Image Forensics

Hany Farid
University of California, Berkeley, CA, USA

Synonyms

Digital forensics; Image authentication

Related Concepts

- ▶ Image Processing
- ▶ Iconic Matching
- ▶ Sensing

Definition

Image forensics refers to the analysis of an image to determine if it has been manipulated from the time of its recording. The techniques described here – so called passive techniques – operate in the absence of digital watermarks, signatures, or specialized hardware. Instead, these techniques analyze physical, geometric, optical, sensor, and file properties for inconsistencies that may arise from image manipulation.

Background

History has shown that many autocratic leaders had photographs manipulated in an attempt to rewrite history. These men understood the power of photography and that if they changed photographs they could change history. Cumbersome and time-consuming darkroom techniques were required to alter the historical record on behalf of Stalin and others. Today, powerful and low-cost digital technology coupled with sophisticated rendering and synthesis techniques and the broad and rapid reach of social media have made it far easier to alter and disseminate digital content. The resulting fakes are often very difficult to detect and are having a significant impact in many different areas of society.

Doctored photographs are appearing in tabloid and fashion magazines, government media, mainstream media, social media, online auction sites, online dating sites, political ad campaigns, and scientific journals. More recently, the coupling of fake news with fake imagery has been used by individuals and state-sponsored entities to disrupt democratic elections, incite civil and political discord, and fuel horrific violence.

The technology that can distort and manipulate digital media is developing rapidly, and the implications of not authenticating content quickly and accurately are becoming more pronounced. The goal of the field of image (and video/audio) forensics is to develop techniques for quickly and accurately authenticating digital content.

At their foundation, most image forensic techniques rely on understanding the imaging pipeline, from the interaction of light with the physical 3-D world, the refraction of light as it passes through the camera lenses, the

K. Ikeuchi, *Computer Vision*,
https://doi.org/10.1007/978-3-030-03243-2_877-1

transformation of light to electrical signals in the camera sensor, to the conversion of electrical signals into a digital image file. This entry is organized according to this pipeline, with each section describing a representative set of forensic techniques built on understanding and modeling some aspect of the imaging process. Portions of this entry are adapted from [1].

Application

Physics-Based Forensics

A political ad shows a presidential candidate covertly meeting with a foreign agent. Is the image real or is it a composite created by splicing together two images? The lighting and reflections often hold the answer. Unless the candidate and agent were photographed under identical lighting conditions, there may be discrepancies in the shadows and lighting created by the light source and in the reflections of each actor on nearby shiny surfaces. This section describes a forensic technique for reasoning about the physical plausibility of shadows and reflections [2, 3].

Shadows

Let's start with the simplest situation: A 3-D scene is illuminated by a single, small light source. Consider the scene depicted in Fig. 1a in which a box casts a shadow on the ground. For every point in this cast shadow, there must be a line to the light source that passes through the box. For every point outside the shadow, there must be a line to the light source that is unobstructed by the box. Consider now a line connecting the point at the corner of the shadow and its corresponding point at the corner of the box: Follow this line, and it will intersect the light.

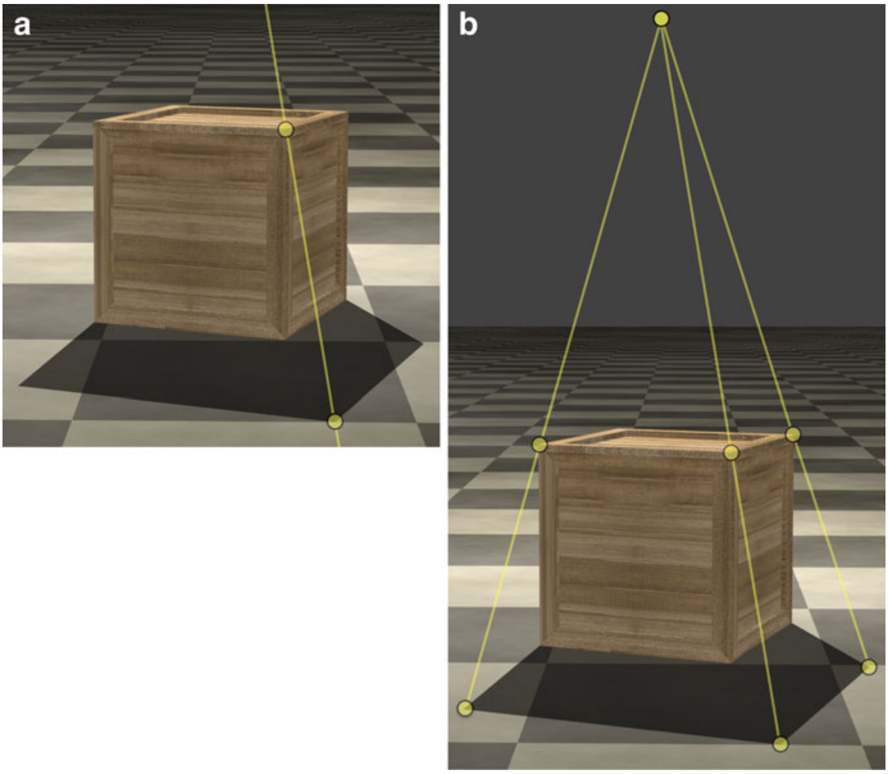


Image Forensics, Fig. 1 A cast shadow constraint connects (**a**) a point on the box's shadow with the corresponding point on the box. Multiple such constraints (**b**) intersect at the projection of the light source

Because straight lines in the physical scene are imaged as straight lines (assuming no lens distortion), the location constraint in the 3-D scene also holds in an image of the scene. Just as the shadow corner, the corresponding box corner, and the light source are all constrained to lie on a single line in the scene, the image of the shadow corner, the image of the box corner, and the image of the light source are all constrained to lie on a single line in the image. This idea is illustrated in Fig. 1a, which shows a line that connects a point on the edge of the shadow to the corresponding point on the box. In the image, the projection of the light source lies somewhere on this line. Now let's connect two more points on the cast shadow to their corresponding points on the box, as in Fig. 1b. We will continue to use the corners of the box because they are distinctive. These three lines intersect at a single point above the box. This intersection is the projection of the light source in the image.

The geometric constraint relating the shadow, the object, and the light holds whether the light source is nearby (a desk lamp) or distant (the sun). This constraint also holds regardless of the location and orientation of the surfaces onto which the shadow is cast. Regardless of the scene geometry, all of the constraint lines intersect at the same point.

The boundary of the image plane in Fig. 1b had to be extended to see the intersection of the three lines. This is because the light source is not visible in the original image of the scene. This will typically be the case, and, depending on where the light is, the lines may have to be extended beyond the image's left, right, top, or (counterintuitively) bottom boundary.

If one or more of the cast shadow constraint lines in an image do not converge on a common intersection, the image may be a fake [3].

Reflections

Somewhat surprisingly, reflections in flat mirror surfaces lend themselves to the same type of analysis as cast shadows. The basic geometry of a reflection is shown in the bottom panel of Fig. 2. Shown on the left is a bird's eye view of three boxes, and shown on the right is their mirror reflection, with the mirror shown in the middle. The reflections are equal in size and equal in distance from the mirror so that corresponding points on the virtual and real objects are connected by parallel lines.

This scene geometry changes when the scene is projected onto a camera sensor. Lines that were parallel when viewed from the plane of the mirror are no longer parallel. Instead, due to perspective projection, these parallel lines converge to a single point, as shown in the top panel of Fig. 2. Because the lines connecting corresponding points in a scene and its reflection are always parallel, these lines should have a common intersection in the image.

If one or more of the reflection constraint lines in an image do not converge on a common intersection, the image may be a fake [2].

Sensor-Based Forensics

Gun barrels are grooved to impart spin to a bullet for increased accuracy and range. Because these grooves introduce distinct markings to the bullet, ballistic techniques can link a bullet to a specific handgun. Similarly, photo forensic techniques can use the distinctive artifacts introduced by a camera's sensor to link an image to a specific device. In addition, inconsistencies in these artifacts can provide evidence of tampering. This section describes forensic techniques for estimating image artifacts introduced by camera sensors [4–6].

Color Filter Array

Modern digital cameras have three channels with different peak sensitivities: R (red), G (green), and B (blue). This means that each pixel in a digital image is represented as three values: R, G, and B. The camera sensor, however, is sensitive to all visible light. To measure the light in one color channel, it is, therefore, necessary to restrict the wavelength of the light that impinges on the sensor. This restriction is accomplished by a color filter array (CFA) that sits atop the sensor. Most CFAs use a Bayer pattern as shown in Fig. 3. The color of each square in the pattern indicates the part of the visible spectrum (R, G, or B) that the filter transmits at that location. Note that the

Image Forensics, Fig. 2 Three boxes are reflected in a mirror (top) and a virtual bird's eye view (bottom) of this scene with the boxes on the left, mirror in the middle, and reflection on the right. The yellow lines connect points on a box to their reflection. When such a scene is imaged under linear perspective projection, lines connecting any point in the scene with its reflection will intersect at a single point, as shown in the top panel

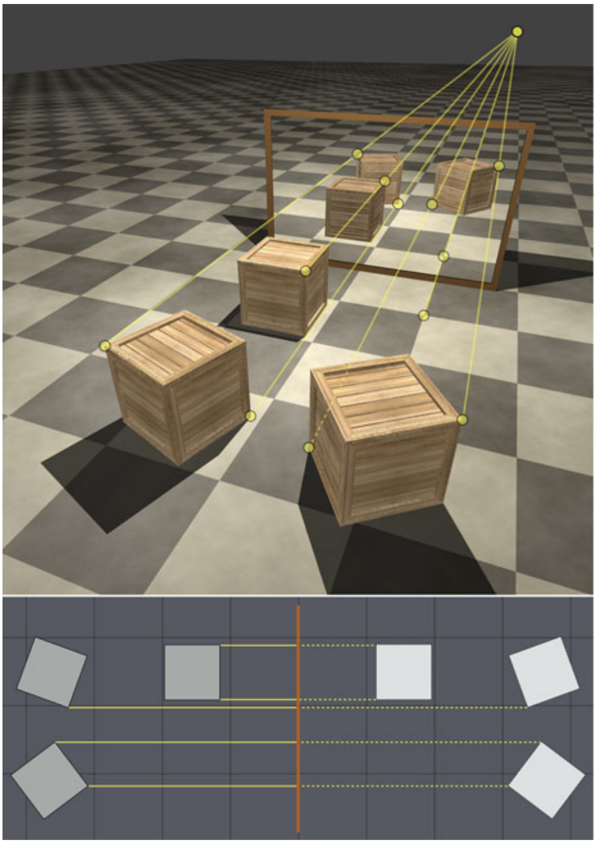
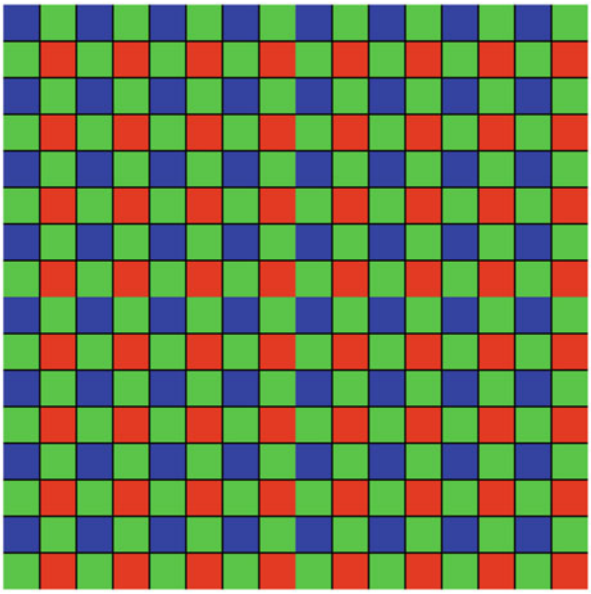


Image Forensics, Fig. 3 A Bayer pattern used to record a subset of RGB pixels



R, G, and B filters of the CFA are distributed in a consistent, periodic pattern. This periodicity is important because it is the key to this next forensic technique.

The CFA transmits only one part of the visible spectrum to each cell on the sensor. For a full color image, it is necessary to have all three measurements for each pixel. Since each sensor cell makes only one measurement, the other two values must be reconstructed. This process – CFA interpolation – reconstructs the missing RGB values by interpolating the surrounding values. Consider, for example, the cell in the second row and second column in the Bayer pattern shown in Fig. 3. This sensor cell measures the R-channel, but not the G-channel or B-channel. Values for the G-channel are measured by cells immediately above, below, and on either side. We can make a reasonable guess about the missing G-channel value from an average of its four neighbors. This basic process underlies all CFA interpolation algorithms.

As a result of CFA interpolation, two-thirds of all RGB values have been reconstructed by interpolating neighboring measurements. And, because the CFA pattern is periodic, the interpolation introduces periodic correlations between these values. These periodic correlations are highly distinctive, and so their presence provides a reliable sign that the image is authentic. We do not know, however, which pixels are CFA interpolated, nor do we know the precise form of the CFA correlations. The classic expectation/maximization (EM) algorithm can be used to simultaneously estimate both and localize parts of an image that violate the expected correlations [4,6].

Photo Response Nonuniformity

In an ideal imaging device, the pixel values of the digital image would accurately reflect the amount of light recorded by each photo detector. Real devices, however, have imperfections that introduce noise in the image. One source of noise arises when stray electrons occur sporadically within sensor cells. These stray electrons introduce noise when they combine with the electrons generated by the photo detector as it responds to light. The resulting noise pattern is random, fluctuating from image to image. Another source of noise arises from slight variations in the size and material properties of the sensor cells themselves. Physical inconsistencies across the sensor cells lead to differences in the efficiency with which the cells convert light into digital pixel values. Some cells consistently underreport the amount of light, while others consistently over-report the amount of light. These variations, termed photo-response nonuniformity (PRNU), lead to a stable noise pattern that is distinctive to the device.

To illustrate how PRNU noise might alter an image, imagine that we point our camera at a perfectly uniform gray wall. A noise-free sensor will record an image with exactly the same value at every pixel. Let’s say that this pixel value is 128 (on a scale of 0 (black) to 255 (white)). PRNU noise modulates the value of each pixel by multiplying 128 by a value slightly less than or slightly greater than 1.0. Unlike sensor noise, which additively modulates the pixel regardless of its value, PRNU modulates the pixel proportional to its value. Also unlike sensor noise, PRNU is a fixed property of the sensor and does not vary from image to image.

With some modest assumptions, a maximum likelihood estimator can be used to estimate the PRNU. Although it is possible to get a crude estimate of a device’s PRNU from a single image, a reliable estimate requires 10–20 images (the exact number depends on the quality of the camera, as well as the quality and content of the images).

The PRNU associated with a particular device is not only stable, it is also distinctive. Even devices of the same make and model have different PRNUs. The stable and distinctive properties of the PRNU allow it to serve two forensic functions: It can be used to detect localized image tampering, and it can be used to link an image to a specific device [5].

File-Based Forensics

The first rule of any forensic analysis must surely be “preserve the evidence.” Because JPEG and other lossy image compression schemes discard and distort image information, they would seem to be a forensic analyst's worst enemy. However, because the details of compression differ across devices, JPEG compression may provide an opportunity for the analyst. This section describes forensic techniques that exploit features of the image file that differ across devices. These features can be used to link an image to a device or to determine whether an image has been re-saved after its initial recording [7, 8].

Quantization

By way of background, the JPEG image format has emerged as the standard for devices that capture digital images. This image format uses a lossy compression scheme that allows for a trade-off between memory size and visual quality.

Specifically, given a three-channel color image, JPEG encoding consists of four basic steps: (1) transform the image from a three-channel color image (RGB) to a three-channel luminance/chrominance image (YCbCr); (2) convert the image into a spatial frequency representation by partitioning the individual channels into non-overlapping 8 × 8 pixel blocks. Each block is then converted to frequency space using a 2-D discrete cosine transform (DCT); (3) quantize the DCT values in each 8 × 8 block by an amount that depends on the frequency and channel (to quantize a value c by an amount q, divide c by q and round up or down to the nearest integer); and (4) perform entropy-encoding on the quantized DCT coefficients. The decoding of a JPEG image follows the same steps but in reverse order.

Device and software engineers fine-tune this trade-off to suit their individual tastes and needs. The resulting variation in JPEG settings provides a distinctive signature for each type of device. These settings can be used to link an image to a specific camera type or to determine whether an image has been re-saved after its initial recording.

Decoding requires knowing the quantization values used to encode the image, and so the quantization values must be stored as part of a JPEG file. The quantization values are specified as a set of 192 integer values organized as three 8 × 8 tables. Each table contains the quantization values for 64 frequencies for one of the three image channels (YCbCr). Because the JPEG standard does not impose the use of specific quantization tables, engineers are free to select all 192 values resulting in quantization tables that vary greatly across devices and software.

An image that is edited and re-saved acquires the JPEG quantization tables of the editing software. If it can be determined that an image's quantization tables are not the same as those of the original recording device, this may indicate that the image has been altered. To test whether the image and device quantization tables match, we need the camera make and model which is typically embedded in the image metadata. (The metadata for a digital image contains data about the camera make and model, the camera settings (e.g., exposure time and focal length), the date and time of image capture, the GPS location of image capture, and much more. The metadata is stored along with the image data in the image file, and it is readily extracted with various programs.) We also need to know the set of possible quantization tables for this device.

For most devices, the range of possible quantization tables can be determined by recording images at all possible pairings of quality, resolution, and, when it is adjustable, aspect ratio. Most devices have a relatively small number of these settings, yielding a small number of possible quantization tables. The tables associated with a particular device can then be compared with the quantization tables of an image purportedly taken with that device [7].

This image-to-device comparison can be made more specific by including the dimensions of the image. Because the compression settings are often associated with a particular resolution, the quantization table and image resolution may be combined into a single device signature. This device signature can be honed further by considering the dimensions and quantization tables of the embedded thumbnail (which is stored as a separate JPEG image, typically with different quantization tables than the full-resolution image).

Markers

A JPEG image file contains data corresponding to the compressed image and thumbnail as well as their quantization tables. The way this data is organized within the file varies across devices and programs adding to the device signature details that can link an image to a specific camera type. A JPEG file consists of multiple labeled segments of data. Each data segment is labeled with a unique *marker*. Although the JPEG standard specifies that certain information must be stored in the JPEG file, it does not specify the location or order of these segments. Camera and software engineers are, therefore, free to organize file data in any way that they choose. As with the JPEG signature described above, the JPEG markers associated with a particular device can then be compared with the markers of an image purportedly taken with that device [8].

Pixel-Based Forensics

In the hands of a talented forger, the methods for manipulating images may be applied so skillfully that the faked image appears authentic, even to a trained eye. But these same methods often leave anomalous patterns that are too regular to be accidental. This section describes forensic techniques that can detect pixel-level anomalies that arise from different forms of tampering [9–11].

Double Compression

Because most digital images are initially recorded in the JPEG format, any image manipulation will lead to multiple compressions: A first compression on the camera and a second compression by the photo editing software. Multiple compressions – and in particular, multiple DCT quantizations – may leave behind a telltale artifact in the underlying DCT coefficients.

Consider a simple example of this double quantization (to quantize a value c by an amount q, divide c by q and round up or down to the nearest integer). Shown in Fig. 4 are the distributions for each of the following stages: (a) initial DCT values, (b) after quantization with $q_1 = 3$, (c) after re-scaling with $q_1 = 3$, and (d) after a second quantization with $q_2 = 2$. The shaded gray areas in panel (d) for bins 1 and 3

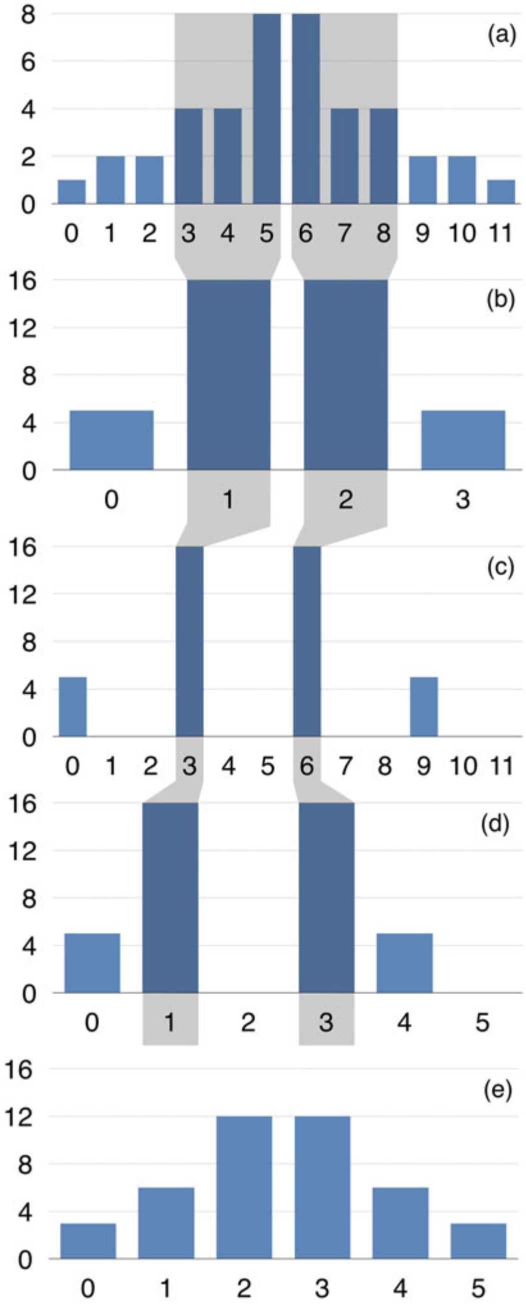


Image Forensics, Fig. 4 Double quantization with first quantization of $q_1 = 3$ and second quantization of $q_2 = 2$ (panels (**a**)-(**d**)). The shaded gray regions illustrate how values in the original histogram (**a**) are transformed through a single quantization (**b**), re-scaling (**c**), and a second quantization (**d**). The empty bins in the resulting distribution (**d**) are a telltale sign of double compression, as compared to the singly compressed distribution (with $q_1 = 2$) in panel (**e**) show how these values are transformed by the first quantization, the re-scaling, and the second quantization. Note that bins 2 and 5 are empty in the final distribution (d). This shouldn't be surprising because we initially condensed 12 bins into four and then later expanded the range to five bins.

For comparison, shown in Fig. 4e is the original distribution after a single quantization with $q_1 = 2$. This singly compressed distribution has the same number of bins as in panel (d), but the content of the bins is strikingly different. Unlike the doubly compressed distribution, the single compression produces no empty bins. The anomalous distribution after double compression is the telltale sign that the image was recompressed after the original recording.

When present, the anomalous pattern is repeated across the entire distribution of DCT values. In some cases, this periodic pattern is as simple as a populated bin followed by an unpopulated bin (e.g., $q_1 = 2$ and $q_2 = 1$). In other cases, this periodic pattern is a bit more complex. For example, with $q_1 = 5$ and $q_2 = 2$ the anomalous pattern is one populated bin, followed by two unpopulated bins, followed by one populated bin, followed by one unpopulated bin. This pattern of five then repeats. Regardless of the specific pattern, the detection of double compression is based on the presence of a periodic pattern in the distribution of DCT values. Because the double compression artifacts are so distinct, it is relatively straightforward to detect their presence [9, 11].

Cloning

In 2008, a photo of Iran's provocative missile test appeared on the front page of newspapers around the world. It was quickly revealed, however, that the photo of four airborne missiles had been doctored. To conceal the launch failure of one of the missiles, the image of a successful missile had been copied and pasted over the failed missile. This cloning was easily detected because of the suspicious similarity in the billowing dust clouds. Although the cloning in this image was detectable by eye, carefully executed clones can be visually imperceptible.

Detecting a clone involves two steps: the identification of potential matches and the verification of a match. The verification step is straightforward: If two regions are clones, their pixel values will be highly correlated. In contrast to the simplicity of verifying matches, the problem of identifying potential matches can easily lead to a combinatorial explosion. To see why this problem explodes, let's assume that we are searching a 1000×1000 pixel image for a cloned region of known size and shape. An image of this size contains $1,000,000$ pixels and yields 500 billion region pairs that could be potential matches. (Ignoring overlapping regions and the edges of the image, there are $1,000,000$ choose 2, equal to $(1,000,000 \times 999,998)/2 = 500,000,000,000$ possible pairings of pixels, each of which may be the center of a pair of cloned regions.) We do not typically know the cloned region's size and shape, nor do we know whether the cloned region has been resized or rotated before being added back into the image. Clearly, an exhaustive search of the potential matches in an image is computationally intractable.

With an easy verification step but a prohibitive search space, the task of detecting cloning reduces to finding an efficient and accurate way to search an image for two nearly identical regions. One particularly effective algorithm [10] consists of three basic steps: (1) the identification of distinctive features in the image; (2) the extraction of a compact descriptor of each feature; and (3) the search for two clusters of features that have pairwise similar descriptors and that are related by a translation (and optionally a scaling and rotation).

The first step is to identify salient features in the image. These salient features should be sufficiently distinctive that they would have relatively few matches in the image. One such approach is the *Harris detector* which assigns a value to each pixel that is proportional to the amount of spatial variation in the pixels that surround it. Once the salient features in the image have been identified, the next step is to describe them in a compact way to allow for efficient matching. This description should retain the distinctiveness of the feature, but it should also be unaffected by common image transformations such as scaling, rotation, brightness and contrast adjustment, and compression. The *scale-invariant feature transform (SIFT)* or *histogram of oriented gradients (HOG)* descriptor offers a reasonable compromise between specificity and tolerance to transformations.

The third step in detecting potential matches is to search for two sets of similar features that are related by a translation (and optionally a scaling and rotation). Isolating these corresponding sets in a sea of features requires model-fitting in the presence of outliers. The *random sample consensus (RANSAC)* algorithm can be used to simultaneously extract the matching features and estimate the relationship between them.

The output of this clone detection algorithm will typically require a human analyst to review the purported matches to determine if they are semantically meaningful.

Recent Trends

There are many forensic techniques that can detect image tampering, and new techniques are constantly being developed. Each of these techniques, however, can be circumvented. A determined and skilled forger can, for example, build a custom JPEG coder that exactly mimics a camera's file packaging, carefully remove any DCT artifacts that arise from multiple compressions, reinsert the expected color filter array interpolation correlations, analyze all shadows and reflections to ensure that they are physically consistent, and carefully work through all of the other traces used by dozens of different forensic techniques. The number, variety, and complexity of these techniques, however, make it difficult and time-consuming (but not impossible) for the average forger to create a fake that is completely indistinguishable from an authentic image.

Despite significant advances over the past two decades in the field of image forensics, much work remains to be done. While many forensic techniques are highly effective, many of them also require manual and careful oversight to apply them. This means that they require some expertise to apply and that they are not yet ready to be deployed at Internet-scale (to the tune of millions or billions of uploads a day).

Authentication will also be made more difficult by rapid advances in machine learning that have made it easier than ever to create sophisticated and compelling fakes [12–14]. These technologies have removed many of the time and skill barriers previously required to create high-quality fakes. Not only can these automatic tools be used to create compelling fakes, they can be turned against our forensic techniques in the form of generative adversarial networks (GANs) that modify fake content to bypass forensic detection [15]. There is little doubt that this arms race will continue into the foreseeable future.

References

1. Farid H (2016) Photo forensics. MIT Press, Cambridge
2. O'Brien J, Farid H (2012) Exposing photo manipulation with inconsistent reflections. ACM Trans Graph 31(1):4:1–4:11
3. Kee E, O'Brien J, Farid H (2014) Exposing photo manipulation from shading and shadows. ACM Trans Graph 33(5):165:1–165:21
4. Popescu A, Farid H (2010) Exposing digital forgeries in color filter array interpolated images. IEEE Trans Signal Process 53(10):3948–3959
5. Fridrich J, Lukas J, Goljan M (2006) Digital camera identification from sensor noise. IEEE Trans Inf Secur Forensic 1(2):205–214
6. Kirchner M (2010) Efficient estimation of CFA pattern configuration in digital camera images. In: Media forensics and security II. International Society for Optics and Photonics, vol 7541, p 754111
7. Kee E, Johnson M, Farid H (2011) Digital image authentication from JPEG headers. IEEE Trans Inf Forensics Secur 7(3):1066–1075
8. Gloe T (2012) Forensic analysis of ordered data structures on the example of JPEG files. In: IEEE workshop on information forensics and security
9. Popescu A, Farid H (2004) Statistical tools for digital forensics. In: International workshop on information hiding
10. Pan X, Lyu S (2010) Region duplication detection using image feature matching. IEEE Trans Inf Forensics Secur 5(4):857–867
11. Bianchi T, Piva A (2011) Analysis of non-aligned double JPEG artifacts for the localization of image forgeries. In: IEEE workshop on information forensics and security
12. Liu M-Y, Breuel T, Kautz J (2018) Unsupervised image-to-image translation networks. In: Neural information processing systems, pp 700–708

13. Suwajanakorn S, Seitz SM, Kemelmacher-Shlizerman I (2017) Synthesizing Obama: learning lip sync from audio. ACM Trans Graph 36(4):95
14. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE international conference on computer vision
15. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Wade-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680