Photo Fakery and Forensics

Hany Farid
Dartmouth College

Photographs can no longer be trusted. From the tabloid magazines to the fashion industry, mainstream media outlets, political campaigns, and the photo hoaxes that land in our email in-boxes, doctored photographs are appearing with a growing frequency and sophistication. I will briefly describe the impact of all of this photographic tampering and recent technological advances that have the potential to return some trust to photographs. Specifically, I will describe a representative sample of image forensics techniques for detecting inconsistencies in lighting, pixel correlations, and compression artifacts.

Contents

1 Photo Fakery

History is riddled with the remnants of photographic fakery. Stalin, Mao, Hitler, Mussolni, Castro, and Brezhnev each had photographs manipulated in an attempt to alter history. Cumbersome and time-consuming darkroom techniques were required to alter history on behalf of Stalin and others. Today, powerful and low-cost digital technology has made it far easier for nearly anyone to alter digital images. And the resulting fakes are often very difficult to detect. This photographic fakery is having a significant impact in many different areas.

Media: For the past decade, Adnan Hajj has produced striking war photographs from the on-going struggle in the Middle East. On August 7^{th} of 2006, the Reuters news agency published one of Hajj's photographs showing the remnants of an Israeli bombing of a Lebanese town. In the week that followed, hundreds of bloggers and nearly every major news organization reported that the photograph had been doctored with the addition of more smoke. The general consensus was one of outrage and anger – Hajj was accused of doctoring the image to exaggerate the impact of the Israeli shelling. An embarrassed Reuters quickly retracted the photograph and removed from its archives nearly 1,000 photographs contributed by Hajj. The case of Hajj is, of course, by no means unique. In 2003 Brian Walski, a veteran photographer of numerous wars doctored a photograph that appeared

on the cover of the Los Angeles Times. After discovering the fake, the outraged editors of the LA Times fired Walski. The news magazines Time and Newsweek have each been rocked by scandal after it was revealed that photographs appearing on their covers had been doctored. And, in the past few years, countless news organizations around the world have been shaken by similar experiences.

Science: Those in the media are not alone in succumbing to the temptation to manipulate photographs. In 2004, Professor Hwang Woo-Suk and colleagues published what appeared to be groundbreaking advances in stem cell research. This paper appeared in one of the most prestigious scientific journals, *Science*. Evidence slowly emerged that these results were manipulated and/or fabricated. After months of controversy, Hwang retracted the *Science* paper and resigned his position at the University. An independent panel investigating the accusations of fraud found, in part, that at least nine of the eleven customized stem cell colonies that Hwang had claimed to have made were fakes. Much of the evidence for those nine colonies, the panel said, involved doctored photographs of two other, authentic, colonies. While this case garnered international coverage and outrage, it is by no means unique. In an increasingly competitive field, scientists are succumbing to the temptation to exaggerate or fabricate their results. Mike Rossner, the managing editor of the *Journal of Cell Biology* estimates that as many as 20% of accepted manuscripts to his journal contain at least one figure that has to be remade because of inappropriate image manipulation [35].

Law: The child pornography charges against its Police Chief shocked the small town of Wapakoneta, Ohio. At his trial, the defendant's lawyer argued that if the State could not prove that the seized images were real, then the defendant was within his rights in possessing the images. In 1996 the Child Pornography Prevention Act (CPPA) extended the existing federal criminal laws against child pornography to include certain types of "virtual porn". In 2002 the United States Supreme Court found that portions of the CPPA, being overly broad and restrictive, violated First Amendment rights. The Court ruled that "virtual" or "computer generated" images depicting a fictitious minor are constitutionally protected. The burden of proof in this case, and countless others, shifted to the State who had to prove that the images were real and not computer generated. Given the sophistication of computer generated images, several state and federal rulings have further found that juries should not be asked to make the determination between real or virtual. And at least one federal judge questioned the ability of even expert witnesses to make this determination. This example highlights the general complexities that exist at the intersection of digital technology and the law.

Politics: "Fonda Speaks to Vietnam Veterans At Anti-War Rally" read the headline with an accompanying photograph purportedly showing Senator John Kerry sharing a stage with the then controversial Jane Fonda. The faux article was also a fake – a composite of two separate and unrelated photographs. And just days after being selected as a running mate to U.S. presidential hopeful John McCain, doctored images of a bikini clad and gun-toting Sarah Palin were widely distributed on the Internet. The pairing of one's political enemies with controversial figures is certainly not new. It is believed that a doctored photograph contributed to Senator Millard Tydings' electoral defeat in 1950. The photo of Tydings conversing with Earl Browder, a leader of the American Communist party, was meant to suggest that Tydings had communist sympathies. Recent political ads have seen a startling number of doctored photographs pitting candidates in a flattering or damaging light.

National Security: With tensions mounting between the United States and Iran, the Iranian Government announced the successful testing of ballistic missiles. As evidence, the government released a

photograph showing the simultaneous launch of four missiles. Shortly after its world-wide publication, it was revealed that the image had been doctored. In reality, only three missiles had launched, while the fourth missile, which failed to launch, was digitally inserted. This example highlighted the importance of image authentication, and showed the potential implications of photo tampering on a geo-political stage.

While historically they may have been the exception, doctored photographs today are increasingly impacting nearly every aspect of our society. While the technology to distort and manipulate digital media is developing at break-neck speeds, the technology to detect such alterations is lagging behind. To this end, I will describe some recent innovations for detecting photo tampering that have the potential to return some trust to photographs.

2 Photo Forensics

Digital watermarking has been proposed as a means by which an image can be authenticated (see, for example, [27, 8] for general surveys). The drawback of this approach is that a watermark must be inserted at the time of recording, which would limit this approach to specially equipped digital cameras. This method also relies on the assumption that the digital watermark cannot be easily removed and reinserted — it is not yet clear whether this is a reasonable assumption (e.g., [9]). In contrast to these approaches, we have proposed techniques for detecting tampering in digital images that work in the absence of any digital watermark or signature.

Given the variety of images and forms of tampering, the forensic analysis of images benefits from a variety of tools that can detect various forms of tampering. Over the past eight years my students, colleagues and I have developed a suite of computational and mathematical techniques for detecting tampering in digital images. Our approach in developing each forensic tool is to first understand how a specific form of tampering disturbs certain statistical or geometric properties of an image, and then to develop a computational techniques to detect these perturbations. Within this framework, I describe five of such techniques [1].

Specifically, I will describe three techniques for detecting inconsistencies in lighting, the first two of which estimate the direction to a light source, and the third of which estimates a more complex lighting environment consisting of multiple light sources. The fourth technique exploits pixel correlations that are introduced into an image as a result of the specific design of digital camera sensors. And the final technique leverages the artifacts introduced by the JPEG compression algorithm. These techniques were chosen as a representative sample of a larger body of image forensic techniques.

2.1 Lighting Direction (2-D)

Consider the creation of a forgery showing two movie stars, rumored to be romantically involved, walking down a sunset beach. Such an image might be created by splicing together individual images of each movie star. In so doing, it is often difficult to exactly match the lighting effects due to directional lighting (e.g., the sun on a clear day). Differences in lighting can, therefore, be a telltale sign of digital tampering. To the extent that the direction of the light source can be estimated for different objects/people in an image, inconsistencies in the lighting direction can be used as evidence of digital tampering.

[1] Portions of this chapter have appeared in [24, 26, 25, 38].

The standard approaches for estimating light source direction begin by making some simplifying assumptions: (1) the surface of interest is Lambertian (the surface reflects light isotropically); (2) the surface has a constant reflectance value; (3) the surface is illuminated by a point light source infinitely far away; and (4) the angle between the surface normal and the light direction is in the range 0° to 90°. Under these assumptions, the image intensity can be expressed as:

$$I(x,y) \quad = \quad R(\vec{N}(x,y) \cdot \vec{L}) + A, \tag{1}$$

where R is the constant reflectance value, $\vec{L}$ is a 3-vector pointing in the direction of the light source, $\vec{N}(x,y)$ is a 3-vector representing the surface normal at the point (x,y), and A is a constant ambient light term [14], Figure 1 (left). If we are only interested in the direction of the light source, then the reflectance term, R, can be considered to have unit-value, understanding that the estimation of $\vec{L}$ will only be within an unknown scale factor. The resulting linear equation provides a single constraint in four unknowns, the three components of $\vec{L}$ and the ambient term A.

With at least four points with the same reflectance, R, and distinct surface normals, $\vec{N}$, the light source direction and ambient term can be solved for using standard least-squares estimation. To begin, a quadratic error function, embodying the imaging model of Equation (1), is given by:

$$\begin{aligned} E(\vec{L},A) \quad &= \quad \left|\left| M \begin{pmatrix} L_x \\ L_y \\ L_z \\ A \end{pmatrix} - \begin{pmatrix} I(x_1,y_1) \\ I(x_2,y_2) \\ \vdots \\ I(x_p,y_p) \end{pmatrix} \right|\right|^2 \\ &= \quad \left|\left| M\vec{v} - \vec{b} \right|\right|^2, \end{aligned} \tag{2}$$

where $||\cdot||$ denotes vector norm, L_x, L_y, and L_z denote the components of the light source direction $\vec{L}$, and

$$M = \begin{pmatrix} N_x(x_1,y_1) & N_y(x_1,y_1) & N_z(x_1,y_1) & 1 \\ N_x(x_2,y_2) & N_y(x_2,y_2) & N_z(x_2,y_2) & 1 \\ \vdots & \vdots & \vdots & \vdots \\ N_x(x_p,y_p) & N_y(x_p,y_p) & N_z(x_p,y_p) & 1 \end{pmatrix}, \tag{3}$$

where $N_x(x_i,y_i)$, $N_y(x_i,y_i)$, and $N_z(x_i,y_i)$ denote the components of the surface normal $\vec{N}$ at image coordinate (x_i,y_i). The quadratic error function above is minimized by differentiating with respect to the unknown, $\vec{v}$, setting the result equal to zero, and solving for $\vec{v}$ to yield the least-squares estimate:

$$\vec{v} \quad = \quad (M^T M)^{-1} M^T \vec{b}. \tag{4}$$

Note that this solution requires knowledge of 3-D surface normals from at least four distinct points ($p \geq 4$) on a surface with the same reflectance. With only a single image and no objects of known geometry in the scene, it is unlikely that this will be possible. Most approaches to overcome this problem rely on acquiring multiple images [37] or placing an object of known geometry in the scene (e.g., a sphere) [4]. For forensic applications, these solutions are not practical.

In [32], the authors suggest a clever solution for estimating two components of the light source direction (L_x and L_y) from only a single image. While their approach clearly provides less information

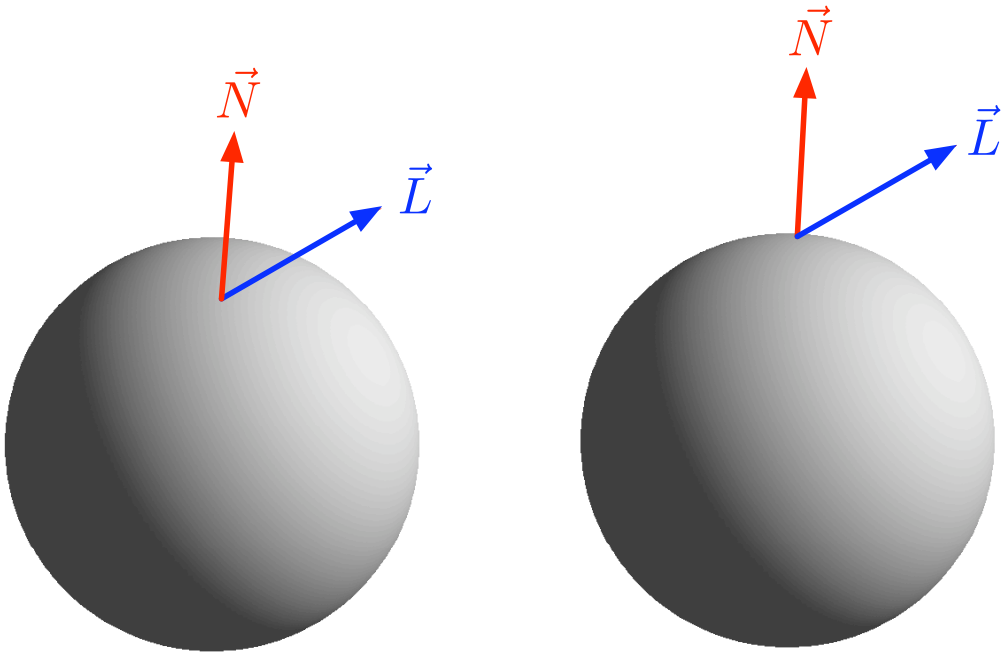


Figure 1: Schematic diagram of the imaging geometry for: 3-D surface normals (left), and 2-D surface normals (right). In the 2-D case, the z-component of the surface normal ($\vec{N}$) is zero.

regarding the light source direction, it does make the problem tractable from a single image. The authors note that at the occluding boundary of a surface, the z-component of the surface normal is zero, $N_z = 0$. In addition, the x- and y-components of the surface normal, N_x and N_y, can be estimated directly from the image, Figure 1 (right).

With this assumption, the error function of Equation (2) takes the form:

$$\begin{aligned} E(\vec{L}, A) &= \left\| M \begin{pmatrix} L_x \\ L_y \\ A \end{pmatrix} - \begin{pmatrix} I(x_1, y_1) \\ I(x_2, y_2) \\ \vdots \\ I(x_p, y_p) \end{pmatrix} \right\|^2 \\ &= \left\| M\vec{v} - \vec{b} \right\|^2, \end{aligned} \tag{5}$$

where,

$$M = \begin{pmatrix} N_x(x_1, y_1) & N_y(x_1, y_1) & 1 \\ N_x(x_2, y_2) & N_y(x_2, y_2) & 1 \\ \vdots & \vdots & \vdots \\ N_x(x_p, y_p) & N_y(x_p, y_p) & 1 \end{pmatrix}. \tag{6}$$

This error function is minimized, as before, using standard least-squares to yield the same solution as in Equation (4), but with the matrix M taking the form given in Equation (6). In this case, the solution requires knowledge of 2-D surface normals from at least three distinct points ($p \geq 3$) on a surface with the same reflectance.

The intensity, $I(x_i, y_i)$, at a boundary point, (x_i, y_i), cannot be directly measured from the image as the surface is occluded. The authors in [32] note, however, that the intensity can be extrapolated by considering the intensity profile along a ray coincident to the 2-D surface normal. They also found that simply using the intensity close to the border of the surface is often sufficient.

We extend this basic formulation in two ways. First, we estimate the two-dimensional light source direction from local patches along an object's boundary (as opposed to along extended boundaries as in [32]). This is done to relax the assumption that the reflectance along the entire surface is constant. Then, a regularization (smoothness) term is introduced to better condition the final estimate of light source direction.

The constant reflectance assumption is relaxed by assuming that the reflectance for a local surface patch (as opposed to the entire surface) is constant. This requires us to estimate individual light source directions, $\vec{L}^i$, for each patch along a surface. Under the infinite light source assumption, the orientation of these estimates should not vary, but their magnitude may (recall that the estimate of the light source is only within a scale factor, which depends on the reflectance value R, Equation (1)).

Consider a surface partitioned into n patches, and, for notational simplicity, assume that each patch contains p points. The new error function to be minimized is constructed by packing together, for each patch, the 2-D version of the constraint of Equation (1):

$$\begin{aligned} E_1(\vec{L}^1, \ldots, \vec{L}^n, A) &= \left\| M \begin{pmatrix} L_x^1 \\ L_y^1 \\ \vdots \\ L_x^n \\ L_y^n \\ A \end{pmatrix} - \begin{pmatrix} I(x_1^1, y_1^1) \\ \vdots \\ I(x_p^1, y_p^1) \\ \vdots \\ I(x_1^n, y_1^n) \\ \vdots \\ I(x_p^n, y_p^n) \end{pmatrix} \right\|^2 \\ &= \left|\left| M\vec{v} - \vec{b} \right|\right|^2, \end{aligned} \tag{7}$$

where,

$$M = \begin{pmatrix} N_x(x_1^1, y_1^1) & N_y(x_1^1, y_1^1) & & 0 & 0 & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ N_x(x_p^1, y_p^1) & N_y(x_p^1, y_p^1) & & 0 & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & & N_x(x_1^n, y_1^n) & N_y(x_1^n, y_1^n) & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & & N_x(x_p^n, y_p^n) & N_y(x_p^n, y_p^n) & 1 \end{pmatrix}. \tag{8}$$

The above quadratic error function is minimized, as before, using least-squares with the solution taking on the same form as in Equation (4). In this case, the solution provides n estimates of the 2-D light directions, $\vec{L}^1$, ..., $\vec{L}^n$, and an ambient term A. Note that while individual light source directions are estimated for each surface patch, a single ambient term is assumed.

While the local estimation of light source directions allows for the relaxation of the constant reflectance assumption, it could potentially yield less stable results. Note that under the assumption

of an infinite point light source, the orientation of the n light directions should be equal. With the additional assumption that the change in reflectance from patch to patch is relatively small (i.e., the change in the magnitude of neighboring $\vec{L}^i$'s is small), we can condition the individual estimates with the following regularization term:

$$E_2(\vec{L}^1, \ldots, \vec{L}^n) \quad = \quad \sum_{i=2}^{n} \left|\left| \vec{L}^i - \vec{L}^{i-1} \right|\right|^2. \tag{9}$$

This additional error term penalizes neighboring estimates that are different from one another. The quadratic error function $E_1(\cdot)$, Equation (7), is conditioned by combining it with the regularization term $E_2(\cdot)$, scaled by a factor λ, to yield the final error function:

$$E(\vec{L}^1, \ldots, \vec{L}^n, A) \quad = \quad E_1(\vec{L}^1, \ldots, \vec{L}^n, A) + \lambda E_2(\vec{L}^1, \ldots, \vec{L}^n). \tag{10}$$

This combined error function can still be minimized using least-squares minimization. The error function $E_2(\cdot)$ is first written in a more compact and convenient form as:

$$E_2(\vec{v}) \quad = \quad ||C\vec{v}||^2, \tag{11}$$

where the $2n - 2 \times 2n + 1$ matrix C is given by:

$$C = \begin{pmatrix} -1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & \ddots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 \end{pmatrix}, \tag{12}$$

with $\vec{v} = \begin{pmatrix} L_x^1 & L_y^1 & L_x^2 & L_y^2 & \ldots & L_x^n & L_y^n & A \end{pmatrix}^T$. The error function of Equation (10) then takes the form:

$$E(\vec{v}) \quad = \quad ||M\vec{v} - \vec{b}||^2 + \lambda ||C\vec{v}||^2. \tag{13}$$

Differentiating this error function yields:

$$\begin{aligned} E'(\vec{v}) \quad &= \quad 2M^T M\vec{v} - 2M^T\vec{b} + 2\lambda C^T C\vec{v} \\ &= \quad 2(M^T M + \lambda C^T C)\vec{v} - 2M^T\vec{b}. \end{aligned} \tag{14}$$

Setting this result equal to zero and solving for $\vec{v}$ yields the least-squares estimate:

$$\vec{v} \quad = \quad (M^T M + \lambda C^T C)^+ M^T\vec{b}, \tag{15}$$

where $+$ denotes pseudo-inverse. The final light direction estimate is computed by averaging the n resulting light direction estimates, $\vec{L}^1, \ldots, \vec{L}^n$.

The light direction estimation requires the localization of an occluding boundary. These boundaries are extracted by manually selecting points in the image along an occluding boundary. This rough estimate of the position of the boundary is used to define its spatial extent. The boundary is then partitioned into approximately eight small patches. Three points near the occluding boundary

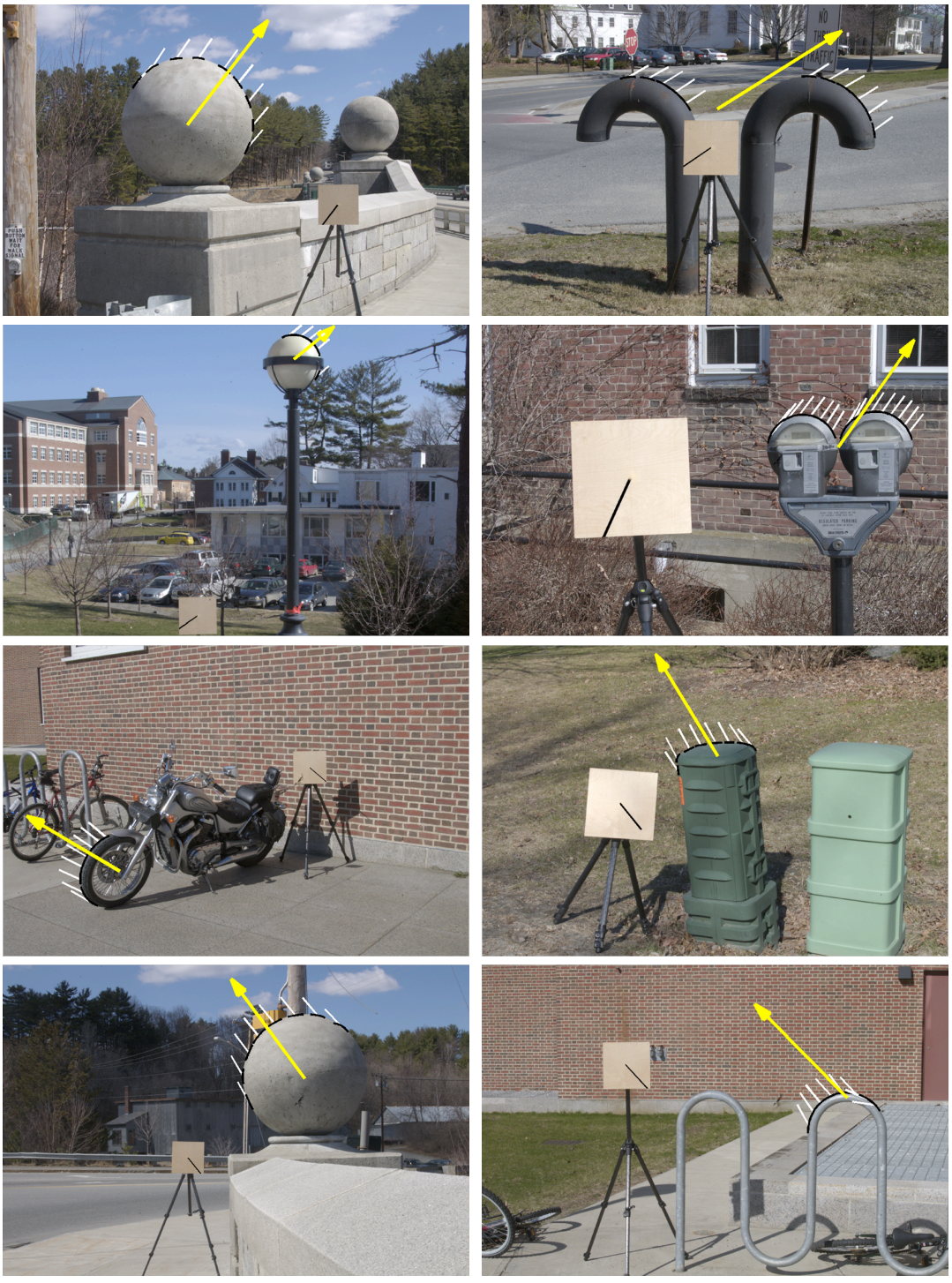


Figure 2: Shown are eight images with the extracted occluding boundaries (black), individual light source estimates (white), and the final average light source direction (large arrow). In each image, the cast shadow on the calibration target indicates the direction to the illuminating sun, and has been darkened to enhance visibility.

are manually selected for each patch, and fit with a quadratic curve. The surface normals along each patch are then estimated analytically from the resulting quadratic fit.

Shown in Figure 2 are eight images of objects illuminated by the sun on a clear day. In order to determine the accuracy of our approach, a calibration target, consisting of a flat surface with a rod extending from the center, was placed in each scene. The target was approximately parallel to the image plane, so that the shadow cast by the rod indicated the direction of the sun. Errors in our estimated light source direction are given relative to this orientation. The average estimation error is 4.8° with a minimum and maximum error of 0.6° and 10.9°. The image returning the largest error is the parking meters. There are probably at least three reasons for this larger error, and for errors in general. The first is that the metallic surface violates the Lambertian assumption. The second is that the paint on the meter is worn in several spots causing the reflectance to vary, at times, significantly from patch to patch. And the third is that we did not calibrate the camera so as to remove luminance non-linearities (e.g., gamma correction) in the image.

The creation of a digital forgery often involves combining objects/people from separate images. In so doing, it is difficult to exactly match the lighting effects due to directional lighting (e.g., the sun on a clear day). At least one reason for this is that such a manipulation may require the creation or removal of shadows and lighting gradients. And while large inconsistencies in lighting direction may be fairly obvious, there is evidence from the human psychophysics literature that we are surprisingly insensitive to differences in lighting across an image [43, 34]. To the extent that the direction of the light source can be estimated for different objects/people in an image, inconsistencies in lighting can be used as evidence of digital tampering.

2.2 Lighting Direction (3-D)

In the previous section, we described how to estimate the light source direction in 2-D. While this approach has the benefit of being applicable to arbitrary objects, it has the drawback that it can only determine the direction to the light source within one degree of ambiguity. Next we describe how the full 3-D light source direction can be estimated by leveraging a 3-D model of the human eye. Specifically, we describe how to estimate the 3-D direction to a light source from specular highlights on the eyes.

The position of a specular highlight is determined by the relative positions of the light source, the reflective surface and the viewer (or camera). In Figure 3, for example, is a diagram showing the creation of a specular highlight on an eye. In this diagram, the three vectors $\vec{L}$, $\vec{N}$ and $\vec{R}$ correspond to the direction to the light, the surface normal at the point at which the highlight is formed, and the direction in which the highlight will be seen. For a perfect reflector, the highlight is seen only when the view direction $\vec{V} = \vec{R}$. For an imperfect reflector, a specular highlight can be seen for viewing directions $\vec{V}$ near $\vec{R}$, with the strongest highlight seen when $\vec{V} = \vec{R}$.

An algebraic relationship between the vectors $\vec{L}$, $\vec{N}$, and $\vec{V}$ is first derived. We then show how the 3-D vectors $\vec{N}$ and $\vec{V}$ can be estimated from a single image, from which the direction to the light source $\vec{L}$ is determined.

The law of reflection states that a light ray reflects off of a surface at an angle of reflection θ_r equal to the angle of incidence θ_i, where these angles are measured with respect to the surface normal $\vec{N}$, Figure 3. Assuming unit-length vectors, the direction of the reflected ray $\vec{R}$ can be described in terms

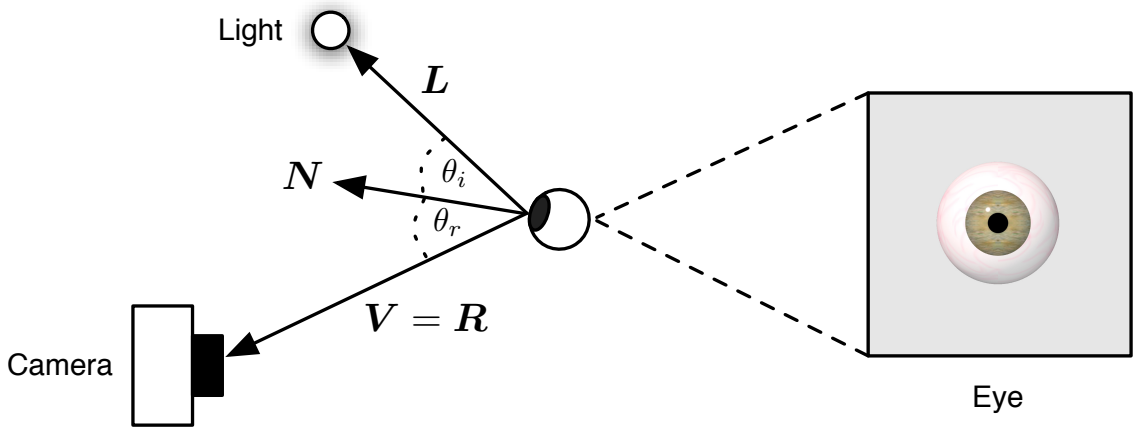


Figure 3: The formation of a specular highlight on an eye (small white dot on the iris). The position of the highlight is determined by the surface normal $\vec{N}$ and the relative directions to the light source $\vec{L}$ and viewer $\vec{V}$.

of the light direction $\vec{L}$ and the surface normal $\vec{N}$:

$$\begin{aligned} \vec{R} &= \vec{L} + 2(\cos(\theta_i)\vec{N} - \vec{L}) \\ &= 2\cos(\theta_i)\vec{N} - \vec{L}. \end{aligned} \tag{16}$$

By assuming a perfect reflector ($\vec{V} = \vec{R}$), the above constraint yields:

$$\begin{aligned} \vec{L} &= 2\cos(\theta_i)\vec{N} - \vec{V} \\ &= 2\left(\vec{V}^T\vec{N}\right)\vec{N} - \vec{V}. \end{aligned} \tag{17}$$

The light direction $\vec{L}$ can therefore be estimated from the surface normal $\vec{N}$ and view direction $\vec{V}$ at a specular highlight. Note that the light direction is specified with respect to the eye, and not the camera. In practice, all of these vectors will be placed in a common coordinate system, allowing us to compare light directions across the image.

In order to estimate the surface normal $\vec{N}$ and view direction $\vec{V}$ in a common coordinate system, we first need to estimate the projective transform that describes the transformation from world to image coordinates. With only a single image, this calibration is generally an under-constrained problem. In our case, however, the known geometry of the eye can be exploited to estimate this required transform. Throughout, upper-case symbols will denote world coordinates and lower-case will denote camera/image coordinates.

The limbus, the boundary between the sclera (white part of the eye) and the iris (colored part of the eye), can be well modeled as a circle [33]. The image of the limbus, however, will be an ellipse except when the eye is directly facing the camera. Intuitively, the distortion of the ellipse away from a circle will be related to the pose and position of the eye relative to the camera. We therefore seek the transform that aligns the image of the limbus to a circle.

In general, a projective transform that maps 3-D world coordinates to 2-D image coordinates can be represented, in homogeneous coordinates, as a 3×4 matrix. We assume that points on a limbus are coplanar, and define the world coordinate system such that the limbus lies in the $Z = 0$ plane. With

this assumption, the projective transformation reduces to a 3×3 planar projective transform [19], where the world points $\vec{X}$ and image points $\vec{x}$ are represented by 2-D homogeneous vectors.

Points on the limbus in our world coordinate system satisfy the following implicit equation of a circle:

$$f(\vec{X}; \vec{\alpha}) = (X_1 - C_1)^2 + (X_2 - C_2)^2 - r^2 = 0, \tag{18}$$

where $\vec{\alpha} = \begin{pmatrix} C_1 & C_2 & r \end{pmatrix}^T$ denotes the circle center and radius.

Consider a collection of points, $\vec{X}_i$, $i = 1, \ldots, m$, each of which satisfy Equation (18). Under an ideal pinhole camera model, the world point $\vec{X}_i$ maps to the image point $\vec{x}_i$ as follows:

$$\vec{x}_i = H\vec{X}_i, \tag{19}$$

where H is a 3×3 projective transform matrix.

The estimation of H can be formulated in an orthogonal distance fitting framework. Let $E(\cdot)$ be an error function on the parameter vector $\vec{\alpha}$ and the unknown projective transform H:

$$E(\vec{\alpha}, H) = \sum_{i=1}^{m} \min_{\vec{\hat{X}}} \left\| \vec{x}_i - H\vec{\hat{X}} \right\|^2, \tag{20}$$

where $\vec{\hat{X}}$ is on the circle parametrized by $\vec{\alpha}$. The error embodies the sum of the squared errors between the data, $\vec{x}_i$, and the closest point on the model, $\vec{\hat{X}}$. This error function is minimized using non-linear least squares via the Levenberg-Marquardt iteration [41].

Once estimated, the projective transform H can be decomposed in terms of intrinsic and extrinsic camera parameters [19]. The intrinsic parameters consist of the camera focal length, camera center, skew and aspect ratio. For simplicity, we will assume that the camera center is the image center, that the skew is 0 and the aspect ratio is 1, leaving only the focal length f. The extrinsic parameters consist of a rotation matrix R and translation vector $\vec{t}$ that define the transformation between the world and camera coordinate systems. Since the world points lie on a single plane, the projective transform can be decomposed in terms of the intrinsic and extrinsic parameters as:

$$H = \lambda K \begin{pmatrix} \vec{r_1} & \vec{r_2} & \vec{t} \end{pmatrix}, \tag{21}$$

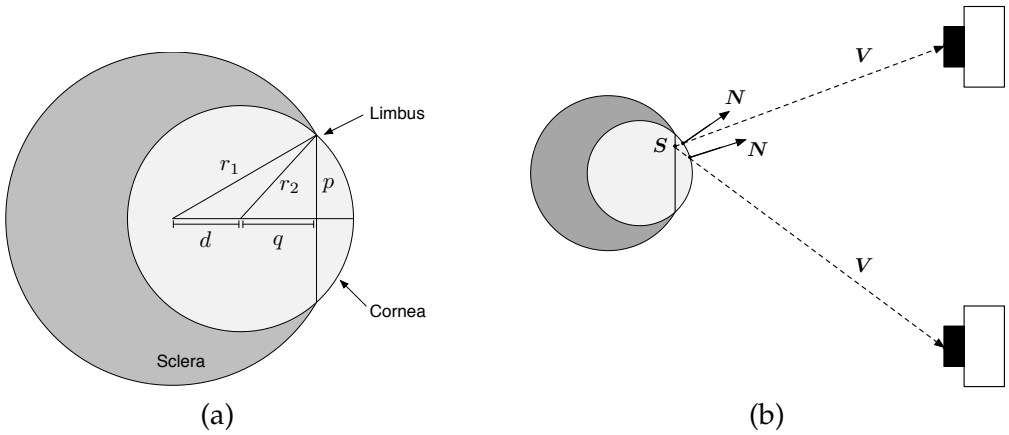
where the 3×3 intrinsic matrix K is:

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{22}$$

λ is a scale factor, the column vectors $\vec{r_1}$ and $\vec{r_2}$ are the first two columns of the rotation matrix R, and $\vec{t}$ is the translation vector.

With a known focal length f, and hence a known matrix K, the world to camera coordinate transform $\hat{H}$ can be estimated directly:

$$\begin{aligned} \frac{1}{\lambda} K^{-1} H &= \begin{pmatrix} \vec{r_1} & \vec{r_2} & \vec{t} \end{pmatrix} \\ \hat{H} &= \begin{pmatrix} \vec{r_1} & \vec{r_2} & \vec{t} \end{pmatrix}, \end{aligned} \tag{23}$$



(a) (b)

Figure 4: (a) A side view of a 3-D model of the human eye. The larger sphere represents the sclera and the smaller sphere represents the cornea. The limbus is defined by the intersection of the two spheres. (b) The surface normal at a point $\vec{S}$ in the plane of the limbus depends on the view direction $\vec{V}$.

where the scale factor λ is chosen so that $\vec{r_1}$ and $\vec{r_2}$ are unit vectors. The complete rotation matrix is given by:

$$R = \begin{pmatrix} \vec{r_1} & \vec{r_2} & \vec{r_1} \times \vec{r_2} \end{pmatrix}, \tag{24}$$

where $\times$ denotes cross product. If the focal length is unknown, it can be directly estimated as described in [26].

Recall that the minimization of Equation (20) yields both the transform H and the circle parameters $\vec{\alpha}$ for the limbus. The unit vector from the center of the limbus to the origin of the camera coordinate system is the view direction, $\vec{v}$. Let $\vec{X_c} = \begin{pmatrix} C_1 & C_2 & 1 \end{pmatrix}$ denote the estimated center of a limbus in world coordinates. In the camera coordinate system, this point is given by:

$$\vec{x_c} = \hat{H}\vec{X_c}. \tag{25}$$

The view direction, as a unit vector, in the camera coordinate system is then given by:

$$\vec{v} = -\frac{\vec{x_c}}{\|\vec{x_c}\|}, \tag{26}$$

where the negative sign reverses the vector so that it points from the eye to the camera.

The 3-D surface normal $\vec{N}$ at a specular highlight is estimated from a 3-D model of the human eye [30]. The model consists of a pair of spheres as illustrated in Figure 4(a). The larger sphere, with radius $r_1 = 11.5$ mm, represents the sclera and the smaller sphere, with radius $r_2 = 7.8$ mm, represents the cornea. The centers of the spheres are displaced by a distance $d = 4.7$ mm. The limbus, a circle with radius $p = 5.8$ mm, is defined by the intersection of the two spheres. The distance between the center of the smaller sphere and the plane containing the limbus is $q = 5.25$ mm. These measurements vary slightly among adults, and the radii of the spheres are approximately 0.1 mm smaller for female eyes [21, 30].

Consider a specular highlight in world coordinates at location $\vec{S} = \begin{pmatrix} S_x & S_y \end{pmatrix}$, measured with respect to the center of the limbus. The surface normal at $\vec{S}$ depends on the view direction $\vec{V}$. In

Figure 4(b) is a schematic showing this relationship for two different positions of the camera. The surface normal $\vec{N}$ is determined by intersecting the ray leaving $\vec{S}$, along the direction $\vec{V}$, with the edge of the sphere. This intersection can be computed by solving a quadratic system for k, the distance between $\vec{S}$ and the edge of the sphere,

$$\begin{aligned}(S_x + kV_x)^2 + (S_y + kV_y)^2 + (q + kV_z)^2 &= r_2^2 \\ k^2 + 2(S_xV_x + S_yV_y + qV_z)k + (S_x^2 + S_y^2 + q^2 - r_2^2) &= 0, \end{aligned} \tag{27}$$

where q and r_2 are specified by the 3-D model of the eye. The view direction $\vec{V} = \begin{pmatrix} V_x & V_y & V_z \end{pmatrix}$ in the world coordinate system is given by:

$$\vec{V} = R^{-1}\vec{v}, \tag{28}$$

where $\vec{v}$ is the view direction in camera coordinates and R is the estimated rotation between the world and camera coordinate systems. The surface normal $\vec{N}$ in the world coordinate system is then given by:

$$\vec{N} = \begin{pmatrix} S_x + kV_x \\ S_y + kV_y \\ q + kV_z \end{pmatrix}, \tag{29}$$

and in camera coordinates: $\vec{n} = R\vec{N}$.

Consider a specular highlight $\vec{x_s}$ specified in image coordinates and the estimated projective transform H from world to image coordinates. The inverse transform H^{-1} maps the coordinates of the specular highlight into world coordinates:

$$\vec{X_s} = H^{-1}\vec{x_s} \tag{30}$$

The center $\vec{C}$ and radius r of the limbus in the world coordinate system determine the coordinates of the specular highlight, $\vec{S}$, with respect to the model:

$$\vec{S} = \frac{p}{r}\left(\vec{X_s} - \vec{C}\right), \tag{31}$$

where p is specified by the 3-D model of the eye. The position of the specular highlight $\vec{S}$ is then used to determine the surface normal $\vec{N}$. Combined with the estimate of the view direction $\vec{V}$, the light source direction $\vec{L}$ can be estimated from Equation (17). In order to compare light source estimates in the image, the light source estimate is converted to camera coordinates: $\vec{l} = R\vec{L}$

To test the efficacy of this light estimation, synthetic images of eyes were rendered using the `pbrt` environment [36]. The shape of the eyes conformed to the 3-D model described above and the eyes were placed in one of 12 different locations. For each location, the eyes were rotated by a unique amount relative to the camera. The eyes were illuminated with two light sources: a fixed light directly in line with the camera, and a second light placed in one of four different positions. The twelve locations and four light directions gave rise to 48 images, Figure 5. Each image was rendered at a resolution of 1200×1600 pixels, with the cornea occupying less than 0.1% of the entire image. Shown in Figure 5 are several examples of the rendered eyes, along with a schematic of the imaging geometry. The limbus and position of the specular highlight(s) were automatically extracted from the rendered image. For each highlight, the projective transform H, the view direction $\vec{v}$ and surface

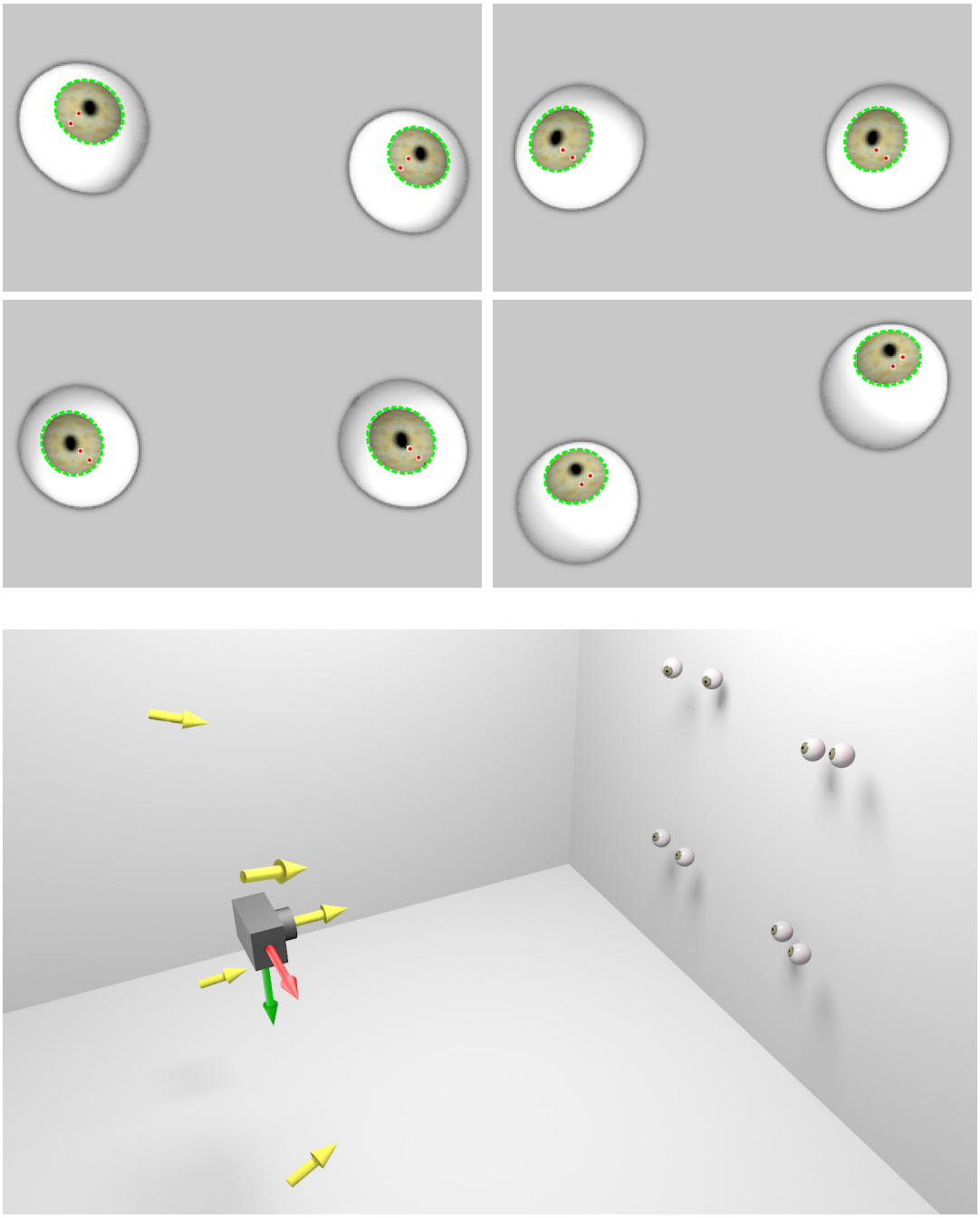


Figure 5: Synthetically generated eyes. Each of the upper panels corresponds to different positions and orientations of the eyes and locations of the light sources. The ellipse fit to each limbus is shown with a dashed line, and the small dots denote the positions of the specular highlights. Shown below is a schematic of the imaging geometry: the position of the lights, camera and a subset of the eye positions.

Figure 6: A subject at different locations and orientations relative to the camera and two light sources. Shown to the right are magnified views of the eyes. The ellipse fit to each limbus is shown with a dashed line and the small dots denote the positions of the specular highlights. See also Table 1.

normal $\vec{n}$ were estimated, from which the direction to the light source $\vec{l}$ was determined. The angular error between the estimated $\vec{l}$ and actual $\vec{l_0}$ light directions is computed as $\phi = \cos^{-1}\left(\vec{l}^T\vec{l_0}\right)$, where the vectors are normalized to be unit length. With a known focal length, the average angular error in estimating the light source direction was 2.8° with a standard deviation of 1.3° and a maximum error of 6.8°. With an unknown focal length, the average error was 2.8° with a standard deviation of 1.3° and a maximum error of 6.3°.

To further test the efficacy of our technique, we photographed a subject under controlled lighting. A camera and two lights were arranged along a wall, and the subject was positioned 250 cm in front of the camera and at the same elevation. The first light L_1 was positioned 130 cm to the left of and 60 cm above the camera. The second light L_2 was positioned 260 cm to the right and 80 cm above the camera. The subject was placed in five different locations and orientations relative to the camera and lights, Figure 6. A six mega-pixel Nikon D100 camera with a 35 mm lens was set to capture in the highest quality JPEG format. For each image, an ellipse was manually fit to the limbus of each eye. In these images, the limbus did not form a sharp boundary – the boundary spanned roughly 3 pixels. As such, we fit the ellipses to the better defined inner outline [23], Figure 6. The radius of each limbus was approximately 9 pixels, and the cornea occupied 0.004% of the entire image. Each specular highlight was localized by specifying a bounding rectangular area around each highlight and computing the centroid of the selection. The weighting function for the centroid computation

	left eye		right eye		left eye		right eye	
image	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
1	5.8	7.6	3.8	1.6	5.8	7.7	3.9	1.7
2	–	8.7	–	0.8	–	10.4	–	18.1
3	9.3	–	11.0	–	17.6	–	10.1	–
4	12.5	16.4	7.5	7.3	10.4	13.6	7.4	5.6
5	14.0	–	13.8	–	17.4	–	16.5	–

Table 1: Angular errors (degrees) in estimating the light direction for the images shown in Figure 6. On the left are the errors for a known focal length, and on the right are the errors for an unknown focal length. A '–' indicates that the specular highlight for that light was not visible on the cornea.

was chosen to be the squared (normalized) pixel intensity. The location to the light source(s) was estimated for each pair of eyes assuming a known and unknown focal length. The angular errors for each image are given in Table 1. Note that in some cases an estimate for one of the light sources was not possible when the highlight was not visible on the cornea. With a known focal length, the average angular error was 8.6°, and with an unknown focal length, the average angular error was 10.5°.

When creating a composite of two or more people it is often difficult to match the lighting conditions under which each person was originally photographed. Specular highlights that appear on the eye are a powerful cue as to the shape, color and location of the light source(s). Inconsistencies in these properties of the light can be used as evidence of tampering. We can measure the 3-D direction to a light source from the position of the highlight on the eye. While we have not specifically focused on it, the shape and color of a highlight are relatively easy to quantify and measure and should also prove helpful in exposing digital forgeries. Since specular highlights tend to be relatively small on the eye, it is possible to manipulate them to conceal traces of tampering. To do so, the shape, color and location of the highlight would have to be constructed so as to be globally consistent with the lighting in other parts of the image. Inconsistencies in this lighting may be detectable using the technique described in the previous section.

2.3 Lighting Environment

In the previous two sections, we have shown how to estimate the direction to a light source, and how inconsistencies in the illuminant direction can be used to detect tampering. This approach is appropriate when the lighting is dominated by a single light source, but is less appropriate in more complex lighting environments containing multiple light sources or non-directional lighting. Here we describe how to quantify such complex lighting environments and how to use inconsistencies in lighting to detect tampering.

The lighting of a scene can be complex—any number of lights can be placed in any number of positions, creating different lighting environments. In order to model such complex lighting, we assume that the lighting is distant and that surfaces in the scene are convex and Lambertian. To use this model in a forensic setting, we also assume that the surface reflectance is constant and that the camera response is linear.

Under the assumption of distant lighting, an arbitrary lighting environment can be expressed as a non-negative function on the sphere, $L(\vec{V})$, where $\vec{V}$ is a unit vector in Cartesian coordinates and

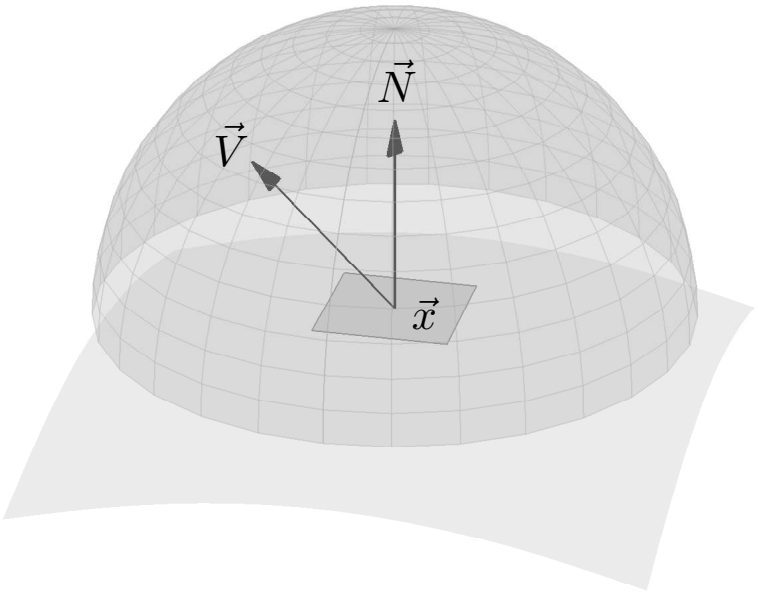


Figure 7: The irradiance (light received) at a point $\vec{x}$ is determined by integrating the amount of incoming light from all directions $\vec{V}$ in the hemisphere about the surface normal $\vec{N}$.

the value of $L(\vec{V})$ is the intensity of the incident light along direction $\vec{V}$, Figure 7. If the object being illuminated is convex, the irradiance (light received) at any point on the surface is due to only the lighting environment; i.e., there are no cast shadows or interreflections [39]. As a result, the irradiance, $E(\vec{N})$, can be parametrized by the unit length surface normal $\vec{N}$ and written as a convolution of the reflectance function of the surface, $R(\vec{V}, \vec{N})$, with the lighting environment $L(\vec{V})$:

$$E(\vec{N}) \quad = \quad \int_{\Omega} L(\vec{V})\, R(\vec{V}, \vec{N})\, d\Omega, \tag{32}$$

where Ω represents the surface of the sphere and $d\Omega$ is an area differential on the sphere. For a Lambertian surface, the reflectance function is a clamped cosine:

$$R(\vec{V}, \vec{N}) \quad = \quad \max\left(\vec{V} \cdot \vec{N}, 0\right), \tag{33}$$

which is either the cosine of the angle between vectors $\vec{V}$ and $\vec{N}$, or zero when the angle is greater than 90 degrees. This reflectance function effectively limits the integration in Equation (32) to the hemisphere about the surface normal $\vec{N}$, Figure 7. In addition, while we have assumed no cast shadows, Equation (33) explicitly models attached shadows, i.e., shadows due to surface normals facing away from the direction $\vec{V}$.

The convolution in Equation (32) can be simplified by expressing both the lighting environment and the reflectance function in terms of spherical harmonics. Spherical harmonics form an orthonormal basis for piecewise continuous functions on the sphere and are analogous to the Fourier basis on

the line or plane. The first three orders of spherical harmonics are shown in Figure 8 and defined as:

$$\begin{array}{lllllllll} Y_{0,0}(\vec{N}) & = & \frac{1}{\sqrt{4\pi}} & & Y_{1,-1}(\vec{N}) & = & \sqrt{\frac{3}{4\pi}}y & & Y_{1,0}(\vec{N}) & = & \sqrt{\frac{3}{4\pi}}z \\ Y_{1,1}(\vec{N}) & = & \sqrt{\frac{3}{4\pi}}x & & Y_{2,-2}(\vec{N}) & = & 3\sqrt{\frac{5}{12\pi}}xy & & Y_{2,-1}(\vec{N}) & = & 3\sqrt{\frac{5}{12\pi}}yz \\ Y_{2,0}(\vec{N}) & = & \frac{1}{2}\sqrt{\frac{5}{4\pi}}(3z^2-1) & & Y_{2,1}(\vec{N}) & = & 3\sqrt{\frac{5}{12\pi}}xz & & Y_{2,2}(\vec{N}) & = & \frac{3}{2}\sqrt{\frac{5}{12\pi}}(x^2-y^2) \end{array}$$

where $\vec{N} = \begin{pmatrix} x & y & z \end{pmatrix}$ in Cartesian coordinates.

The lighting environment expanded in terms of these spherical harmonics is:

$$L(\vec{V}) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} l_{n,m}Y_{n,m}(\vec{V}), \tag{34}$$

where $Y_{n,m}(\cdot)$ is the m^{th} spherical harmonic of order n, and $l_{n,m}$ is the corresponding coefficient of the lighting environment. Similarly, the reflectance function for Lambertian surfaces, $R(\vec{V},\vec{N})$, can be expanded in terms of spherical harmonics, and due to its symmetry about the surface normal, only harmonics with $m = 0$ appear in the expansion:

$$R(\vec{V},\vec{N}) = \sum_{n=0}^{\infty} r_n Y_{n,0}\left(\begin{pmatrix} 0 & 0 & \vec{V}\cdot\vec{N} \end{pmatrix}^T\right). \tag{35}$$

Note that for $m = 0$, the spherical harmonic $Y_{n,0}(\cdot)$ depends only on the z-component of its argument.

Convolutions of functions on the sphere become products when represented in terms of spherical harmonics [39, 2]. As a result, the irradiance, Equation (32), takes the form:

$$E(\vec{N}) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} \hat{r}_n l_{n,m}Y_{n,m}(\vec{N}), \tag{36}$$

where

$$\hat{r}_n = \sqrt{\frac{4\pi}{2n+1}}r_n. \tag{37}$$

The key observation in [39] and [2] was that the coefficients $\hat{r}_n$ for a Lambertian reflectance function decay rapidly, and thus the infinite sum in Equation (36) can be well approximated by the first nine terms:

$$E(\vec{N}) \approx \sum_{n=0}^{2}\sum_{m=-n}^{n} \hat{r}_n l_{n,m}Y_{n,m}(\vec{N}). \tag{38}$$

Since the constants $\hat{r}_n$ are known for a Lambertian reflectance function, the irradiance of a convex Lambertian surface under arbitrary distant lighting can be well modeled by the nine lighting environment coefficients $l_{n,m}$ up to order two.

Irradiance describes the total amount of light reaching a point on a surface. For a Lambertian surface, the reflected light, or radiosity, is proportional to the irradiance by a reflectance term ρ. In addition, Lambertian surfaces emit light uniformly in all directions, so the amount of light received by a viewer (i.e., camera) is independent of the view direction.

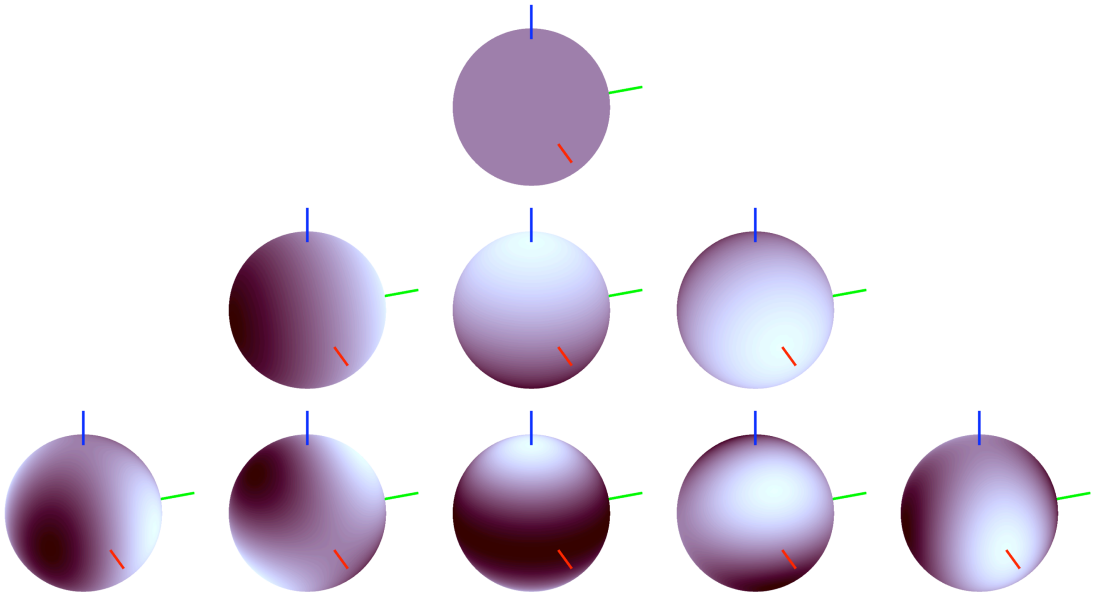


Figure 8: The first three orders of spherical harmonics as functions on the sphere. Shown from top to bottom are the order zero spherical harmonic, $Y_{0,0}(\cdot)$; the three order one spherical harmonics, $Y_{1,m}(\cdot)$; and the five order two spherical harmonics, $Y_{2,m}(\cdot)$.

A camera maps its received light to intensity through a camera response function $f(\cdot)$. Assuming the reflectance term ρ is constant across the surface, the measured intensity at a point $\vec{x}$ in the image is given by [10]:

$$I(\vec{x}) \quad = \quad f(\rho t E(\vec{N}(\vec{x}))). \tag{39}$$

where $E(\cdot)$ is the irradiance, $\vec{N}(\vec{x})$ is the surface normal at point $\vec{x}$, and t is the exposure time. For simplicity, we assume a linear camera response, and thus the intensity is related to the irradiance by an unknown multiplicative factor, which is assumed to have unit value—this assumption implies that the lighting coefficients can only be estimated to within an unknown scale factor. Under these assumptions, the relationship between image intensity and irradiance is simply:

$$I(\vec{x}) \quad = \quad E(\vec{N}(\vec{x})). \tag{40}$$

Since, under our assumptions, the intensity is equal to irradiance, Equation (40) can be written in terms of spherical harmonics by expanding Equation (38):

$$\begin{aligned} I(\vec{x}) \quad &= \quad l_{0,0}\pi Y_{0,0}(\vec{N}) + l_{1,-1}\tfrac{2\pi}{3} Y_{1,-1}(\vec{N}) + l_{1,0}\tfrac{2\pi}{3} Y_{1,0}(\vec{N}) + l_{1,1}\tfrac{2\pi}{3} Y_{1,1}(\vec{N}) \\ &+ \quad l_{2,-2}\tfrac{\pi}{4} Y_{2,-2}(\vec{N}) + l_{2,-1}\tfrac{\pi}{4} Y_{2,-1}(\vec{N}) + l_{2,0}\tfrac{\pi}{4} Y_{2,0}(\vec{N}) \\ &+ \quad l_{2,1}\tfrac{\pi}{4} Y_{2,1}(\vec{N}) + l_{2,2}\tfrac{\pi}{4} Y_{2,2}(\vec{N}). \end{aligned} \tag{41}$$

Note that this expression is linear in the nine lighting environment coefficients, $l_{0,0}$ to $l_{2,2}$. As such, given 3-D surface normals at $p \geq 9$ points on the surface of an object, the lighting environment

coefficients can be estimated as the least-squares solution to the following system of linear equations:

$$\begin{pmatrix} \pi Y_{0,0}(\vec{N}(\vec{x}_1)) & \frac{2\pi}{3} Y_{1,-1}(\vec{N}(\vec{x}_1)) & \dots & \frac{\pi}{4} Y_{2,2}(\vec{N}(\vec{x}_1)) \\ \pi Y_{0,0}(\vec{N}(\vec{x}_2)) & \frac{2\pi}{3} Y_{1,-1}(\vec{N}(\vec{x}_2)) & \dots & \frac{\pi}{4} Y_{2,2}(\vec{N}(\vec{x}_2)) \\ \vdots & \vdots & \ddots & \vdots \\ \pi Y_{0,0}(\vec{N}(\vec{x}_p)) & \frac{2\pi}{3} Y_{1,-1}(\vec{N}(\vec{x}_p)) & \dots & \frac{\pi}{4} Y_{2,2}(\vec{N}(\vec{x}_p)) \end{pmatrix} \begin{pmatrix} l_{0,0} \\ l_{1,-1} \\ \vdots \\ l_{2,2} \end{pmatrix} = \begin{pmatrix} I(\vec{x}_1) \\ I(\vec{x}_2) \\ \vdots \\ I(\vec{x}_p) \end{pmatrix}$$

$$M\vec{v} = \vec{b}, \tag{42}$$

where M is the matrix containing the sampled spherical harmonics, $\vec{v}$ is the vector of unknown lighting environment coefficients, and $\vec{b}$ is the vector of intensities at p points. The least-squares solution to this system is:

$$\vec{v} = \left(M^T M\right)^{-1} M^T \vec{b}. \tag{43}$$

This solution requires 3-D surface normals from at least nine points on the surface of an object. Without multiple images or known geometry, however, this requirement may be difficult to satisfy from an arbitrary image.

As in [24] and [32], we observe that under an assumption of orthographic projection, the z-component of the surface normal is zero along the occluding contour of an object. Therefore, the intensity profile along an occluding contour simplifies to:

$$I(\vec{x}) = A + l_{1,-1}\tfrac{2\pi}{3} Y_{1,-1}(\vec{N}) + l_{1,1}\tfrac{2\pi}{3} Y_{1,1}(\vec{N}) + l_{2,-2}\tfrac{\pi}{4} Y_{2,-2}(\vec{N}) + l_{2,2}\tfrac{\pi}{4} Y_{2,2}(\vec{N}), \tag{44}$$

where

$$A = l_{0,0}\tfrac{\pi}{2\sqrt{\pi}} - l_{2,0}\tfrac{\pi}{16}\sqrt{\tfrac{5}{\pi}}. \tag{45}$$

Note that the functions $Y_{i,j}(\cdot)$ depend only on the x and y components of the surface normal $\vec{N}$. That is, the five lighting coefficients can be estimated from only 2-D surface normals, which are relatively simple to estimate from a single image.[2] In addition, Equation (44) is still linear in its now five lighting environment coefficients, which can be estimated as the least-squares solution to:

$$\begin{pmatrix} 1 & \frac{2\pi}{3} Y_{1,-1}(\vec{N}(\vec{x}_1)) & \frac{2\pi}{3} Y_{1,1}(\vec{N}(\vec{x}_1)) & \frac{\pi}{4} Y_{2,-2}(\vec{N}(\vec{x}_1)) & \frac{\pi}{4} Y_{2,2}(\vec{N}(\vec{x}_1)) \\ 1 & \frac{2\pi}{3} Y_{1,-1}(\vec{N}(\vec{x}_2)) & \frac{2\pi}{3} Y_{1,1}(\vec{N}(\vec{x}_2)) & \frac{\pi}{4} Y_{2,-2}(\vec{N}(\vec{x}_2)) & \frac{\pi}{4} Y_{2,2}(\vec{N}(\vec{x}_2)) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \frac{2\pi}{3} Y_{1,-1}(\vec{N}(\vec{x}_p)) & \frac{2\pi}{3} Y_{1,1}(\vec{N}(\vec{x}_p)) & \frac{\pi}{4} Y_{2,-2}(\vec{N}(\vec{x}_p)) & \frac{\pi}{4} Y_{2,2}(\vec{N}(\vec{x}_p)) \end{pmatrix} \begin{pmatrix} A \\ l_{1,-1} \\ l_{1,1} \\ l_{2,-2} \\ l_{2,2} \end{pmatrix} = \begin{pmatrix} I(\vec{x}_1) \\ I(\vec{x}_2) \\ \vdots \\ I(\vec{x}_p) \end{pmatrix} \tag{46}$$

$$M\vec{v} = \vec{b}, \tag{47}$$

which has the same least-squares solution as before:

$$\vec{v} = \left(M^T M\right)^{-1} M^T \vec{b}. \tag{48}$$

Note that this solution only provides five of the nine lighting environment coefficients. We will show, however, that this subset of coefficients is still sufficiently descriptive for forensic analysis.

[2] The 2-D surface normal is the gradient vector of an implicit curve fit to the edge of an object.

When analyzing the occluding contours of objects in real images, it is often the case that the range of surface normals is limited, leading to an ill-conditioned matrix M. This limitation can arise from many sources, including occlusion or object geometry. As a result, small amounts of noise in either the surface normals or the measured intensities can cause large variations in the estimate of the lighting environment vector $\vec{v}$. To better condition the estimate, an error function $E(\vec{v})$ is defined that combines the least-squares error of the original linear system with a regularization term:

$$E(\vec{v}) \quad = \quad \|M\vec{v} - \vec{b}\|^2 + \lambda\|C\vec{v}\|^2, \tag{49}$$

where λ is a scalar, and the matrix C is diagonal with $\begin{pmatrix} 1 & 2 & 2 & 3 & 3 \end{pmatrix}$ on the diagonal. The matrix C is designed to dampen the effects of higher order harmonics and is motivated by the observation that the average power of spherical harmonic coefficients for natural lighting environments decreases with increasing harmonic order [12]. For the full lighting model when 3-D surface normals are available, Equation (42), the matrix C has $\begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 \end{pmatrix}$ on the diagonal.

The error function to be minimized, Equation (49), is a least-squares problem with a Tikhonov regularization [16]. The analytic minimum is found by differentiating with respect to $\vec{v}$:

$$\begin{aligned} \frac{dE(\vec{v})}{d\vec{v}} &= 2M^TM\vec{v} - 2M^T\vec{b} + 2\lambda C^TC\vec{v} \\ &= 2(M^TM + \lambda C^TC)\vec{v} - 2M^T\vec{b}, \end{aligned} \tag{50}$$

setting the result equal to zero, and solving for $\vec{v}$:

$$\vec{v} \quad = \quad (M^TM + \lambda C^TC)^{-1}M^T\vec{b}. \tag{51}$$

In practice, we have found that the conditioned estimate in Equation (51) is appropriate if less than 180° of surface normals are available along the occluding contour. If more than 180° of surface normals are available, the least-squares estimate, Equation (48), can be used, though both estimates will give similar results for small values of λ.

The estimated coefficient vector $\vec{v}$, Equation (51), is a low-order approximation of the lighting environment. For forensics purposes, we would like to differentiate between lighting environments based on these coefficients. Intuitively, coefficients from objects in different lighting environments should be distinguishable, while coefficients from objects in the same lighting environment should be similar. In addition, measurable differences in sets of coefficients should be mostly due to differences in the lighting environment and not to other factors such as object color or image exposure. Taking these issues into consideration, we propose an error measure between two estimated lighting environments. Let $\vec{v}_1$ and $\vec{v}_2$ be two vectors of lighting environment coefficients. From these coefficients, the irradiance profile along a circle (2-D) or a sphere (3-D) is synthesized, from which the error is computed. The irradiance profiles corresponding to $\vec{v}_1$ and $\vec{v}_2$ are given by:

$$\vec{x}_1 \quad = \quad M\vec{v}_1, \tag{52}$$

$$\vec{x}_2 \quad = \quad M\vec{v}_2, \tag{53}$$

where the matrix M is of the form in Equation (42) (for 3-D normals) or Equation (47) (for 2-D normals). After subtracting the mean, the correlation between these zero-meaned profiles is:

$$\mathrm{corr}(\vec{x}_1, \vec{x}_2) \quad = \quad \frac{\vec{x}_1^T\vec{x}_2}{\|\vec{x}_1\|\|\vec{x}_2\|} \tag{54}$$

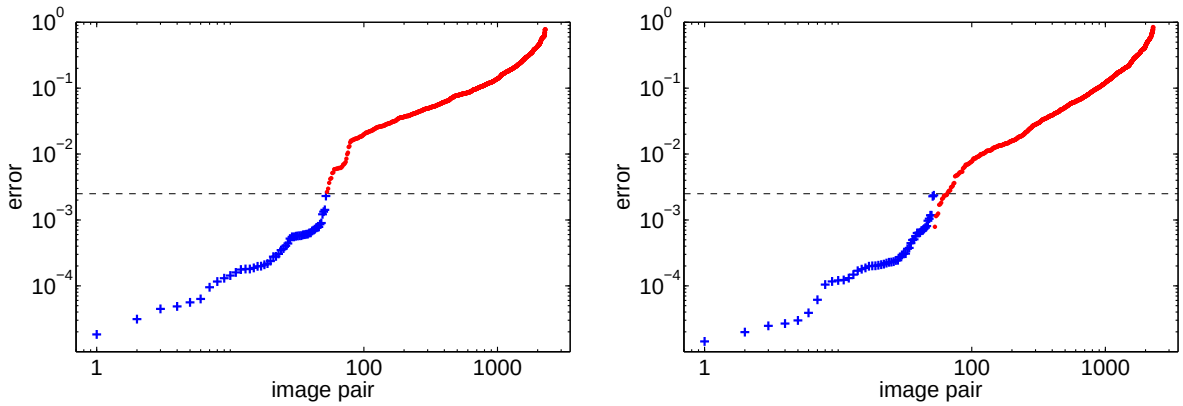


Figure 9: Errors between image pairs corresponding to the same ('+') and different ('·') locations using the full 9-parameter model with 3-D surface normals (left) and using the 5-parameter model with 2-D surface normals (right). Both the horizontal and vertical axes are scaled logarithmically.

In practice, this correlation can be computed directly from the lighting environment coefficients:

$$\mathrm{corr}(\vec{v}_1, \vec{v}_2) \quad = \quad \frac{\vec{v}_1^T Q \vec{v}_2}{\sqrt{\vec{v}_1^T Q \vec{v}_1}\sqrt{\vec{v}_2^T Q \vec{v}_2}}, \tag{55}$$

where the matrix Q for both the 2-D and 3-D cases is derived in [25]. By design, this correlation is invariant to both additive and multiplicative factors on the irradiance profiles $\vec{x}_1$ and $\vec{x}_2$. Recall that our coefficient vectors $\vec{v}_1$ and $\vec{v}_2$ are estimated to within an unknown multiplicative factor. In addition, different exposure times under a nonlinear camera response function can introduce an additive bias. The correlation is, therefore, invariant to these factors and produces values in the interval $[-1, 1]$. The final error is then given by:

$$D(\vec{v}_1, \vec{v}_2) \quad = \quad \frac{1}{2}\left(1 - \mathrm{corr}(\vec{v}_1, \vec{v}_2)\right), \tag{56}$$

with values in the range $[0, 1]$.

To test our ability to discriminate between lighting environments we photographed a diffuse sphere in 28 different locations with a 6.3 mega-pixel Nikon D-100 digital camera set to capture in high-quality JPEG mode. The focal length was set to 70 mm, the f-stop was fixed at $f/8$, and the shutter speed was varied to capture two or three exposures per location. In total, there were 68 images. For each image, the Adobe Photoshop "Quick Selection Tool" was used to locate the occluding contour of the sphere from which both 2-D and 3-D surface normals could be estimated. The 3-D surface normals were used to estimate the full set of nine lighting environment coefficients and the 2-D surface normals along the occluding contour were used to estimate five coefficients. For both cases, the regularization term λ in Equation (51) was set to 0.01. For each pair of images, the error, Equation (56), between the estimated coefficients was computed. In total, there were 2278 image pairs: 52 pairs were different exposures from the same location, and 2226 pairs were captured in different locations. The errors for all pairs for both models (3-D and 2-D) are shown in Figure 9. In both plots, the 52 image pairs from the same location are plotted first ('+'), sorted by error. The 2226 pairs from different locations are plotted next ('·'). Note that the axes are scaled logarithmically. For the 3-D

Figure 10: Superimposed on each image are the contours from which the surface normals and intensity values are extracted to form the matrix M and the corresponding vector $\vec{b}$, Equation (47).

case, the minimum error between an image pair from different locations is 0.0027 and the maximum error between an image pair from the same location is 0.0023. Therefore, the two sets of data, same location versus different location, are separated by a threshold of 0.0025. For the 2-D case, thirteen image pairs (0.6%) fell below 0.0025. These image pairs correspond to lighting environments that are indistinguishable based on the five coefficient model.

To be useful in a forensic setting, lighting estimates from objects in the same lighting environment should be robust to differences in color and material type, as well as to geometric differences, since arbitrary objects may not have the full range of surface normals available. To test our algorithm under these conditions, we downloaded twenty images of multiple objects in natural lighting environments from Flickr [22], Figure 10. In each image, occluding contours of two to four objects were specified using a semi-automated approach. A coarse contour was defined by painting along the edge of the object using Adobe Photoshop. Each stroke was then automatically divided into quadratic segments, or regions, which were fit to nearby points with large gradients. The analyzed regions for all images are shown in Figure 10. Analytic surface normals and intensities along the occluding contour were measured from the regions. With the 2-D surface normals and intensities, the five lighting environment coefficients were estimated, Equation (51). The regularization term λ in Equation (51) was increased to 0.1, which is larger than in the simulation due to an increasing sensitivity to noise.

Across all twenty images, there were 49 pairs of objects from the same image and 1329 pairs of objects from different images. For each pair of objects, the error between the estimated coefficients was computed. For objects in the same image, the average error was 0.009 with a standard deviation of 0.007 and a maximum error of 0.027. For comparison, between objects in different images the average error was 0.295 with a standard deviation of 0.273. There were, however, 196 pairs of objects (15%) from different images that fell below 0.027. The lighting environments in these images (e.g., the

two police images, the trees and skiers images, etc.) were indistinguishable using the five coefficient model. For objects from the same image, the pair with the maximum error of 0.027 is the basketball and basketball player. The sweaty skin of the basketball player is somewhat shiny, a violation of the Lambertian assumption. In addition, the shoulders and arms of the basketball player provide only a limited extent of surface normals, making the linear system somewhat ill-conditioned. In contrast, the objects from the same image with the minimum error of 0.0001 are the left and right pumpkins on the bench. Both pumpkins provide a large extent of surface normals, over 200°, and the surfaces are fairly diffuse. Since the surfaces fit the assumptions and the linear systems are well-conditioned, the error between the estimated coefficients is small.

We created three forgeries by mixing and matching several of the images in Figure 10. These forgeries are shown in Figure 11. Regions along the occluding contour of two to four objects in each image were selected for analysis. These regions are superimposed on the images in the right column of Figure 11. Surface normals and intensities along these occluding contour were extracted, from which the five lighting environment coefficients were estimated, Equation (51), with the regularization term $\lambda = 0.1$. Shown in each panel is a sphere rendered with the estimated coefficients. These spheres qualitatively show discrepancies between the lighting. For all pairs of objects originally in the same lighting environment, the average error is 0.005 with maximum error of 0.01. For pairs of objects from different lighting environments, the average error is 0.15 with a minimum error of 0.03.

The ability to estimate complex lighting environments was motivated by our earlier work in which we showed how to detect inconsistencies in the direction to an illuminating light source, Section 2.1. The work described here generalizes this approach by allowing us to estimate more complex models of lighting and in fact can be adapted to estimate the direction to a single light source. Specifically, by considering only the two first-order spherical harmonics, $Y_{1,-1}(\cdot)$ and $Y_{1,1}(\cdot)$, the direction to a light source can be estimated as $\tan^{-1}(l_{1,-1}/l_{1,1})$.

When creating a composite of two or more people, it is often difficult to exactly match the lighting, even if the lighting seems perceptually consistent. The reason for this is that complex lighting environments (multiple light sources, diffuse lighting, directional lighting) give rise to complex and subtle lighting gradients and shading effects in the image. Under certain simplifying assumptions (distant light sources and diffuse surfaces), arbitrary lighting environments can be modeled with a 9-dimensional model. This model approximates the lighting with a linear combination of spherical harmonics. We have shown how to approximate a simplified 5-dimensional version of this model from a single image, and how to stabilize the model estimation in the presence of noise. Inconsistencies in the lighting model across an image are then used as evidence of tampering.

2.4 Color Filter Array

The previous three sections described forensic analysis based on detecting inconsistencies in lighting. These tools are particularly useful in determining if a photograph was created by compositing several photographs together. Next, we describe a complementary forensic analysis that can determine if any part of an image was manipulated or changed (e.g., air-brushing) from the time of its original recording.

Most digital cameras capture color images using a single sensor in conjunction with an array of color filters. As a result, only one third of the samples in a color image are captured by the camera, the other two thirds being interpolated. This interpolation introduces specific correlations between the samples of a color image. When creating a digital forgery these correlations may be destroyed or altered. We describe the form of these correlations, and propose a method that quantifies and detects

Figure 11: Shown on the left are three forgeries: the ducks, swans, and football coach were each added into their respective images. Shown on the right are the analyzed regions superimposed in white, and spheres rendered from the estimated lighting coefficients.

$r_{1,1}$	$g_{1,2}$	$r_{1,3}$	$g_{1,4}$	$r_{1,5}$	$g_{1,6}$	
$g_{2,1}$	$b_{2,2}$	$g_{2,3}$	$b_{2,4}$	$g_{2,5}$	$b_{2,6}$	
$r_{3,1}$	$g_{3,2}$	$r_{3,3}$	$g_{3,4}$	$r_{3,5}$	$g_{3,6}$	
$g_{4,1}$	$b_{4,2}$	$g_{4,3}$	$b_{4,4}$	$g_{4,5}$	$b_{4,6}$	$\cdots$
$r_{5,1}$	$g_{5,2}$	$r_{5,3}$	$g_{5,4}$	$r_{5,5}$	$g_{5,6}$	
$g_{6,1}$	$b_{6,2}$	$g_{6,3}$	$b_{6,4}$	$g_{6,5}$	$b_{6,6}$	
			$\vdots$			$\ddots$

Figure 12: The top-left portion of a CFA image obtained from a Bayer array. The red, $r_{2i+1,2j+1}$, and blue, $b_{2i,2j}$, pixels are sampled on rectilinear lattices, while the green, $g_{2i+1,2j}$ and $g_{2i,2j+1}$, pixels are sampled twice as often on a quincunx lattice. Notice that at each pixel location only a single color sample is recorded.

them in any portion of an image.

A digital color image consists of three channels containing samples from different bands of the color spectrum, e.g., red, green, and blue. Most digital cameras, however, are equipped with a single CCD or CMOS sensor, and capture color images using a color filter array (CFA). The most frequently used CFA, the Bayer array [3], 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, Figure 12. 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 Figure 12, 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:

$$\tilde{R}(x,y) = \begin{cases} S(x,y) & \text{if } S(x,y) = r_{x,y} \\ 0 & \text{otherwise} \end{cases} \tag{57}$$

$$\tilde{G}(x,y) = \begin{cases} S(x,y) & \text{if } S(x,y) = g_{x,y} \\ 0 & \text{otherwise} \end{cases} \tag{58}$$

$$\tilde{B}(x,y) = \begin{cases} S(x,y) & \text{if } B(x,y) = b_{x,y} \\ 0 & \text{otherwise} \end{cases}, \tag{59}$$

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 estimation of the missing color samples is referred to as CFA interpolation or demosaicking. CFA interpolation has been extensively studied and many methods have been proposed (see, for example, [40] for a survey, and [17, 20, 31] for more recent methods).

The simplest methods for demosaicking 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:

$$R(x,y) = \sum_{u,v=-N}^{N} h_r(u,v)\tilde{R}(x-u,y-v) \quad (60)$$

$$G(x,y) = \sum_{u,v=-N}^{N} h_g(u,v)\tilde{G}(x-u,y-v) \quad (61)$$

$$B(x,y) = \sum_{u,v=-N}^{N} h_b(u,v)\tilde{B}(x-u,y-v), \quad (62)$$

where $\tilde{R}(\cdot)$, $\tilde{G}(\cdot)$, $\tilde{B}(\cdot)$ are defined in Equations (57)-(59), and $h_r(\cdot)$, $h_g(\cdot)$, $h_b(\cdot)$ are linear filters of size $(2N+1)\times(2N+1)$. Different forms of interpolation (nearest neighbor, bilinear, bicubic [28], etc.) differ in the form of the interpolation filter used. For the Bayer array, the bilinear and bicubic filters for the red and blue channels are separable. The 1-D filters are given by:

$$h_l = \begin{bmatrix} 1/2 & 1 & 1/2 \end{bmatrix}$$
$$h_c = \begin{bmatrix} -1/16 & 0 & 9/16 & 1 & 9/16 & 0 & -1/16 \end{bmatrix}.$$

There are many other CFA interpolation algorithms including smooth hue transition [6], median filter [15], gradient-based [29], adaptive color plane [18], and threshold-based variable number of gradients [5]. Regardless of their specific implementations, each CFA interpolation algorithm introduces specific statistical correlations between a subset of pixels in each color channel. Since the color filters in a CFA are typically arranged in a periodic pattern, these correlations are periodic. Consider, for example, the red channel, $R(x,y)$, that has been sampled on a Bayer array, Figure 12, then CFA interpolated using bilinear interpolation. In this case, 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:

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

Note that in this simple case, the estimated samples are perfectly correlated to their neighbors. As such, a CFA interpolated image can be detected (in the absence of noise) by noticing, for example, that every other sample in every other row or column is perfectly correlated to its neighbors. At the same time, the non-interpolated samples are less likely to be correlated in precisely the same manner. Furthermore, it is likely that tampering will destroy these correlations, or that the splicing together of two images from different cameras will create inconsistent correlations across the composite image. As such, the presence or lack of correlations produced by CFA interpolation can be used to authenticate an image, or expose it as a forgery.

We begin by assuming a simple linear model for the periodic correlations introduced by CFA interpolation. That is, each interpolated pixel is correlated to a weighted sum of pixels in a small neighborhood centered about itself. While perhaps overly simplistic when compared to the highly non-linear

nature of most CFA interpolation algorithms, this simple model is both easy to parametrize and can reasonably approximate each of the CFA interpolation algorithms described above. Note that most CFA algorithms estimate a missing color sample from neighboring samples in all three color channels. For simplicity, however, we ignore these inter-channel correlations and treat each color channel independently.

If the specific form of the correlations is known (i.e., the parameters of the linear model), then it would be straightforward to determine which samples are correlated to their neighbors. On the other hand, if it was known which samples are correlated to their neighbors, the specific form of the correlations could be easily determined. In practice, of course, neither are known. To simultaneously estimate both we employ the expectation/maximization (EM) algorithm [11], as described below.

Let $f(x, y)$ denote a color channel (red, green, or blue) of a CFA interpolated image. We begin by assuming that each sample in $f(x, y)$ belongs to one of two models: (1) M_1 if the sample is linearly correlated to its neighbors, satisfying:

$$f(x,y) \quad = \quad \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) + n(x,y), \tag{63}$$

where the model parameters are given by the linear coefficients $\vec{\alpha} = \{\alpha_{u,v} | -N \leq u, v \leq N\}$ (N is an integer and $\alpha_{0,0} = 0$), and $n(x, y)$ denotes independent and identically distributed samples drawn from a Gaussian distribution with zero mean and unknown variance σ^2; or (2) M_2 if the sample is not correlated to its neighbors, i.e., is generated by an "outlier process".

The expectation-maximization algorithm (EM) is a two-step iterative algorithm: (1) in the E-step the probability of each sample belonging to each model is estimated; and (2) in the M-step the specific form of the correlations between samples is estimated. More specifically, in the E-step, the probability of each sample of $f(x, y)$ belonging to model M_1 is estimated using Bayes' rule:

$$\Pr\{f(x,y) \in M_1 \mid f(x,y)\} \quad = \quad \frac{\Pr\{f(x,y) \mid f(x,y) \in M_1\} \Pr\{f(x,y) \in M_1\}}{\sum_{i=1}^{2} \Pr\{f(x,y) \mid f(x,y) \in M_i\} \Pr\{f(x,y) \in M_i\}}, \tag{64}$$

where the prior probabilities $\Pr\{f(x, y) \in M_1\}$ and $\Pr\{f(x, y) \in M_2\}$ are assumed to be equal to $1/2$. The probability of observing a sample $f(x, y)$ knowing it was generated from model M_1 is given by:

$$\Pr\{f(x,y) \mid f(x,y) \in M_1\} \quad = \quad \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{\left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) \right)^2}{2\sigma^2} \right]. \tag{65}$$

The variance, σ^2, of this Gaussian distribution is estimated in the M-step. A uniform distribution is assumed for the probability of observing a sample generated by the outlier model, M_2, i.e., $\Pr\{f(x, y) \mid f(x, y) \in M_2\}$ is equal to the inverse of the range of possible values of $f(x, y)$. Note that the E-step requires an estimate of the coefficients $\vec{\alpha}$, which on the first iteration is chosen randomly. In the M-step, a new estimate of $\vec{\alpha}$ is computed using weighted least squares, by minimizing the following quadratic error function:

$$E(\vec{\alpha}) \quad = \quad \sum_{x,y} w(x,y) \cdot \left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) \right)^2, \tag{66}$$

where the weights $w(x,y) \equiv \Pr\{f(x,y) \in M_1 \mid f(x,y)\}$, Equation (64). This error function is minimized by computing the gradient with respect to $\vec{\alpha}$, setting this gradient equal to zero, and solving the resulting linear system of equations. Setting equal to zero the partial derivative with respect to one of the coefficients, $\alpha_{s,t}$, yields:

$$\frac{\partial E}{\partial \alpha_{s,t}} = 0 \tag{67}$$

$$\sum_{x,y} w(x,y) f(x+s,y+t) \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u,y+v) = \sum_{x,y} w(x,y) f(x+s,y+t) f(x,y). \tag{68}$$

Re-ordering the terms on the left-hand side yields:

$$\sum_{u,v=-N}^{N} \alpha_{u,v} \left(\sum_{x,y} w(x,y) f(x+s,y+t) f(x+u,y+v) \right) = \sum_{x,y} w(x,y) f(x+s,y+t) f(x,y) \; . \tag{69}$$

This process is repeated for each component, $\alpha_{s,t}$, of $\vec{\alpha}$, to yield a system of linear equations that can be solved using standard techniques. The E-step and the M-step are iteratively executed until a stable estimate of $\vec{\alpha}$ is achieved. The final $\vec{\alpha}$ has the property that it maximizes the likelihood of the observed samples.

We collected one hundred images, fifty of resolution 512×512, and fifty of resolution 1024×1024. Each of these images were cropped from a smaller set of twenty 1600×1200 images taken with a Nikon Coolpix 950 camera, and twenty 3034×2024 images taken with a Nikon D100 camera. The Nikon Coolpix 950 employs a four-filter (yellow, cyan, magenta, green) CFA, and was set to store the images in uncompressed TIFF format. The Nikon D100 camera employs a Bayer array, and was set to store the images in RAW format. To avoid interference with the CFA interpolation of the cameras, each color channel of these images was independently blurred with a 3×3 binomial filter, and down-sampled by a factor of two in each direction. These down-sampled color images, of size 256×256 or 512×512, were then re-sampled onto a Bayer array, and CFA interpolated using the algorithms described above.

Shown in Figures 13 and 14 are the results of running the EM algorithm on eight 256×256 color images that were CFA interpolated. The parameters of the EM algorithm were: $N = 1$, $\sigma_0 = 0.0075$, and $p_0 = 1/256$. Shown in the left column are the images, in the middle column the estimated probability maps from the green channel (the red and blue channels yield similar results), and in the right column the magnitude of the Fourier transforms of the probability maps [3]. Note that, although the periodic patterns in the probability maps are not visible at this reduced scale, they are particularly salient in the Fourier domain in the form of localized peaks. Note also that these peaks do not appear in the images that are not CFA interpolated (last row of Figures 13 and 14).

Since it is likely that tampering will destroy the periodicity of the CFA correlations, it may be possible to detect and localize tampering in any portion of an image. To illustrate this, consider the leftmost image in Figure 15, taken with a Nikon D100 digital camera and saved in RAW format.

[3] For display purposes, the probability maps were up-sampled by a factor of two before Fourier transforming. The periodic patterns introduced by CFA interpolation have energy in the highest horizontal, vertical and diagonal frequencies, which corresponds to localized frequency peaks adjacent to the image edges. Up-sampling by a factor of two shrinks the support of a probability map's spectrum, and shifts these peaks into the mid-frequencies, where they are easier to see. Also for display purposes, the Fourier transforms of the probability maps were high-pass filtered, blurred, scaled to fill the range $[0, 1]$, and gamma corrected with an exponent of 2.0.

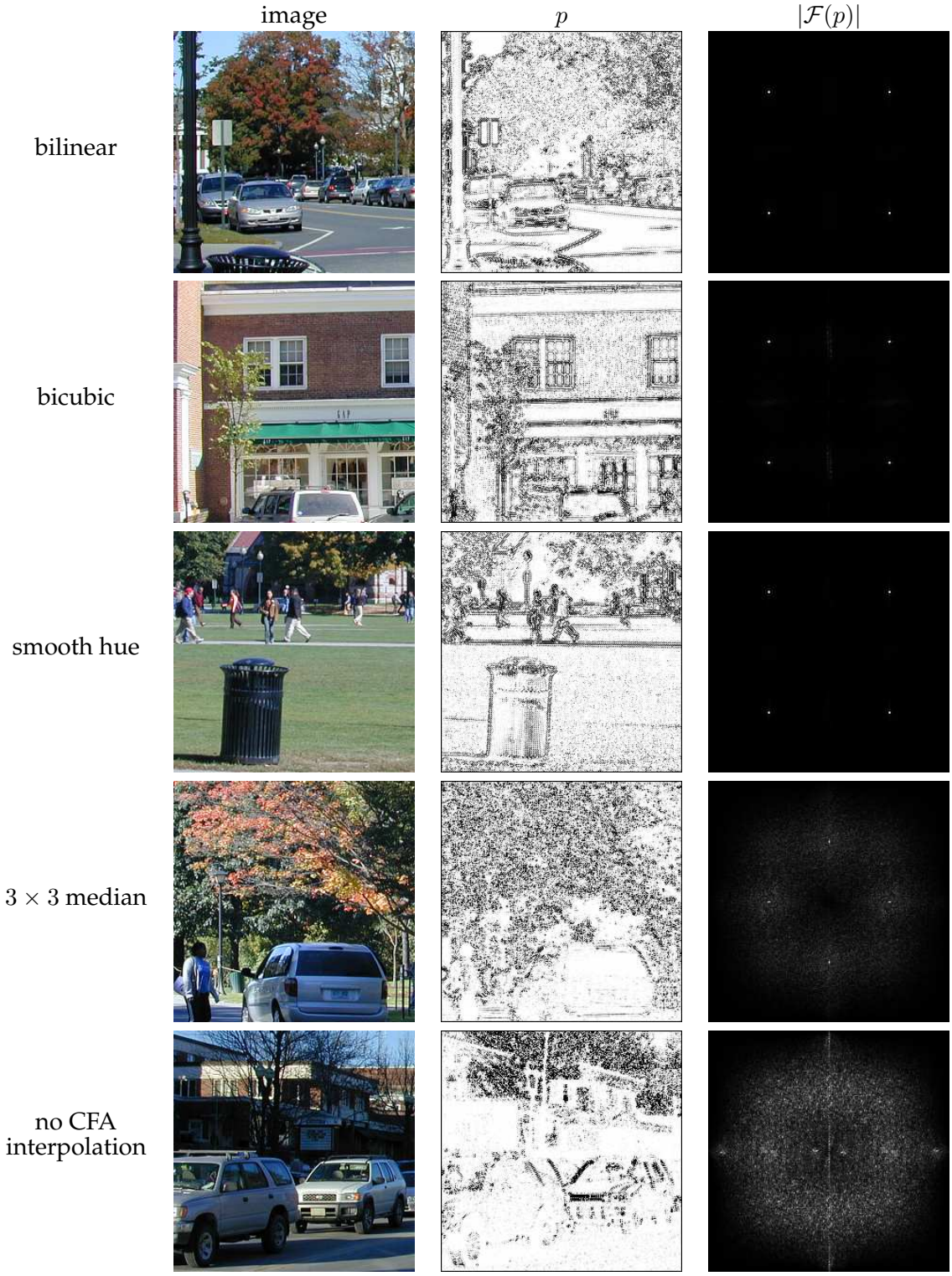


Figure 13: Shown in each row is an image interpolated with the specified algorithm, the probability map of the green channel as output by the EM algorithm, and the magnitude of the Fourier transform of the probability map. Note the peaks in $|\mathcal{F}(p)|$ corresponding to periodic correlations in the CFA interpolated images, and the lack of such peaks in the non-CFA interpolated image (last row).

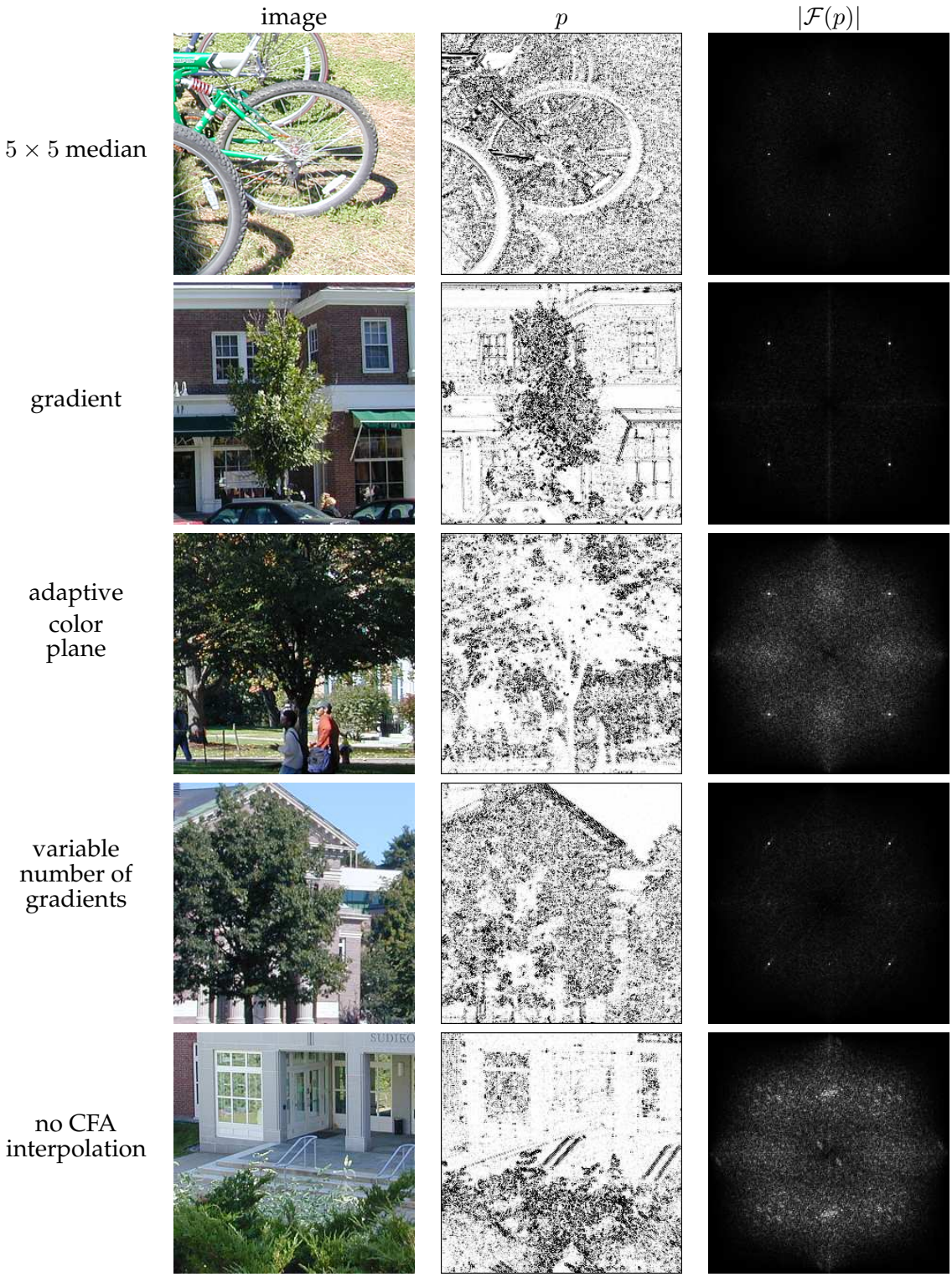


Figure 14: Shown in each row is an image interpolated with the specified algorithm, the probability map of the green channel as output by the EM algorithm, and the magnitude of the Fourier transform of the probability map. Note the peaks in $|\mathcal{F}(p)|$ corresponding to periodic correlations in the CFA interpolated images, and the lack of such peaks in the non-CFA interpolated image (last row).

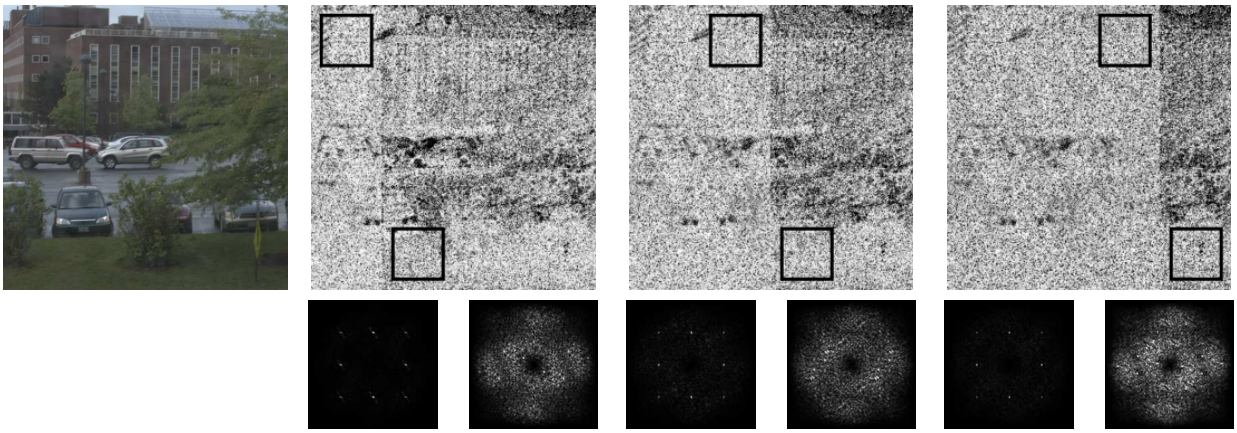


Figure 15: Shown are an image, and the probability maps of composite images obtained by splicing together the CFA interpolated image (left portion of each image) and the same image without CFA interpolation (right portion). Also shown are the magnitudes of the Fourier transforms of windows from the two regions. Note that windows from the CFA interpolated regions (left) have localized peaks in the Fourier domain, while the windows from the "tampered" regions (right) do not.

Each color channel of this image (initially interpolated using the adaptive color plane technique) was blurred with a 3×3 binomial filter and down-sampled by a factor of two in order to destroy the CFA periodic correlations. The 512×512 down-sampled image was then re-sampled on a Bayer array and CFA interpolated. Next, composite images, 512×512 pixels in size, were created by splicing, in varying proportions ($1/4$, $1/2$, and $3/4$), the non-CFA interpolated image and the same image CFA interpolated with the bicubic algorithm. Shown in Figure 15 are the probability maps obtained from running EM on the red channel of the composite images. Notice that these probability maps clearly reveal the presence of two distinct regions. Shown below each probability map are the magnitudes of the Fourier transforms of two windows, one from the non-CFA interpolated portion (right), and one from the CFA interpolated portion (left). Notice the presence of localized frequency peaks in the CFA interpolated portion, and the lack of such peaks in the non-CFA interpolated portion.

Shown in Figure 16 is a perceptually plausible forgery created using Adobe Photoshop (top row, right). The tampering consisted of hiding the damage on the car hood using air-brushing, smudging, blurring, and duplication. Also shown is the original image CFA interpolated using bicubic interpolation (top row, left), the estimated probability map of the tampered image (middle row), and the magnitude of the Fourier transforms of two windows one from a tampered portion (bottom row, left), and one from an unadulterated portion (bottom row, right). Note that even though the periodic pattern is not visible in the probability map, localized frequency peaks reveal its presence. Note also that the window from the tampered region does not contain frequency peaks.

The benefit of this approach is that nearly any manipulation to the image can be detected. The drawback is that an original resolution version of the image is required for authentication. And, as with any authentication scheme, our forensic analysis is vulnerable to counter-attack. A tampered image could, for example, be re-sampled onto a color filter array, and then re-interpolated. This attack, however, requires knowledge of the camera's CFA pattern and interpolation algorithm, and may be beyond the reaches of a novice forger.

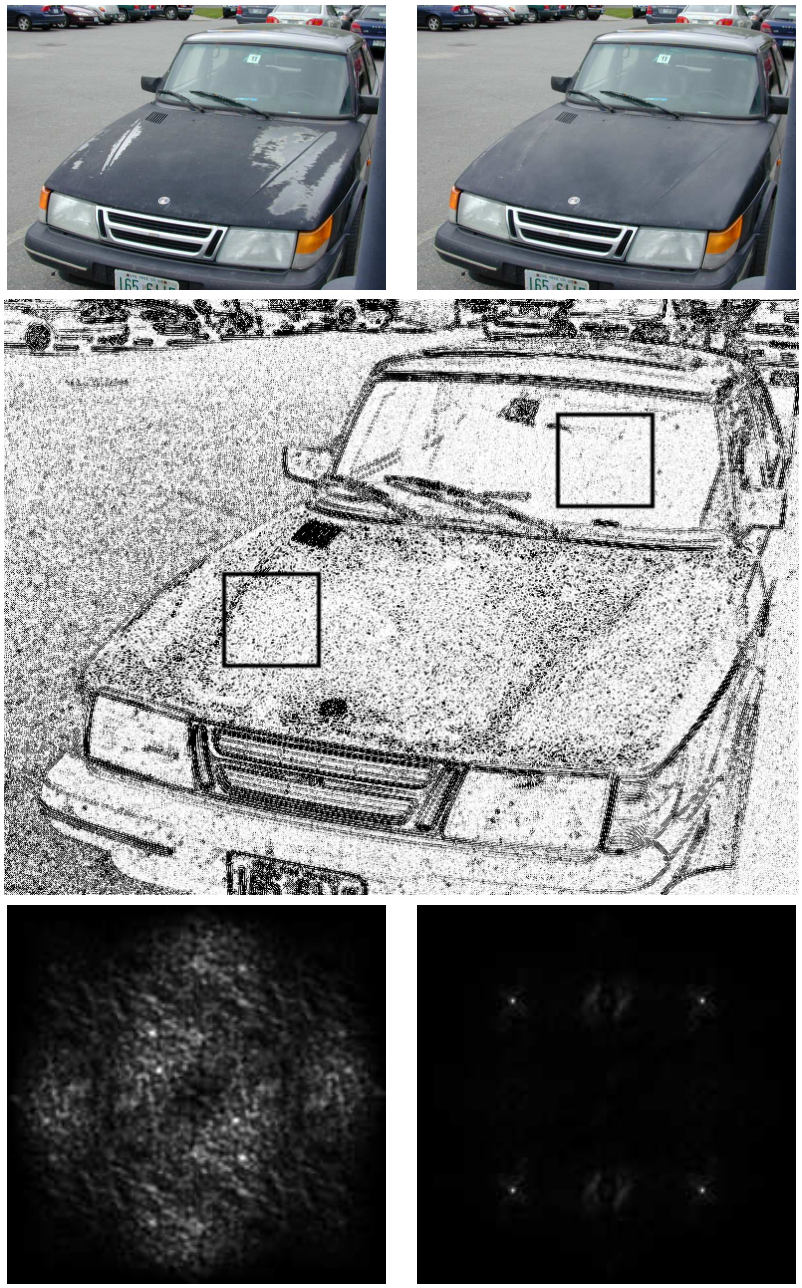


Figure 16: Shown are: (top) an original and a tampered image, (middle) the probability map of the tampered image's green channel, and (bottom) the magnitude of the Fourier transform of two windows from the probability map. The windows correspond to a tampered and an unadulterated portions of the forgery. Note the lack of peaks in the tampered region signifying the absence of CFA interpolation.

2.5 JPEG Ghosts

Although highly effective in some situations, the techniques described above are not applicable when analyzing low quality images. Detecting tampering in low quality images is particularly challenging since low quality images often destroy many artifacts that could be used to detect tampering. One such technique is described next that detects tampering which results when part of a JPEG image is inserted into another higher quality JPEG image. For example, when one person's head is spliced onto another person's body, or when two separately photographed people are combined into a single composite. This approach works by explicitly determining if part of an image was originally compressed at a lower quality relative to the rest of the image.

In the standard JPEG compression scheme [1, 44], a color image (RGB) is first converted into luminance/chrominance space (YCbCr). The two chrominance channels (CbCr) are typically subsampled by a factor of two relative to the luminance channel (Y). Each channel is then partitioned into 8×8 pixel blocks. These values are converted from unsigned to signed integers (e.g., from $[0, 255]$ to $[-128, 127]$). Each block is converted to frequency space using a 2-D discrete cosine transform (DCT). Each DCT coefficient, c, is then quantized by an amount q:

$$\hat{c} \;=\; \text{round}(c/q), \tag{70}$$

where the quantization q depends on the spatial frequency and channel. Larger quantization values q yield better compression at the cost of image degradation. Quantization values are typically larger in the chrominance channels, and in the higher spatial frequencies, roughly modeling the sensitivity of the human visual system.

Consider now a set of coefficients c_1 quantized by an amount q_1, which are subsequently quantized a second time by an amount q_2 to yield coefficients c_2. With the exception of $q_2 = 1$ (i.e., no quantization), the difference between c_1 and c_2 will be minimal when $q_2 = q_1$. It is obvious that the difference between c_1 and c_2 increases for quantization value $q_2 > q_1$ since the coefficients become increasingly more sparse as q_2 increases. For values of $q_2 < q_1$, the difference between c_1 and c_2 also increases because although the second quantization does not affect the granularity of the coefficients, it does cause a shift in their values. Shown in Figure 17(a), for example, is the sum of squared differences between c_1 and c_2 as a function of the second quantization q_2, where $q_1 = 17$, and where the coefficients c_1 are drawn from a normal zero-mean distribution. Note that this difference increases as a function of increasing q_2, with the exception of $q_2 = q_1$, where the difference is minimal. If q_1 is not prime, unlike the above example, then multiple minima may appear at quality values q_2 that are integer multiples of q_1. As will be seen below, this issue can be circumvented by averaging over all of the JPEG DCT coefficients.

Consider next a set of coefficients c_0 quantized by an amount q_0, followed by quantization by an amount $q_1 < q_0$ to yield c_1. Further quantizing c_1 by q_2 yields the coefficients c_2. As before, the difference between c_1 and c_2 will be minimal when $q_2 = q_1$. But, since the coefficients were initially quantized by q_0, where $q_0 > q_1$, we expect to find a second minimum when $q_2 = q_0$. Shown in Figure 17(b) is the sum of squared differences between c_1 and c_2, as a function of q_2, where $q_0 = 23$ and $q_1 = 17$. As before, this difference increases as a function of increasing q_2, reaches a minimum at $q_2 = q_1 = 17$, and most interestingly has a second local minimum at $q_2 = q_0 = 23$. We refer to this second minimum as a JPEG ghost, as it reveals that the coefficients were previously quantized (compressed) with a larger quantization (lower quality).

Recall that the JPEG compression scheme separately quantizes each spatial frequency within a 8×8 pixel block. One approach to detecting JPEG ghosts would be to separately consider each spatial fre-

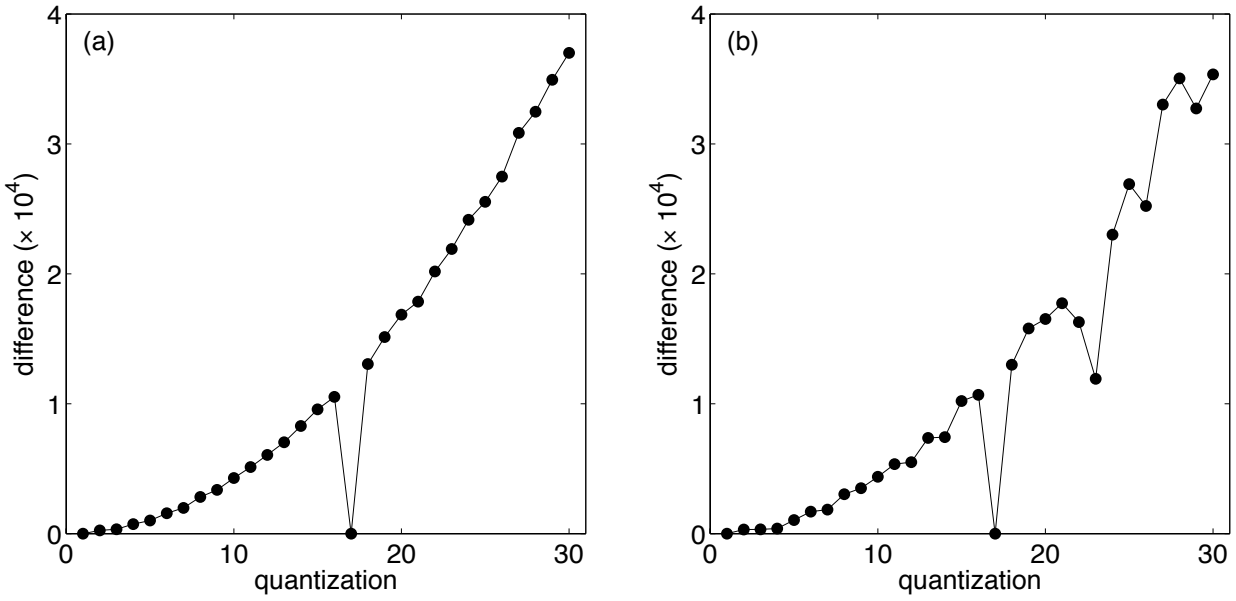


Figure 17: Shown in panel (a) is the sum of squared differences between coefficients quantized by an amount $q_1 = 17$, followed by a second quantization in the range $q_2 \in [1, 30]$ (horizontal axis) – this difference reaches a minimum at $q_2 = q_1 = 17$. Shown in panel (b) is the sum of squared differences between coefficients quantized initially by an amount $q_0 = 23$ followed by $q_1 = 17$, followed by quantization in the range $q_2 \in [1, 30]$ (horizontal axis) – this difference reaches a minimum at $q_2 = q_1 = 17$ and a local minimum at $q_2 = q_0 = 23$, revealing the original quantization.

quency in each of the three luminance/color channels. However, recall that multiple minima are possible when comparing integer multiple quantization values. If, on the other hand, we consider the cumulative effect of quantization on the underlying pixel values, then this issue is far less likely to arise (unless all 192 quantization values at different JPEG qualities are integer multiples of one another – an unlikely scenario). Therefore, instead of computing the difference between the quantized DCT coefficients, we consider the difference computed directly from the pixel values, as follows:

$$d(x, y, q) = \frac{1}{3} \sum_{i=1}^{3} [f(x, y, i) - f_q(x, y, i)]^2, \tag{71}$$

where $f(x, y, i)$, $i = 1, 2, 3$, denotes each of three RGB color channels[4], and $f_q(\cdot)$ is the result of compressing $f(\cdot)$ at quality q.

Shown in the top left panel of Figure 18 is an image whose central 200×200 pixel region was extracted, compressed at a JPEG quality of $65/100$, and re-inserted into the image whose original quality was 85. Shown in each subsequent panel is the sum of squared differences, Equation (71), between this manipulated image, and a re-saved version compressed at different JPEG qualities. Note that the central region is clearly visible when the image is re-saved at the quality of the tampered region (65). Also note that the overall error reaches a minimum at the saved quality of 85. There are some

[4]The detection of JPEG ghosts is easily adapted to grayscale images by simply computing $d(x, y, q)$, Equation (71), over a single image channel.

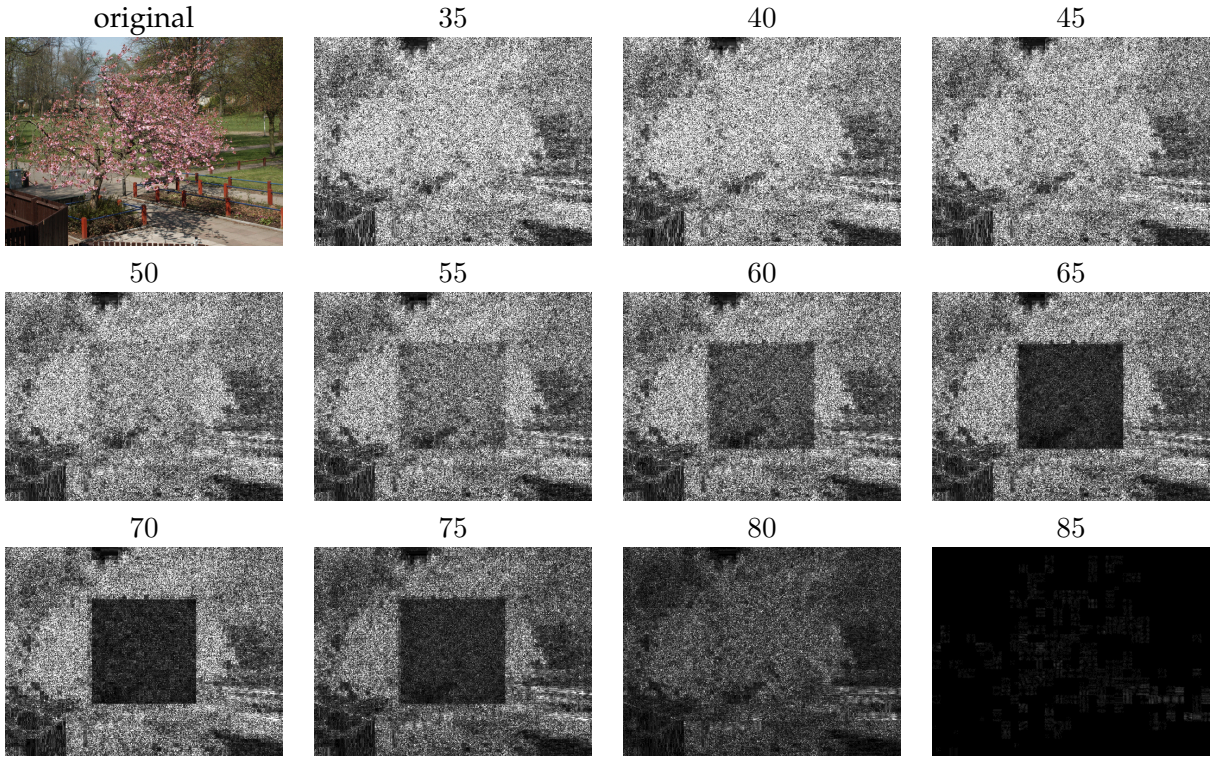


Figure 18: Shown in the top left panel is the original image from which a central 200×200 region was extracted, saved at JPEG quality 65, and re-inserted into the image whose original quality was 85. Shown in each subsequent panel is the difference between this image and a re-saved version compressed at different JPEG qualities in the range $[35, 85]$. At the originally saved quality of 65, the central region has a lower difference than the remaining image.

variations in the difference images within and outside of the tampered region which could possibly confound a forensic analysis. These fluctuations are due to the underlying image content. Specifically, because the image difference is computed across all spatial frequencies, a region with small amounts of high spatial frequency content (e.g., a mostly uniform sky) will have a lower difference as compared to a highly textured region (e.g., grass). In order to compensate for these differences, we consider a spatially averaged and normalized difference measure. The difference image is first averaged across a $b \times b$ pixel region:

$$\delta(x, y, q) \quad = \quad \frac{1}{3} \sum_{i=1}^{3} \frac{1}{b^2} \sum_{b_x=0}^{b-1} \sum_{b_y=0}^{b-1} [f(x + b_x, y + b_y, i) - f_q(x + b_x, y + b_y, i)]^2, \tag{72}$$

and then normalized so that the averaged difference at each location (x, y) is scaled into the range $[0, 1]$:

$$d(x, y, q) \quad = \quad \delta(x, y, q) - \frac{\min_q[\delta(x, y, q)]}{\max_q[\delta(x, y, q)] - \min_q[\delta(x, y, q)]}. \tag{73}$$

Although the JPEG ghosts are often visually highly salient, it is still useful to quantify if a specified region is statistically distinct from the rest of the image. To this end, the two-sample Kolmogorov-

Smirnov statistic [7] is employed to determine if the distribution of differences, Equation (73), in two regions are similar or distinct. The K-S statistic is defined as:

$$k = \max_u |C_1(u) - C_2(u)|, \tag{74}$$

where $C_1(u)$ and $C_2(u)$ are the cumulative probability distributions of two specified regions in the computed difference $d(x, y, q)$, where each value of q is considered separately.

There are two potential complicating factors that arise when detecting JPEG ghosts in a general forensic setting. First, it is likely that different cameras and photo-editing software packages will employ different JPEG quality scales and hence quantization tables [13]. When iterating through different qualities it would be ideal to match these qualities and tables, but this may not always be possible. Working to our advantage, however, is that the difference images are computed by averaging across all spatial frequencies. As a result small differences in the original and subsequent quantization tables will likely not have a significant impact. The second practical issue is that in the above examples we have assumed that the tampered region remains on its original 8×8 JPEG lattice after being inserted and saved. If this is not the case, then the mis-alignment may destroy the JPEG ghost since new spatial frequencies will be introduced by saving on a new JPEG block lattice. This problem can be alleviated by sampling all 64 possible alignments (a 0 to 7 pixel shift in the horizontal and vertical directions). Specifically, an image is shifted to each of these 64 locations prior to saving at each JPEG quality. Although this increases the complexity of the analysis, each comparison is efficient, leading to a minimal impact in overall run-time complexity.

To test the efficacy of detecting JPEG ghosts, $1,000$ uncompressed TIFF images were obtained from the Uncompressed Colour Image Database (UCID) [42]. These color images are each of size 512×384 and span a wide range of indoor and outdoor scenes. A central portion from each image was removed, saved at a specified JPEG quality of Q_0, re-inserted into the image, and then the entire image was saved at the same or different JPEG quality of Q_1. The MatLab function `imwrite` was used to save images in the JPEG format. This function allows for JPEG qualities to be specified in the range of 1 to 100. The size of the central region ranged from 50×50 to 200×200 pixels. The JPEG quality Q_1 was selected randomly in the range 40 to 90, and the difference between JPEG qualities Q_0 and Q_1 ranged from 0 to 25, where $Q_0 \leq Q_1$ (i.e., the quality of the central region is less than the rest of the image, yielding quantization levels for the central region that are larger than for the rest of the image). Note that this manipulation is visually seamless, and does not disturb any JPEG blocking statistics.

Note that is assumed here that the same JPEG qualities/tables were used in the creation and testing of an image, and that there is no shift in the tampered region from its original JPEG block lattice. The impact of these assumptions will be explored below, where it is shown that they are not critical to the efficacy of the detection of JPEG ghosts.

After saving an image at quality Q_1, it was re-saved at qualities Q_2 ranging from 30 to 90 in increments of 1. The difference between the image saved at quality Q_1 and each image saved at quality Q_2 was computed as specified by Equation (73), with $b = 16$. The K-S statistic, Equation (74), was used to compute the statistical difference between the image's central region, and the rest of the image. If the K-S statistic for any quality Q_2 exceeded a specified threshold, the image was classified as manipulated. This threshold was selected to yield a less than 1% false positive rate (an authentic image incorrectly classified as manipulated).

Many of the images in the UCID database have significant regions of either saturated pixels, or largely uniform intensity patches. These regions are largely unaffected by varying JPEG compression

Table 2: JPEG ghost detection accuracy (%)

	$Q_1 - Q_0$					
size	0	5	10	15	20	25
200×200	99.2	14.8	52.6	88.1	93.8	99.9
150×150	99.2	14.1	48.5	83.9	91.9	99.8
100×100	99.1	12.6	44.1	79.5	91.1	99.8
50×50	99.3	5.4	27.9	58.8	77.8	97.7

qualities, and therefore exhibit little variation in the computed difference images, Equation (73). As such, these regions provide unreliable statistics and were ignored when computing the K-S statistic, Equation (74). Specifically, regions of size $b \times b$ with an average intensity variance less than 2.5 gray values were simply not included in the computation of the K-S statistic.

Shown in Table 2 are the estimation results as a function of the size of the manipulated region (ranging from 200×200 to 50×50) and the difference in JPEG qualities between the originally saved central region, Q_0, and the final saved quality, Q_1 (ranging from 0 to 25 – a value of $Q_1 - Q_0 = 0$ denotes no tampering). Note that accuracy for images with no tampering (first column) is greater than 99% (i.e., a less than 1% false positive rate). Also note that the detection accuracy is above 90% for quality differences larger than 20 and for tampered regions larger than 100×100 pixels. The detection accuracy degrades with smaller quality differences and smaller tampered regions. Shown in Figure 19(a) are ROC curves for a tampered region of size 150×150 and a quality difference of 15. Shown in Figure 19(b) are ROC curves for a tampered region of size 100×100 and a quality difference of 10. In each panel, the solid curve corresponds to the accuracy of detecting the tampered region, and the dashed curve corresponds to the accuracy of correctly classifying an authentic image. The vertical dotted lines denote false positive rates of 10%, 5%, and 1%. As expected, there is a natural trade-off between the detection accuracy and the false positives which can be controlled with the threshold on the K-S statistic.

The next few examples illustrate the efficacy of detecting JPEG ghosts in visually plausible forgeries. In each example, the forgery was created and saved using Adobe Photoshop CS3 which employs a 12-point JPEG quality scale. The MatLab function `imwrite` was then used to re-compress each image on a 100-point scale. In order to align the original JPEG block lattice with the re-saved lattice, the image was translated to each of 64 possible spatial locations (between 0 and 7 pixels in the horizontal and vertical directions). The shift that yielded the largest K-S statistic was then selected.

Shown in Figure 20 are an original and doctored image. The inserted flying car was originally of JPEG quality $4/12$ and the final image was saved at quality $10/12$. Shown in the bottom portion of Figure 20 are the difference images between the tampered image saved at JPEG qualities 60 through 98 in steps of 2. The maximal K-S statistic for the car was 0.92. Regions of low variance are coded with mid-level gray in each panel. A second example is shown in Figure 21. The inserted dolphin was originally of JPEG quality $5/12$ and the final image was saved at quality $8/12$. Shown in the bottom portion of Figure 21 are the difference images between the tampered image saved at JPEG qualities 60 through 100 in steps of 2. The maximal K-S statistic for the dolphin was 0.84. In both examples, the JPEG ghosts of the inserted car and dolphin are visually salient and statistically distinct from the rest of the image.

The advantage of this approach is that it is effective on low quality images and can detect relatively

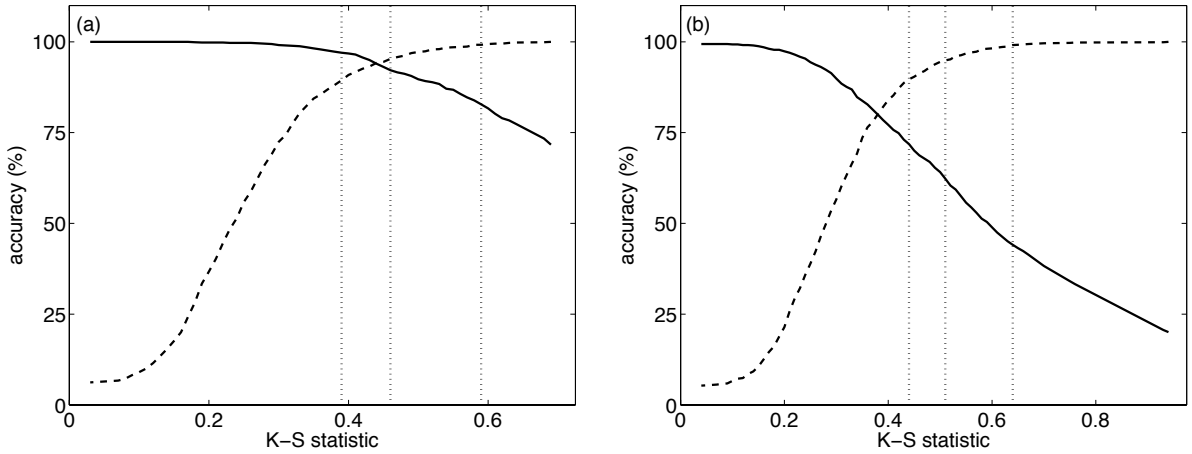


Figure 19: Shown are ROC curves for (a): a tampered region of size 150×150 and a quality difference of 15; and (b) a tampered region of size 100×100 and a quality difference of 10. The solid curve corresponds to the accuracy of detecting the tampered region, and the dashed curve corresponds to the accuracy of correctly classifying an authentic image. The vertical dotted lines denote (from left to right) false positive rates of 10%, 5%, and 1%. See also Table 2.

small regions that have been altered. The disadvantage of this approach is that it is only effective when the tampered region is of lower quality than the image into which it was inserted.

3 Discussion

Today's technology allows digital media to be altered and manipulated in ways that were simply impossible twenty years ago. Tomorrow's technology will almost certainly allow for us to manipulate digital media in ways that today seem unimaginable. And as this technology continues to evolve it will become increasingly more important for the science of digital forensics to try to keep pace.

There is little doubt that as we continue to develop techniques for exposing photographic frauds, new techniques will be developed to make better and harder to detect fakes. And while some of the forensic tools may be easier to fool than others, some tools will be difficult for the average user to circumvent. For example, three of the techniques described here leverage complex and subtle lighting and geometric properties of the image formation process that are non-trivial to correct in a standard photo-editing software. As with the spam/anti-spam and virus/anti-virus game, an arms race between the forger and forensic analyst is somewhat inevitable. The field of image forensics however has and will continue to make it harder and more time consuming (but never impossible) to create a forgery that cannot be detected. It is my hope that this new technology, along with awareness and sensible policy and law, will help the media, the courts, and our society contend with this exciting and at times puzzling digital age.

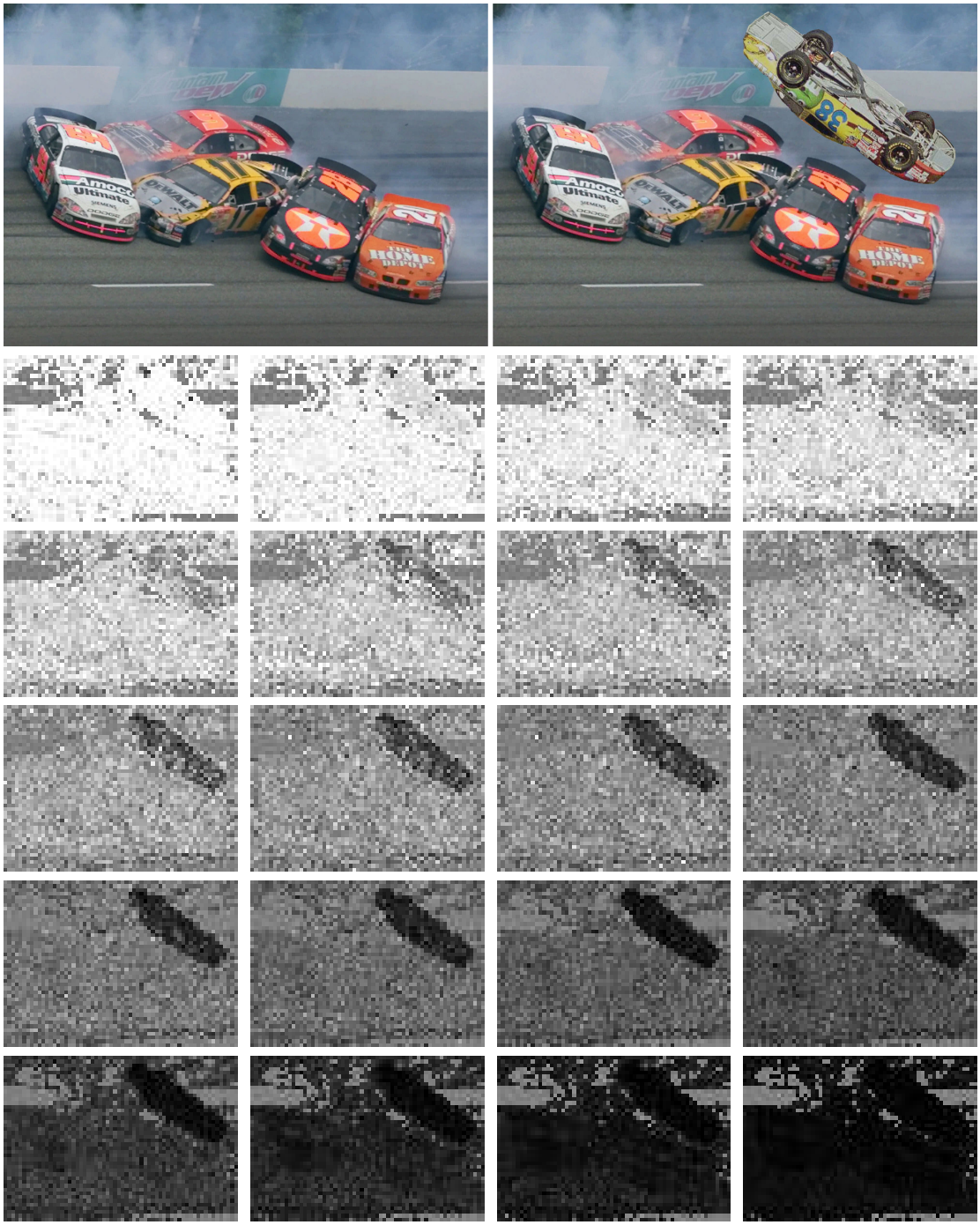


Figure 20: Shown are the original (left) and doctored (right) image. Shown below are the difference images at qualities 60 through 98 in steps of 2.

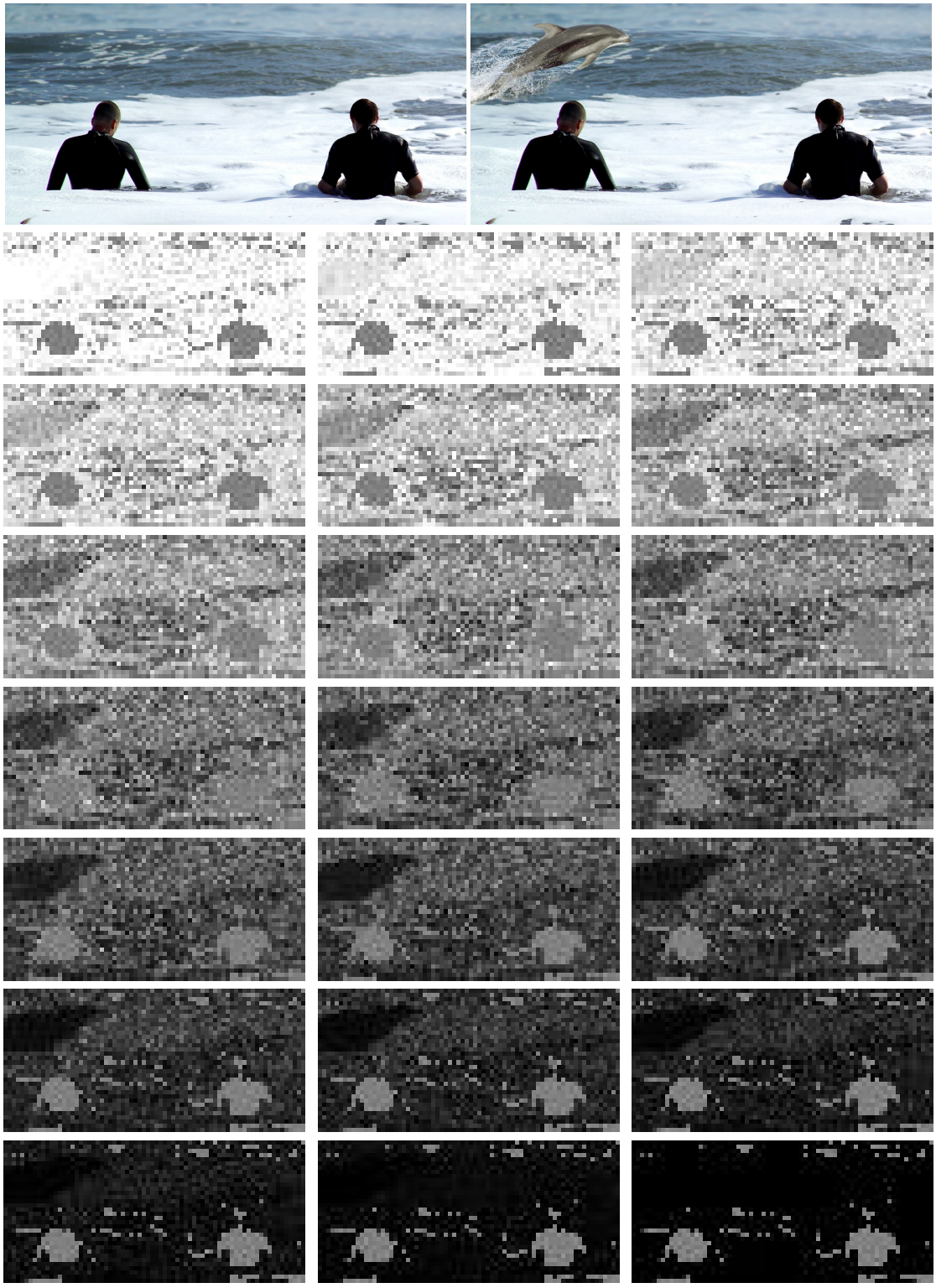


Figure 21: Shown are the original (left) and doctored (right) image. Shown below are the difference images at qualities 60 through 100 in steps of 2.

Acknowledgment

Thanks to my students and colleagues with whom I have worked over the years to develop these and other digital forensic methods, in particular Micah K. Johnson, Eric Kee, Siwei Lyu, Alin Popescu, Weihong Wang and Jeffrey Woodward. This work was supported by a gift from Adobe Systems, Inc., a gift from Microsoft, Inc., a grant from the National Science Foundation (CNS-0708209), a grant from the U.S. Air Force (FA8750-06-C-0011), and by the Institute for Security Technology Studies at Dartmouth College under grants from the Bureau of Justice Assistance (2005-DD-BX-1091) and the U.S. Department of Homeland Security (2006-CS-001-000001). Points of view or opinions in this document are those of the author and do not represent the official position or policies of the U.S. Department of Justice, the U.S. Department of Homeland Security, or any other sponsor.

About the Author

Hany Farid received his undergraduate degree in Computer Science and Applied Mathematics from the University of Rochester in 1989. He received his Ph.D. in Computer Science from the University of Pennsylvania in 1997. Following a two year post-doctoral position in Brain and Cognitive Sciences at MIT, he joined the Dartmouth faculty in 1999. Hany is the David T. McLaughlin Distinguished Professor of Computer Science and Associate Chair of Computer Science. He is also affiliated with the Institute for Security Technology Studies at Dartmouth. Hany is the recipient of an NSF CAREER award, a Sloan Fellowship and a Guggenheim Fellowship. Hany can be reached at farid@cs.dartmouth.edu, and more information about his work can be found at `www.cs.dartmouth.edu/farid`.

References

[1] Digital compression and coding of continuous-tone still images, Part 1: Requirements and guidelines. ISO/IEC JTC1 Draft International Standard 10918-1, 1991.

[2] R. Basri and D.W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern and Machine Intelligence*, 25(2):218–233, 2003.

[3] B. E. Bayer. Color imaging array. US Patent, 3971065, 1976.

[4] R. Brunelli. Estimation of pose and illuminant direction for face processing. *Image and Vision Computing*, 15(10):741–748, 1997.

[5] E. Chang, S. Cheung, and D. Y. Pan. Color filter array recovery using a threshold-based variable number of gradients. In N. Sampat and T. Yeh, editors, *Sensors, Cameras, and Applications for Digital Photography, Proceedings of the SPIE*, volume 3650, pages 36–43, March 1999.

[6] D. R. Cok. Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. US Patent, 4642678, 1987.

[7] W.J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, 1980.

[8] I.J. Cox, M.L. Miller, and J.A. Bloom. *Digital Watermarking*. Morgan Kaufmann Publishers, 2002.

[9] S.A. Craver, M. Wu, B. Liu, A. Stubblefield, B. Swartzlander, and D.S. Wallach. Reading between the lines: Lessons from the SDMI challenge. In *10th USENIX Security Symposium*, Washington DC, 2001.

[10] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, 1997.

[11] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 99(1):1–38, 1977.

[12] R.O. Dror, A.S. Willsky, and E.H. Adelson. Statistical characterization of real-world illumination. *Journal of Vision*, 4(9):821–837, 2004.

[13] H. Farid. Digital image ballistics from JPEG quantization. Technical Report TR2006-583, Department of Computer Science, Dartmouth College, 2006.

[14] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, Inc., 2nd edition, 1993.

[15] W. T. Freeman. Median filter for reconstructing missing color samples. US Patent, 4724395, 1988.

[16] G.H. Golub, P.C. Hansen, and D.P. O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

[17] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11(9):997–1013, September 2002.

[18] J. F. Hamilton and J. E. Adams. Adaptive color plan interpolation in single sensor color electronic camera. US Patent, 5629734, 1997.

[19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[20] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 669–672, September 2003.

[21] M.J. Hogan, J.A. Alvarado, and J.E. Weddell. *Histology of the Human Eye*. W.B Saunders Company, 1971.

[22] http://www.flickr.com.

[23] D.R Iskander. A parametric approach to measuring limbus corneae from digital images. *IEEE Transactions on Biomedical Engineering*, 53(6):1134–1140, June 2006.

[24] M.K. Johnson and H. Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *ACM Multimedia and Security Workshop*, New York, NY, 2005.

[25] M.K. Johnson and H. Farid. Exposing digital forgeries in complex lighting environments. *IEEE Transactions on Information Forensics and Security*, 3(2):450–461, 2007.

[26] M.K. Johnson and H. Farid. Exposing digital forgeries through specular highlights on the eye. In *9th International Workshop on Information Hiding*, Saint Malo, France, 2007.

[27] S. Katzenbeisser and F.A.P. Petitcolas. *Information Techniques for Steganography and Digital Watermarking*. Artec House, 2000.

[28] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29(6):1153–1160, December 1981.

[29] C. A. Laroche and M. A. Prescott. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. US Patent, 5373322, 1994.

[30] A. Lefohn, R. Caruso, E. Reinhard, B. Budge, and P. Shirley. An ocularist's approach to human iris synthesis. *IEEE Computer Graphics and Applications*, 23(6):70–75, 2003.

[31] D. D. Muresan and T. W. Parks. Adaptively quadratic (AQua) image interpolation. *IEEE Transactions on Image Processing*, 13(5):690–698, May 2004.

[32] P. Nillius and J.O. Eklundh. Automatic estimation of the projected light source direction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[33] K. Nishino and S.K. Nayar. Eyes for relighting. *ACM Transactions on Graphics*, 23(3):704–711, 2004.

[34] Y. Ostrovsky, P. Cavanagh, and P. Sinha. Perceiving illumination inconsistencies in scenes. Technical Report AI Memo 2001-029, Massachusetts Institute of Technology, 2001.

[35] H. Pearson. Image manipulation: CSI: Cell biology. *Nature*, 434:952–953, 2005.

[36] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2004.

[37] J.M. Pinel, H. Nicolas, and C. Le Bris. Estimation of 2D illuminant direction and shadow segmentation in natural video sequences. In *Proceedings of VLBV*, pages 197–202, 2001.

[38] A.C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.

[39] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *Journal of the Optical Society of America A*, 18:2448–2559, 2001.

[40] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander III. Demosaicking methods for Bayer color arrays. *Journal of Electronic Imaging*, 11(3):306–315, July 2002.

[41] A. Ruszczyński. *Nonlinear Optimization*. Princeton University Press, 2006.

[42] G. Schaefer and M. Stich. UCID - an uncompressed colour image database. Technical report, School of Computing and Mathematics, Nottingham Trent University, U.K., 2003.

[43] P. Sinha. Perceiving illumination inconsistencies. In *Investigative Ophthalmology and Visual Science*, volume 41/4:1192, 2000.

[44] G.K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 34(4):30–44, 1991.