic) differ in the form of the interpolation filter $h(k)$ in step b. Since all three steps in the re-sampling of a signal are linear, this process can be described with a single linear equation.

Denoting the original and re-sampled signals in vector form, f and g, respectively, re-sampling takes the form:

$$g = A_{p/q} f \tag{20.27}$$

where the $n \times m$ matrix $A_{p/q}$ embodies the entire re-sampling process. Depending on the re-sampling rate, the re-sampling process will introduce correlations of varying degrees between neighboring samples. For example, consider the up-sampling of a signal by a factor of two using linear interpolation. In this case, the re-sampling matrix takes the form:

$$A_{1/2} = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0.5 & 0.5 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0.5 & 0.5 & \cdots \end{bmatrix} \tag{20.28}$$

According to (20.27), the odd samples of the re-sampled signal g take on the values of the original signal f, i.e., $g_{2i-1} = f_i, i = 1,2,\ldots m$. The even samples, on the other hand, are the average of adjacent neighbors of the original signal:

$$g_{2i} = 0.5 f_i + 0.5 f_{i+1} \tag{20.29}$$

where $i = 1,2,\ldots m-1$. Note that since each sample of the original signal can be found in the re-sampled signal, i.e., $f_i = g_{2i-1}$ and $f_{i+1} = g_{2i+1}$, the above relationship can be expressed in terms of the re-sampled samples only:

$$g_{2i} = 0.5 g_{2i-1} + 0.5 g_{2i+1} \tag{20.30}$$

In this simple case, each even sample is precisely the same linear combination of its adjacent two neighbors across the entire re-sampled signal. As a result, a re-sampled signal could be detected (in the absence of noise) by noticing that every other sample is perfectly correlated to its neighbors.

In order to detect re-sampling signal in a general forensic setting, consider re-sampling a signal by an arbitrary amount p/q. Firstly, it is necessary to check when the i^{th} sample of a re-sampled signal is equal to a linear combination of its $2N$ neighbors, that is:

$$g_i \stackrel{?}{=} \sum_{k=-N}^{k=N} a_k g_{i+k} \tag{20.31}$$

where α_k are scalar weights (and $\alpha_0 = 0$). Re-ordering terms, and re-writing the above constraint in terms of the re-sampling matrix yields:

$$g_i - \sum_{k=-N}^{k=N} \alpha_k g_{i+k} = 0 \tag{20.32}$$

Let a_i denote the i^{th} row of the re-sampling matrix $A_{p/q}$, and f denotes the original signal. Eq. (20.32) turns to be

$$\begin{aligned} a_i \bullet f - \sum_{k=-N}^{k=N} \alpha_k a_{i+k} \bullet f &= 0 \\ \left(a_i - \sum_{k=-N}^{k=N} \alpha_k a_{i+k} \right) \bullet f &= 0 \end{aligned} \tag{20.33}$$

where we see now that the i^{th} sample of a re-sampled signal is equal to a linear combination of its neighbors when the i^{th} row of the re-sampling matrix, a_i, is equal to a linear combination of the neighboring rows, $\sum_{k=-N}^{k=N} \alpha_k a_{i+k}$..

For example, in the case of up-sampling by a factor of two, Eq. (20.28), the even rows are a linear combination of the two adjacent odd rows. Note also that if the i^{th} sample is a linear combination of its neighbors then the $(i - kp)^{th}$ sample (k an integer) will be the same combination of its neighbors, that is, the correlations are periodic. It is, of course, possible for the constraint of Eq. (20.34) to be satisfied when the difference on the left-hand side of the equation is orthogonal to the original signal f. While this may occur on occasion, these correlations are unlikely to be periodic.

Example 20.1 The matrix for up-sampling by a factor of 4/3 using linear interpolation has the form:

$$A_{4/3} = \begin{bmatrix} 1 & 0 & 0 & 0 & \\ 0.25 & 0.75 & 0 & 0 & \\ 0 & 0.5 & 0.5 & 0 & \\ 0 & 0 & 0.75 & 0.25 & \\ 0 & 0 & 0 & 1 & \\ & & & & \ddots \end{bmatrix} \tag{20.34}$$

Describe how the third row of Eq. (20.34) is correlated to the first, second, fourth, and fifth rows? Are the fourth and fifth rows similarly correlated to their neighboring rows? How about the seventh and eleventh rows?

Solution. Let $\alpha = (\alpha_{-2}, \alpha_{-1}, \alpha_1, \alpha_2)$*denote the weights of the combination. According to Eq. (20.33), the third row can be written as the combination of the first, second, fourth, and fifth rows in the following matrix equation:*

$$\begin{aligned}(0 \quad 0.5 \quad 0.5 \quad 0) = \alpha_{-2}(1 \quad 0 \quad 0 \quad 0) + \alpha_{-1}(0.25 \quad 0.75 \quad 0 \quad 0) \\ + \alpha_1(0 \quad 0 \quad 0.75 \quad 0.25) + \alpha_2(0 \quad 0 \quad 0 \quad 1)\end{aligned}$$

Rewritten the equation yields:

$$\begin{aligned}0 &= \alpha_{-2} + 0.25\alpha_{-1} \\ 0.5 &= 0.75\alpha_{-1} \\ 0.5 &= 0.75\alpha_1 \\ 0 &= 0.25\alpha_1 + \alpha_2\end{aligned} \tag{20.35}$$

Solving the linear equation array (20.35) gets $\alpha_{-2} = -\frac{1}{6}, \alpha_{-1} = \frac{2}{3}, \$\$\alpha_1 = \frac{2}{3}, \alpha_2 = -\frac{1}{6}$.

It can be found that the fourth and fifth rows do not have similarly correlated to their neighboring rows while the seventh and eleventh rows have the similar correlated to their neighboring rows. The example shows the periodic characteristics of the correlations.

If the specific form of the correlations, α, are priori knowledge, it is straightforward to determine which samples satisfy Eq. (20.33). While neither the re-sampling amount nor the specific form of the correlations are typically known in practice. In this case, the expectation/maximization (EM) algorithm is usually employed to determine if a signal has been re-sampled [9, 10]. EM can estimate a set of periodic samples that are correlated to their neighbors, and the specific form of these correlations.

Resampling in two dimensions is a straight forward application of the above mentioned operations in both spatial directions.

Blurring Detection

Blurring is a common process in digital image manipulation which is used to reduce the degree of discontinuity or to remove unwanted defects. Furthermore, blur operation is one of the commonly used methods to hide the presence of forgery. So identifying blur inconsistencies in various image regions can be helpful in detection image forgeries [13].

Blurring Model

Many factors can extrinsically or intrinsically cause image blur. Blur is generally one of five types: Object motion blur, camera shake blur, defocus blur, atmospheric turbulence blur, and intrinsic physical blur. These types of blur degrade an image in

different ways. An accurate estimation of the sharp image and the blur kernel requires an appropriate modeling of the digital image formation process. Hence we first focus on analyzing the image formation model. Recall that image formation encompasses the radiometric and geometric processes by which a 3D world scene is projected onto a 2D focal plane. In a typical camera system, light rays passing through a lens's finite aperture are concentrated onto the focal point. This process can be modeled as a concatenation of the perspective projection and the geometric distortion. Due to the non-linear sensor response, the photons are transformed into an analog image, from which the final digital image is formed by discretization.

Mathematically, the above process can be formulated as:

$$y = S(f(D(P(s) * h_{ex}) * h_{in})) + n \tag{20.36}$$

where y is the observed blurry image plane, S is the real planar scene, $P(\bullet)$ denotes the perspective projection, $D(\bullet)$ is the geometric distortion operator, $f(\bullet)$ denotes the nonlinear camera response function (CRF) that maps the scene irradiance to intensity, h_{ex} is the extrinsic blur kernels caused by external factors such as motion, h_{in} denotes the blur kernels determined by intrinsic elements such as imperfect focusing, $S(\bullet)$ denotes the sampling operator due to the sensor array, and n models the noise.

The above process explicitly describes the mechanism of blur generation. However, what we are interested here is the recovery of a sharp image having no blur effect, rather than the geometry of the real scene. Hence, focusing on the image plane and ignoring the sampling errors, we obtain

$$Y \approx f(x^* h) + n \tag{20.37}$$

where x is the latent sharp image induced from $D(P(s))$ and h is an approximated blur kernel combining h_{ex} and h_{in}, as assumed by most approaches. CRF in this formulation has a significant influence on the deblurring process if it is not appropriately addressed. For simplification, most researchers neglect the effect of the CRF, or explore it as a preprocessing step. Let us remove the effect of the CRF to obtain a further simplified formulation:

$$y = x^* h + n \tag{20.38}$$

This equation is the most commonly-used formulation in image deblurring. Given the above formulation, the general objective is to recover an accurate x (non-blind deblurring), or to recover x and h (blind deblurring), from the observation y, while simultaneously removing the effects of noises n. Taking into account a whole image, the above equation is often represented as a matrix-vector form:

$$y = Hx + n \tag{20.39}$$

where y, $x$and n are lexicographically ordered column vectors, respectively. H is a Block Toeplitz with Toeplitz Blocks (BTTB) matrix derived from h.

While the noise may originate during image acquisition, processing, or transmission, and is dependent on the imaging system, term n is often modeled as Gaussian noise, Poisson noise or impulse noise. The above equation is not suitable for describing these noises since it only characterizes the plus case and the signal-uncorrelated case. On the other hand, Poisson and impulse noise are usually signal-correlated.

Traditionally, the blur kernel in image deblurring methods is usually assumed to be spatially invariant (aka uniform blur), which means that the blurry image is the convolution of a sharp image and a single kernel [14–17]. However in practice, it has been noted that the invariance is violated by complex motion or other factors. Thus spatially variant blur (aka non- uniform blur) is more practical [18], but is hard to address. In this case, the matrix H in the above equation is no longer a BTTB matrix since different pixels in the image correspond to different kernels.

Actually, h can be expressed in different forms for different kinds of blurring. For example, as a result of imperfect focusing by the imaging system or different depths of scene, the fields outside the focus field are defocused, giving rise to defocus blur, or out of focus blur. Generally, a crude approximation of a defocus blur is made as a uniform circular model:

$$h(i,j) = \begin{cases} \dfrac{1}{\pi R^2}, & if \quad \sqrt{i^2 + j^2} \le R \\ 0, & otherwise \end{cases} \tag{20.40}$$

where R is the radius of the circle. This is valid if the depth of scene does not have significant variation and R is properly selected.

Blur analysis and deblurring methods. Image deblurring is a traditional inverse problem whose aim is to recover a sharp image from the corresponding degraded (blurry and/or noisy) image. Maximum a posteriori is a traditional deblurring method.

In statistics, Bayesian inference updates the states of a probability hypothesis by exploiting additional evidence. Bayes' rule is the critical foundation of Bayesian inference and can be expressed as

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{20.41}$$

where A stands for the hypothesis set and B corresponds to the evidence set. This rule states that the true posterior probability $p(A|B)$ is based on our prior knowledge of the problem, i.e. $p(A)$, and is updated according to the compatibility of the evidence and the given hypothesis, i.e. the likelihood $p(B|A)$. In our scenario for the non-blind deblurring problem, A is then the underlying sharp image x to be estimated, while B denotes the blurry observation y. For the blind case, a slight difference is that

A means the pair of $(x; h)$ since h is also a hypothesis in which we are interested. It can be written for both cases as

$$p(x|y,h) = \frac{p(y|x,h)p(x)}{p(y)} \tag{20.42}$$

$$p(x,h|y) = \frac{p(y|x,h)p(x)p(h)}{p(y)} \tag{20.43}$$

Note that either x or y and h are usually assumed to be uncorrelated. Irrespective of case, the likelihood $p(y|x,h)$ is dependent on the noise assumption.

The most commonly-used estimator in a Bayesian inference framework is the maximum a posteriori (MAP). This strategy attempts to find the optimal solution $A*$ which maximizes the distribution of the hypothesis set A given the evidence set B. In the blind case,

$$\begin{aligned}(x*,h*) &= \arg\max_{x,h} \; p(x,h|y) \\ &= \arg\max_{x,h} \; p(y|x,h)p(x)p(h)\end{aligned} \tag{20.44}$$

while in the non-blind scenario, the term $p(h)$ is discarded.

20.2.3.2 Device-Based Image Forgery Detection

Digital image may come from various imaging devices, e.g., various cameras and scanners. Consequently, identifying the device used for its acquisition is an interesting method to determine integrity and authenticity of a given image. The sensor noise, chromatic aberration or color filter array (CFA) can be used to identify the source of the image for detecting image forgery.

Sensor Noise

Imaging sensors used in capturing devices tends to introduce various defects and to create noise in the pixel values. The sensor noise is generally composed by three parts, i.e. pixel defects, fixed pattern noise (FPN), and photo response non-uniformity (PRNU). FPN and PNRU depend on dark currents in the sensor and pixel non-uniformities, respectively. They are the so-called pattern noise. Usually, PRNU is factor used for forensic and ballistic purposes.

Specifically, the image imperfections can be modeled as [9]:

$$I(x,y) = I_0(x,y) + \gamma I_0(x,y)K(x,y) + N(x,y), \tag{20.45}$$

where $I_0(\bullet)$ is the noise-free image, γ is a multiplicative constant, $K(\bullet)$ is the multiplicative PRNU, and $N(\bullet)$ is an additive noise term.

In order to make the estimation of the PRNU more reliable, it is estimated from a series of authentic images $I_k(x, y)$ ($k = 1,2,\ldots N$) taken from the camera in question. Each image is denoised with any standard denoising filter and subtracted from the original image. Let

$$W_k(x, y) = I_k(x, y) - \widehat{I}_k(x, y), \tag{20.46}$$

where $\widehat{I}_k(x, y)$ is the denoised images. The term $W_k(x, y)$ suppress the underlying image content. Then, the PRNU is estimated as:

$$K(x, y) = \frac{\sum_{k=1}^{n} W_k(x, y) I_k(x, y)}{\sum_{k=1}^{n} I_k^2(x, y)}. \tag{20.47}$$

An image in question $I(x, y)$ is denoised and subtracted from itself to yield $W(x, y)$ as described in (20.44). The PRNU $K(x, y)$ is estimated from a set of images known to have originated from the same camera. The correlation between the PRNU and the image being analyzed is given by:

$$\rho = I(x, y)K(x, y) \bigotimes W(x, y) \tag{20.48}$$

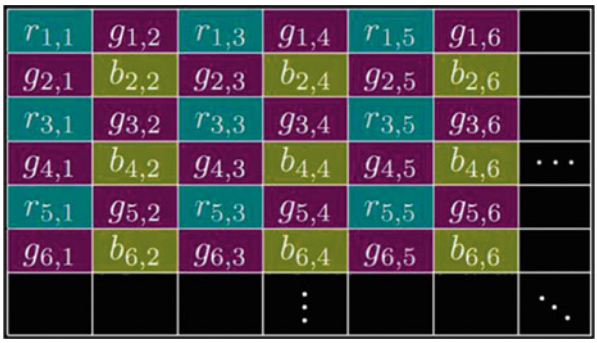
where $\bigotimes$ denotes normalized correlation. The correlation ρ is used as a measure of authenticity and can be computed locally in order to detect localized tampering.

Color Filter Array

A digital color image consists of three channels containing samples from different bands of the color spectrum, e.g., red, green, and blue. However, Most digital cameras are equipped with a single charge-coupled device (CCD) or complementary metal oxide semiconductor (CMOS) sensor, which captures color images using a color filter array (CFA). The CFA consists of an array of color sensors, each of which captures the corresponding color scene at an appropriate pixel location. The missing color samples are obtained by CFA interpolating process. Image forgery can be detected by identifying correlation introduced by the interpolation process.

The most frequently used CFA is the Bayer array [9]. It employs three color filters: red, green, and blue. The red and blue pixels are sampled on rectilinear lattices, while the green pixels are sampled on a quincunx lattice, as shown in Fig. 20.14. Since only a single color sample is recorded at each pixel location, the other two color samples must be estimated from the neighboring samples in order to obtain a three-channel color image. Let $S(x, y)$ denote the CFA image in Fig. 20.14, and $\tilde{R}(x, y)$, $\tilde{G}(x, y)$, $\tilde{B}(x, y)$ denote the red, green, and blue channels constructed from $S(x, y)$ as follows:

Fig. 20.14 Bayer array [9]



$$\tilde{R}(x,y) = S(x,y) \; if \;\; S(x,y) = r_{x,y} \\ = 0 \; otherwise \tag{20.49}$$

$$\tilde{G}(x,y) = S(x,y) \; if \;\; S(x,y) = g_{x,y} \\ = 0 \; otherwise \tag{20.50}$$

$$\tilde{B}(x,y) = S(x,y) \; if \;\; S(x,y) = b_{x,y} \\ = 0 \; otherwise \tag{20.51}$$

where (x, y) span an integer lattice. A complete color image, with channels $R(x, y)$, $G(x, y)$, and $B(x, y)$ needs to be estimated. These channels take on the non-zero values of $\tilde{R}(x, y)$, $\tilde{G}(x, y)$, and $\tilde{B}(x, y)$, and replace the zeros with estimates from neighboring samples.

The simplest methods for CFA interpolating are kernel-based interpolation methods that act on each channel independently. These methods can be efficiently implemented as linear filtering operations on each color channel:

$$R(x,y) = \sum_{u,v=-N}^{N} h_r(u,v)\tilde{R}(x-u, y-v) \tag{20.52}$$

$$G(x,y) = \sum_{u,v=-N}^{N} h_g(u,v)\tilde{G}(x-u, y-v) \tag{20.53}$$

$$B(x,y) = \sum_{u,v=-N}^{N} h_b(u,v)\tilde{B}(x-u, y-v) \tag{20.54}$$

where $\tilde{R}(\bullet)$, $\tilde{G}(\bullet)$, $\tilde{B}(\bullet)$ are defined in Eqs. (20.49), (20.50), and (20.51), and $h_r(\bullet)$, $h_g(\bullet)$, $h_b(\bullet)$ are linear filters of size $(2N + 1) \times (2N + 1)$. Different forms of interpolation (nearest neighbor, bilinear, bicubic, etc.) differ in the form of the interpolation filter used. For the Bayer array, the bilinear filter for the red and blue channels are separable.

The correlations are periodic because the color filters in a CFA are typically arranged in a periodic pattern. For example, in Fig. 20.14, the red samples in the odd rows and even columns are the average of their closest horizontal neighbors, the red samples in the even rows and odd columns are the average of their closest vertical neighbors, and the red samples in the even rows and columns are the average of their closest diagonal neighbors:

$$R(2x+1,2y) = \frac{R(2x+1,2y-1)}{2} + \frac{R(2x+1,2y+1)}{2} \tag{20.55}$$

$$R(2x,2y+1) = \frac{R(2x-1,2y+1)}{2} + \frac{R(2x+1,2y+1)}{2} \tag{20.56}$$

$$R(2x,2y) = \frac{R(2x-1,2y-1)}{4} + \frac{R(2x-1,2y+1)}{4} + \frac{R(2x+1,2y-1)}{4} + \frac{R(2x+1,2y+1)}{4} \tag{20.57}$$

In the above simple case, the estimated samples are perfectly correlated to their neighbors. Consequently, a CFA interpolated image can be detected (in the absence of noise) by noticing that every other sample in every other row or column is perfectly correlated to its neighbors while the non-interpolated samples are less likely to be correlated in precisely the same manner. The lack of correlations produced by CFA interpolation can be used to expose it as a forgery since tampering or splicing of two images from different cameras will create inconsistent correlations.

If the specific form of the correlations, α, are priori knowledge, it is straightforward to determine which samples satisfy Eq. (20.31). However, neither the re-sampling amount nor the specific form of the correlations are typically known in practice. In this case, the expectation/maximization (EM) algorithm is usually employed to determine if a signal has been re-sampled [9, 10]. EM can estimate a set of periodic samples that are correlated to their neighbors, and the specific form of these correlations.

Chromatic Aberration

In an ideal imaging system, light passes through the lens and is focused to a single point on the sensor. However, real optical imaging systems deviate from such ideal models. They fail to perfectly focus light of all wavelengths due to their imperfections, which is known as chromatic aberration. There are two types of chromatic aberration: longitudinal and lateral. Longitudinal aberration causes different wavelengths to focus at different distances from the lens while lateral aberration is attributed to different wavelengths focusing at different positions on the sensor.

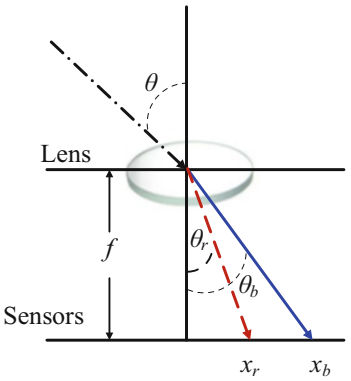


Fig. 20.15 1-D Aberration—The refraction of light in one dimension [11]

When tampering with an image, these aberrations are often disturbed and fail to be consistent across the image. This reveals the presence of forgery.

In classical optics, the refraction of light at the boundary between two media is described by Snell's Law [11]:

$$n \ \sin(\theta) = n_f \ \sin(\theta_f) \tag{20.58}$$

where θ is the angle of incidence, θ_f is the angle of refraction. n and n_f are the refractive indices of the media through which the light passes. The refractive index of glass, n_f, depends on the wavelength of the light that traverses it, which results in polychromatic light being split according to wavelength when it passes through the lens and strikes the sensor. Figure 20.15 gives an example of the splitting of short wavelength (solid blue ray) and long wavelength (dashed red ray) light. The result of this splitting of light is termed lateral chromatic aberration. The incident light reaches the lens at an angle θ. Then it is split into short wavelength (solid blue ray) and long wavelength (dashed red ray) light with an angle of refraction of θ_r and θ_b. These rays strike the sensor at positions x_r and x_b. By Snell's law, yielding:

$$n \ \sin(\theta) = n_r \sin(\theta_r) \tag{20.59}$$

$$n \ \sin(\theta) = n_b \sin(\theta_b) \tag{20.60}$$

Combing (20.56) and (20.57), yielding:

$$n_r \ \sin(\theta_r) = n_b \sin(\theta_b) \tag{20.61}$$

Dividing both sides by $\cos(\theta_b)$ gives:

$$\begin{aligned} n_r \sin(\theta_r)/\cos(\theta_b) &= n_b \ \tan(\theta_b) \\ &= n_b x_b / f \end{aligned} \tag{20.62}$$

where f is the lens-to-sensor distance. If we assume that the differences in angles of refraction are relatively small, then $\cos(\theta_r) \approx \cos(\theta_b)$. Equation (20.60) turns to be:

$$\begin{aligned} n_r \sin(\theta_r)/\cos(\theta_r) &\approx n_b x_b/f \\ n_r \tan(\theta_r) &\approx n_b x_b/f \\ n_r x_r/f &\approx n_b x_b/f \\ x_r &\approx \alpha x_b \end{aligned} \tag{20.63}$$

where $\alpha = n_b/n_r$.

For a two-dimensional lens and sensor, the distortion caused by lateral chromatic aberration takes a form similar to Eq. (20.63). An incident ray reaches the lens at angles θ and ϕ, relative to the $x = 0$ and $y = 0$ planes, respectively. The application of Snell's law yields:

$$n_r \sin(\theta_r) = n_b \sin(\theta_b) \tag{20.64}$$

$$n_r \sin(\phi_r) = n_b \sin(\phi_b) \tag{20.65}$$

Following the above 1-D derivation yields the following 2-D model:

$$(x_r, y_r) \approx \alpha(x_b, y_b) \tag{20.66}$$

Note that this model is simply an expansion/contraction about the center of the image.

In real lenses, the center of optical aberrations is often different from the image center due to the complexities of multi-lens systems. The previous model can therefore be augmented with an additional two parameters, (x_0, y_0), to describe the position of the expansion/contraction center. The model now takes the form:

$$x_r = \alpha(x_b - x_0) + x_0 \tag{20.67}$$

$$y_r = \alpha(y_b - y_0) + y_0 \tag{20.68}$$

Taking lateral chromatic aberration of green channel as an example, the aberration between the red and green channels, and between the blue and green channels can be estimated. Then, deviations or inconsistencies in these models can be used as evidence of tampering. Let (x_1, y_1, α_1) and (x_2, y_2, α_2) denote the red to green and blue to green distortions, respectively. Lateral chromatic aberration results in an expansion or contraction between the color channels, and hence a misalignment between the color channels. There are several metrics that may be used to quantify the alignment of the color channels. An metric based on mutual information is usually used to help contend with the inherent intensity differences across the color channels.

Let $R(x, y)$ and $G(x, y)$ denote the red channel and the green channel of an RGB image, respectively. A corrected version of the red channel is denoted as $R(x_r, y_r)$ where:

$$x_r = \alpha_1(x - x_1) + x_1 \tag{20.69}$$

$$y_r = \alpha_1(y - y_1) + y_1 \tag{20.70}$$

The model parameters are determined by maximizing the mutual information between $R(x_r, y_r)$ and $G(x, y)$ as follows:

$$\text{argmax}_{x_1, y_1, \alpha_1} I(R; G) \tag{20.71}$$

where R and G are the random variables from which the pixel intensities of $R(x_r, y_r)$ and $G(x, y)$ are drawn. The mutual information between these random variables is defined to be:

$$I(R; G) = \sum_{r \in R} \sum_{g \in G} \Pr(r, g) \log\left(\frac{\Pr(r, g)}{\Pr(r)\Pr(g)}\right) \tag{20.72}$$

where $\Pr(\bullet, \bullet)$ is the joint probability distribution, and $\Pr(\bullet)$ is the marginal probability distribution.

Usually, by using a brute-force iterative search, the maximal metric of mutual information can be obtained. Specifically, a relatively course sampling of the parameter space for x_1,y_1,α_1 is searched on the first iteration. On the second iteration, a refined sampling of the parameter space is performed about the maximum from the first stage. This process is repeated for a specified number of iterations. The brute-force search is computationally demanding, while it ensures to reach global minimum value.

20.2.3.3 Format-Based Image Forgery Detection

When working with larger images of greater bit depth, the images tend to become too large to transmit over a standard Internet connection. In order to display an image in a reasonable amount of time, techniques must be incorporated to reduce the image's file size. These techniques make use of mathematical formulas to analyze and condense image data, resulting in smaller file sizes. This process is called compression.

In images there are two types of compression: Lossy and lossless. Both methods save storage space, but the implemented procedures are different. Lossy compression creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate, resulting in close approximations of the original image, although not an exact duplicate. An example of an image format that uses this compression technique is JPEG (Joint Photographic Experts Group). Lossless compression, on the other hand, never removes any information from the original image, but instead represents data in mathematical formulas. The original image's integrity is maintained and the decompressed image output is bit-by-bit identical to the original.

JPEG (Joint Photographic Experts Group) is the most popular and commonly used compression standard, which has been found in variety of applications. Most digital cameras export JPEG file format, the suffixes of which are ".jpg" and ".jpeg". Consequently, JPEG compression properties based digital image forgery detection is an important method for digital image forensics. In this section, the JPEG compression standard and its processes are firstly described. Then, JPEG Compression Properties based image forgery detection is presented from the aspects of JPEG header, JPEG blocking, and Double JPEG compression.

JPEG Compression

JPEG compression is widely used in the still continuous-tone compression including grey-scale image and full color image. It supports lossless compression and lossy compression. The techniques include DCT, Quantization, Huffman, Run Length Encoding, Entropy coding and so on.

Now most images are compressed based on the JPEG baseline system, which is the core of the JPEG algorithm, as shown in Fig. 20.16. It is based on the sequential DCT-based model.

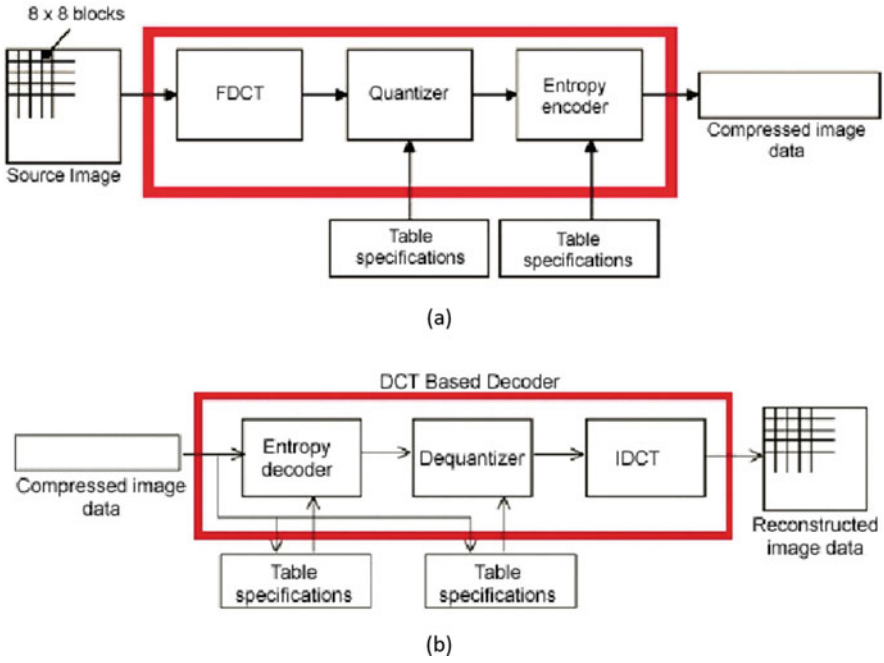


Fig. 20.16 JPEG baseline system. (**a**) Encoding Schematic. (**b**) Decoding Schematic

Data Unit

To prepare for processing, the images matrix is broken up into 8×8 squares (the size was determined when the JPEG standard was created as a balance between image quality and the processing power of the time).

The JPEG deals with the grayscale image, so for the color images they are separated into the different channels (each equivalent to a greyscale channel) and treated individually. Usually the RGB image should transform into YUV space before the encoding process. YUV, also called YCrCb, is a color encoder method. Y represents Luminance that is the greyscale value, U and V represent Chrominance that describe the image tone (also use Cr representation) and saturation level (also use Cb representation).

Compared with RGB, YUV occupy little bandwidth while human eye is insensitive to the small changes in brightness and is insensitive to the Chroma. Consequently, a high compression ratio can be obtained by throwing a lot of chrominance data. The transition formulas is as follows:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{20.73}$$

Additionally, before being fed into the DCT, every value in the matrix is shifted from an unsigned integer with range $[0, 2^{p-1} - 1]$ to a signed integer with range $\lfloor -(2^{p-1}), 2^{p-1} - 1 \rfloor$ by subtracting 2^{p-1} from the value, where p is the number of bits per channel. In the case of the standard 8-bits channel, the numbers are shifted from [0,255] to [−128,127] by subtracting 128. This centers the activity around on 0 for easier handling with cosine functions.

DCT and IDCT

The discrete cosine transform (DCT) is closely related to the Discrete Fourier Transform (DFT). Both take a set of points from the spatial domain and transform them into an equivalent representation in the frequency domain. The difference is that while the DFT takes a discrete signal in one spatial dimension and transforms it into a set of points in one frequency the discrete cosine transform (for an 8×8 block of values) takes a 64-point discrete signal, which can be thought of as a function of two spatial dimensions x and y, and turn them into 64 basis-signal amplitudes (also called DC coefficients) which are in terms of the 64 unique orthogonal two-dimensional "spatial frequencies". The DCT coefficient values are the relative amounts of the 64 spatial frequencies in both directions is the "DC coefficient" and the rest are called "AC coefficients."

After DCT, the original data has been organized in term of importance. The human eye has more difficulty discriminating between higher frequencies than low and most computer data is relatively low frequency. Low frequency data carries

more important information than the higher frequencies. The data in the DCT matrix is organized from lowest frequency in the upper left to highest frequency in the lower right. This prepares the data for the next step, quantization.

Quantization

Quantization achieves two goals: It allows more important information to keep in the low frequency data; It makes the high frequency coefficient values close to zero. By quantization, every element in the 8×8 DCT matrix is divided by a corresponding element in a quantization matrix S to yield a matrix Q according to the formula:

$$Q(u, v) = round\left[\frac{F(u, v)}{S(u, v)}\right] \tag{20.74}$$

The matrix S generally has lower values in the upper left. It increases as they get closer to the lower righter. S could be any matrix while the JPEG committee has recommended certain ones that seem to work well, the following are the quantization table used for Luminance and Chrominance.

Luminance Quantization Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Chrominance Quantization Table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

From the above two tables, we can know the Chroma quantization step is bigger than Luma', that is to say the Luma quantization precision is finer than Chroma'.

It is convenient for user to customize the level of compression at runtime to fine tune the quality or compression ratio since the quantization table can be defined by

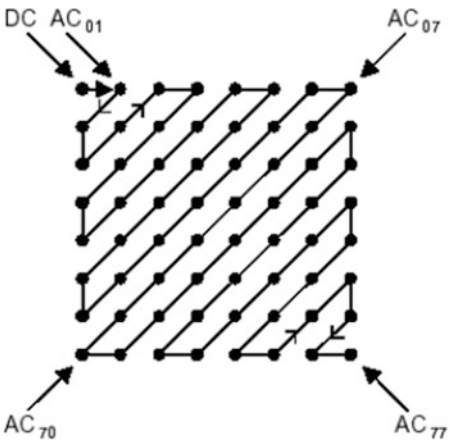


Fig. 20.17 The zig-zag pattern of entropy encoding

the user. If the user wants better quality at the price of compression, he can lower the values in the quantization table. If he wants higher compression with less image quality he can raise the values in the matrix.

Entropy Encoding

Instead of reading across the rows, row after row, JPEG compression reads along the diagonals. This tends to group the lower coefficients in the beginning of the string and distribute the zeros in longer continuous strings, as shown in Fig. 20.17.

The coefficients are scaled-except for the DC component, which is the scaled average intensity value of the entire image.

Usually, the AC coefficients begin with a string of zeros. Before delivering the coefficients to the receiving decoder, the string of zeros is replaced with a code that describes how many zeros there are in a line. For example, a string of 15 zeros 000000000000000 might be described as 01111 (1111 is a binary for the number 15). Then, the condensed string of numbers is passed on to a Huffman encoder, which is lossless. After encoding, the header information should be added to the data, including the quantization factor, the scale factors and the Huffman tables.

Note: JPEG supports two kinds of image reconstruction models: sequential model and progressive model. Sequential model completes the image coding and transporting at one time while progressive model finishes the task through several times.

JPEG Compression Properties Based Image Forgery Detection

The JPEG file format has emerged as a universal image standard employed by nearly all commercial digital cameras. As such, the details of this encoding scheme, and the variations among cameras of different make and software engineers yield images

with different JPEG compression parameters. Therefore, the JPEG parameters used by cameras of different make and model form the camera signature, which can be used for image authentication.

JPEG Header Based Image Forgery Detection

A camera signature extracted from a JPEG image header consists of information about quantization tables, Huffman codes, thumbnails, and EXIF (Exchangeable Image File) format.

The JPEG standard does not enforce any specific quantization table or Huffman code. Therefore, camera and software engineers are free to balance compression and quality to their own needs, which can be used for authentication. Specifically, the first three components of the JPEG header are the image dimensions, quantization table, and Huffman code. The image dimensions are used to distinguish between cameras with different sensor resolution. The set of three 8×8 quantization tables are specified as a one dimensional array of 192 values. The Huffman code is specified as six sets of 15 values corresponding to the number of codes of length 1, 2 ... 15: Each of three channels requires two codes, one for the DC coefficients and one for the AC coefficients. This representation eschews the actual code for a more compact representation that distinguishes codes based on the distribution of code lengths. In total, 284 values are extracted from the full resolution image: 2 image dimensions, 192 quantization values, and 90 Huffman codes [19].

A thumbnail version of the full resolution image is often embedded in the JPEG header. The next three components of the camera signature are extracted from this thumbnail image, which is created by cropping, filtering and down-sampling the full-resolution image. The thumbnail is then typically compressed and stored in the header as a JPEG image. Some camera manufacturers do not create a thumbnail image, or do not encode them as a JPEG image. In such cases, a value of zero can be assigned to all of the thumbnail parameters. Rather than being a limitation, the lack of a thumbnail is considered as a characteristic property of a camera. In total, 284 values are extracted from the thumbnail image: 2 thumbnail dimensions, 192 quantization values, and 90 Huffman codes.

The final component of the camera signature in the JPEG header is extracted from an image's EXIF metadata. The metadata stores a variety of information about the camera and image. Generally, there are five main image file directories (IFDs) in the metadata: Primary; EXIF; Interoperability; Thumbnail; and GPS. Camera manufacturers are free to embed any (or no) information into each IFD. A compact representation of their choice can be extracted by counting the number of entries in each of these five IFDs. The total number of any additional IFDs, and the total number of entries in each of these are used as an additional feature because the EXIF standard allows for the creation of additional IFDs. Some camera manufacturers customize their metadata in ways that do not conform to the EXIF standard, yielding errors when parsing the metadata. These errors are considered to be a feature of camera design and the total number of parser errors are used as an additional feature. In total,

8 values are extracted from the metadata: 5 entry counts from the standard IFDs, 1 for the number of additional IFDs, 1 for the number of entries in these additional IFDs, and 1 for the number of parser errors [19].

Any manipulation of the JPEG image will alter the original signature, and can therefore be detected. Specifically, by extracting the signature from an image and comparing it to a database of known authentic camera signatures, the photo alteration can be detected. Any matching camera make and model can be compared to the make and model specified in the image's EXIF metadata. Any mismatch is strong evidence of some form of tampering.

JPEG Blocking Based Image Forensics

The basis for JPEG compression is the block DCT transform. Because each 8×8 pixel image block is individually transformed and quantized, artifacts appear at the border of neighboring blocks in the form of horizontal and vertical edges [20]. These blocking artifacts may be disturbed when an image is manipulated.

Blocking artifact characteristics matrix (BACM) is developed in [21] to recognize whether an image is an original JPEG image or it has been cropped from another JPEG image and re-saved as a JPEG image. For uncompressed images, this matrix is random, while for a compressed image, this matrix has a specific pattern. When an image is cropped and recompressed, this pattern is disrupted. Specifically, the BACM exhibits regular symmetrical shape in the original JPEG image. The regular symmetrical property of the BACM is destroyed if the images are cropped from another JPEG image and re-saved as JPEG images.

Another way to detect JPEG image forgery based on JPEG blocking artifacts is measuring its quality inconsistency [22]. Blocking artifact measure is calculated based on the estimated table, which is estimated based on power spectrum of the histogram of the DCT coefficients. The inconsistencies of the JPEG blocking artifacts are then checked as a trace of image forgery. This approach is able to detect spliced image forgeries using different quantization table, or forgeries which would result in the blocking artifact inconsistencies in the whole images, such as block mismatching and object retouching.

Double JPEG Compression

When manipulating an image, it requires to load the image into a photo-editing software program and resaved. If the original image is JPEG format, it is likely that the manipulated images is also stored in this format. In this scenario, the manipulated image is compressed twice. This double compression introduces specific artifacts, which is not presented in singly compressed images [10]. Therefore, the presence of these artifacts can be used as evidence of some manipulation. Note that double JPEG compression does not necessarily prove malicious tampering [20].

Consider the example of a generic discrete 1-D signal $f(x)$. Quantization is a point-wise operation that is described by a one-parameter family of functions:

$$q_a(u) = \left\lfloor \frac{u}{a} \right\rfloor \tag{20.76}$$

where a is the quantization step (a strictly positive integer), and u denotes a value in the range of $f(x)$.Dequantization brings the quantized values back to their original range: $q_a^{-1}(u) = au$. Note that quantization is not invertible, and that dequantization is not the inverse function of quantization. Double quantization that results from double compression is given by:

$$q_{ab}(u) = \left\lfloor \left\lfloor \frac{u}{b} \right\rfloor \frac{b}{a} \right\rfloor \tag{20.77}$$

where a and b are the quantization steps. Double quantization can be represented as a sequence of three steps: (1) Quantization with step b, followed by (2) dequantization with step b, followed by (3) quantization with step a.

Consider a set of coefficients normally distributed in the range [0,127]. Fig. 20.18c is an example of the histograms of the coefficients double-quantized with steps 3 followed by 2 and Fig. 20.18d is the histograms of the coefficients double-quantized with steps 2 followed by 3 [18]. When the step size decreases (Fig. 20.18c), some bins in the histogram are empty, because the first quantization places the samples of the original signal into 42 bins, while the second quantization redistributes them into 64 bins. When the step size increases (Fig. 20.18d), some bins contain more samples than their neighboring bins, because the even bins receive samples from four original histogram bins while the odd bins receive samples from only two. The periodicity of the artifacts introduced into the histograms can be used to detect double JPEG compression.

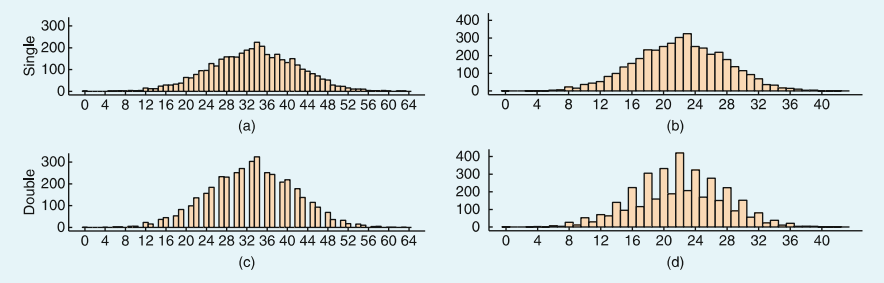


Fig. 20.18 Distribution of single (**a**, **b**) and (**c**, **d**) double quantized coefficients [18]

Review Questions

1. What is sampling and quantization in Digital Image Processing?
2. Compute the discrete cosine transform matrix for n = 4.
3. What are the differences between active image forgery detection and passive-blind image forgery detection?
4. What is Copy-Move Forgery and Image-Splicing Forgery in digital image forgery?
5. Describe in your own words, how does the keypoint-based copy-move forgery detection technique work?
6. Which of the following is not an active approach for digital image forgery detection?

   (a) Digital Watermarking
   (b) Digital Signature
   (c) Hash function
   (d) None of the above

7. Describe passive-blind image forgery detection process.
8. Digital signature provides _____.

   (a) confidentiality
   (b) integrity
   (c) availability
   (d) authentication

20.3 Practice Exercise

The objective of this exercise is to practice basic image manipulation techniques using Matlab and to learn how to detect image forgery through the implementation of a copy-move forgery detection algorithm based on DCT, which has been introduced in [24]. For more details on the algorithm, please refer to [24], particularly Sect. 4.2 in [24].

20.3.1 Setting Up Practical Exercise Environment

For this exercise, we will Matlab. Download and install Matlab from the following web site onto your computer.

https://cn.mathworks.com/products/matlab.html

Also, download CoMoFoD—Image Database for Copy-Move Forgery Detection [26, 27] from the following web site.

http://www.vcl.fer.hr/comofod/

20.3.2 Exercises

Part A: Compute DCT

Table 20.2 is an 8×8 block data of an original image. Please compute the DCT of the data in Table 20.2 by using Matlab and fill in all the blanks of Table 20.3.

Note that the Matlab code you developed for this part of exercise can be used in feature extraction process of the following copy-move forgery detection exercises.

Part B: Implement DCT-Based Detection of Copy-Move Forgery

In this part of exercise, we focus on detection of copy-move forgery in which a part of the image is copied and pasted somewhere else in the image with the intent to cover an important image feature. This exercise is about realizing a robust match detection for copy-move forgery which matches the pixel representation of the blocks that consists of quantized DCT coefficients, called DCT-based detection of copy-move forgery in images [24]. We firstly review the algorithm framework of copy-move forgery detection shown in Fig. 20.19. Note that this is a generic framework for DCT-based detection of copy-move forgery. Then, we describe the flow of the algorithm used in the exercise in detail. Finally, we implement each step of the algorithm by using Mtalab.

Next, we describe the algorithm flow for detecting copy-move forgery and identifying two or more blocks with similar features [24]:

Table 20.2 Data of original image

120	137	146	147	143	138	131	123
111	125	140	135	156	138	135	119
99	114	146	147	150	155	144	135
94	105	129	153	146	138	149	143
97	97	119	140	155	144	137	141
88	102	125	141	134	134	147	146
85	103	122	126	143	140	141	147
76	97	117	128	126	140	143	135

Table 20.3 DCT of the data in Table 20.2

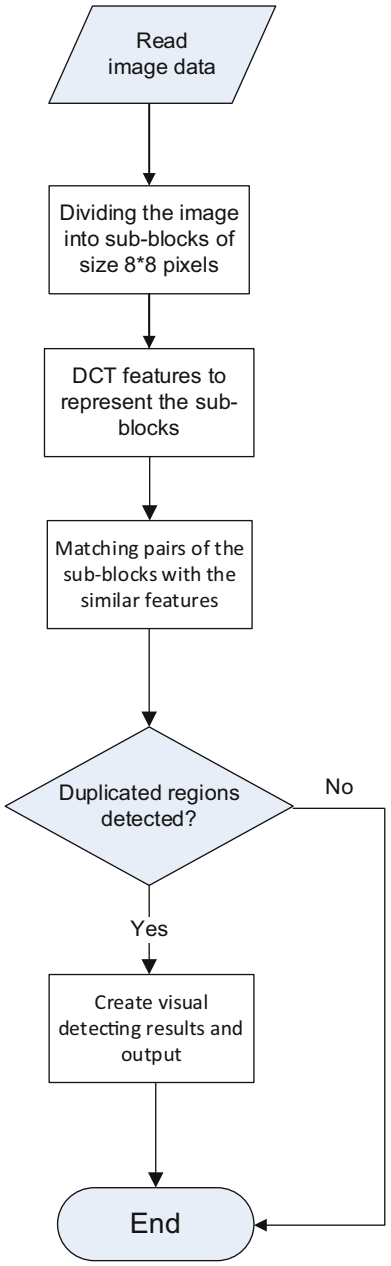


Fig. 20.19 The algorithm framework of copy-move forgery detection [23]

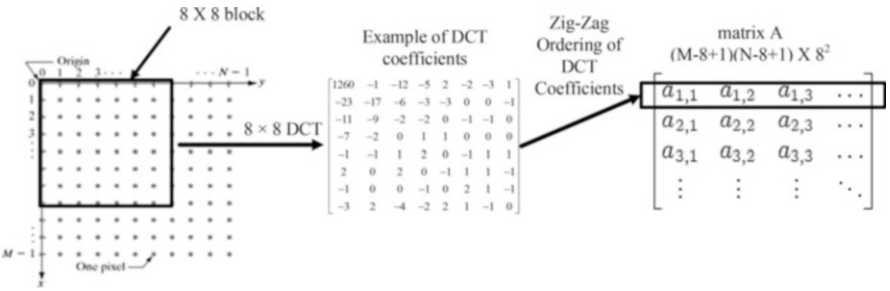


Fig. 20.20 Representing B X B block using DCT coefficients (B = 8)

1. **The input image is a grayscale image I of the size $M \times N$. If it is a color image, it should be converted to a grayscale image firstly using a standard formula $\mathbf{I = 0.299R + 0.587G + 0.114B}$.**
2. **Slide the $B \times B$ pixels window from the upper left corner to the button right, and split the image into $(M - B + 1) \times (N - B + 1)$ blocks.**
3. **Compute DCT of each block and extract the DCT coefficients. The quantization step is controlled by a user-specified parameter Q, and each DCT coefficient will be quantized by Q and then rounded to its nearest integer. This parameter is equivalent to the quality factor in JPEG compression. The larger Q-factor leads to finer quantization, the blocks must match more closely in order to be identified as similar. Lower values of the Q-factor produce more matching blocks, thereby possibly having some false matches. In this exercise, we set the Q as 0.1. Note that the Q will need to be adjusted if you use another image from the dataset.**

And reshape the $B \times B$ quantized coeffcient matrix to a row (e.g., in the zig-zag order) in the matrix A. So form a $(M - B + 1)(N - B + 1) \times B^2$ matrix A. Also, the rows of matrix A are lexicographically sorted (Fig. 20.20).

Note that after you divide the image into blocks, you can simply compare all blocks pixel by pixel to identify some copied and moved blocks, also known as exact match. However, exact match isn't very reliable, and can be easily defeated by using image manipulation. The advantage of using the DCT is the ability to reliably identify copied and moved blocks.

4. **Compare every row to its adjacent row in matrix A. If they are equal, the positions of these matching blocks are saved into a list. For simplicity, the position of a block can be defined using its top-left pixel. Assume that (x_i, y_i) and (x_j, y_j) are the locations of two matching blocks. Then, calculate a shift vector s between the two matching blocks as follows**

$$s = (s_1, s_2) = \left(x_i - x_j, y_i - y_j\right).$$

Also, a shift-vector counter C will be used to record a number of matched occurrences, and increment on each matching pair of blocks

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

Note that the shift vectors s and –s correspond to the same shift. So we can multiply it by -1 if $s_1 \leq 0$. It is known as normalization. Also, the shift-vector counter is initialized to zero when the algorithm starts.

5. **Loop over the counts of all the shift vectors and identify these shift vectors whose counter exceeds a pre-defined threshold *T* by the user. Then, find all the matching blocks of a specific shift vector and color them using the same color.**

Now you are ready to implement the algorithm defined above step-by-step using Matlab. In this exercise, we select the image file "029_F.png" from CoMoFoD as the example. For other images in the dataset, you may need to adjust the parameters suggested above in order to detect the copy-move forgery accurately.

The follow is the basic skeleton of your Matlab code. The actual code for your program goes in place of **Your Program Goes Here**.

```
%Read Image Data from the image file 029_F.png
Your Program Goes Here
%Convert Image to grayscale
Your Program Goes Here
%Dividing the image into 8*8 sub-blocks and compute DCT of each block
Your Program Goes Here
%Computing the quantized DCT coefficients using the Q-factor (Q=0.1)
Your Program Goes Here
%Create and sort the matrix A
Your Program Goes Here
%Find matching blocks and construct shift vectors and record their matching occurrences into their counters
Your Program Goes Here
%Loop over the counts of all the shift vectors and identify these shift vectors whose counter exceeds a pre-defined threshold T (T=100)
Your Program Goes Here
%Color duplicated regions and display the colored image
Your Program Goes Here
```

Fig. 20.21 Original image and forged image used in the exercise. (**a**) The original image, (**b**) The forged image

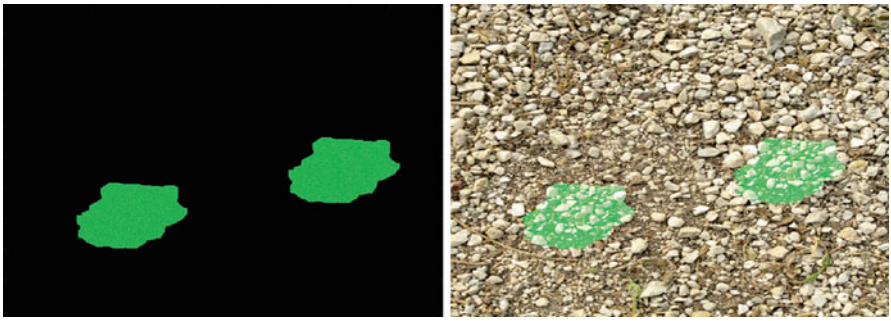


Fig. 20.22 The detection result

In the end, you should be able to observe the following results after your program has completed. Fig. 20.21a is the original image, Fig. 20.21b is the forged image, and Fig. 20.22 is the detection result.

References

1. https://www.bbc.com/news/blogs-trending-31970420
2. https://www.cnn.com/2014/09/19/us/california-lawyer-suspension-fake-celebrity-photos/index.html
3. H. Farid. Digital Image Forensics, http://www.cs.dartmouth.edu/farid/downloads/tutorials/digitalimageforensics.pdf

4. J. Redi, W. Taktak, J.-L. Dugelay. Digital Image Forensics: a booklet for beginners Multimedia Tools and Applications, vol. 51, pp. 133–162, October 2011
5. Gajanan K. Birajdar, Vijay H. Mankar,Digital image forgery detection using passive techniques: A survey, Digital Investigation, 2013, vol. 10, pp. 226–245.
6. C. I. Podilchuk and E. J. Delp, Digital watermarking: Algorithms and applications, IEEE Signal Processing Magazine, 2001, pp. 33–46.
7. C. Paar and J. Pelzl, Understanding Cryptography—A Textbook for Students and Practitioners. Berlin, Germany: Springer-Verlag, 2010.
8. X. Hou, J. Harel, and C. Koch, Image Signature: Highlighting Sparse Salient Regions, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 1, 2012.
9. N. Warif, A. Wahab, M. Idris, R. Ramli, R. Salleh, S. Shamshirband, K.-K. Choo, Copy-move forgery detection: Survey, challenges and future directions, Journal of Network and Computer Applications, vol. 75, pp. 259–278, 2016.
10. W. Luo, Z. Qu, F. Pan, J. Huang. A survey of passive technology for digital image forensics. Frontiers of Computer Science in China, vol. 1, no. 2, pp. 166-179, 2007.
11. M.K. Johnson and H. Farid. Exposing Digital Forgeries Through Chromatic Aberration. ACM Multimedia and Security Workshop, Geneva, Switzerland, 2006
12. A. Popescu, H. Farid, Exposing digital forgeries by detecting traces of re-sampling. IEEE Transactions on Signal Process 2005, vol. 53, no. 2, pp. 758–67.
13. C. Song, X. Lin. Natural Image Splicing Detection Based on Defocus Blur at Edges. Proc. IEEE/CIC International Conference on Communications in China (ICCC), Shanghai, China, 2014.
14. L. B. Lucy. An iterative technique for the rectification of observed distributions. The astronomical journal, vol. 79, no. 6, pp. 745–754, 1974
15. N. Wiener. Extrapolation, interpolation, and smoothing of stationary time series, vol 2. Cambridge, MA: MIT press, 1949
16. R. Fergus, B. Singh, A. Hertzmann, S. Roweis, W. Freeman. Removing camera shake from a single photograph. In: Proceedings of ACM SIGGRAPH, pp 787–794, 2006
17. T. Kenig, Z. Kam, A. Feuer. Blind image deconvolution using machine learning for three-dimensional microscopy. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 12. pp. 2191–2204, 2010
18. Anat Levin, Yair Weiss, Frédo Durand, William T. Freeman: Understanding Blind Deconvolution Algorithms. IEEE Trans. Pattern Anal. Mach. Intell. vol. 33, no. 12, pp. 2354-2367, 2011
19. E. Kee, M.K. Johnson, and H. Farid. Digital image authentication from JPEG headers. IEEE Transactions on Information Forensics and Security, 2011, vol. 6, no. 3, pp. 1066-1075.
20. H. Farid, A survey of image forgery detection, IEEE Signal Processing Magazine, vol. 2, no. 26, pp. 16–25, 2009.
21. W. Luo, Z. Qu, J. Huang, and G. Qiu, A novel method for detecting cropped and recompressed image block, IEEE Conference on Acoustics, Speech and Signal Processing, Honolulu, HI, 2007, pp. 217–220.
22. S. Ye, Q. Sun, and E. C. Chang, Detecting digital image forgeries by measuring inconsistencies of blocking artifact, IEEE International Conference on Multimedia and Expo, Beijing, China, 2007, pp. 12–15.
23. Y. Huang, W. Lu, W. Sun, D. Long. Improved DCT-based detection of copy-move forgery in images. Forensic Science International, vol. 206, no. 1-3, pp. 178–184, 2011
24. J. Fridrich, D. Soukalm, J. Luka, Detection of Copy-Move Forgery in Digital Images, Proc. of DFRWS 2003, Cleveland, OH, USA, August 5-8 2003
25. http://jaco-watermark.sourceforge.net/
26. D. Tralic, I. Zupancic, S. Grgic, M. Grgic, "CoMoFoD - New Database for Copy-Move Forgery Detection", in Proc. 55th International Symposium ELMAR-2013, pp. 49-54, September 2013
27. CoMoFoD - Image Database for Copy-Move Forgery Detection. http://www.vcl.fer.hr/comofod/