

# Lighting and Optical Tools for Image Forensics

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Micah Kimo Johnson

DARTMOUTH COLLEGE

Hanover, New Hampshire

September 21, 2007

Examining Committee:

---

(chair) Hany Farid, Ph.D.

---

Jessica Fridrich, Ph.D.

---

Fabio Pellacini, Ph.D.

---

Peter Winkler, Ph.D.

---

Charles Barlowe, Ph.D.

Dean of Graduate Studies



## **Abstract**

We present new forensic tools that are capable of detecting traces of tampering in digital images without the use of watermarks or specialized hardware. These tools operate under the assumption that images contain natural properties from a variety of sources, including the world, the lens, and the sensor. These properties may be disturbed by digital tampering and by measuring them we can expose the forgery. In this context, we present the following forensic tools: (1) illuminant direction, (2) specularities, (3) lighting environment, and (4) chromatic aberration. The common theme of these tools is that they exploit lighting or optical properties of images. Although each tool is not applicable to every image, they add to a growing set of image forensic tools that together will complicate the process of making a convincing forgery.

## Acknowledgments

First, I would like to thank my advisor Hany Farid. It is because of him that I finished this Ph.D., but he also deserves credit for helping me begin it. He was instrumental in initially accepting me into the program despite my nontraditional background. Through his advice and guidance, I learned to take the skills I had obtained from years in the classroom and apply them to real-world problems. He exposed me to new research areas and gave me the freedom to find the solutions on my own. His influence can be seen throughout this work.

I would also like to thank my committee members Jessica Fridrich, Fabio Pellacini and Peter Winkler. Their comments and suggestions brought up directions I had not considered and helped me refine my thinking about the decisions I had made.

Beyond my committee, I would like to acknowledge some of the people in the Computer Science department. I am grateful for the many excellent professors—I fully enjoyed the courses I took from Javed Aslam, Tom Cormen, Prasad Jayanti, Dan Rockmore and Sean Smith. I feel I am leaving Dartmouth with a strong and broad background in computer science. The staff and sysadmins, including Sandy Brash, Kelly Clark, Wayne Cripps, and Tim Tregubov, kept everything running smoothly and I appreciate the work they do every day that is often taken for granted. I would also like to acknowledge past and current members of the image science group: Siwei Lyu, Alin Popescu, Weihong Wang and Jeff Woodward. I certainly learned as much from our conversations as I did reading countless papers and books. And a special thanks goes to Elena Davidson and John Thomas for dragging me out of my office several times a week for “pizza day.”

I would like to thank my family and in-laws for supporting me and for providing a bit of perspective. When things at school were busy, it was nice to have a life outside of the department to remind me that there are important things beyond the laptop screen.

Finally, I would like to thank my wife Amity for her patience and understanding through the years of graduate-student life. Her unconditional love and support is truly appreciated.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Forgeries . . . . .	1
1.2	Watermarking . . . . .	4
1.3	Forensics . . . . .	4
1.4	Contributions . . . . .	6
<b>2</b>	<b>Illuminant direction</b>	<b>7</b>
2.1	Methods . . . . .	7
2.1.1	Infinite light source (3-D) . . . . .	7
2.1.2	Infinite light source (2-D) . . . . .	9
2.1.3	Local light source (2-D) . . . . .	12
2.1.4	Multiple light sources . . . . .	15
2.2	Results . . . . .	16
2.2.1	Infinite light source . . . . .	17
2.2.2	Local light source . . . . .	17
2.2.3	Multiple light sources . . . . .	20
2.2.4	Sensitivity . . . . .	20
2.2.5	Forgeries . . . . .	22
2.3	Discussion . . . . .	24
<b>3</b>	<b>Specularity</b>	<b>25</b>
3.1	Methods . . . . .	25
3.1.1	Camera calibration . . . . .	27
3.1.2	View direction . . . . .	30
3.1.3	Surface normal . . . . .	30
3.1.4	Light direction . . . . .	32
3.1.5	Consistency of estimates . . . . .	32
3.2	Results . . . . .	34
3.2.1	Synthetic images . . . . .	34
3.2.2	Real images: controlled lighting . . . . .	36
3.2.3	Real images: unknown lighting . . . . .	38

3.2.4	Sensitivity . . . . .	39
3.2.5	Forgeries . . . . .	40
3.3	Discussion . . . . .	42
<b>4</b>	<b>Lighting environment</b>	<b>44</b>
4.1	Methods . . . . .	44
4.1.1	Representing lighting environments . . . . .	44
4.1.2	From irradiance to intensity . . . . .	48
4.1.3	Estimating lighting environments . . . . .	49
4.1.4	Comparing lighting environments . . . . .	51
4.2	Results . . . . .	54
4.2.1	Simulation . . . . .	54
4.2.2	Spheres . . . . .	55
4.2.3	Photographs . . . . .	56
4.2.4	Sensitivity . . . . .	58
4.2.5	Forgeries . . . . .	60
4.3	Discussion . . . . .	61
<b>5</b>	<b>Chromatic aberration</b>	<b>64</b>
5.1	Methods . . . . .	64
5.1.1	2-D Aberration . . . . .	66
5.1.2	Estimating Chromatic Aberration . . . . .	67
5.2	Results . . . . .	69
5.2.1	Synthetic images . . . . .	69
5.2.2	Calibrated images . . . . .	70
5.2.3	Forgeries . . . . .	71
5.3	Discussion . . . . .	74
<b>6</b>	<b>Discussion</b>	<b>76</b>
<b>A</b>	<b>Curve fitting</b>	<b>79</b>
A.1	Minimization . . . . .	79
A.2	Affine transforms . . . . .	81
A.2.1	Error function . . . . .	82
A.3	Planar projective transforms . . . . .	86
A.3.1	Error function . . . . .	87
A.4	Constraints . . . . .	90
A.5	Multiple curves . . . . .	91
A.6	Examples . . . . .	91

# Chapter 1

## Introduction

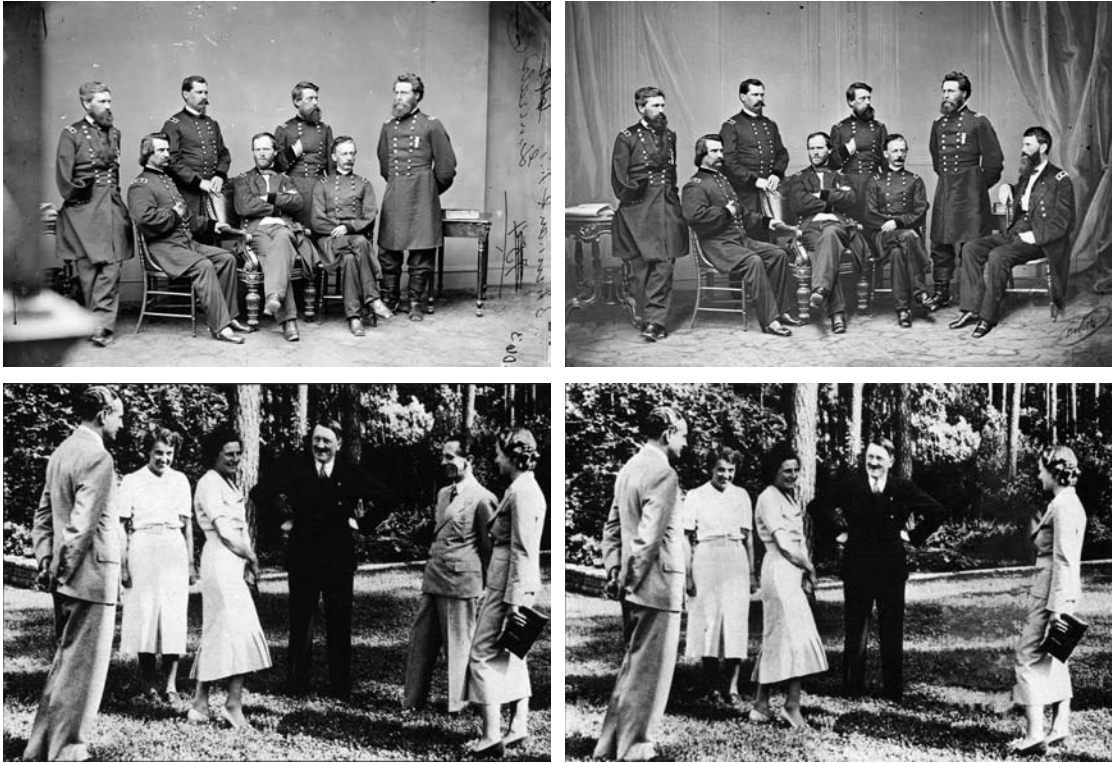
Digital images are everywhere: on the covers of magazines, in newspapers, in courtrooms, and all over the internet. We are exposed to them throughout the day and most of the time, we trust what we see. But given the ease with which images can be manipulated, we need to be aware that seeing is not always believing.

In recent years, tampered images have affected science, law, politics, the media, and business. Some cases have made national and international headlines, tarnishing the public's perception of images. While forgeries are not a new problem, the tools for making forgeries, such as digital cameras, computers, and software, have increased in sophistication, bringing the ability to make forgeries out of the hands of specialists to anyone. The tools for detecting forgeries, on the other hand, are only beginning to be developed. There is a clear need for these tools if the public is to regain trust in published images.

### 1.1 Forgeries

The art of making an image forgery is as old as photography itself. In its early years, photography quickly became the chosen method for making portraits, and portrait photographers learned that they could improve sales by retouching their photographs to please the sitter [5]. During the Civil War, many photos were retouched with additional details for dramatic effect. The photographers of the era also experimented with compositing, i.e., combining multiple images into one. An early example of compositing appears in the top panel of Figure 1.1. The general on the far right, General Francis P. Blair, was not present in the original photograph (left), but is present in a version available from the Library of Congress (right). There are many more examples from the early years of photography, and in most cases, the forgeries were made either to enhance insufficient details or for humorous effects; they were not designed to deceive. By the early to mid 20<sup>th</sup> century, however, photographers found that image forgeries could be powerful tools for changing public perception and even history.

Nazi Germany is famous for its propaganda and there are many examples of image manipulation with the deliberate intention to deceive. In the bottom panel of Figure 1.1, is an image forgery (right) where Joseph Goebbels, Hitler's propaganda minister, was removed from the original image



**Figure 1.1:** Top: A forgery showing General Sherman posing with his generals before (left) and after (right) manipulation—General Blair was added to the original photograph. Bottom: A forgery showing Hitler with several people before (left) and after (right) manipulation—Joseph Goebbels, Hitler’s minister of propaganda, was removed from the original photograph.

(left) [28]. There are similar examples from Soviet Russia and the United States where unfavorable people were removed from images, or where people were added to images for political reasons. Despite countless examples from history to the contrary, many still believe the old adage “the camera never lies.”

More recently, there have been numerous examples of tampered images in newspapers and on magazine covers. Figure 1.2, for example, shows covers from three popular magazines where the images have been manipulated. The first example, from *New York* magazine, is perhaps the least believable and to its credit the following disclaimer appears on the cover: “Note: This is a manipulated image. They’re not actually crazy enough to pose for a picture like this.” The next two images were more controversial for two reasons: the images were more believable and the disclaimer was found not on the cover, but on a page within the magazine.<sup>1</sup> To make matters worse, *Newsweek* is considered by many to be a trustworthy source of news and the public was shocked to learn they were using techniques similar to *Star*. While these images might tarnish the public opinion of a celebrity, cases involving manipulated images with more serious implications have

<sup>1</sup>*Newsweek* refers to the image of Martha Stewart as a “photo illustration” and *Star* refers to the image of Brad Pitt and Angelina Jolie as a “composite of two photographs.”





**Figure 1.2:** Manipulated images appearing on the covers of popular magazines. From left to right: *New York* from July 25, 2005; *Newsweek* from March 7, 2005; and *Star* from May 2005.

arisen in science and law.

In 2004, a team lead by Korean scientist Dr. Hwang Woo-Suk published groundbreaking results in stem cell research in the journal *Science*. Their results showed the successful production of stem cells from 11 patients, offering hope that new cures for diseases were around the corner. But other researchers began to find flaws in their work and by late 2005, one of the co-authors of the paper admitted that photographs in the paper had been doctored [26, 31, 56]. Hwang soon retracted the *Science* paper and resigned from his position at Seoul National University. After this scandal, other journals realized the importance of investigating images in submitted papers. The editors of the *Journal of Cell Biology* have been testing images since 2002 and they estimate that 25 percent of accepted manuscripts have images that are modified beyond their standards, while one percent contain fraudulent images [8].

In law, the Child Pornography Prevention Act of 1996 (CPPA) outlawed virtual child pornography, i.e., images that appear to depict minors engaged in sexual acts but were created by computer or by other means. In 2002, the United States Supreme Court declared the CPPA to be in violation of the First Amendment. Their decision was based on the fact that no children are directly harmed in the production of virtual child pornography, and therefore, such images are protected under the right to freedom of speech. An unfortunate side-effect of this ruling is that people accused of producing child pornography can claim that the images are computer-generated; the burden of proving the images are real, a non-trivial problem, is on the prosecution [13].

In all of these examples, the authenticity of images is in question. How are we to prove that images are authentic, or similarly, how can we prove that images have been modified or are computer-generated? There is a need for technology to address this problem and current solutions typically fall in one of two categories: watermarking or forensics.

## 1.2 Watermarking

One solution to image authentication problem is digital watermarking [9, 30]. The idea of digital watermarking is to embed information into an image that can be extracted later to verify authenticity. Watermarking requires specialized cameras, such as the Canon EOS-1D Mark II or the Nikon D2Xs. Both cameras generate an image-specific digest and bundle it with the image at the time of recording. The image can be authenticated at a later date by regenerating a digest and checking against the original; a difference indicates that the image was modified since recording. While these cameras could be useful in some settings, such as law enforcement, the limitations are significant. The most obvious limitation is that currently only a few cameras, and typically the expensive models, have this feature. But further, these systems do not allow modifications to an image, including modifications that could improve the image, such as sharpening or enhancing contrast.

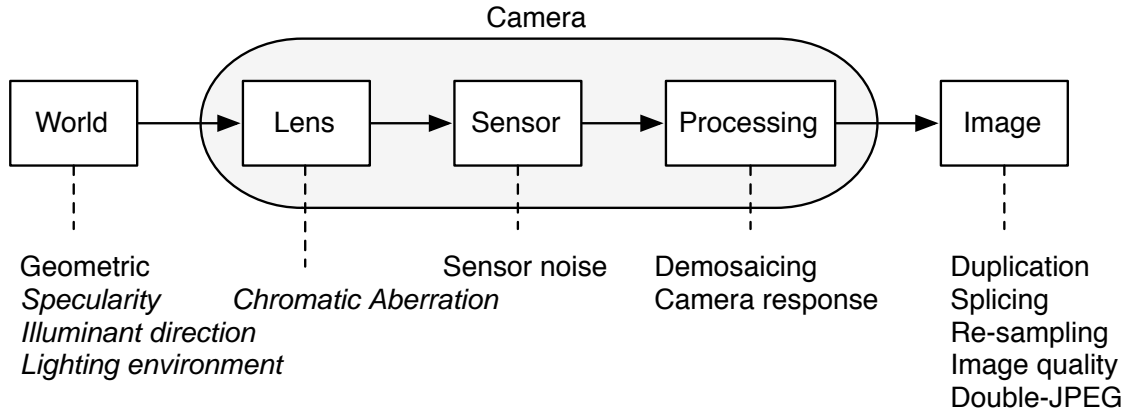
There are many other watermarking schemes, some designed to permit modifications and others designed to reveal modifications if they have occurred [20, 32, 34, 61]. For example, semi-fragile watermarks allow for simple modifications to an image, such as JPEG compression, while tell-tale watermarks can be analyzed to reveal possible tampering. All watermarking schemes, however, require specialized hardware or software to embed the watermark in the image and it is unlikely that all camera manufacturers will agree to include watermarking technology in every camera they make. Digital watermarking is therefore limited to problem domains where the make and model of camera can be controlled.

## 1.3 Forensics

Over the last few years, there has been a growing body of work on tools for digital image forensics. These tools are capable of detecting tampering in images from any camera, without relying on watermarks or specialized hardware. Instead of watermarks, these tools assume that images possess certain regularities that are disturbed by tampering. These regularities can come from a variety of sources, including the world, the camera, or the image itself, Figure 1.3. The common approach taken by these tools is to measure the regularities and detect differences in the measurements. Most of the current forensic tools target specific types of tampering since a single manipulation may disturb only some of the regularities. While there is no single tool that can detect all types of tampering, the current tools can detect many common manipulations. These tools together are a powerful way to detect forgeries.

One of the most basic image manipulations is copy-move or cloning. This manipulation is necessary if a forger needs to cover part of an image and it can be successful if a homogeneous texture is available (e.g., grass, sand, or water). Although different regions of a homogeneous texture may look similar qualitatively, it is highly unlikely that they will be exactly the same numerically. Two different forensic tools exploit this basic observation to detect cloning [19, 44].

Another basic image manipulation is splicing, otherwise known as photomontage. For this manipulation, a forger combines regions from different images into a single image. One technique



**Figure 1.3:** Sources of regularities in the imaging process and current forensic tools that exploit these regularities. The tools printed in italics constitute this thesis.

for detecting splicing searches for the presence of abrupt discontinuities in the image [37]. Several other techniques use estimates of the camera response function from different regions of an image to detect splicing and possibly other manipulations [25, 35, 45].

Often, a forger may need to resize or rotate regions of an image. These manipulations generally involve re-sampling the image data onto a new lattice. This process introduces statistical correlations, which are detectable under certain conditions [46]. Another approach uses image quality metrics detect re-sampling and other common image-processing operations [2]. In addition, if the forger needs to save the forgery as a JPEG image and it was originally captured in JPEG format, the resulting image will have been double-JPEG compressed. Double-JPEG compression also introduces statistical correlations which can be detected for image forensics [45].

The sensor in a digital camera has been exploited to detect tampering. A typical sensor only captures one of the three color channels at each pixel. To create RGB values for each pixel, the missing color channels are interpolated from neighboring pixels using a demosaicing algorithm. As with re-sampling and double-JPEG compression, demosaicing introduces statistical correlations, which are detectable and useful for forensics [47]. The unique noise patterns of digital sensors are also useful for forensics. These noise patterns are similar to a digital fingerprint for a particular camera, and they can be estimated from a collection of images taken by the same camera [18, 36]. Once the noise has been estimated, it can be used for camera identification and forgery detection.

Finally, though typically used for robot navigation or 3-D modeling, geometric techniques can be useful for image forensics. When known geometries are present in a scene (e.g., circles, lines, rectangles), they can be used to make measurements under perspective projection [10, 29]. Two example uses are measuring the height of a person in an image relative to an object of known length or measuring the distance between objects on the same plane.

Most of the current forensic tools have focused on regularities from sources that are inherently digital, e.g., the sensor and quantization. These sources occur on the right side of Figure 1.3. But,

the imaging process introduces regularities from non-digital sources as well: the world and the lens. Although these sources are rich with regularities, few computational tools exist for exploiting these regularities for image forensics.

## 1.4 Contributions

In this thesis, we present four new tools for image forensics. These tools measure regularities in images that arise from the world and the lens, i.e., sources on the left side of Figure 1.3. For each tool, we describe the conditions under which it is applicable, give a physical model for the property being analyzed, provide a method for estimating the property from a single image, and demonstrate results on real images and forgeries. In this context, we present the following four tools:

1. **Illuminant direction.** When creating a digital composite of, for example, two people standing side-by-side, it is often difficult to match the lighting conditions from the individual photographs. Lighting inconsistencies can therefore be a useful tool for revealing traces of digital tampering. The illuminant direction tool estimates the direction to the light source from several objects in an image; widely varying estimates are evidence of tampering.
2. **Specularity.** Human eyes are reflective and provide information about the lighting environment under which a person was photographed. The specularity tool estimates a 3-D direction to the light source from a specular highlight on the eye; strong inconsistencies in estimates from different highlights across the image are evidence of tampering.
3. **Lighting environment.** Although the lighting of a scene can be arbitrarily complex, the appearance of a diffuse object in any scene is well represented by a low-dimensional model. The lighting environment tool estimates parameters of a low-dimensional model of lighting and is applicable to more complex lighting environments than the illuminant direction or specularity tools. As with the other lighting tools, inconsistencies in estimates across the image are evidence of tampering.
4. **Chromatic Aberration.** Chromatic aberration results from the failure of an optical system to perfectly focus light of different wavelengths. When tampering with an image, this aberration is often disturbed and fails to be consistent across the image. Large inconsistencies in estimates of chromatic aberration from different parts of an image are evidence of tampering.

Although tampering with images is not a new phenomenon, the availability of digital image technology and image processing software makes it easy for anyone to make a forgery. From internet hoaxes, to fake magazine covers, to manipulated scientific results, these images can have a profound effect on society. While each tool in this thesis targets a specific type of tampering, they add to a growing set of image forensic tools that together will detect a wide variety of forgeries.

## Chapter 2

# Illuminant direction

Consider the creation of a forgery showing two celebrities, rumored to be romantically involved, walking down a sunset beach. Such an image might be created by splicing together individual images of each celebrity. In doing so, it is often difficult to match the lighting effects due to directional lighting (e.g., the sun on a clear day). Therefore, differences in lighting can be a telltale sign of digital tampering. Shown in Figure 2.1, for example, is a composite image where the two people were originally photographed with the light in significantly different positions. While this particular forgery is fairly obvious, more subtle differences in light direction may be harder to detect by simple visual inspection [40, 52].

To the extent that the direction to the light source can be estimated from different objects or people in an image, inconsistencies in these estimates can be used as evidence of digital tampering. In this chapter, we describe a technique for estimating the light direction from a single image, and show its efficacy in real-world settings.

### 2.1 Methods

The general problem of estimating the illuminant direction has been widely studied in the field of computer vision (e.g., [7, 38, 41]). In this section, we define the general problem, review a standard solution and then show how some additional simplifying assumptions make the problem more tractable. We then extend this solution to provide for a more effective and broadly applicable forensic tool.

#### 2.1.1 Infinite light source (3-D)

The standard approaches for estimating light direction begin by making some simplifying assumptions about the surface of interest: (1) it is Lambertian (i.e., it reflects light isotropically); (2) it has a constant reflectance value; (3) it is illuminated by a point light source infinitely far away; and (4) the angle between the surface normal and the light direction<sup>1</sup> is in the range  $0^\circ$  to  $90^\circ$ . Under these

---

<sup>1</sup>The assumption that the angle between the surface and light is bounded between  $0^\circ$  to  $90^\circ$  can be relaxed by replacing  $(\vec{N}(x, y) \cdot \vec{L})$  in Equation (2.1) with  $\max(\vec{N}(x, y) \cdot \vec{L}, 0)$ , which is not used here to avoid the non-linear max operator.



**Figure 2.1:** A digital composite of celebrities Cher and Brad Pitt. Note that Cher was originally photographed with a diffuse non-directional light source, whereas Brad Pitt was photographed with a directional light positioned to his left.

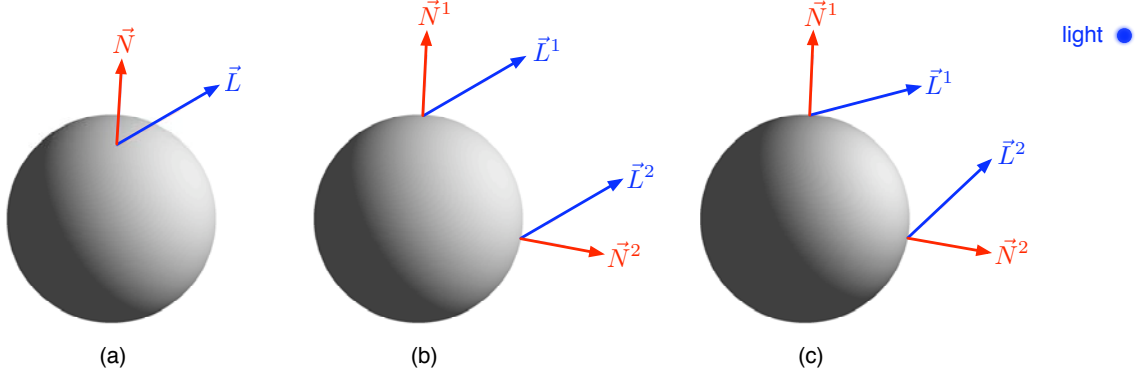
assumptions, the image intensity can be expressed as:

$$I(\vec{x}) = R(\vec{N}(\vec{x}) \cdot \vec{L}) + A, \quad (2.1)$$

where  $R$  is the constant reflectance value,  $\vec{L}$  is a 3-vector pointing towards the light source,  $\vec{N}(\vec{x})$  is a 3-vector representing the surface normal at the point  $\vec{x}$ , and  $A$  is a constant ambient light term [17], Figure 2.2(a). If we are only interested in the direction to 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 direction and ambient term can be estimated using least-squares. To begin, a quadratic error function embodying the imaging model of Equation (2.1) is given by:

$$\begin{aligned} E(\vec{L}, A) &= \left\| M \begin{pmatrix} L_x \\ L_y \\ L_z \\ A \end{pmatrix} - \begin{pmatrix} I(\vec{x}_1) \\ I(\vec{x}_2) \\ \vdots \\ I(\vec{x}_p) \end{pmatrix} \right\|^2, \\ &= \left\| M\vec{v} - \vec{b} \right\|^2, \end{aligned} \quad (2.2)$$



**Figure 2.2:** Diagram of the imaging geometry for (a) an infinite light source (3-D); (b) an infinite light source (2-D); and (c) a local light source (2-D). In the 2-D cases, the  $z$ -component of the surface normal  $\vec{N}$  is zero. (c) For a local light source, the direction to the light source  $\vec{L}$  varies across the sphere's surface.

where  $L_x$ ,  $L_y$ , and  $L_z$  denote the components of the light direction  $\vec{L}$ , and:

$$M = \begin{pmatrix} N_x(\vec{x}_1) & N_y(\vec{x}_1) & N_z(\vec{x}_1) & 1 \\ N_x(\vec{x}_2) & N_y(\vec{x}_2) & N_z(\vec{x}_2) & 1 \\ \vdots & \vdots & \vdots & \vdots \\ N_x(\vec{x}_p) & N_y(\vec{x}_p) & N_z(\vec{x}_p) & 1 \end{pmatrix}, \quad (2.3)$$

where  $N_x(\vec{x}_i)$ ,  $N_y(\vec{x}_i)$ , and  $N_z(\vec{x}_i)$  denote the components of the surface normal  $\vec{N}$  at the point  $\vec{x}_i$ . The quadratic error function in Equation (2.2) is minimized by differentiating with respect to the unknown vector  $\vec{v}$ , setting the result equal to zero, and solving for  $\vec{v}$  to yield the least-squares estimate:

$$\vec{v} = (M^T M)^{-1} M^T \vec{b}. \quad (2.4)$$

Note that this solution requires knowledge of 3-D surface normals from at least four distinct points on the surface of an object ( $p \geq 4$ ). With only a single image and no objects of known geometry in the scene, it is unlikely that this will be possible. To overcome this problem, most approaches acquire multiple images [43] or place an object of known geometry in the scene (e.g., a sphere) [6]. For forensic applications, these solutions are not practical.

### 2.1.2 Infinite light source (2-D)

In [38], the authors suggest a clever solution for estimating two components of the light direction ( $L_x$  and  $L_y$ ) from only a single image. While their approach clearly provides less information regarding the light direction, it does make the problem tractable from a single image. The authors note that, under an assumption of orthographic projection, the  $z$ -component of the surface normal is zero,  $N_z = 0$ , along the occluding boundary of a surface. In addition, the  $x$ - and  $y$ -components

of the surface normal,  $N_x$  and  $N_y$ , can be estimated directly from the image, Figure 2.2(b).

With this assumption, the error function of Equation (2.2) takes the form:

$$\begin{aligned} E(\vec{L}, A) &= \left\| M \begin{pmatrix} L_x \\ L_y \\ A \end{pmatrix} - \begin{pmatrix} I(\vec{x}_1) \\ I(\vec{x}_2) \\ \vdots \\ I(\vec{x}_p) \end{pmatrix} \right\|^2, \\ &= \left\| M\vec{v} - \vec{b} \right\|^2, \end{aligned} \quad (2.5)$$

where:

$$M = \begin{pmatrix} N_x(\vec{x}_1) & N_y(\vec{x}_1) & 1 \\ N_x(\vec{x}_2) & N_y(\vec{x}_2) & 1 \\ \vdots & \vdots & \vdots \\ N_x(\vec{x}_p) & N_y(\vec{x}_p) & 1 \end{pmatrix}. \quad (2.6)$$

As before, this error function is minimized using standard least-squares to yield the same solution as in Equation (2.4), but with the matrix  $M$  taking the form given in Equation (2.6). In this case, the solution requires knowledge of 2-D surface normals from at least three distinct points ( $p \geq 3$ ) along the boundary of an object with constant reflectance.

The intensity,  $I(\vec{x}_i)$ , at a boundary point  $\vec{x}_i$  cannot be directly measured from the image as the surface is occluded. The authors in [38] 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 (see section 2.2 for a more detailed description).

We extend this basic formulation in three ways. First, we estimate the two-dimensional light direction from local patches along an object's boundary (as opposed to along extended boundaries as in [38]). This is done to relax the assumption that the reflectance across the entire surface is constant. Next, we introduce a regularization (smoothness) term to better condition the final estimate of the light direction. Finally, this formulation is extended to accommodate a local directional light source (e.g., a desk lamp).

### Relaxing the constant reflectance assumption

To relax the constant reflectance assumption, we assume that the reflectance for a local surface patch is constant (as opposed to the entire surface). This requires us to estimate individual light directions,  $\vec{L}^i$ , for each patch along a surface. Because we have assumed an infinite light source, these light direction estimates should be parallel, though their magnitudes may vary; recall that the light direction estimate is only within a scale factor that depends on the reflectance value  $R$ , Equation (2.1).



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 (2.1):

$$\begin{aligned}
 E_1(\vec{L}^1, \dots, \vec{L}^n, A) &= \left\| M \begin{pmatrix} L_x^1 \\ L_y^1 \\ \vdots \\ L_x^n \\ L_y^n \\ A \end{pmatrix} - \begin{pmatrix} I(\vec{x}_1^1) \\ \vdots \\ I(\vec{x}_p^1) \\ \vdots \\ I(\vec{x}_1^n) \\ \vdots \\ I(\vec{x}_p^n) \end{pmatrix} \right\|^2, \\
 &= \left\| M\vec{v} - \vec{b} \right\|^2,
 \end{aligned} \tag{2.7}$$

where:

$$M = \begin{pmatrix} N_x(\vec{x}_1^1) & N_y(\vec{x}_1^1) & 0 & 0 & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ N_x(\vec{x}_p^1) & N_y(\vec{x}_p^1) & 0 & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & N_x(\vec{x}_1^n) & N_y(\vec{x}_1^n) & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & N_x(\vec{x}_p^n) & N_y(\vec{x}_p^n) & 1 \end{pmatrix}. \tag{2.8}$$

As before, the above quadratic error function is minimized using least-squares with the solution taking on the same form as in Equation (2.4). In this case, the solution provides  $n$  estimates of the 2-D light directions,  $\vec{L}^1, \dots, \vec{L}^n$ , and an ambient term  $A$ . Note that while individual light directions are estimated for each surface patch, a single ambient term is assumed.

While the local estimation of light directions allows for the relaxation of the constant reflectance assumption, it could potentially yield less stable results. 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 light direction estimates small), we can condition the individual estimates with the following regularization term:

$$E_2(\vec{L}^1, \dots, \vec{L}^n) = \sum_{i=2}^n \left\| \vec{L}^i - \vec{L}^{i-1} \right\|^2. \tag{2.9}$$

This additional error term penalizes neighboring estimates that are different from one another. The quadratic error function  $E_1(\cdot)$ , Equation (2.7), is conditioned by combining it with the regularization

term  $E_2(\cdot)$ , scaled by a factor  $\lambda$ , to yield the final error function:

$$E(\vec{L}^1, \dots, \vec{L}^n, A) = E_1(\vec{L}^1, \dots, \vec{L}^n, A) + \lambda E_2(\vec{L}^1, \dots, \vec{L}^n). \quad (2.10)$$

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:

$$E_2(\vec{v}) = \|C\vec{v}\|^2, \quad (2.11)$$

where  $\vec{v} = (L_x^1 \ L_y^1 \ L_x^2 \ L_y^2 \ \dots \ L_x^n \ L_y^n \ A)^T$  and where the  $2n - 2 \times 2n + 1$  matrix  $C$  is given by:

$$C = \begin{pmatrix} -1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & \ddots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 0 & 1 & 0 \end{pmatrix}. \quad (2.12)$$

The error function of Equation (2.10) then takes the form:

$$E(\vec{v}) = \|M\vec{v} - \vec{b}\|^2 + \lambda \|C\vec{v}\|^2. \quad (2.13)$$

Differentiating this error function yields:

$$\begin{aligned} \frac{\partial E(\vec{v})}{\partial \vec{v}} &= 2M^T M\vec{v} - 2M^T \vec{b} + 2\lambda C^T C\vec{v} \\ &= 2(M^T M + \lambda C^T C)\vec{v} - 2M^T \vec{b}. \end{aligned} \quad (2.14)$$

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

$$\vec{v} = (M^T M + \lambda C^T C)^{-1} M^T \vec{b}. \quad (2.15)$$

The final light direction estimate is computed by averaging the  $n$  resulting light direction estimates from  $\vec{L}^1$  to  $\vec{L}^n$ .

### 2.1.3 Local light source (2-D)

Inherent to the formulation of the previous two sections was the assumption that the light source was infinitely far away (i.e.,  $\vec{L}$  does not depend on the image coordinates). With a local light source, however, this assumption is no longer valid, Figure 2.2(c). The model for an infinite light source, Equation (2.1), can be rewritten to accommodate a local light source as follows:

$$I(\vec{x}) = R(\vec{N}(\vec{x}) \cdot \vec{L}(\vec{x})) + A. \quad (2.16)$$

Note that the light direction is now a function of the position  $\vec{x}$ .

We begin by assuming that the light direction for a local surface patch is constant across the patch. The light direction for each surface patch is then estimated using the solution of Equation (2.7). The previous section introduced a regularization term that encouraged neighboring estimates to be equal, Equation (2.9). In the case of a local light source, a different regularization term is needed as neighboring directions are expected to converge to a single nearby point. This regularization term takes the form:

$$E_2(\vec{L}^1, \dots, \vec{L}^n) = \sum_{i=1}^n \left\| C_i \vec{L}^i \right\|^2, \quad (2.17)$$

where the matrix  $C_i$  is derived below. As in the previous section, the final error function to be minimized is given by:

$$E(\vec{L}^1, \dots, \vec{L}^n, A) = E_1(\vec{L}^1, \dots, \vec{L}^n, A) + \lambda E_2(\vec{L}^1, \dots, \vec{L}^n), \quad (2.18)$$

where  $E_1(\cdot)$  is given by Equation (2.7), and  $\lambda$  is a scaling factor. Unlike the previous section, this error function cannot be minimized analytically, and is instead minimized using an iterative conjugate gradient minimization. Although the functional form of the error function appears similar to that of the previous section, the matrices  $C_i$  depend on the light direction estimate  $\vec{L}^i$ , hence the need for an iterative minimization.

### Local light source regularization

The matrix  $C_i$  in Equation (2.17) is designed to penalize divergence in the light direction estimate  $\vec{L}^i$ . It is derived by first estimating a light position  $\vec{L}$  and then using this position to form a projection matrix for each region of an object.

Consider the local light direction estimates from a pair of objects estimated by minimizing the quadratic error function of Equation (2.7). Denote these estimates as  $\vec{L}_1$  and  $\vec{L}_2$ , and denote  $\vec{c}_1$  and  $\vec{c}_2$  as the center pixel along each boundary. Assuming that these estimates are not parallel, the intersection of the individual light directions is determined by solving:

$$\vec{c}_1 + \alpha_1 \vec{L}_1 = \vec{c}_2 + \alpha_2 \vec{L}_2, \quad (2.19)$$

for the scalar values  $\alpha_1$  and  $\alpha_2$ , using standard least-squares estimation. This intersection yields an estimate of the position of the local light source,  $\vec{L} = \vec{c}_1 + \alpha_1 \vec{L}_1$ .

Consider now the collection of individual estimates along each patch of an occluding boundary,  $\vec{L}^1, \dots, \vec{L}^n$ . Under the model of a single local light source, each of these estimates should be in the direction  $\vec{L} - \vec{c}_i$ , where  $\vec{c}_i$  is the center pixel of patch  $i$ . The regularization term, therefore, penalizes each estimate  $\vec{L}^i$  proportional to its deviation from this direction. Specifically, the penalty is proportional to the difference between the initial estimate  $\vec{L}^i$  and the projection of the estimate

onto  $\vec{L} - \vec{c}_i$ :

$$\begin{aligned}\vec{R}_i &= \vec{L}^i - \vec{\Delta}_i \left( \vec{\Delta}_i^T \vec{L}^i \right), \\ &= \left( I - \vec{\Delta}_i \vec{\Delta}_i^T \right) \vec{L}^i, \\ &= C_i \vec{L}^i,\end{aligned}\tag{2.20}$$

where  $I$  is the identity matrix and where:

$$\vec{\Delta}_i = \frac{\vec{L} - \vec{c}_i}{\left\| \vec{L} - \vec{c}_i \right\|}.\tag{2.21}$$

The penalty for  $\vec{L}^i$  is then simply the magnitude of  $\vec{R}_i$ , Equation (2.17). Note that  $\vec{L}$ , and hence the matrix  $C_i$ , is re-estimated on each iteration of the conjugate gradient minimization [51].

### Minimization

The error function of Equation (2.18) is composed of two terms:

$$E_1(\vec{v}) = \left\| M\vec{v} - \vec{b} \right\|^2,\tag{2.22}$$

and the regularization term:

$$E_2(\vec{L}^1, \dots, \vec{L}^n) = \sum_{i=1}^n \left\| C_i \vec{L}^i \right\|^2,\tag{2.23}$$

where the matrix  $M$  is given by Equation (2.8), the vector  $\vec{v}$  contains the individual light estimates  $\vec{L}^i$  and the ambient term  $A$  given in Equation (2.7), the vector  $\vec{b}$  is given in Equation (2.7), and the matrix  $C_i$  is given in Equation (2.20). The error function  $E_2(\cdot)$  may be written in a more compact and convenient form as:

$$E_2(\vec{v}) = \left\| C\vec{v} \right\|^2,\tag{2.24}$$

where the block-diagonal matrix  $C$  is:

$$C = \begin{pmatrix} C_1 & & & 0 \\ & C_2 & & 0 \\ & & \ddots & \vdots \\ & & & C_n & 0 \end{pmatrix}.\tag{2.25}$$

The error function of Equation (2.18) then takes the form:

$$E(\vec{v}) = \left\| M\vec{v} - \vec{b} \right\|^2 + \lambda \left\| C\vec{v} \right\|^2,\tag{2.26}$$

and it can be minimized with the following approach, based on the conjugate gradient method.

The minimization begins at a point  $\vec{v}_0$ , and searches along a direction  $\vec{\Delta}$  for a point  $\vec{v}_1$  such that  $E(\vec{v}_1) < E(\vec{v}_0)$ . This search direction is opposite the direction of the gradient of  $E(\vec{v})$  at  $\vec{v}_0$ . At each iteration, the process is repeated with the search proceeding from the previous stopping point. The process terminates when a maximum number of iterations,  $i_{\max}$ , has been reached, or if on the  $i^{\text{th}}$  iteration the gradient is below a tolerance  $\epsilon$ . The initial point,  $\vec{v}_0$ , is determined from the least-squares solution of  $E_1(\cdot)$ , Equation (2.7). Described next is the computation of the required gradient and Hessian.

The gradient is found by differentiating Equation (2.26) with respect to  $\vec{v}$ :

$$\frac{\partial E(\vec{v})}{\partial \vec{v}} = 2M^T M \vec{v} - 2M^T \vec{b} + 2\lambda C^T C \vec{v}, \quad (2.27)$$

and the Hessian is found by differentiating twice with respect to  $\vec{v}$ :

$$\frac{\partial^2 E(\vec{v})}{\partial \vec{v}^2} = 2M^T M + 2\lambda C^T C. \quad (2.28)$$

Note that the matrix  $C$  is recomputed at every iteration of the minimization (i.e., it depends on the estimate of each  $\vec{L}^i$  at each iteration).

### 2.1.4 Multiple light sources

In the previous sections, it was assumed that a single directional light source was illuminating the scene (plus a constant ambient term). This is a reasonable assumption for outdoor images where the sun is typically the single source of illumination. For indoor images, however, this assumption is less reasonable because multiple light sources may be present.

Light is linear. As such, a scene illuminated with two infinite light sources takes the form:

$$\begin{aligned} I(\vec{x}) &= R((\vec{N}(\vec{x}) \cdot \vec{L}_1) + (\vec{N}(\vec{x}) \cdot \vec{L}_2)) + A, \\ &= R(\vec{N}(\vec{x}) \cdot (\vec{L}_1 + \vec{L}_2)) + A, \\ &= R(\vec{N}(\vec{x}) \cdot \vec{L}^+) + A, \end{aligned} \quad (2.29)$$

where  $\vec{L}^+$  is the vector sum of the individual vectors  $\vec{L}_1$  and  $\vec{L}_2$ . Note that this model reduces to the same form as a single light source, Equation (2.1). Using the same approach as in the previous sections will result in an estimate of a “virtual” light source, the vector sum of the individual light sources. This relationship trivially extends to three or more individual light sources.

Although not terribly likely, it is possible that different combinations of light sources will sum to the same “virtual” light source, in which case this approach would be unable to detect an inconsistency in the lighting.

## 2.2 Results

We tested our technique on both synthetically generated images and natural photographs. The synthetic images consisted of one or more spheres of constant reflectance rendered under either the infinite or local imaging models of Equation (2.1) or (2.16). The natural photographs were taken outdoors on a clear sunny day (approximating an infinite point light source), or in a controlled lab setting with a single directional light source (approximating a local point light source). These images were captured with a 6.3 megapixel Nikon D100 digital camera in uncompressed RAW format.

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 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.

The intensity from the occluding boundary cannot be directly measured from the image as the surface is occluded. The authors in [38] note, however, that simply using the intensity close to the border is often sufficient. In this case, the intensity is measured at a fixed number of pixels from the boundary in the direction opposite to the surface normal. More precisely, the intensity at a boundary point  $\vec{x}$  with surface normal  $\vec{N}$  is determined by evaluating the 1-D intensity profile:

$$P(t) = I(\vec{x} + t\vec{N}), \quad (2.30)$$

at an offset of  $t = \delta$  pixels, where  $\delta > 0$ .

But, under certain conditions it is advantageous to extrapolate the intensity by considering the intensity profile along a ray coincident to the 2-D surface normal. In the case of extrapolation, we would like to evaluate  $P(t)$  at  $t = 0$  (i.e., at a boundary point), but the intensity at the boundary is unreliable due to the occlusion. This value can, however, be extrapolated from  $P(t)$  with values  $t > 0$ . We assume that the intensity profile can be modeled with an exponential:

$$P(t) = \alpha t^\beta. \quad (2.31)$$

The model parameters,  $\alpha$  and  $\beta$ , are determined using least-squares estimation<sup>2</sup> on  $\log(P(t))$ . In our results, we consider  $P(t)$  for  $t = 1, \dots, 15$ , for this estimation. The intensity at the boundary,  $P(0)$ , is then simply determined by evaluating Equation (2.31) at  $t = 0$ . This entire process is repeated for each point along the occluding boundary. For objects of constant reflectance across the entire object, the extrapolation method is desirable, as it yields more accurate intensity estimates.

---

<sup>2</sup>The model parameters in Equation (2.31) are determined using least-squares estimation on  $\log(P(t))$ . Specifically, in the log domain we have  $\log(P(t)) = \log(\alpha) + \beta \log(t)$ . A quadratic error function in the model unknowns then takes the form  $E(\vec{v}) = \left\| M\vec{v} - \vec{b} \right\|^2$ , where each row of the matrix  $M$  is  $[1 \ \log(t_i)]$ , each corresponding entry of the vector  $\vec{b}$  is  $\log(P(t_i))$ , and  $\vec{v} = (\log(\alpha) \ \beta)^T$ . The least-squares estimate of this error function is  $\vec{v} = (M^T M)^{-1} M^T \vec{b}$ .

In the results of section 2.2.1 (infinite light source), the method of simply measuring the intensity near the boundary was employed—measurements were made 1 pixel from the boundary. In the results of section 2.2.2 and 2.2.3 (local and multiple light source), the extrapolation technique was employed. The reason for this difference is that the objects in our local and multiple light source experiments consisted of spheres of constant reflectance, which lend themselves to the extrapolation method. On the other hand, the objects in our infinite light source experiments did not have constant reflectance across the entire object, making it unlikely that the extrapolation method would yield accurate results.

In all cases, we converted the original color (RGB) image to grayscale ( $\text{gray} = 0.299R + 0.587G + 0.114B$ ) from which the intensity measurements were made.

Finally, the values of  $\lambda$  in the error functions of Equation (2.10) and (2.18) were empirically determined to be 10 (infinite light source), and 1 (local light source). These values were held fixed for all examples given in the next four sections.

### 2.2.1 Infinite light source

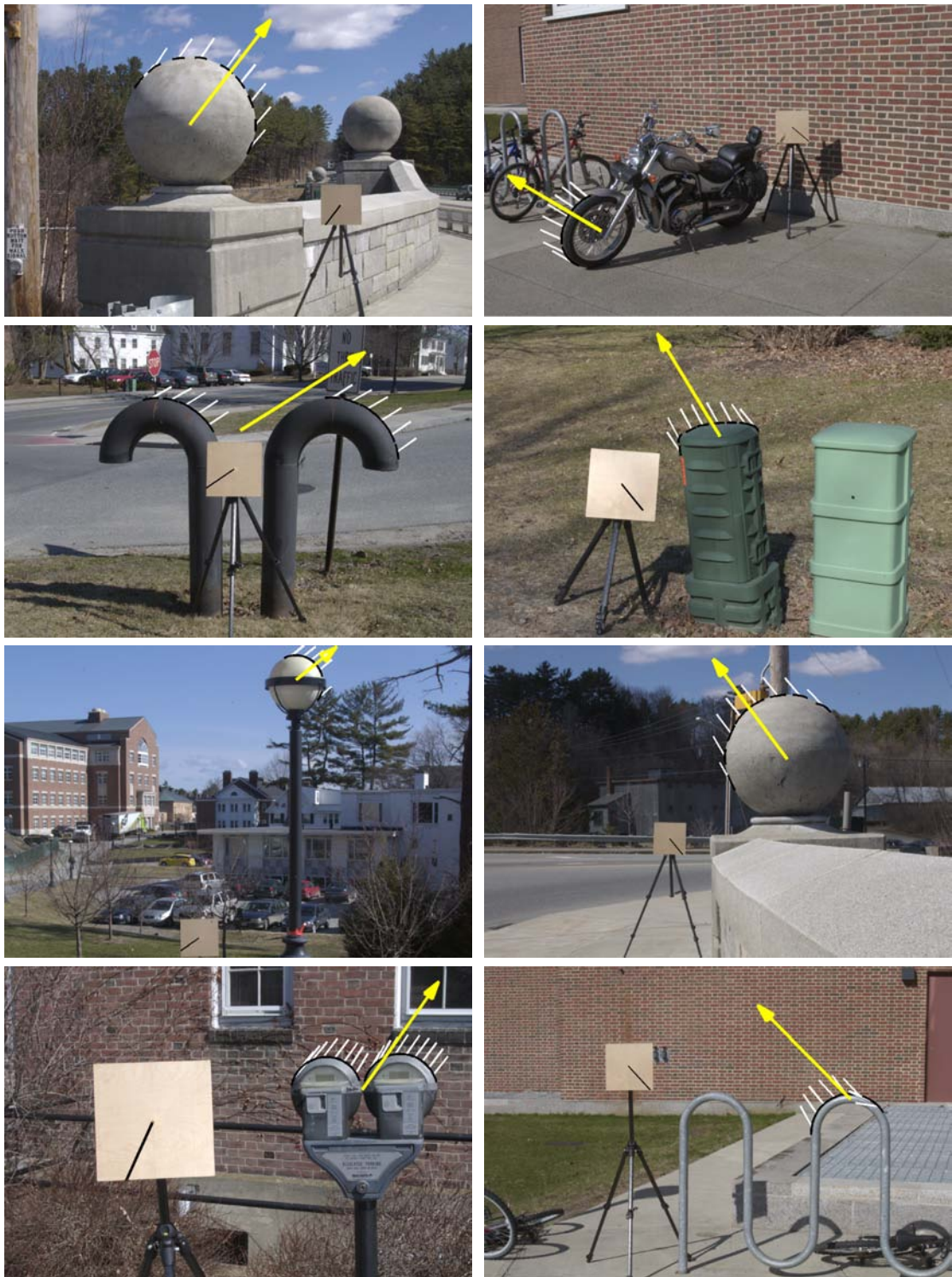
Shown in Figure 2.3 are eight images of objects illuminated by the sun on a clear day. In order to determine the accuracy of our approach, a calibration target, consisting of a flat surface with a rod extending from the center, was placed in each scene. The target was approximately parallel to the image plane, so that the shadow cast by the rod indicated the direction of the sun. Errors in the estimated light direction are given relative to this orientation.

The average estimation error is  $4.8^\circ$  with a minimum and maximum error of  $0.6^\circ$  and  $10.9^\circ$ . The image returning the largest error, the parking meter, is shown in the bottom-left panel of Figure 2.3. There are several reasons for this error, and for errors in general. First, the metallic surface violates the Lambertian assumption. Second, the paint on the meter is worn in several spots causing the reflectance to vary, at times, significantly from patch to patch. Finally, we did not calibrate the camera so as to remove luminance non-linearities (e.g., gamma correction) in the image and gamma can skew the estimate.

Shown in Figure 2.4 is an authentic, although perhaps unlikely, image of Richard Nixon and Elvis Presley. The estimated light directions for each person are consistent, with Nixon at  $98^\circ$  and Presley at  $93^\circ$ . Also shown in Figure 2.4 is a forgery that was entered in the Worth1000 “Celebrity Mini-Me” contest [58]. The light direction estimates are, not surprisingly, consistent:  $147^\circ$  for the larger Superman and  $148^\circ$  for the smaller Superman. This example demonstrates that it is certainly possible for a forgery to have consistent lighting. If the forger carefully selects the source images, the illuminant direction tool will not be able to detect the tampering.

### 2.2.2 Local light source

Shown in Figure 2.5 (top) is a diagram of our experimental setup for testing the local light source estimation. Shown in Figure 2.5 (bottom) are 2 of the 34 images from this experiment. The light

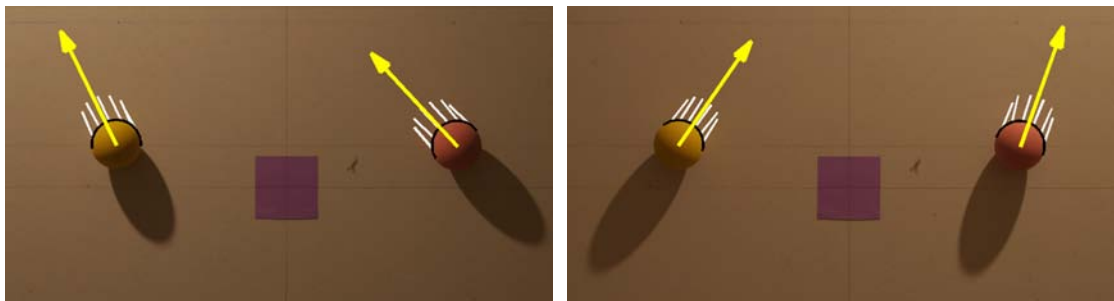
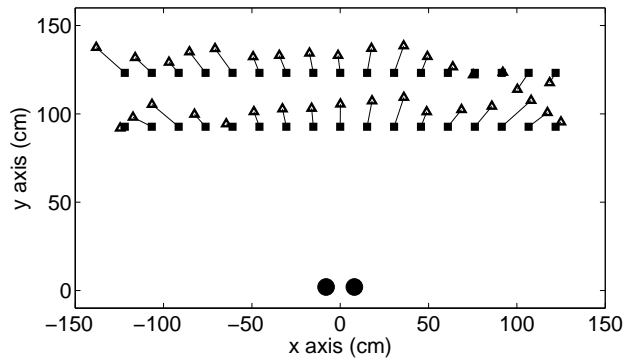


**Figure 2.3:** Eight images with the extracted occluding boundaries (black), individual light direction estimates (white), and the final average light direction (yellow arrow). In each image, the cast shadow on the calibration target indicates the direction to the sun and has been darkened to enhance visibility.

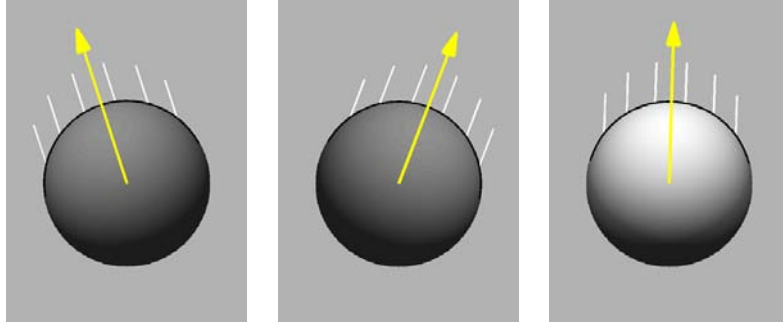




**Figure 2.4:** From left to right: an authentic image of Richard Nixon and Elvis Presley, and a forgery with Superman and a miniature clone. The estimated directions are  $98^\circ$  and  $93^\circ$  for Nixon and Presley, and  $147^\circ$  and  $148^\circ$  for the Superman and the miniature clone. While the light direction estimates are expected to be consistent for authentic images, it is possible for forgeries to have consistent lighting as well.



**Figure 2.5:** (Top) A schematic of the local light source experimental setup along with the actual (squares) and estimated (triangles) light source positions. (Bottom) Two spheres illuminated by a local light source with the extracted boundaries (black), individual light direction estimates (white), and the final average light source direction (yellow arrow) for each sphere.



**Figure 2.6:** From left to right: a sphere illuminated with a single light source positioned at  $-20^\circ$  and  $+20^\circ$  from vertical, and with two light sources positioned at  $\pm 20^\circ$ . Note that in the latter case, the estimate of the light direction (yellow arrow) corresponds to the vector sum of the individual light directions.

consisted of a lamp with an exposed bulb, and the room was otherwise devoid of any light. With the pair of spheres being placed on either side of the origin of a world coordinate system, the light was placed at 93 cm or 124 cm from the origin along the  $y$ -axis, and moved from  $-123$  to  $+123$  cm along the  $x$ -axis, in 17 steps. In Figure 2.5 (top), the squares represent the actual light positions (squares) and the triangles represent the estimated light positions. On average, the position of the light source is estimated within 11.2 cm, with a minimum and maximum error of 0.7 and 22.3 cm. These values correspond to an average, minimum, and maximum error of 9.0%, 0.4%, and 18% as a percentage of the distance from the light source to the origin. With respect to estimating only the orientation of the light source, the average error is  $0.4^\circ$  with a minimum and maximum error of  $0.04^\circ$  and  $1.1^\circ$ .

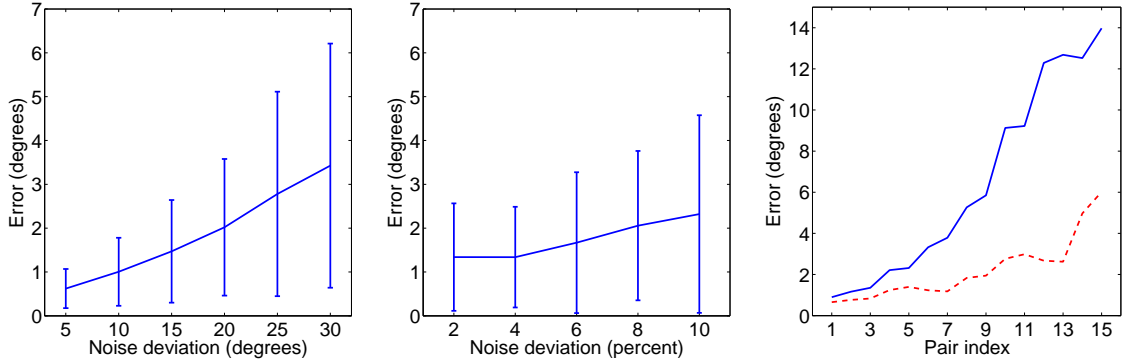
### 2.2.3 Multiple light sources

Shown in Figure 2.6 are three synthetically generated images. In each case, the 3-D scene consisted of a single sphere illuminated with one or two infinite point light sources. In the two left-most panels the sphere was illuminated with a single light source positioned at  $-20^\circ$  and  $+20^\circ$  from vertical ( $90^\circ$ ). In the right-most panel, the sphere was illuminated with two lights positioned at both  $+20^\circ$  and  $-20^\circ$  from vertical. Shown in each panel is the final estimated light direction (yellow arrow). The actual directions to the individual light sources are  $70^\circ$  and  $110^\circ$  yielding a virtual light source at  $90^\circ$  for the scene illuminated by both of these lights. The estimated directions are  $69^\circ$ ,  $108^\circ$ , and  $88^\circ$ , yielding an average error of  $1.7^\circ$ . Note that in the case of the sphere illuminated with two light sources, the estimated direction is, as expected, the vector sum of the individual light sources.

### 2.2.4 Sensitivity

In this section, we explore the sensitivity of the estimated direction to errors in surface normals, errors in intensities, limited surface normal extent, and JPEG compression.

To test the sensitivity of the estimated direction to errors in the surface normals and intensi-



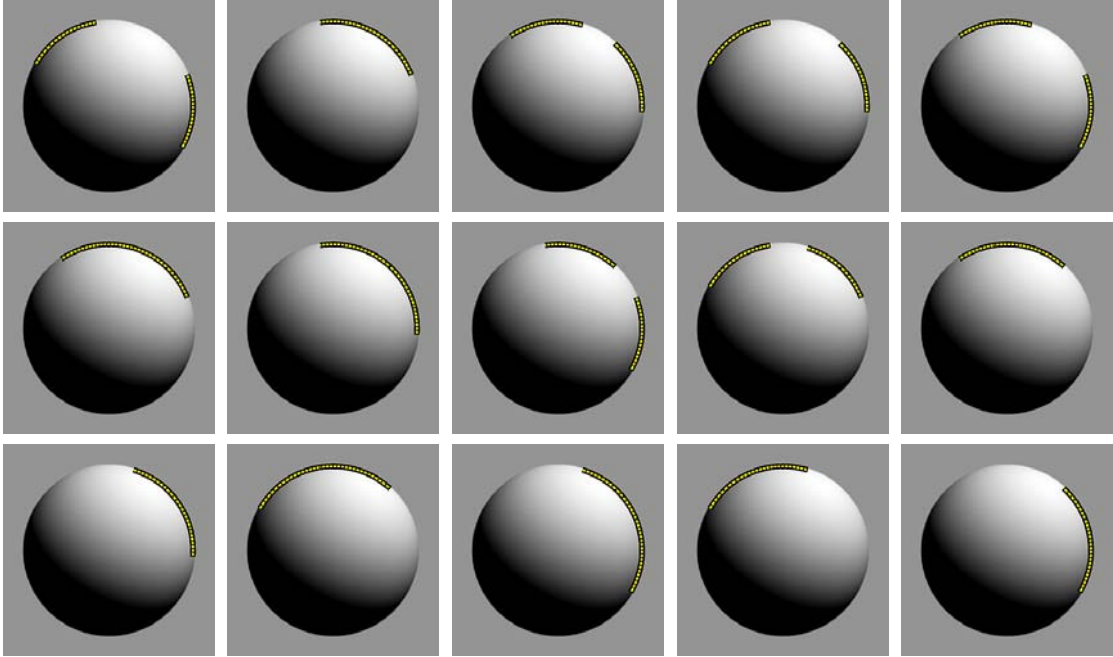
**Figure 2.7:** Average errors in the estimated direction due to noise in the surface normals (left), intensities (center), and surface normal extent (right). In all plots, the averages are computed over 216 images and the error bars show one standard deviation above and below the average. In the surface normal extent plot (right), the pair index indicates the pair of regions used, following the order in Figure 2.8.

ties, we generated 216 images of a sphere illuminated from different directions. For each image, the intensities and surface normals along the occluding contour were measured, using only those measurements that were within  $\pm 90^\circ$  of the location with the maximum intensity. The resulting  $180^\circ$  contour was segmented into six overlapping regions and Gaussian noise of varying amounts was added to both the surface normals and the intensities. The direction to the light source was estimated using Equation (2.13) with  $\lambda$  set to 10. The average errors between the estimated and known light directions over 216 images is shown in Figure 2.7 for noise in surface normals (left) and intensities (center).

For the surface normal experiment, Figure 2.7 (left), Gaussian noise was added to the surface normals in the matrix  $M$ , Equation (2.8). The standard deviation of the noise varied between  $5^\circ$  and  $30^\circ$ , and it modified the direction, but not the magnitude, of each normal. Gaussian noise was added to the intensities as well, with the standard deviation fixed at 5% of the intensity range in the image. Overall, the noise had a small effect on the estimate, causing an average error of  $3.4^\circ$  when the standard deviation of the noise was  $30^\circ$ .

For the intensity experiment, Figure 2.7 (center), the Gaussian noise added to the intensities varied between 2% and 10% of the intensity range of the image. The standard deviation of the noise added to the surface normals was fixed at  $10^\circ$ . As with the surface normal experiment, the additive noise had little effect on the estimate, causing an average error of  $2.3^\circ$  for a standard deviation of 10%.

To test the sensitivity of the estimated direction to surface normal extent, the range of surface normals was limited by using only two of the six regions, Figure 2.8. All fifteen pairs of regions were tested across all 216 images. As with the previous experiments, Gaussian noise was added to the surface normals and intensities, with standard deviations of  $5^\circ$  and 5%, respectively. Unlike the previous experiments, however, the estimated direction was sensitive to these changes, Figure 2.7 (right). The experiment was run both with and without regularization, Equation (2.13), by setting  $\lambda$



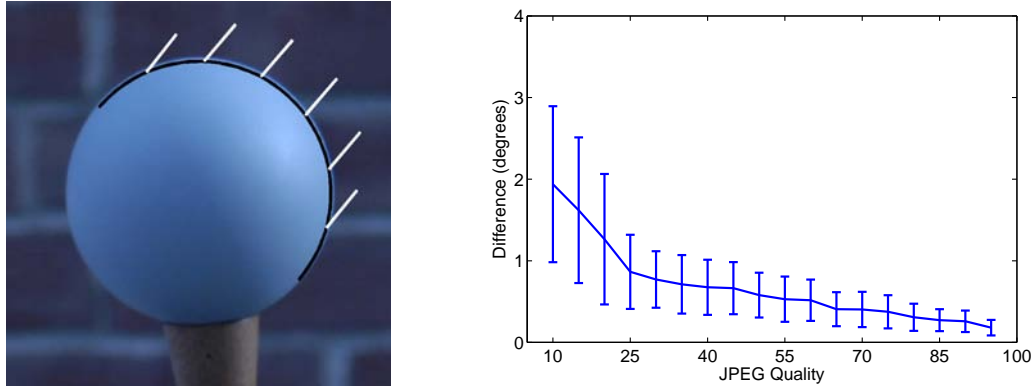
**Figure 2.8:** All 15 pairs of regions used to test sensitivity to surface normal extent. The order, from left-to-right and top-to-bottom, corresponds to the pair index in Figure 2.7.

to 0 or 10. In both cases, greater errors occurred when the pair of regions was skewed to one side of the actual light direction, and lower errors occurred for pairs of regions that were balanced about the light direction. Without regularization, the effect was increased, with an average error of  $14^\circ$  in the worst case. With regularization ( $\lambda = 10$ ), the errors in the worst case were  $6^\circ$ . This experiment demonstrates that the light direction estimates will be unreliable if the extent of surface normals is too small or if the distribution of surface normals is skewed with respect to the actual light direction.

To test the sensitivity of the estimated direction to JPEG compression, we captured 110 images of a diffuse gray sphere in natural environments with primarily directional lighting. For each image, the occluding contour of the sphere was divided into six overlapping regions spanning  $180^\circ$  on the brightest side of the sphere. The illuminant direction was estimated by solving Equation (2.13) with  $\lambda = 10$ . One of the images is shown in Figure (2.9) (left). The images were saved a varying JPEG qualities from 10 to 95 on scale of 0 to 100. Since the actual directions were unknown, we computed the difference between the estimate at the reduced quality and the estimate at full quality. Figure 2.9 (right) shows the average change in the estimated direction due to JPEG compression with error bars at half the standard deviation. This experiment demonstrates that the errors introduced by JPEG compression are small, even at very poor qualities.

### 2.2.5 Forgeries

To demonstrate our approach in a forensic setting, we analyzed three forgeries, Figure 2.10. The first forgery is an image of John Kerry and Jane Fonda sharing a stage at an antiwar rally. This



**Figure 2.9:** (Left) An image of a gray sphere with estimated regions and light directions. (Right) Average difference in the estimated direction due to JPEG compression over 110 images.



**Figure 2.10:** From left to right: a known forgery of John Kerry and Jane Fonda sharing a stage at an antiwar rally, a composite of actor Buster Keaton and actress Tara Reid, and a composite with two statues. The estimated light directions are  $123^\circ$  for Kerry and  $86^\circ$  for Fonda,  $120^\circ$  for Buster and  $62^\circ$  for Tara, and  $63^\circ$  for the left statue and  $42^\circ$  for the right statue.

image was circulated in February of 2004 in an attempt to discredit John Kerry during his campaign for the U.S. presidency. Shortly after its release, however, this image was determined to be a fake, having been created by digitally compositing two separate images. Although we do not know the true illuminant direction, we found an inconsistency in the estimated light direction:  $123^\circ$  for Kerry and  $86^\circ$  for Fonda.

The other two forgeries in Figure 2.10 were downloaded from Worth1000, a website that hosts Photoshop contests [58]. Both images are composites from multiple images. The image of Buster Keaton and Tara Reid gives estimates of  $120^\circ$  for Buster and  $62^\circ$  for Tara. For the image of the two statues, the left statue gives an estimate of  $63^\circ$  and the right statue gives an estimate of  $42^\circ$ . In all cases, the estimated errors are larger than the  $11^\circ$  error observed on real images.

## 2.3 Discussion

The creation of a digital forgery often involves combining objects or people from separate images. In doing so, it is often difficult to match the lighting effects due to directional lighting. At least one reason for this difficulty is that such a manipulation may require the creation or removal of shadows and lighting gradients. And while large inconsistencies in light direction may be fairly obvious, there is evidence from the human psychophysics literature that human subjects are surprisingly insensitive to differences in lighting across an image [40, 52]. The illuminant direction tool enables a user to estimate the direction to a light source from objects in an image—strong inconsistencies in estimates from different objects in the image are evidence of tampering.

While the tool can estimate the direction to a light source with reasonable accuracy, it is only applicable under certain lighting conditions, e.g., outside on a clear day. The limitations of the tool are mostly related to the assumptions of the illumination model: the surface is Lambertian, the illumination is a point light source infinitely far away, and the analyzed intensities must be on a region of the surface that is within  $90^\circ$  of the illuminant direction. In addition, the tool only returns a 2-D estimate of the illuminant direction.

In the next two chapters, we address these limitations with two other tools that also estimate properties of the lighting environment from images. The specular tool estimates a 3-D light direction by assuming a known geometry is in the scene. The lighting environment tool addresses some of the limitations of the illumination model mentioned above, making it applicable to images with more complex lighting. But the fundamental idea behind these tools is the same: to the extent that properties of the lighting environment can be estimated from different objects or people in an image, inconsistencies in lighting can be used as evidence of digital tampering.

## Chapter 3

# Specularity

The photograph in Figure 3.1 of the host and judges for the popular television show *American Idol* was scheduled for publication when it caught the attention of a photo-editor. Coming on the heels of several scandals involving tampered images at major news organizations, the photo-editor was concerned that the image had been doctored. There was good reason to worry: the image was a composite of several photographs. Shown in Figure 3.1 are magnifications of the host’s and judge’s eyes. The inconsistencies in the shape and location of the specular highlight on the eyes suggest that the people were originally photographed under different lighting conditions. In this chapter, we show how the location of a specular highlight can be used to determine the direction to the light source. Inconsistencies in the estimates from different eyes, as well as differences in the shape and color of the highlights, can be used to reveal traces of digital tampering.

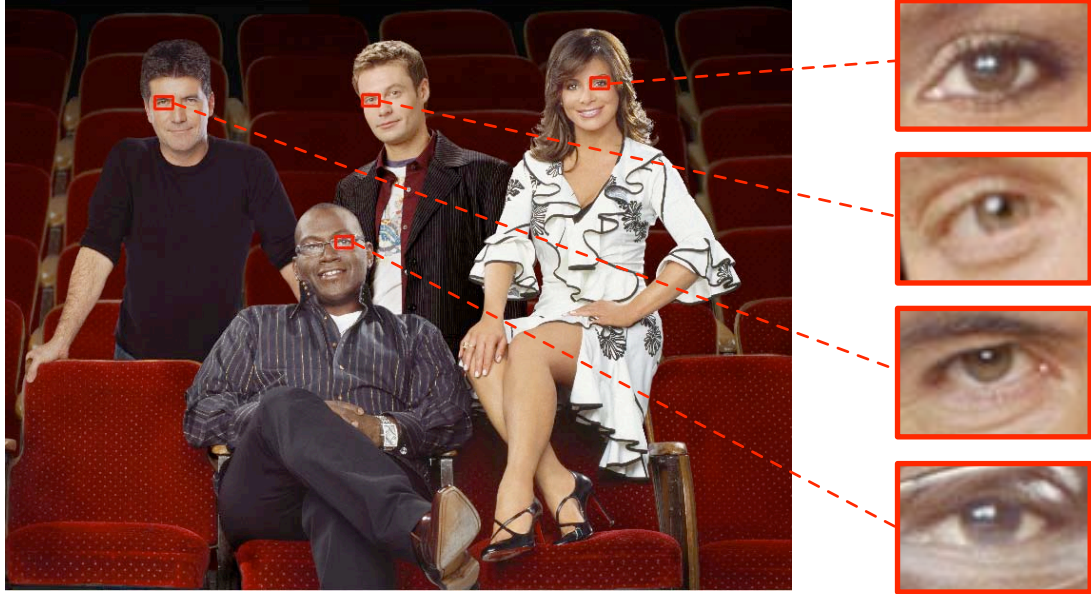
In the previous chapter, we showed how to estimate the light source direction in 2-D from the occluding boundary of an object in an image. While this approach has the benefit of being applicable to arbitrary objects, it has the drawback that it can only determine the direction to the light source within one degree of ambiguity. In contrast, we estimate the full 3-D light source direction by leveraging a 3-D model of the human eye. Although not specifically developed for a forensic setting, the authors of [39] described a technique for computing an environment map from eyes that embodies the illumination in the scene. While the environment map provides a rich source of information about the lighting, it has the drawback of requiring a relatively high-resolution image of the eye.

In this chapter, we describe how to estimate the 3-D direction to a light source from specular highlights on eyes and show the efficacy of this approach on synthetic and real images, as well as on visually plausible forgeries.

### 3.1 Methods

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 3.2, 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





**Figure 3.1:** This photograph of the *American Idol* host and judges is a digital composite of multiple photographs. The inconsistencies in the shape and location of the specular highlight on the eyes suggest that these people were originally photographed under different lighting conditions. Photo courtesy of Fox News and the Associated Press.

to the direction to the light, the surface normal at the highlight, and the direction in which the highlight will be seen. For a perfect reflector, the highlight is seen only when the view direction is equal to the direction of reflection,  $\vec{V} = \vec{R}$ . For an imperfect reflector, a specular highlight can be seen for view directions  $\vec{V}$  near  $\vec{R}$ , with the strongest highlight seen when  $\vec{V} = \vec{R}$ .

We will first derive an algebraic relationship between the vectors  $\vec{L}$ ,  $\vec{N}$ , and  $\vec{V}$ . 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.

### Reflection

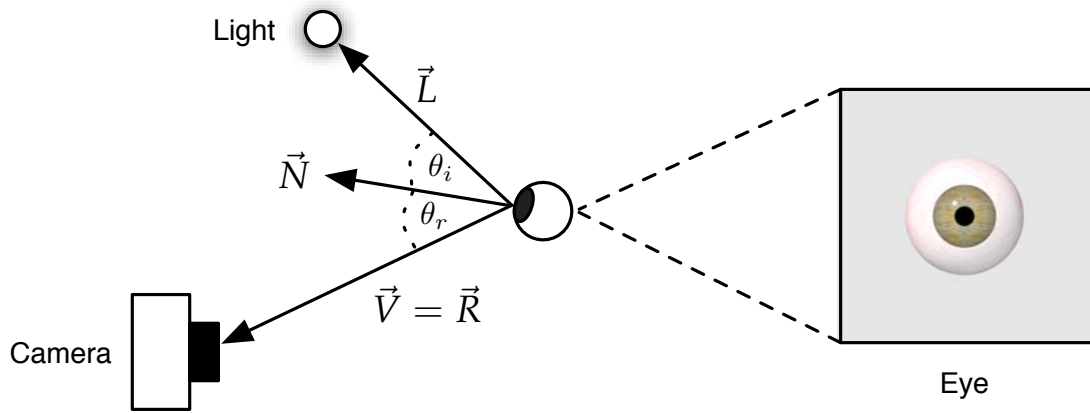
The law of reflection states that a light ray reflects off of a surface at an angle of reflection  $\theta_r$  equal to the angle of incidence  $\theta_i$ , where these angles are measured with respect to the surface normal  $\vec{N}$ , Figure 3.2. Assuming unit-length vectors, the direction of the reflected ray  $\vec{R}$  can be described in terms of the light direction  $\vec{L}$  and the surface normal  $\vec{N}$ :

$$\begin{aligned}\vec{R} &= \vec{L} + 2(\cos(\theta_i)\vec{N} - \vec{L}), \\ &= 2\cos(\theta_i)\vec{N} - \vec{L}.\end{aligned}\tag{3.1}$$

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

$$\vec{L} = 2\cos(\theta_i)\vec{N} - \vec{V},$$





**Figure 3.2:** 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  $\vec{V}$ .

$$= 2 \left( \vec{V}^T \vec{N} \right) \vec{N} - \vec{V}. \quad (3.2)$$

The light direction  $\vec{L}$  can therefore be estimated from the surface normal  $\vec{N}$  and view direction  $\vec{V}$  at a specular highlight. In the following sections, we describe how to estimate these two 3-D vectors from a single image.

Note that the light direction is specified with respect to the eye, and not the camera. In practice, all vectors will be placed in a common coordinate system, allowing us to compare light directions across the image.

### 3.1.1 Camera calibration

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 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 [39]. 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  $3 \times 4$  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  $3 \times 3$  planar projective

transform [22], 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:

$$f(\vec{X}; \vec{a}) = (X_1 - C_1)^2 + (X_2 - C_2)^2 - r^2 = 0, \quad (3.3)$$

where vector  $\vec{a} = (C_1 \ C_2 \ r)^T$  denotes the circle center and radius.

Consider a collection of points,  $\vec{X}_i, i = 1, \dots, m$ , each of which satisfy Equation (3.3). Under an ideal pinhole camera model, the world point  $\vec{X}_i$  maps to the image point  $\vec{x}_i$  as follows:

$$\vec{x}_i = H\vec{X}_i, \quad (3.4)$$

where  $H$  is a  $3 \times 3$  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  $\vec{a}$  and the unknown projective transform  $H$ :

$$E(\vec{a}, H) = \sum_{i=1}^m \min_{\vec{X}^*} \|\vec{x}_i - H\vec{X}^*\|^2, \quad (3.5)$$

where  $\vec{X}^*$  is on the circle parameterized by  $\vec{a}$ . The error embodies the sum of the squared errors between the data,  $\vec{x}_i$ , and the closest point on the model,  $\vec{X}^*$ . This error function is a nonlinear least-squares problem, which is solved using a Gauss-Newton or Levenberg-Marquardt iteration. With only a single circle, there is not a unique projective transform  $H$  that minimizes Equation (3.5). With two coplanar circles, however, the transform can be uniquely determined up to a similarity [59]. Therefore, an error function incorporating both eyes is used to estimate the transform  $H$ . The details of this error function are described in Appendix A.

Once estimated, the projective transform  $H$  can be decomposed in terms of intrinsic and extrinsic camera parameters [22]. The intrinsic parameters consist of the camera focal length, camera center, skew and aspect ratio. For simplicity, we 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:

$$H = \lambda K \begin{pmatrix} \vec{r}_1 & \vec{r}_2 & \vec{t} \end{pmatrix}, \quad (3.6)$$

where the  $3 \times 3$  intrinsic matrix  $K$  is:

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3.7)$$

$\lambda$  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:

$$\begin{aligned} \frac{1}{\lambda} K^{-1} H &= \begin{pmatrix} \vec{r}_1 & \vec{r}_2 & \vec{t} \end{pmatrix}, \\ \hat{H} &= \begin{pmatrix} \vec{r}_1 & \vec{r}_2 & \vec{t} \end{pmatrix}, \end{aligned} \quad (3.8)$$

where the scale factor  $\lambda$  is chosen so that  $\vec{r}_1$  and  $\vec{r}_2$  are unit vectors. The complete rotation matrix is given by:

$$R = \begin{pmatrix} \vec{r}_1 & \vec{r}_2 & \vec{r}_1 \times \vec{r}_2 \end{pmatrix}, \quad (3.9)$$

where  $\times$  denotes cross product.

In the case of an unknown focal length, we estimate the focal length first by decomposing the projective transform  $H$ . The transform  $H$  has eight unknowns: the focal length  $f$ , the scale factor  $\lambda$ , the three rotation angles  $\theta_x$ ,  $\theta_y$  and  $\theta_z$  for the rotation matrix  $R$ , and the three coordinates of the translation vector  $\vec{t}$ . By multiplying the matrices on the right-hand side of Equation (3.6),  $H$  can be expressed in terms of these unknowns:

$$H = \lambda \begin{pmatrix} f c_y c_z & f c_y s_z & f t_x \\ f(s_x s_y c_z - c_x s_z) & f(s_x s_y s_z + c_x c_z) & f t_y \\ c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & t_z \end{pmatrix}, \quad (3.10)$$

where  $c_x = \cos(\theta_x)$ ,  $s_x = \sin(\theta_x)$ , etc., and where the rotation matrix follows the “x-y-z” convention.

Consider the upper-left  $2 \times 2$  sub-matrix of  $H$  rewritten in terms of the four unknowns  $\theta_x$ ,  $\theta_y$ ,  $\theta_z$ , and  $\hat{f} = \lambda f$ . These unknowns are estimated by minimizing the following error function using non-linear least-squares:

$$\begin{aligned} E(\theta_x, \theta_y, \theta_z, \hat{f}) &= (\hat{f} c_y c_z - h_1)^2 + (\hat{f} c_y s_z - h_2)^2 + (\hat{f}(s_x s_y c_z - c_x s_z) - h_4)^2 \\ &\quad + (\hat{f}(s_x s_y s_z + c_x c_z) - h_5)^2, \end{aligned} \quad (3.11)$$

where  $h_i$  corresponds to the  $i^{th}$  entry of  $H$  in row-major order. A Gauss-Newton iterative approach is employed to minimize  $E(\cdot)$ . In practice, we have found that  $\theta_z = \tan^{-1}(h_2/h_1)$ ,  $f = 1$  and

random values for  $\theta_x$  and  $\theta_y$  provide good starting conditions for this minimization. These estimated parameters then yield two possible estimates of the focal length:

$$f_1 = \frac{\hat{f}(c_x s_y c_z + s_x s_z)}{h_7} \quad \text{and} \quad f_2 = \frac{\hat{f}(c_x s_y s_z - s_x c_z)}{h_8}. \quad (3.12)$$

These two estimates are combined using the following weighted average:

$$f = \frac{h_7^2 f_1 + h_8^2 f_2}{h_7^2 + h_8^2}. \quad (3.13)$$

Note that the focal length  $f$  is undefined for  $h_7 = h_8 = 0$ . In addition, this estimation is vulnerable to numerical instabilities for values of  $h_7$  and  $h_8$  near zero. As such, the weighting was chosen to favor larger values of  $h_7$  and  $h_8$ .

### 3.1.2 View direction

Recall that the minimization of Equation (3.5) yields both the transform  $H$  and the circle parameters  $\vec{a}$  for the limbus. Let  $\vec{X}_c = (C_1 \ C_2 \ 1)^T$  denote the estimated center of a limbus in world coordinates. In the camera coordinate system, this point is given by:

$$\vec{x}_c = \hat{H} \vec{X}_c. \quad (3.14)$$

The view direction is the vector from the center of the limbus to the origin of the camera coordinate system. It is given by:

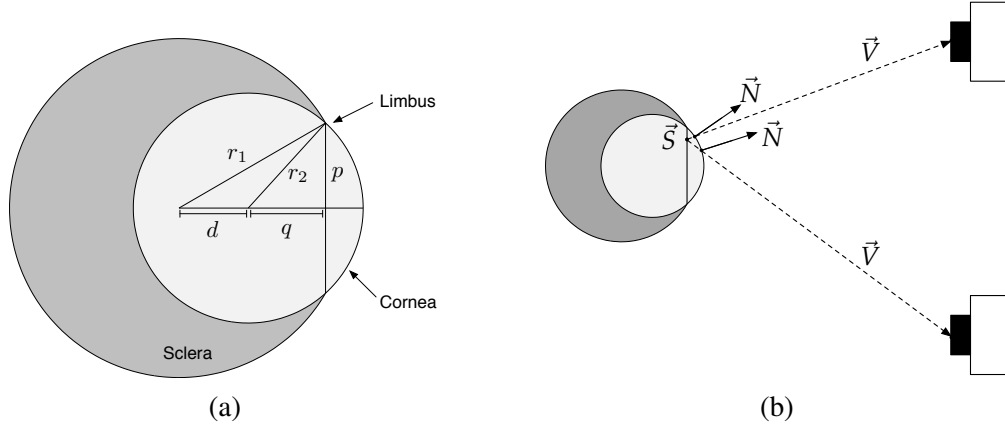
$$\vec{v} = -\frac{\vec{x}_c}{\|\vec{x}_c\|}, \quad (3.15)$$

where it is normalized to unit length and the negative sign reverses the vector so that it points from the eye to the camera.

### 3.1.3 Surface normal

The 3-D surface normal  $\vec{N}$  at a specular highlight is estimated from a 3-D model of the human eye [33]. The model consists of a pair of spheres as illustrated in Figure 3.3(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 [24, 33].

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



**Figure 3.3:** (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  $\vec{V}$ .

Figure 3.3(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:

$$\begin{aligned} (S_x + kV_x)^2 + (S_y + kV_y)^2 + (q + kV_z)^2 &= r_2^2, \\ k^2 + 2(S_xV_x + S_yV_y + qV_z)k + (S_x^2 + S_y^2 + q^2 - r_2^2) &= 0, \end{aligned} \quad (3.16)$$

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

$$\vec{V} = R^{-1}\vec{v}, \quad (3.17)$$

where  $\vec{v}$  is the view direction in camera coordinates, section 3.1.2, and  $R$  is the estimated rotation between the world and camera coordinate systems, section 3.1.1. The surface normal  $\vec{N}$  in the world coordinate system is then given by:

$$\vec{N} = \begin{pmatrix} S_x + kV_x \\ S_y + kV_y \\ q + kV_z \end{pmatrix}, \quad (3.18)$$

and in camera coordinates by:

$$\vec{n} = R\vec{N}. \quad (3.19)$$

### 3.1.4 Light direction

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:

$$\vec{X}_s = H^{-1}\vec{x}_s. \quad (3.20)$$

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:

$$\vec{S} = \frac{p}{r} (\vec{X}_s - \vec{C}), \quad (3.21)$$

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  $\vec{N}$ , as described in the previous section. Combined with the estimate of the view direction  $\vec{V}$ , section 3.1.2, the light source direction  $\vec{L}$  can be estimated from Equation (3.2). In order to compare estimates across the image, the light source direction is converted to camera coordinates:

$$\vec{l} = R\vec{L}. \quad (3.22)$$

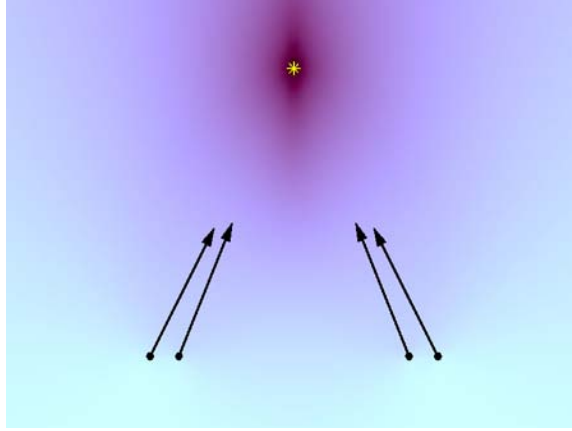
### 3.1.5 Consistency of estimates

In a forensic setting, we would like to determine if the specular highlights in an image are consistent, i.e., they could have arisen from the same light source. The simplest method would be to measure the angle between pairs of light direction estimates: estimates with large angular differences would be deemed inconsistent. But this approach assumes the light source is infinitely far away so that the individual light direction estimates are parallel. In real images, however, we do not expect the estimates to be parallel since specular highlights are usually caused by local light sources (e.g., flashes, lights in the room, windows). Instead, we expect the estimates to converge towards the position of the light source.

The light direction estimates from each specular highlight constrain the position of the light source, Figure 3.4. At the  $i^{\text{th}}$  specular highlight, the angle between the vector to the light source at position  $\vec{x}$  and the estimated direction  $\vec{l}_i$  (a unit vector) is:

$$\theta_i(\vec{x}) = \cos^{-1} \left( \vec{l}_i^T \frac{\vec{x} - \vec{p}_i}{\|\vec{x} - \vec{p}_i\|} \right), \quad (3.23)$$

where  $\vec{p}_i$  is the position of the  $i^{\text{th}}$  specular highlight. Given a set of estimates from  $N$  specular highlights, the position of the light source can be estimated by minimizing the following error



**Figure 3.4:** Error surface for four specular highlights and corresponding light directions in 2-D, Equation 3.24. The error is minimal near the light position (yellow star).

function:

$$E(\vec{x}) = \sum_{i=1}^N \theta_i(\vec{x}). \quad (3.24)$$

Although this error function is intuitive, it is unnecessarily complex due to the  $\cos^{-1}(\cdot)$  nonlinearity. We have also found empirically that this complexity often causes the error function to be more difficult to minimize. To avoid these issues, we note that the term inside the parentheses in Equation (3.23) is a dot product between two unit vectors. This quantity is one if the vectors are parallel; otherwise, it is less than one. Instead of minimizing Equation (3.24), we can maximize:

$$\hat{E}(\vec{x}) = \sum_{i=1}^N \vec{l}_i^T \frac{\vec{x} - \vec{p}_i}{\|\vec{x} - \vec{p}_i\|}, \quad (3.25)$$

which is the sum of the dot products. This function has the advantage that the derivative is simple:

$$\frac{\partial \hat{E}(\vec{x})}{\partial \vec{x}} = \frac{(\|\vec{x} - \vec{p}\|^2 \vec{l}_i - \vec{l}_i^T (\vec{x} - \vec{p})(\vec{x} - \vec{p}))}{\|\vec{x} - \vec{p}\|^3}, \quad (3.26)$$

and it can be maximized using the nonlinear conjugate gradient method [51]. If point  $\vec{x}^*$  is the light source position computed by maximizing Equation (3.25), the angular error for the  $i^{\text{th}}$  specular highlight is given by  $\theta_i(\vec{x}^*)$ .

One approach to detecting inconsistencies using angular errors is with a threshold: angles above the threshold are inconsistent and angles below the threshold are not. But reliable thresholds are often difficult to establish and thresholds provide only a binary result: above or below. A statistical approach, such as a hypothesis test, provides more information since it reports the probability of observing the result (or one more extreme). This approach assumes that the angular errors are

normally distributed and that inconsistent estimates in a forgery will skew the minimization of Equation (3.25) resulting in larger errors for all estimates. The hypothesis test will therefore use all  $N$  angular errors to decide between two hypotheses: (1) that the mean is equal to an expected mean of  $\mu_0$ ; or (2) that the mean is greater than  $\mu_0$ . The test statistic is:

$$z = \frac{\mu - \mu_0}{\sigma_0/\sqrt{N}}, \quad (3.27)$$

where  $\mu$  is the average of the  $N$  angular errors,  $\sigma_0$  is the expected standard deviation, and  $\mu_0$  is the expected mean—the values of  $\mu_0$  and  $\sigma_0$  are determined empirically from authentic images. The significance of the test statistic is given in terms of the standard error function:

$$p(z) = \frac{1}{2} \left( 1 - \operatorname{erf} \left( \frac{z}{\sqrt{2}} \right) \right). \quad (3.28)$$

If the significance of the test statistic is smaller than a level of  $\alpha$  (e.g.,  $\alpha = 1\%$ ), then the average errors from the specular highlights are larger than expected and the estimates can be deemed inconsistent. Otherwise, the estimates cannot be deemed inconsistent.

## 3.2 Results

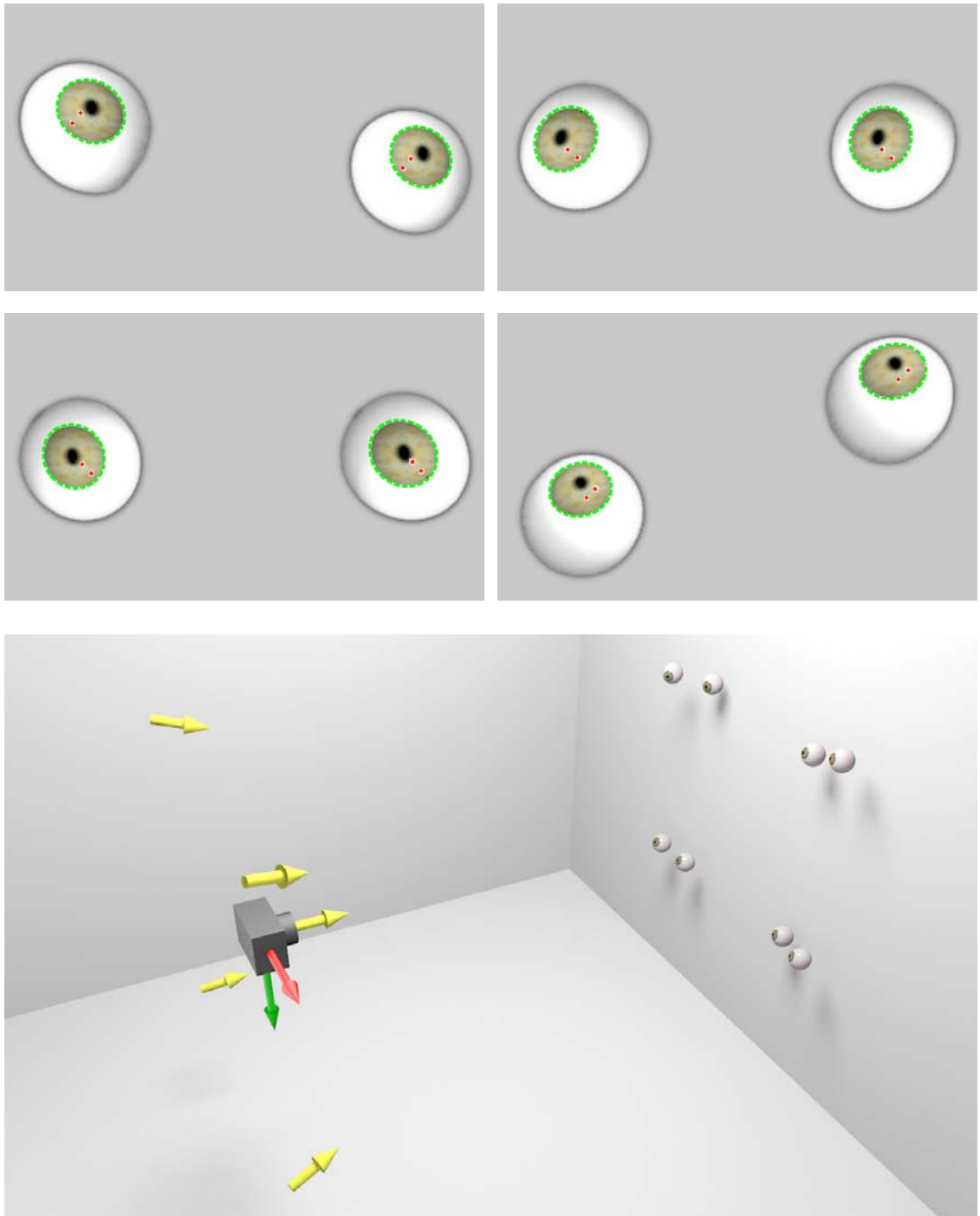
We tested our technique for estimating the 3-D light source direction on both synthetically generated and real images. In all of these results the direction to the light source was estimated from specular highlights in both eyes. This required a slight modification to the minimization in Equation (3.5) which is described in Appendix A. The view direction, surface normal and light direction were then estimated separately for each eye.

### 3.2.1 Synthetic images

Synthetic images of eyes were rendered using the `pbrrt` environment [42]. The shape of the eyes conformed to the 3-D model described in section 3.1.3 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 3.5. Each image was rendered at a resolution of  $1200 \times 1600$  pixels, with the cornea occupying less than 0.1% of the entire image. Shown in Figure 3.5 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 normal  $\vec{n}$  were estimated, from which the direction to the light source  $\vec{l}$  was determined. The angular





**Figure 3.5:** 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 in dashed green, and the red 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.

image	left eye		right eye		left eye		right eye	
	$L_1$	$L_2$	$L_1$	$L_2$	$L_1$	$L_2$	$L_1$	$L_2$
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 3.1:** Angular errors (degrees) in estimating the light direction for the images shown in Figure 3.6. 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.

error between the estimated  $\vec{l}$  and actual  $\vec{l}_0$  light directions is computed as:

$$\phi = \cos^{-1}(\vec{l}^T \vec{l}_0). \quad (3.29)$$

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  $2.8^\circ$  with a standard deviation of  $1.3^\circ$  and a maximum error of  $6.8^\circ$ . With an unknown focal length, the average error was  $2.8^\circ$  with a standard deviation of  $1.3^\circ$  and a maximum error of  $6.3^\circ$ .

### 3.2.2 Real images: controlled lighting

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 3.6. A 6.3 megapixel Nikon D100 digital camera with a 35 mm lens was set to capture in the highest quality JPEG format.

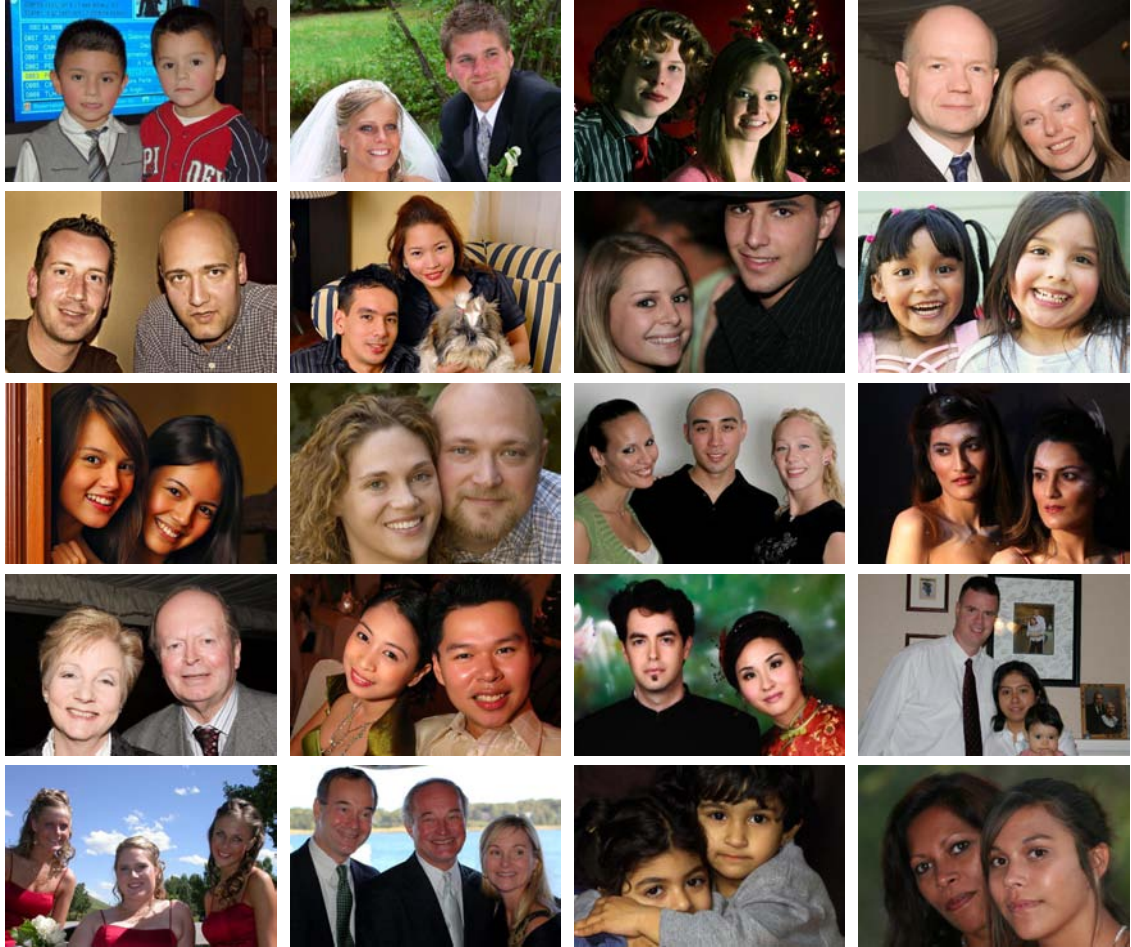
For each image, an ellipse was manually fit to the limbus of each eye. In these images, the limbus did not form a sharp boundary—the boundary spanned roughly 3 pixels. As such, we fit the ellipses to the better defined inner outline [27], Figure 3.6. The radius of each limbus was approximately 9 pixels, 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 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, Equation (3.29), for each image are given in Table 3.1. 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  $8.6^\circ$ , and with an unknown focal length, the average angular error was  $10.5^\circ$ .



**Figure 3.6:** A subject at different locations and orientations relative to the camera and two light sources (left) with magnified views of the eyes (right). The ellipse fit to each limbus is shown in dashed green and the red dots denote the positions of the specular highlights. See also Table 3.1.



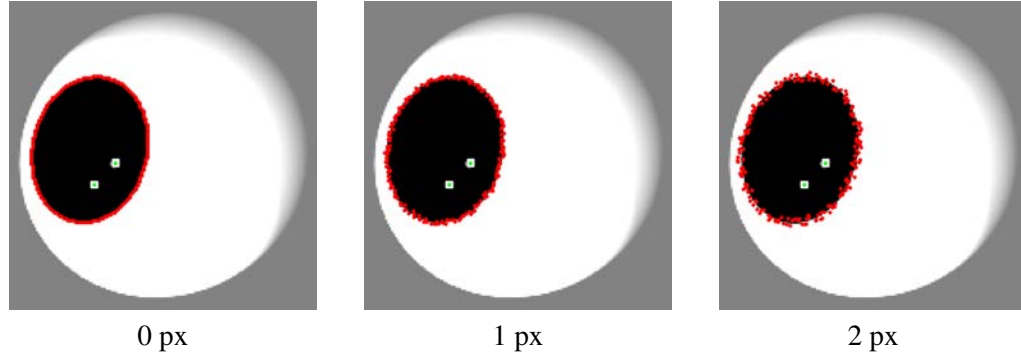
**Figure 3.7:** Twenty images of two or more people with specular highlights in their eyes.

There are several reasons for the increase in error over the synthetic images. First, the average size of the cornea in our real images is much smaller than the size of the cornea in the synthetic images,  $256 \text{ pixels}^2$  versus over  $1000 \text{ pixels}^2$ . Second, the limbus in an adult human eye is slightly elliptical, being 1 mm wider than it is tall [24], while our model assumes a circular limbus. Lastly, the positions of the lights and camera in the room were measured with a tape measure and are almost certainly not exact.

### 3.2.3 Real images: unknown lighting

While it is important to establish the errors in estimating the direction to a known source, it is more important for forensics to establish the consistency of measurements across multiple specular highlights in an image. In an authentic image, we expect light direction estimates from highlights to converge towards the position of the light source. In forgeries, on the other hand, we expect the light direction estimates to diverge if the images were captured under different lighting.

To explore the consistency of light direction estimates in authentic images, we acquired twenty



**Figure 3.8:** Uniform noise added to points on the limbi with maximum deviations of 0 to 2 pixels.

images from Flickr, a photo-sharing website [60]. Shown in Figure 3.7 are the twenty images, each image showing multiple people with specular highlights in their eyes. Following the same approach as the controlled lighting experiment, ellipses were fit by hand to the inner outline of the limbus in each eye and the specular highlights were localized by specifying a bounding rectangular area around each highlight and computing the centroid of the selection.

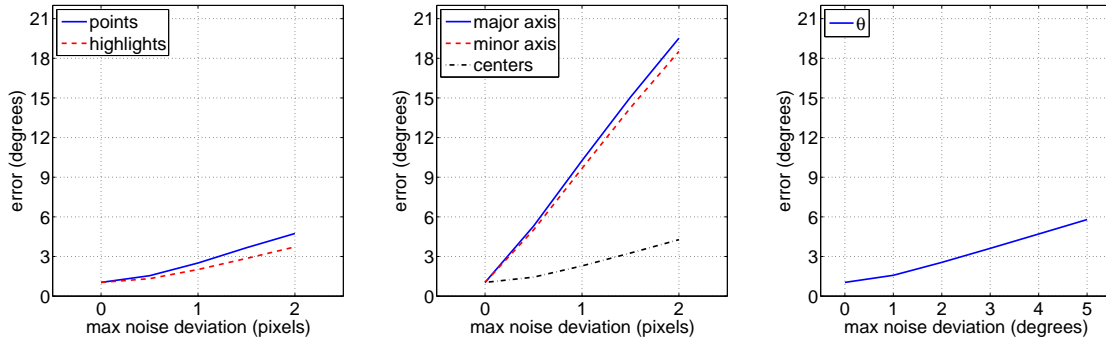
For each image, the light position  $\vec{x}^*$  was estimated by maximizing Equation (3.25) using the light direction estimates from all the specular highlights in the image. The angular errors between each estimated direction and the vector to the point  $\vec{x}^*$  were computed using Equation (3.23). In total, there were 88 light direction estimates (44 people). The average angular error was  $6.4^\circ$ , with a standard deviation of  $2.8^\circ$  and a maximum error of  $12.8^\circ$ .

### 3.2.4 Sensitivity

In this section, we explore the sensitivity of the estimated direction to noise in the points on the limbi, noise in the positions of the highlights, and errors in the shape of the ellipses. A set of 243 simulated images of eyes was used for all experiments. The approximate radius of the eyes in this experiment was 35 pixels and the shape of the limbus was automatically extracted from each eye. In each image, the position and orientation of the eyes varied, and there were two highlights visible in each eye, yielding four measurements from each image. For all experiments, the average errors between the actual and estimated light directions across 972 measurements are given in degrees.

To test the sensitivity of the estimated direction to noise in the points on the limbi, we added uniform random noise of varying amounts to the each coordinate of the points. The maximum displacement of the noise varied between 0 to 2 pixels, Figure 3.8. The average error in the estimated light directions due to the noise is shown as the solid line in the left panel of Figure 3.9. Uniform noise with a 2 pixel maximum deviation caused an average error of  $4.7^\circ$  in the estimated direction.

To test the sensitivity of the estimated direction to noise in the positions of the highlights, we added uniform random noise to the coordinates of the highlights. As with the previous experiment, the maximum deviation of the noise varied between 0 and 2 pixels. The average error in the estimated light directions due to the noise is shown as the dashed line in the left panel of Figure 3.9.



**Figure 3.9:** Sensitivity of the estimated direction to noise. (Left) Average error due to noise in the points on the limbi and noise in the positions of the highlights. (Center) Average error due to noise added to the ellipse center, major axis, and minor axis. (Right) Average error due to noise added to the ellipse angle.

Uniform noise with a 2 pixel maximum deviation caused an average error of  $3.7^\circ$  in the estimated direction.

To test the sensitivity of the estimated direction to the shape of the elliptical limbi, we fit ellipses to each limb and decomposed the ellipses into five parameters: the 2-D center, the lengths of the major and minor axes, and the rotation angle  $\theta$ . Uniform noise between 0 and 2 pixels was added independently to the first four parameters, and noise between  $0^\circ$  and  $5^\circ$  was added to the angular parameter  $\theta$ . The average error in the estimated direction due to the noise is shown in the center and right panels of Figure 3.9. The estimated direction is sensitive to noise added to the lengths of the axes: two pixels of noise added to the major or minor axes caused an average error of  $19.5^\circ$  and  $18.5^\circ$ , respectively. The ratio of the lengths of the axes (i.e., ellipse eccentricity) is directly related to the pose of the eyes and the additive noise skews the pose estimation, causing large errors. The estimated direction is less sensitive to noise added to the center or the angular parameter, with an average error of  $4.3^\circ$  for 2 pixels of noise added to the center and  $5.8^\circ$  for  $5^\circ$  of noise added to  $\theta$ .

### 3.2.5 Forgeries

Shown in Figure 3.10 are four image forgeries. The two images in the left column are the American Idol forgery and a family portrait where the father’s face has been replaced with the face of Gene Simmons from the rock band KISS. In the right column are two forgeries from the “Impossible Celebrity Couples” Photoshop contest hosted by the website Worth1000 [58]. They show, from top to bottom, actor Humphrey Bogart with actress Jessica Alba, and actor George Clooney with actress Claudia Cardinale.

In the American Idol image, there were two specular highlights visible in the eyes of two of the judges. As a result, we tested for consistency in two different ways. First, we minimized Equation (3.25) using the estimates from the left highlight from eyes with two highlights, together with the highlight from eyes with one highlight. Next, we minimized the same equation with the estimates from the right highlight instead of the left. The errors from both approaches are summarized





**Figure 3.10:** Four image forgeries. (Left) the American Idol forgery and a family portrait with rock star Gene Simmons. (Right) two forgeries from the Worth1000 “Impossible Celebrity Couples” Photoshop contest.

in the first two columns of Table 3.2. Using the  $z$ -test, we confirmed that the average error in both cases is statistically greater than the average error of 6.4 and standard deviation of 2.8 measured from the authentic images.

In the KISS image, there were two specular highlights visible in the eyes of the children but only one visible in the eyes of the father. As with the American idol image, we measured consistency between the left and right highlights separately, including the father’s estimates in both sets. We did not estimate the light direction for the mother because we have found that glasses distort the shape and location of the specularities on the eye. The errors for this image are summarized in the third and fourth columns of Table 3.2 and the  $z$ -test confirmed that they are statistically greater than the errors from the authentic images.

In the final two columns of Table 3.2 are the average errors for the celebrity couple images. They too are statistically larger than the errors from the authentic images. Note, however, that it is possible for a forgery to have errors consistent with the errors measured from the authentic images. Two examples of such images are shown in Figure 3.11. The source images for these forgeries were shot under similar lighting so the light direction estimates are close. In fact in the rightmost forgery, the source images come from the same photo shoot so the lighting was the same.

	Idol (L)	Idol (R)	Kiss (L)	Kiss (R)	Bogart	Clooney
mean	11.5	17.8	23.9	12.9	11.2	13.1
std. dev	6.0	8.5	11.0	5.0	3.1	6.1

**Table 3.2:** Average error and standard deviation for the light direction estimates from the image forgeries shown in Figure 3.10. In all cases, the errors were statistically larger than the errors from the authentic images, Figure 3.7.



**Figure 3.11:** Two forgeries from the Worth1000 “Attack of the Clones” Photoshop contest. Each image is a composite of the same person under similar lighting.

### 3.3 Discussion

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 on the eye are a powerful cue as to the shape, color and location of the light source(s). Inconsistencies in these properties can be used as evidence of tampering. We have described how to 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.

This tool is capable of estimating the 3-D direction to a light source, but it depends on two elliptical selections to approximate the shape of the limbi. These ellipses determine the pose of the eyes in the image, and the accuracy of the light direction estimation depends on the accuracy of the pose estimation. If the eyes are sufficiently large and not occluded by eyelids, this selection could be partially automated to improve robustness. But as the eyes become smaller in the image, the selection is difficult to perform manually or automatically. It may be possible to use other sources of pose information, such as the head, to condition the pose of the eyes. Improving the ellipse selection process could make the tool more robust and also simplify the user experience.

Another future direction for this work would be to reconsider, in a forensic setting, the problem of estimating parameters of a full lighting environment from eyes. Earlier work has described this



process for high-resolution images of eyes [39], but perhaps the approach could be simplified to allow for smaller eyes, which would be more realistic for forensics. These estimates would provide more detail about the lighting environment than a single 3-D direction and would be applicable under complex illumination. In the next chapter, we describe a technique for estimating properties of complex lighting environments from diffuse objects, but a similar approach could be used for glossy objects, such as the eye.

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 also be detectable using the technique described in the previous chapter. In addition, small artifacts on the eyes are often visually salient. Nevertheless, as with all forensic tools, it is possible to circumvent this technique.

# Chapter 4

## Lighting environment

In the previous two chapters, we have shown techniques for estimating the direction to a light source, and how inconsistencies in these estimates can be used to detect tampering. These techniques are appropriate when the lighting is dominated by a single light source, but are less appropriate in more complex lighting environments containing multiple light sources or non-directional lighting (e.g., the sky on a cloudy day). Shown in Figure 4.1, for example, is a digital composite of “Katie” and “Kimo.” At first glance, this composite is reasonably compelling. Upon closer examination, however, the lighting on Kimo is seen to be strongly directional while the lighting on Katie is more diffuse. Here we describe how to quantify such complex lighting environments and how to use inconsistencies in lighting to detect tampering.

We leverage earlier work [3, 49] that shows that under some simplifying assumptions, arbitrarily complex lighting environments can be approximated by a low-dimensional model. We show how the parameters of a reduced version of this model can be estimated from a single image, and how this model can be used to detect consistencies and inconsistencies in an image. Results from a broad range of simulated and photographed images as well as visually plausible forgeries are presented.

### 4.1 Methods

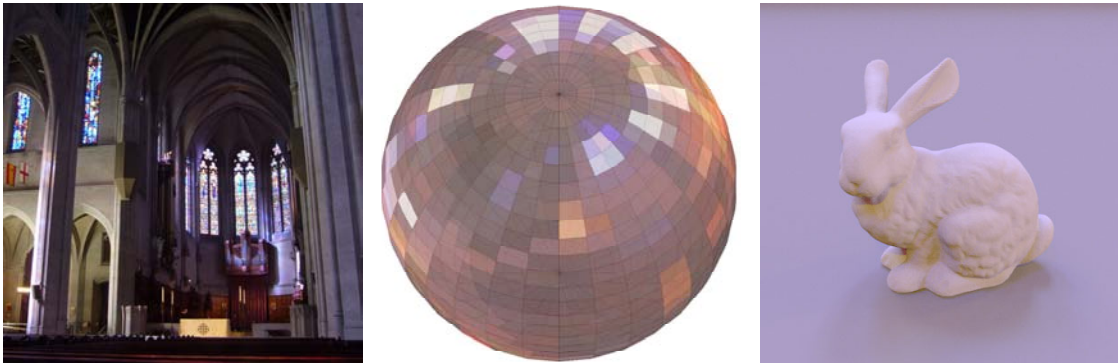
The lighting of a scene can be arbitrarily complex—any number of lights can be placed in any number of positions, creating different lighting environments. In order to model such complex lighting, we assume that the lighting is distant and that surfaces in the scene are convex and Lambertian. To use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

#### 4.1.1 Representing lighting environments

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 the value of  $L(\vec{V})$  is the intensity of the incident light along direction  $\vec{V}$ , Figure 4.2. If the object being illuminated is convex, the irradiance (light received) at any point on the surface is due to



**Figure 4.1:** A fake *Star* magazine cover showing Kimo with actress Katie Holmes. Also shown is a magnified view of this forgery, and the original cover showing Holmes with actor Tom Cruise.



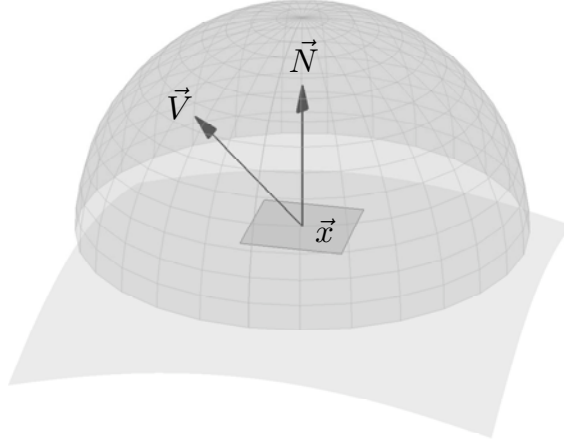
**Figure 4.2:** Shown from left to right are an image taken inside Grace Cathedral in San Francisco, a sphere embodying the lighting environment in Grace Cathedral, and the Stanford bunny rendered under this lighting environment.

only the lighting environment; i.e., there are no cast shadows or interreflections [49]. As a result, the irradiance,  $E(\vec{N})$ , can be parameterized 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  $L(\vec{V})$ :

$$E(\vec{N}) = \int_{\Omega} L(\vec{V}) R(\vec{V}, \vec{N}) d\Omega, \quad (4.1)$$

where  $\Omega$  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:

$$R(\vec{V}, \vec{N}) = \max(\vec{V} \cdot \vec{N}, 0), \quad (4.2)$$



**Figure 4.3:** 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  $\vec{N}$ .

which is either the cosine of the angle between vectors  $\vec{V}$  and  $\vec{N}$ , or zero when the angle is greater than  $90^\circ$ . This reflectance function effectively limits the integration in Equation (4.1) to the hemisphere about the surface normal  $\vec{N}$ , Figure 4.3. In addition, while we have assumed no cast shadows, Equation (4.2) explicitly models *attached* shadows, i.e., shadows due to surface normals facing away from the direction  $\vec{V}$ .

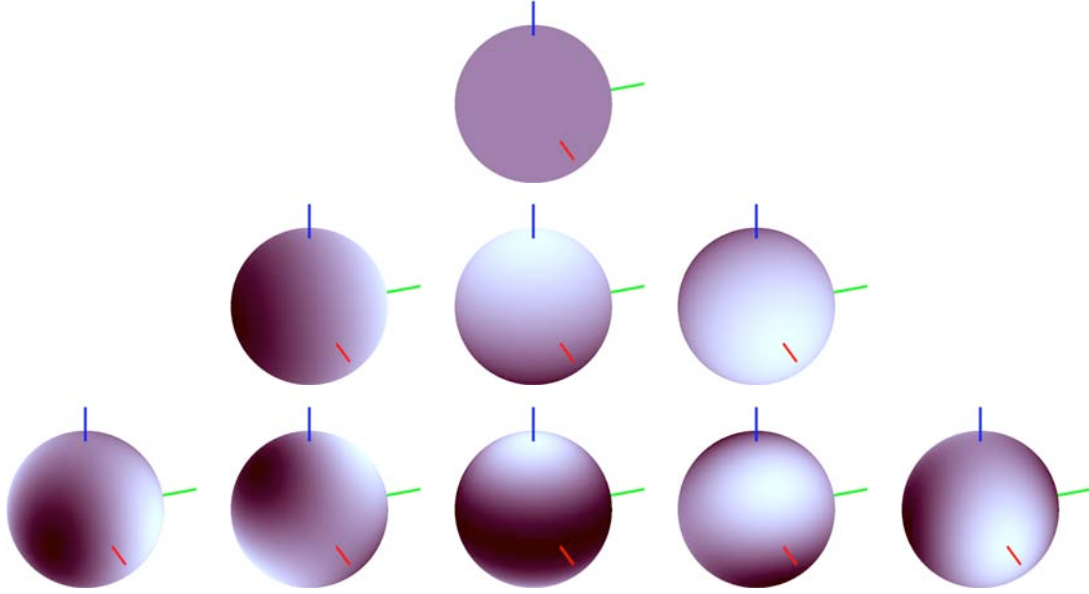
The convolution in Equation (4.1) can be simplified by expressing both the lighting environment and the reflectance function in terms of spherical harmonics. Spherical harmonics form an orthonormal basis for piecewise continuous functions on the sphere and are analogous to the Fourier basis on the line or plane. The first three orders of spherical harmonics in terms of the Cartesian coordinates of the surface normal,  $\vec{N} = (x \ y \ z)^T$ , are defined below and shown in Figure 4.4.

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

The lighting environment expanded in terms of spherical harmonics is:

$$L(\vec{V}) = \sum_{n=0}^{\infty} \sum_{m=-n}^n l_{n,m} Y_{n,m}(\vec{V}), \tag{4.3}$$

where  $Y_{n,m}(\cdot)$  is the  $m^{\text{th}}$  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,



**Figure 4.4:** 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,  $Y_{2,m}(\cdot)$ .

only harmonics with  $m = 0$  appear in the expansion:

$$R(\vec{V}, \vec{N}) = \sum_{n=0}^{\infty} r_n Y_{n,0} \left( (0 \ 0 \ \vec{V} \cdot \vec{N})^T \right). \quad (4.4)$$

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 [3, 49]. As a result, the irradiance, Equation (4.1), takes the form:

$$E(\vec{N}) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \hat{r}_n l_{n,m} Y_{n,m}(\vec{N}), \quad (4.5)$$

where

$$\hat{r}_n = \sqrt{\frac{4\pi}{2n+1}} r_n. \quad (4.6)$$

The key observation in [49] and [3] was that the coefficients  $\hat{r}_n$  for a Lambertian reflectance function decay rapidly, and thus the infinite sum in Equation (4.5) can be well approximated by the first nine terms:

$$E(\vec{N}) \approx \sum_{n=0}^2 \sum_{m=-n}^n \hat{r}_n l_{n,m} Y_{n,m}(\vec{N}). \quad (4.7)$$

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.

#### 4.1.2 From irradiance to intensity

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  $\rho$ . 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.

A camera maps its received light to intensity through a camera response function:

$$I = f(Et), \quad (4.8)$$

where  $I$  is the image intensity and the function  $f(\cdot)$  is often nonlinear. The received light is represented by the product of irradiance  $E$  and exposure time  $t$ , and the dependence of the irradiance and intensity on position  $\vec{x}$  has been dropped for simplicity. In an arbitrary image, we cannot know the exposure time  $t$ , but we show that under an assumption of bounded irradiance for objects in an image, different exposure times cause a change in intensity that can be modeled linearly.

Suppose there is an object in an arbitrary lighting environment with bounded irradiance, and let the minimum and maximum irradiance values for the object be  $E_1$  and  $E_2$ . Let  $t_1$  and  $t_2$  be two different exposure times, and without loss of generality, we assume  $t_1 = 1$  and  $t_2 > t_1$ . The intensities for the first exposure,  $t_1 = 1$ , can be approximated by a truncated Taylor series expanded about the midpoint of irradiance values for the object:

$$f(Et_1) \approx f(m_1) + f'(m_1)(Et_1 - m_1), \quad (4.9)$$

$$= f(m_1) + f'(m_1)(E - m_1), \quad (4.10)$$

where  $m_1 = (E_1 + E_2)/2$ . Similarly, the intensities for the second exposure,  $f(Et_2)$ , can be approximated by a truncated Taylor series expanded about the midpoint of the scaled irradiance values:

$$f(Et_2) \approx f(m_2) + f'(m_2)(Et_2 - m_2), \quad (4.11)$$

where  $m_2 = (t_2E_1 + t_2E_2)/2 = t_2m_1$ .

From Equations (4.10) and (4.11), the relationship between the intensities due to a change in exposure is given by:

$$\begin{aligned} f(Et_2) &\approx f(m_2) + f'(m_2)(Et_2 - m_2), \\ &= f(m_2) + f'(m_2)(Et_2 - m_2) + \left[ \frac{f'(m_2)}{f'(m_1)} t_2 f(Et_1) - \frac{f'(m_2)}{f'(m_1)} t_2 f(Et_1) \right], \end{aligned}$$

$$\begin{aligned}
&= f(m_2) + f'(m_2)(Et_2 - m_2) + \alpha f(Et_1) - \alpha f(m_1) - f'(m_2)t_2(E - m_1), \\
&= f(m_2) + \alpha f(Et_1).
\end{aligned} \tag{4.12}$$

Therefore, the change in the intensity profile due to an increased exposure time  $t_2$  can be modeled by a linear change to the profile of exposure time  $t_1$ .

In general, the intensity at a point  $\vec{x}$  on a Lambertian object is given by

$$I(\vec{x}) = f(\rho t E(\vec{N}(\vec{x}))), \tag{4.13}$$

where  $E(\cdot)$  is the irradiance,  $\vec{N}(\vec{x})$  is the surface normal at point  $\vec{x}$ , and  $t$  is the exposure time, and  $\rho$  is the constant reflectance of the surface. For simplicity, we assume a linear camera response and ignore the effects of exposure time  $t$  and the reflectance term  $\rho$  since their effects on the intensity can be modeled linearly. These assumptions imply that our estimates of the lighting coefficients will be only accurate to within unknown additive and multiplicative terms. Under these assumptions, the relationship between image intensity and irradiance is simply:

$$I(\vec{x}) = E(\vec{N}(\vec{x})). \tag{4.14}$$

### 4.1.3 Estimating lighting environments

Since, under our assumptions, the intensity is equal to irradiance, Equation (4.14) can be written in terms of spherical harmonics by expanding Equation (4.7):

$$\begin{aligned}
I(\vec{x}) &= l_{0,0}\pi Y_{0,0}(\vec{N}) + l_{1,-1}\frac{2\pi}{3}Y_{1,-1}(\vec{N}) + l_{1,0}\frac{2\pi}{3}Y_{1,0}(\vec{N}) + l_{1,1}\frac{2\pi}{3}Y_{1,1}(\vec{N}) \\
&\quad + l_{2,-2}\frac{\pi}{4}Y_{2,-2}(\vec{N}) + l_{2,-1}\frac{\pi}{4}Y_{2,-1}(\vec{N}) + l_{2,0}\frac{\pi}{4}Y_{2,0}(\vec{N}) \\
&\quad + l_{2,1}\frac{\pi}{4}Y_{2,1}(\vec{N}) + l_{2,2}\frac{\pi}{4}Y_{2,2}(\vec{N}).
\end{aligned} \tag{4.15}$$

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:

$$\begin{pmatrix} \pi Y_{0,0}(\vec{N}(\vec{x}_1)) & \frac{2\pi}{3}Y_{1,-1}(\vec{N}(\vec{x}_1)) & \dots & \frac{\pi}{4}Y_{2,2}(\vec{N}(\vec{x}_1)) \\ \pi Y_{0,0}(\vec{N}(\vec{x}_2)) & \frac{2\pi}{3}Y_{1,-1}(\vec{N}(\vec{x}_2)) & \dots & \frac{\pi}{4}Y_{2,2}(\vec{N}(\vec{x}_2)) \\ \vdots & \vdots & \ddots & \vdots \\ \pi Y_{0,0}(\vec{N}(\vec{x}_p)) & \frac{2\pi}{3}Y_{1,-1}(\vec{N}(\vec{x}_p)) & \dots & \frac{\pi}{4}Y_{2,2}(\vec{N}(\vec{x}_p)) \end{pmatrix} \begin{pmatrix} l_{0,0} \\ l_{1,-1} \\ \vdots \\ l_{2,2} \end{pmatrix} = \begin{pmatrix} I(\vec{x}_1) \\ I(\vec{x}_2) \\ \vdots \\ I(\vec{x}_p) \end{pmatrix}, \tag{4.16}$$

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

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:

$$\vec{v} = (M^T M)^{-1} M^T \vec{b}. \quad (4.17)$$

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 chapter 2, we observe that under 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:

$$I(\vec{x}) = A + l_{1,-1} \frac{2\pi}{3} Y_{1,-1}(\vec{N}) + l_{1,1} \frac{2\pi}{3} Y_{1,1}(\vec{N}) + l_{2,-2} \frac{\pi}{4} Y_{2,-2}(\vec{N}) + l_{2,2} \frac{\pi}{4} Y_{2,2}(\vec{N}), \quad (4.18)$$

where:

$$A = l_{0,0} \frac{\pi}{2\sqrt{\pi}} - l_{2,0} \frac{\pi}{16} \sqrt{\frac{5}{\pi}}. \quad (4.19)$$

Note that the functions  $Y_{i,j}(\cdot)$  depend only on the  $x$  and  $y$  components of the surface normal  $\vec{N}$ . Therefore, the five lighting coefficients can be estimated from 2-D surface normals, which are relatively simple to estimate from a single image.<sup>1</sup> In addition, Equation (4.18) is still linear in its now five lighting environment coefficients, which can be estimated as the least-squares solution to:

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

which can be written more simply as:

$$M\vec{v} = \vec{b}. \quad (4.20)$$

This system has the same least-squares solution as before:

$$\vec{v} = (M^T M)^{-1} M^T \vec{b}. \quad (4.21)$$

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.

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

<sup>1</sup>The 2-D surface normal is the gradient vector of an implicit curve fit to the edge of an object.



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:

$$E(\vec{v}) = \|M\vec{v} - \vec{b}\|^2 + \lambda\|C\vec{v}\|^2, \quad (4.22)$$

where  $\lambda$  is a scalar, and the matrix  $C$  is diagonal with ( 1 2 2 3 3 ) 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 (4.16), the matrix  $C$  has ( 1 2 2 2 3 3 3 3 3 ) on the diagonal.

The error function to be minimized, Equation (4.22), is a least-squares problem with a Tikhonov regularization [21]. The analytic minimum is found by differentiating with respect to  $\vec{v}$ :

$$\begin{aligned} \frac{\partial E(\vec{v})}{\partial \vec{v}} &= 2M^T M\vec{v} - 2M^T \vec{b} + 2\lambda C^T C\vec{v}, \\ &= 2(M^T M + \lambda C^T C)\vec{v} - 2M^T \vec{b}, \end{aligned} \quad (4.23)$$

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

$$\vec{v} = (M^T M + \lambda C^T C)^{-1} M^T \vec{b}. \quad (4.24)$$

In practice, we have found that the conditioned estimate in Equation (4.24) 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 (4.21), can be used, though both estimates will give similar results for small values of  $\lambda$ .

#### 4.1.4 Comparing lighting environments

The estimated coefficient vector  $\vec{v}$ , Equation (4.24), is a low-order approximation of the lighting environment. For forensic 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:

$$\vec{x}_1 = M\vec{v}_1, \quad (4.25)$$

$$\vec{x}_2 = M\vec{v}_2, \quad (4.26)$$

where the matrix  $M$  is of the form in Equation (4.16) (for 3-D normals) or Equation (4.20) (for 2-D normals). After subtracting the mean, the correlation between these zero-mean profiles is:

$$\text{corr}(\vec{x}_1, \vec{x}_2) = \frac{\vec{x}_1^T \vec{x}_2}{\|\vec{x}_1\| \|\vec{x}_2\|}. \quad (4.27)$$

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

$$\text{corr}(\vec{v}_1, \vec{v}_2) = \frac{\vec{v}_1^T Q \vec{v}_2}{\sqrt{\vec{v}_1^T Q \vec{v}_1} \sqrt{\vec{v}_2^T Q \vec{v}_2}}, \quad (4.28)$$

where the matrix  $Q$  is derived below for both the 2-D and 3-D cases.

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:

$$D(\vec{v}_1, \vec{v}_2) = \frac{1}{2} (1 - \text{corr}(\vec{v}_1, \vec{v}_2)), \quad (4.29)$$

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

### Matrix $Q$ for 2-D and 3-D correlation

The matrix  $Q$  for the 2-D correlation, notated  $Q_2$ , is derived by integrating products of the functions in the matrix  $M$  from Equation (4.20) about the unit circle. First, consider the average value of the irradiance profile, found by integrating Equation (4.18) around the unit circle. Each of the spherical harmonics integrates to zero, thus the average value is simply the ambient term  $A$ . From this observation, the zero-mean irradiance profile for vector  $\vec{v}$  is given by:

$$\vec{x} = \hat{M}\vec{v}, \quad (4.30)$$

where  $\hat{M}$  is the matrix  $M$  from Equation (4.20) with the first column of ones replaced with zeros. The numerator of Equation (4.28) can then be rewritten as an inner product of irradiance profiles:

$$\vec{x}_1^T \vec{x}_2 = (\hat{M}\vec{v}_1)^T (\hat{M}\vec{v}_2) = \vec{v}_1^T (\hat{M}^T \hat{M}) \vec{v}_2 = \vec{v}_1^T Q_2 \vec{v}_2. \quad (4.31)$$



	$l_{0,0}$	$l_{1,-1}$	$l_{1,0}$	$l_{1,1}$	$l_{2,-2}$	$l_{2,-1}$	$l_{2,0}$	$l_{2,1}$	$l_{2,2}$	3-D	2-D
GRC	0.44	0.35	-0.18	-0.06	-0.05	-0.22	-0.10	0.21	-0.05	2.6	2.7
GAL	0.76	0.34	-0.19	0.54	0.50	-0.10	-0.27	-0.14	0.42	4.4	0.8
EUC	0.43	0.36	0.03	-0.10	-0.06	-0.01	-0.13	-0.05	-0.00	0.2	0.1
STP	0.26	0.14	-0.01	0.02	0.01	-0.03	-0.08	0.00	-0.03	6.4	1.7
UFZ	0.31	0.37	-0.00	-0.01	-0.02	-0.01	-0.27	0.00	-0.24	2.5	1.4

**Table 4.1:** Lighting environment coefficients and estimation errors from different lighting environments. The 3-D and 2-D errors have exponent  $10^{-4}$ .

## 4.2 Results

We tested our technique for estimating lighting environment coefficients on synthetically generated images and real images of natural lighting environments. The synthetic images were rendered using the `pbrt` environment [42] with data from a gallery of light probe images maintained by Paul Debevec [11]. The natural images were obtained in two different ways. For the first set, we photographed a known target in a variety of lighting conditions. For the second set, we downloaded twenty images from Flickr, a popular image sharing website [60]. Results from four visually plausible forgeries are also presented. For all images, the lighting environment coefficients were estimated from the green channel of the image. Although all three color channels could be analyzed, we find that this is often unnecessary since the estimation is invariant to both multiplicative and additive terms.

### 4.2.1 Simulation

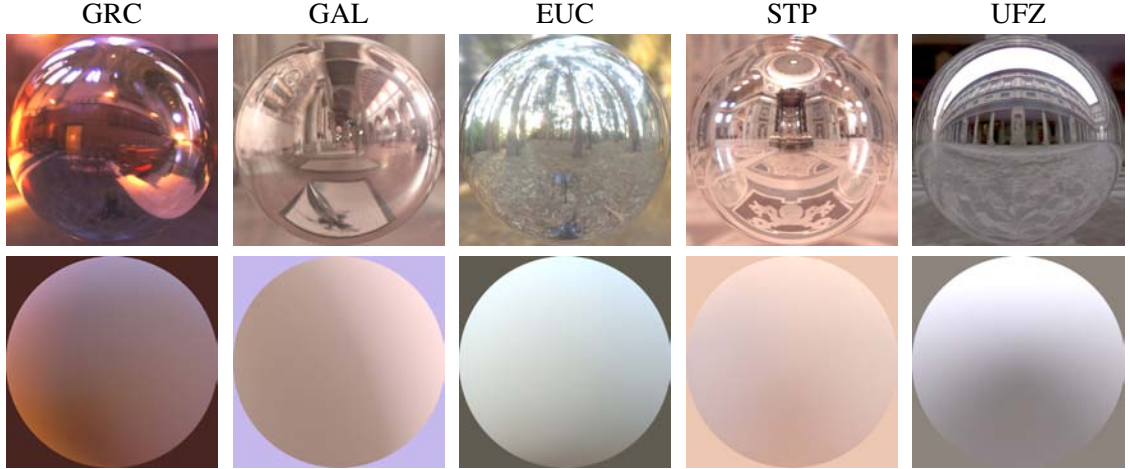
Lighting environments can be captured by a variety of methods, such as photographing a mirror sphere [11], or through panoramic photography techniques. These methods produce high dynamic range images, known as light probe images, that represent the lighting environment function  $L(\vec{V})$ . The spherical harmonic coefficients are computed by integrating the lighting environment function  $L(\vec{V})$  against the corresponding spherical harmonic basis function [48]:

$$l_{n,m} = \int_{\Omega} L(\vec{V}) Y_{n,m}(\vec{V}) d\Omega. \quad (4.36)$$

Shown in Table 4.1 are nine lighting environment coefficients computed from five different light probe images. The light probes, Figure 4.5, were captured in the following locations: Grace Cathedral, San Francisco (GRC); Galileo’s Tomb, Florence (GAL); a Eucalyptus Grove, UC Berkeley (EUC); St. Peter’s Basilica, Rome (STP); and the Uffizi Gallery, Florence (UFZ).<sup>2</sup>

These lighting environment coefficients were used to render a Lambertian sphere in each of the five lighting environments, Figure 4.5. Using the known geometry of these spheres, the lighting environment coefficients were estimated in two different ways: with 3-D surface normals from the visible side of the sphere, and with 2-D surface normals along the occluding contour. In both cases,

<sup>2</sup>Light probe images ©1998, 1999 Paul Debevec, available at <http://www.debevec.org/Probes>.



**Figure 4.5:** Shown along the top row are five light probes from different lighting environments, from which lighting coefficients are computed (Table 4.1). Shown in the bottom row are Lambertian spheres rendered from these coefficients.

the regularization term  $\lambda$  in Equation (4.24) was set to 0.01.

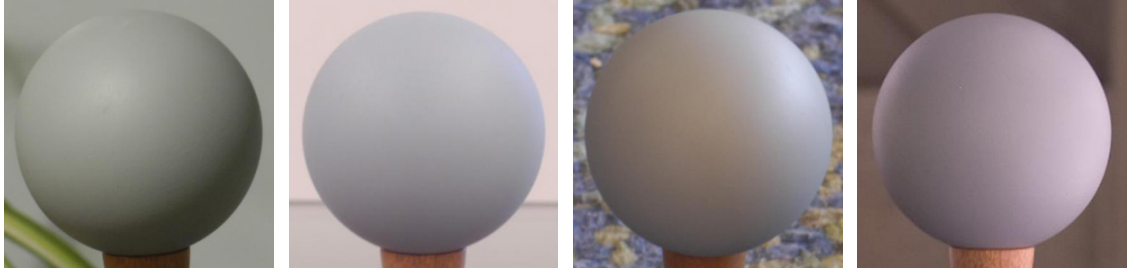
The estimation errors are reported in the last two columns of Table 4.1. For the 2-D case, the errors are computed between the five estimated coefficients and the corresponding subset of actual coefficients ( $l_{0,0}, l_{1,-1}, l_{1,1}, l_{2,-2}, l_{2,2}$ ). Overall, the errors are less than 0.001; for comparison, the average error between all ten pairs of different lighting environments is 0.13 with a minimum of 0.015.

## 4.2.2 Spheres

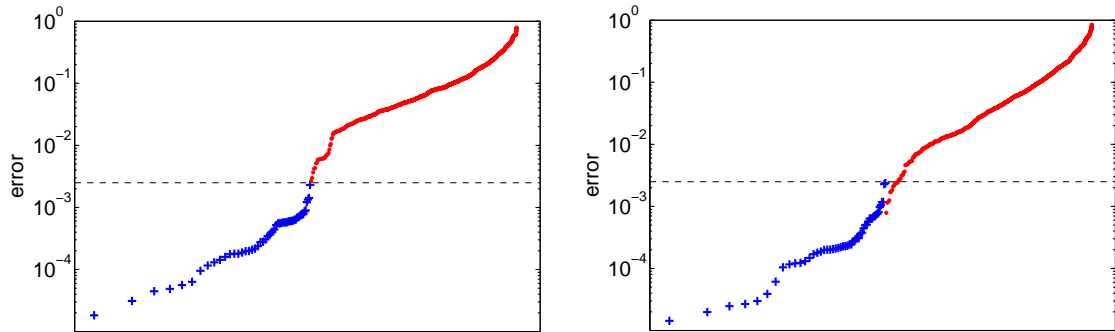
To test our ability to discriminate between lighting environments in real images, we photographed a diffuse sphere in 28 different locations with a 6.3 megapixel Nikon D100 digital camera set to capture in high-quality JPEG mode. The focal length was set to 70 mm, the  $f$ -stop was fixed at  $f/8$ , and the shutter speed was varied to capture two or three exposures per location. In total, there were 68 images, four of which are shown in Figure 4.6.

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  $\lambda$  in Equation (4.24) was set to 0.01.

For each pair of images, the error, Equation (4.29), 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 4.7. In both plots, the 52 image pairs from the same location are plotted first (blue ‘+’), sorted by error. The 2226 pairs from different locations are plotted next (red ‘.’). Note that the axes are scaled logarithmically in both plots.



**Figure 4.6:** A diffuse sphere photographed in four different lighting environments.



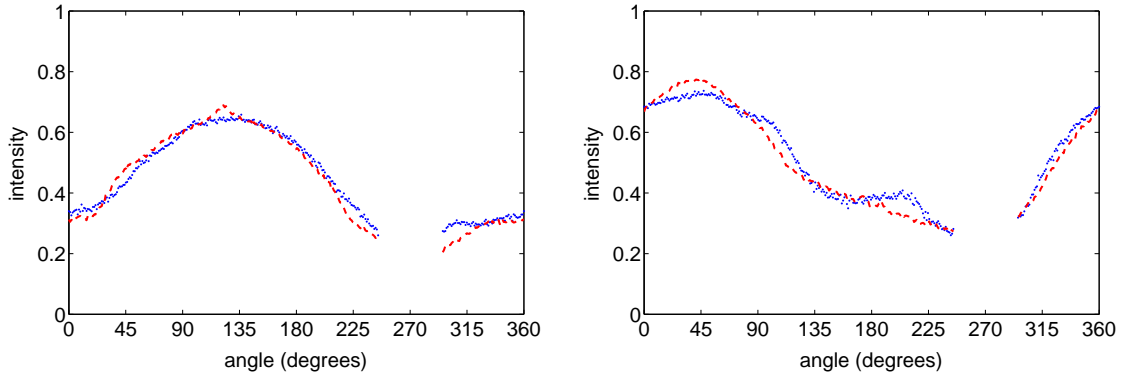
**Figure 4.7:** Errors between image pairs corresponding to the same (blue '+') and different (red '.') locations using the full 9-parameter model with 3-D surface normals (left) and using the 5-parameter model with 2-D surface normals (right). Both the horizontal and vertical axes are scaled logarithmically.

For the 3-D 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 the threshold of 0.0025. These image pairs correspond to lighting environments that are indistinguishable based on the five coefficient model. For example, two of these indistinguishable lighting environment pairs are shown in Figure 4.8. In each plot, the red (dashed) and blue (dotted) lines are from different lighting environments, where the 2-D error between these environments is less than 0.0025. Both plots illustrate that different lighting environments can create similar intensity profiles, and low-order approximations of these profiles will be unable to capture the differences. Therefore, while large errors indicate different lighting environments, small errors can only indicate indistinguishable lighting environments.

### 4.2.3 Photographs

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



**Figure 4.8:** Shown in each panel are intensity profiles from a pair of spheres in indistinguishable lighting environments. In each case, the error between the red dashed and blue dotted profiles is below the threshold of 0.0025. (The gap in the profiles corresponds to the sphere’s mounting stand for which no intensity values are available.)

under these conditions, we downloaded twenty images of multiple objects in natural lighting environments from Flickr [60], Figure 4.9.

In each image, the 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 4.10. 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 (4.24). The regularization term  $\lambda$  in Equation (4.24) was increased to 0.1, which is larger than in the simulation due to sensitivity to noise (see section 4.2.4).

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.

The objects with the maximum error of 0.027 are 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 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  $200^\circ$ , 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.





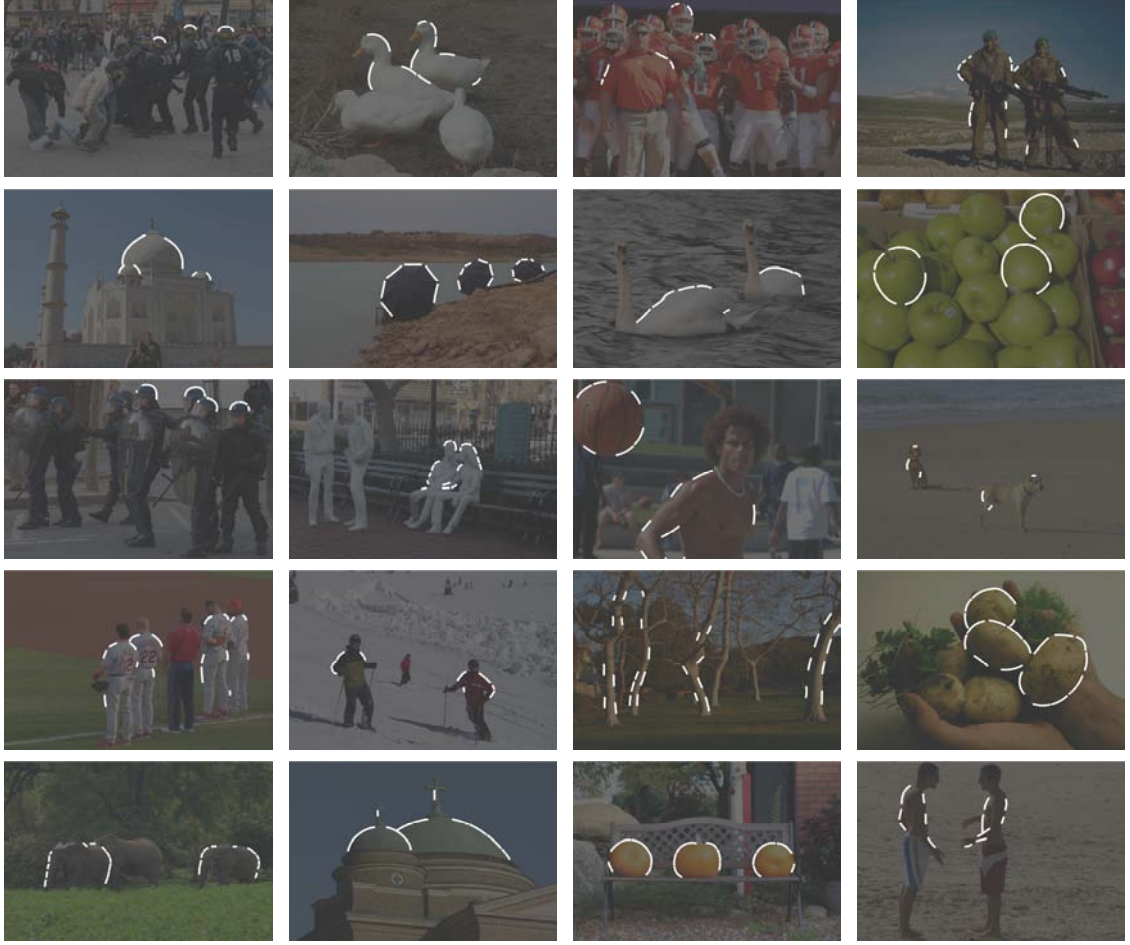
**Figure 4.9:** Twenty images of multiple objects in natural lighting environments, see also Figure 4.10.

#### 4.2.4 Sensitivity

We explored the sensitivity of the estimate to surface normal extent in the presence of additive noise and JPEG compression. Random lighting environments were generated by picking coefficients according to a unit-variance and zero-mean Gaussian distribution. To simulate natural lighting, the coefficients at order  $n$  were scaled so that the average power was proportional to  $1/n^2$  [12]. From each lighting environment, we rendered images of spheres and added Gaussian noise with standard deviation equal to 5% of the intensity range of the image. From each image, the coefficient vector  $\vec{v}$  was estimated, Equation (4.24), with  $\lambda = 0.01$ . The surface normals were limited to a specified extent, from  $30^\circ$  to  $360^\circ$ , about the primary illuminant direction. The surface normal extent affects the stability of the estimate  $\vec{v}$ , which can be formalized by computing the sensitivity of  $\vec{v}$  to perturbations in  $M$  [54]:

$$\kappa(M) + \frac{\kappa(M)^2 \tan \theta}{\eta}, \quad (4.37)$$





**Figure 4.10:** Superimposed on each image from Figure 4.9 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 (4.20).

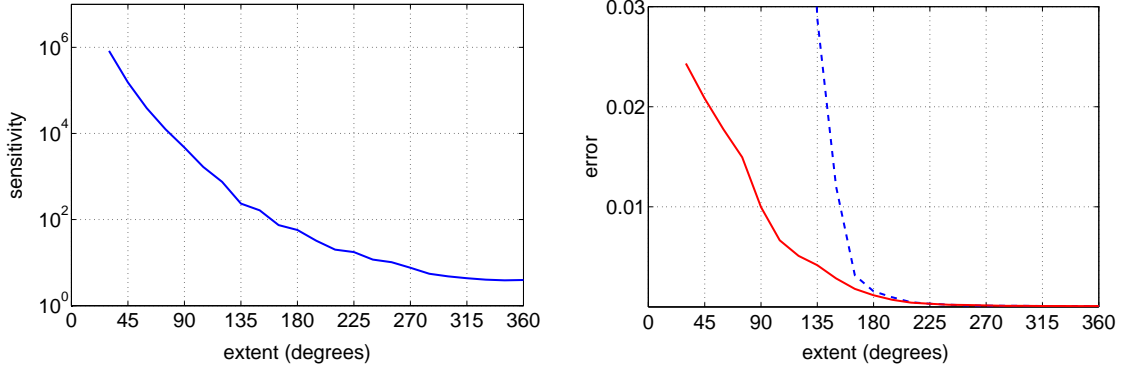
where  $\kappa(M) = \sigma_{\max}/\sigma_{\min}$  is the condition number of the matrix  $M$  (ratio of the largest to smallest singular value), and  $\theta$  and  $\eta$  are:

$$\theta = \cos^{-1} \left( \frac{\|M\vec{v}\|}{\|\vec{b}\|} \right), \quad (4.38)$$

$$\eta = \sigma_{\max} \|\vec{v}\| / \|M\vec{v}\|. \quad (4.39)$$

As shown in the left panel of Figure 4.11, the sensitivity, Equation (4.37), increases dramatically as the extent of surface normals decreases, indicating potential instability of the estimate  $\vec{v}$ . Shown in the right panel of Figure 4.11 is the error averaged over 2000 random environments per surface normal extent for both the conditioned (solid red) and unconditioned systems (dashed blue), Equations (4.21) and (4.24). Note that the conditioned system provides considerably more accurate results when the surface normal extent is below  $180^\circ$ .

The sensitivity to JPEG compression was also tested. As above, we generated random lighting environments and rendered images of spheres in these environments. These images were then saved



**Figure 4.11:** Shown on the left is the sensitivity, Equation (4.37), of the least-squares problem, Equation (4.16), as a function of the surface normal extent (note that the vertical axis is scaled logarithmically). Shown on the right is the average error between the estimated and actual lighting environment vectors as a function of surface normal extent. Each data point corresponds to the error averaged over 2000 random lighting environments. The dashed blue curve corresponds to the unconditioned solution, Equation (4.20), and is largely unstable for a surface normal extent less than  $180^\circ$ . The solid red curve corresponds to the conditioned solution, Equation (4.22), and is substantially more stable.

with a JPEG quality between 5 and 100 (in a range of 0, 100). The lighting environment coefficients were estimated from surface normals spanning a range of  $135^\circ$ . For JPEG quality of 5, the average error over 2000 random trials is 0.03. For a quality between 10–35, the average error is 0.01; for a quality between 40–65, the average error is 0.005; and for a quality between 70–100, the average error is 0.002. Note that for JPEG quality between 40–100, the errors are comparable or less than the errors introduced from additive noise, Figure 4.11.

## 4.2.5 Forgeries

We created three forgeries by mixing several of the images in Figure 4.9, and we downloaded one forgery from Worth1000, a Photoshop contest website [58]. These forgeries are shown in Figure 4.12 and Figure 4.13.

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 4.12 and Figure 4.13. Surface normals and intensities along these occluding contour were extracted, from which the five lighting environment coefficients were estimated, Equation (4.24), with the regularization term  $\lambda = 0.1$ .

Shown in each figure is a sphere rendered with the estimated coefficients. These spheres qualitatively show discrepancies between the estimated lighting environments. The calculated errors between object pairs are summarized in Table 4.2. For all pairs of objects originally in the same lighting environment (above the horizontal line), the average error is 0.005 with maximum error of 0.01. For pairs of objects from different lighting environments (below the horizontal line), the average error is 0.15 with a minimum error of 0.03.

Police		Umbrellas		Soldiers		Snoop Dogg	
pair	error	pair	error	pair	error	pair	error
1, 2	0.006	1, 2	0.010	2, 3	0.002		
3, 4	0.004	3, 4	0.004				
1, 3	0.047	1, 3	0.152	1, 2	0.109	1, 2	0.388
1, 4	0.033	1, 4	0.194	1, 3	0.138		
2, 3	0.076	2, 3	0.229				
2, 4	0.054	2, 4	0.277				

**Table 4.2:** Errors between pairs of objects in the forgeries of Figure 4.12 and Figure 4.13.

### 4.3 Discussion

When creating a composite of two or more people, it is often difficult to exactly match the lighting, even if it seems perceptually consistent. The reason for this difficulty 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.

We showed the efficacy of this approach on a broad range of simulated images, photographic images, and visually plausible forgeries. In each case, the model parameters can be well approximated, from which differences in lighting can typically be detected. There are, however, instances when different lighting environments give rise to similar model coefficients—in these cases the lighting differences are indistinguishable.

The ability to estimate complex lighting environments was motivated by the illuminant direction technique presented in chapter 2. The approach in this chapter generalizes the illuminant direction 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  $\tan^{-1}(l_{1,-1}/l_{1,1})$ .

While any forensic tool is vulnerable to countermeasures, the precise matching of lighting in an image can be difficult, although certainly not impossible. And the forger will need to keep in mind that there are other ways to estimate properties of the lighting, including the highlights in a subject’s eyes.



**Figure 4.12:** 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 (see also Table 4.2).



**Figure 4.13:** Shown on the left is a forgery where the head of rapper Snoop Dogg has been placed on the body of an orchestra conductor. Shown on the right are the analyzed regions superimposed in white, and spheres rendered from the estimated lighting coefficients (see also Table 4.2).

# Chapter 5

## Chromatic aberration

Most images contain a variety of aberrations that result from imperfections and artifacts of the optical imaging system. In an ideal imaging system, light passes through the lens and is focused to a single point on the sensor. Optical systems, however, deviate from such ideal models in that they fail to perfectly focus light of all wavelengths. The resulting effect is known as chromatic aberration and it occurs in two forms: longitudinal and lateral. Longitudinal aberration manifests itself as differences in the focal planes for different wavelengths of light. Lateral aberration manifests itself as a spatial shift in the locations where light of different wavelengths reach the sensor—this shift is proportional to the distance from the optical center. In both cases, chromatic aberration leads to various forms of color imperfections in the image. To a first-order approximation, longitudinal aberration can be modeled as a convolution of the individual color channels with an appropriate low-pass filter. Lateral aberration, on the other hand, can be modeled as an expansion/contraction of the color channels with respect to one another. When tampering with an image, these aberrations are often disturbed and fail to be consistent across the image.

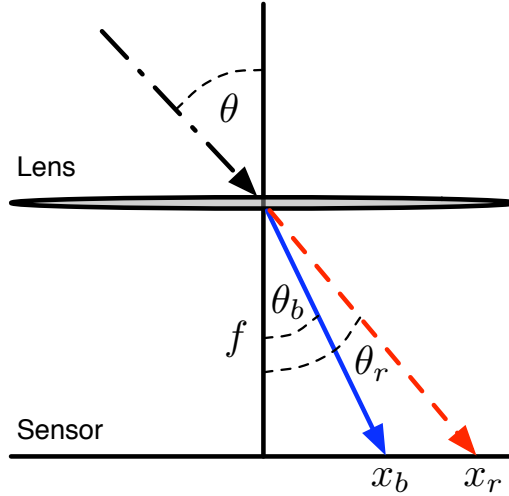
In this chapter, we describe a computational technique based on maximizing mutual information for automatically estimating lateral chromatic aberration. Although we eventually plan to incorporate longitudinal chromatic aberration, only lateral chromatic aberration is considered here. We show the efficacy of this approach for detecting digital tampering in synthetic and real images.

### 5.1 Methods

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

$$n \sin(\theta) = n_f \sin(\theta_f), \quad (5.1)$$

where  $\theta$  is the angle of incidence,  $\theta_f$  is the angle of refraction, and  $n$  and  $n_f$  are the refractive indices of the media through which the light passes, Figure 5.1. The refractive index of glass,  $n_f$ , depends on the wavelength of the light that traverses it. This dependency results in polychromatic light being



**Figure 5.1:** The refraction of light in one dimension. Polychromatic light enters the lens at an angle  $\theta$ , and emerges at an angle which depends on wavelength. As a result, different wavelengths of light, two of which are represented as the red (dashed) and the blue (solid) rays, will be imaged at different points,  $x_r$  and  $x_b$ .

split according to wavelength as it exits the lens and strikes the sensor. Figure 5.1, for example, is a schematic showing the splitting of short wavelength (solid blue ray) and long wavelength (dashed red ray) light. The result of this splitting of light is termed lateral chromatic aberration.

Lateral chromatic aberration can be quantified with a low-parameter model. Consider, for example, the position of the short wavelength (solid blue ray) and the long wavelength (dashed red ray) light on the sensor,  $x_r$  and  $x_b$ , shown in Figure 5.1. The relationship between the angle of incidence and angle of refraction is given by Snell's law, Equation (5.1), yielding:

$$\begin{aligned}\sin(\theta) &= n_r \sin(\theta_r), \\ \sin(\theta) &= n_b \sin(\theta_b),\end{aligned}$$

which are combined to yield:

$$n_r \sin(\theta_r) = n_b \sin(\theta_b). \quad (5.2)$$

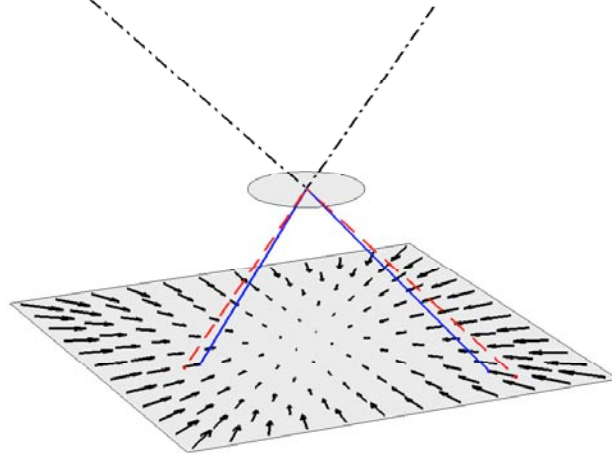
Dividing both sides by  $\cos(\theta_b)$  gives:

$$\begin{aligned}n_r \sin(\theta_r) / \cos(\theta_b) &= n_b \tan(\theta_b), \\ &= n_b x_b / f,\end{aligned} \quad (5.3)$$

where  $f$  is the lens-to-sensor distance. If we assume that the differences in angles of refraction are relatively small, then  $\cos(\theta_b) \approx \cos(\theta_r)$ . Equation (5.3) then takes the form:

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





**Figure 5.2:** The refraction of light in two dimensions. Polychromatic light enters the lens and emerges at an angle which depends on wavelength. As a result, different wavelengths of light, two of which are represented as the red (dashed) and the blue (solid) rays, will be imaged at different points. The vector field shows the amount of deviation across the image.

$$\begin{aligned}
 n_r x_r / f &\approx n_b x_b / f, \\
 n_r x_r &\approx n_b x_b, \\
 x_r &\approx \alpha x_b,
 \end{aligned} \tag{5.4}$$

where  $\alpha = n_b/n_r$ . This low-parameter model generalizes for any two wavelengths of light, where  $\alpha$  is a function of these wavelengths.

### 5.1.1 2-D Aberration

For a two-dimensional lens and sensor, the distortion caused by lateral chromatic aberration takes a form similar to Equation (5.4). In 2-D, an incident ray reaches the lens at angles  $\theta$  and  $\phi$ , relative to the  $x = 0$  and  $y = 0$  planes, respectively. Applying Snell's law yields:

$$\begin{aligned}
 n_r \sin(\theta_r) &= n_b \sin(\theta_b) \\
 n_r \sin(\phi_r) &= n_b \sin(\phi_b),
 \end{aligned}$$

and following the derivation for the 1-D model yields the following 2-D model:

$$(x_r, y_r) \approx \alpha(x_b, y_b). \tag{5.5}$$

Shown in Figure 5.2 is vector-based depiction of this aberration, where each vector  $\vec{v}$  is the difference between the positions of the short wavelength light and long wavelength light,  $\vec{v} = (x_r - x_b, y_r - y_b)$ . Note that this model is simply an expansion/contraction about the center of the image. In real lenses, the center of optical aberrations is often different from the image center due to



the complexities of multi-lens systems [57]. The previous model can therefore be augmented with an additional two parameters,  $(x_0, y_0)$ , to describe the position of the expansion/contraction center. The model now takes the form:

$$x_r = \alpha(x_b - x_0) + x_0, \quad (5.6)$$

$$y_r = \alpha(y_b - y_0) + y_0. \quad (5.7)$$

It is common for lens designers to try to minimize chromatic aberration in lenses. This is usually done by combining lenses with different refractive indices to align the rays for different wavelengths of light. If two wavelengths are aligned, the lens is called an achromatic doublet or achromat. It is not possible for all wavelengths that traverse an achromatic doublet to be aligned and the residual error is known as the secondary spectrum. The secondary spectrum is visible in high-contrast regions of an image as a magenta or green halo [23].

### 5.1.2 Estimating Chromatic Aberration

In the previous section, a model for lateral chromatic aberration was derived, Equations (5.6) and (5.7). This model describes the relative positions at which light of varying wavelengths strike the sensor. With a three color channel RGB image, we assume that the lateral chromatic aberration is constant within each color channel. Using the green channel as reference, we would like to estimate the aberration between the red and green channels, and between the blue and green channels. Deviations or inconsistencies in these models will then be used as evidence of tampering.

Recall that the model for lateral chromatic aberration consists of three parameters, two parameters for the center of the distortion and one parameter for the magnitude of the distortion. These model parameters will be denoted  $(x_1, y_1, \alpha_1)$  and  $(x_2, y_2, \alpha_2)$  for the red to green and blue to green distortions, respectively.

The estimation of these model parameters can be framed as an image registration problem [4]. Specifically, lateral chromatic aberration results in an expansion or contraction between the color channels, and hence a misalignment between the color channels. We, therefore, seek the model parameters that bring the color channels back into alignment. There are several metrics that may be used to quantify the alignment of the color channels. To help contend with the inherent intensity differences across the color channels we employ a metric based on mutual information that has proven successful in such situations [55]. We have found that this metric achieves slightly better results than a simpler correlation coefficient metric (with little difference in the run-time complexity<sup>1</sup>). Other metrics, however, may very well achieve similar or better results.

We will describe the estimation of the red to green distortion parameters (the blue to green estimation follows a similar form). Denote the red channel of a RGB image as  $R(x, y)$  and the

---

<sup>1</sup>The run-time complexity is dominated by the interpolation necessary to generate  $R(x_r, y_r)$ , and not the computation of mutual information.

green channel as  $G(x, y)$ . A corrected version of the red channel is denoted as  $R(x_r, y_r)$  where:

$$x_r = \alpha_1(x - x_1) + x_1, \quad (5.8)$$

$$y_r = \alpha_1(y - y_1) + y_1. \quad (5.9)$$

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

$$\operatorname{argmax}_{x_1, y_1, \alpha_1} I(\mathcal{R}; \mathcal{G}), \quad (5.10)$$

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

$$I(\mathcal{R}; \mathcal{G}) = \sum_{r \in \mathcal{R}} \sum_{g \in \mathcal{G}} P(r, g) \log \left( \frac{P(r, g)}{P(r)P(g)} \right), \quad (5.11)$$

where  $P(\cdot, \cdot)$  is the joint probability distribution, and  $P(\cdot)$  is the marginal probability distribution.

This measure of mutual information is maximized using a brute-force iterative search. On the first iteration, a relatively coarse sampling of the parameter space for  $x_1, y_1, \alpha_1$  is searched. On the second iteration, a refined sampling of the parameter space is performed about the maximum from the first stage. This process is repeated for  $N$  iterations.

In order to quantify the error between the estimated and known model parameters, we compute the average angular error between the displacement vectors at every pixel. Specifically, let  $x_0, y_0, \alpha_0$  be the actual parameters and let  $x_1, y_1, \alpha_1$  be the estimated model parameters. The vector displacement fields for these distortions are:

$$\vec{v}_0(x, y) = \begin{pmatrix} (\alpha_0(x - x_0) + x_0) - x \\ (\alpha_0(y - y_0) + y_0) - y \end{pmatrix}, \quad (5.12)$$

$$\vec{v}_1(x, y) = \begin{pmatrix} (\alpha_1(x - x_1) + x_1) - x \\ (\alpha_1(y - y_1) + y_1) - y \end{pmatrix}. \quad (5.13)$$

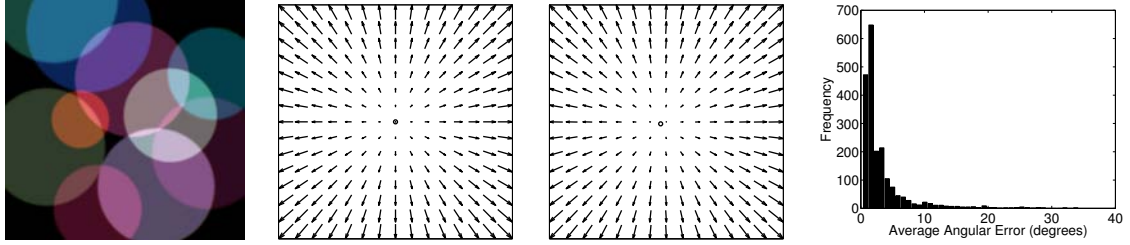
The angular error  $\theta(x, y)$  between any two vectors is:

$$\theta(x, y) = \cos^{-1} \left( \frac{\vec{v}_0(x, y) \cdot \vec{v}_1(x, y)}{\|\vec{v}_0(x, y)\| \|\vec{v}_1(x, y)\|} \right). \quad (5.14)$$

The average angular error,  $\bar{\theta}$ , over all  $P$  pixels in the image is:

$$\bar{\theta} = \frac{1}{P} \sum_{x, y} \theta(x, y). \quad (5.15)$$

To improve reliability, this average is restricted to vectors whose norms are larger than a specified threshold, 0.01 pixels. It is this measure,  $\bar{\theta}$ , that is used to quantify the error in estimating lateral



**Figure 5.3:** Synthetically generated images. Shown are, from left to right, a sample image, the distortion applied to the blue channel (the small circle denotes the distortion center), the estimated distortion, and a histogram of angular errors from 2000 images. For purposes of display, the vector fields are scaled by a factor of 50.

chromatic aberration.

## 5.2 Results

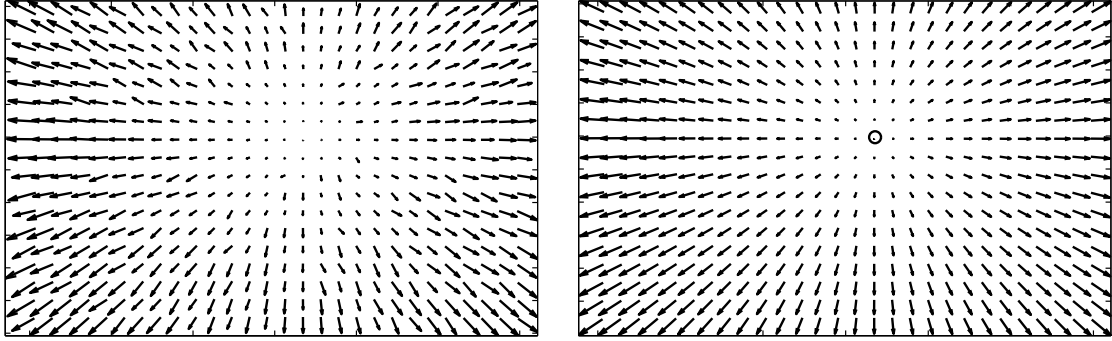
We demonstrate the suitability of the proposed model for lateral chromatic aberration, and the efficacy of estimating this aberration using the mutual information-based algorithm. We first present results from synthetically generated images. Results are then presented for a set of calibrated images photographed under different lenses and lens settings. We also show how inconsistencies in lateral chromatic aberration can be used to detect tampering in visually plausible forgeries.

### 5.2.1 Synthetic images

Synthetic color images of size  $512 \times 512$  were generated as follows. Each image consisted of ten randomly placed anti-aliased discs of various sizes and colors, Figure 5.3. Lateral chromatic aberration was simulated by warping the blue channel relative to the green channel. The center of the distortion,  $(x_2, y_2)$ , was the image center, and the distortion coefficient,  $\alpha_2$ , was chosen between 1.0004 and 1.0078, producing maximum displacements of between 0.1 and 2 pixels. Fifty random images for each of forty values of  $\alpha_2$  were generated for a total of 2000 images.

As described in the previous section, the distortion parameters are determined by maximizing the mutual information for the blue to green distortion. On the first iteration of the brute-force search algorithm, values of  $x_2, y_2$  spanning the entire image were considered, and values of  $\alpha_2$  between 1.0002 to 1.02 were considered. Nine iterations of the search algorithm were performed, with the search space consecutively refined on each iteration.

Shown in the second and third panels of Figure 5.3 are examples of the applied and estimated distortion (the small circle denotes the distortion center). Shown in the fourth panel of Figure 5.3 is the distribution of average angular errors from 2000 images. The average error is  $3.4^\circ$  with 93% of the errors less than  $10^\circ$ . These results demonstrate the general efficacy of the mutual information-based algorithm for estimating lateral chromatic aberration.



**Figure 5.4:** Calibration. In the left panel is an actual red to green chromatic aberration. In the right panel is the best three parameter model fit to this distortion. Note that the actual distortion is well fit by this model. For purposes of display, the vector fields are scaled by a factor of 100.

### 5.2.2 Calibrated images

In order to test the efficacy of our approach on real images, we first estimated the lateral chromatic aberration for two lenses at various focal lengths and apertures. A 6.3 megapixel Nikon D100 digital camera was equipped with a Nikkor 18–35 mm ED lens and a Nikkor 70–300 mm ED lens.<sup>2</sup> For the 18–35 mm lens, focal lengths of 18, 24, 28, and 35 mm with 17  $f$ -stops, ranging from  $f/29$  to  $f/3.5$ , per focal length were considered. For the 70–300 mm lens, focal lengths of 70, 100, 135, 200, and 300 with 19  $f$ -stops, ranging from  $f/45$  to  $f/4$ , per focal length were considered.

A calibration target was constructed of a peg board with 1/4-inch diameter holes spaced one inch apart. The camera was positioned at a distance from the target so that roughly 500 holes appeared in each image. This target was back-illuminated with diffuse lighting, and photographed with each lens and lens setting described above. For each color channel of each calibration image, the center of the holes were automatically computed with sub-pixel resolution. The red to green lateral chromatic aberration was estimated by comparing the relative positions of these centers across the entire image. The displacements between the centers were then modeled as a three parameter expansion/contraction pattern,  $x_1, y_1, \alpha_1$ . These parameters were estimated using a brute force search that minimized the root mean square error between the measured displacements and the model. Shown in the left panel of Figure 5.4 is the actual red to green distortion, and shown in the right panel is the best model fit. Note that while not perfect, the three parameter is a reasonable approximation to the actual distortion. The blue to green aberration was estimated in a similar manner, yielding model parameters  $x_2, y_2, \alpha_2$ . This calibration data was used to quantify the estimation errors from real images of natural scenes.

Images of natural scenes were obtained using the same camera and calibrated lenses. These images of size  $3020 \times 2008$  pixels were captured and stored in uncompressed TIFF format (see below for the effects of JPEG compression). For each of these 205 images, the focal length and  $f$ -stop were extracted from the EXIF data in the image header. The estimated aberration from each

<sup>2</sup>ED lenses help to eliminate secondary chromatic aberration.

image was then compared with the corresponding calibration data with the same lens settings.

The distortion parameters were determined by maximizing the mutual information between the red and green, and blue and green channels. On the first iteration of the brute-force search algorithm, values of  $x_1$ ,  $y_1$  and  $x_2$ ,  $y_2$  spanning the entire image were considered, and values of  $\alpha_1$  and  $\alpha_2$  between 0.9985 to 1.0015 were considered. The bounds on  $\alpha_1$  and  $\alpha_2$  were chosen to include the entire range of the distortion coefficient measured during calibration, 0.9987 to 1.0009. Nine iterations of the search algorithm were performed, with the search space consecutively refined on each iteration.

In the top panel of Figure 5.5 is one of the 205 images. In the second and third panels are the calibrated and estimated blue to green distortions (the small circle denotes the distortion center). In the bottom panel of Figure 5.5 is the distribution of average angular errors, Equation (5.15), from the red to green and blue to green distortions from all 205 images. The average error is  $20.3^\circ$  with 96.6% of the errors less than  $60^\circ$ . Note that the average errors here are approximately six times larger than the synthetically generated images of the previous section. Much of the error is due to other aberrations in the images, such as longitudinal aberration, that are not considered in our current model.

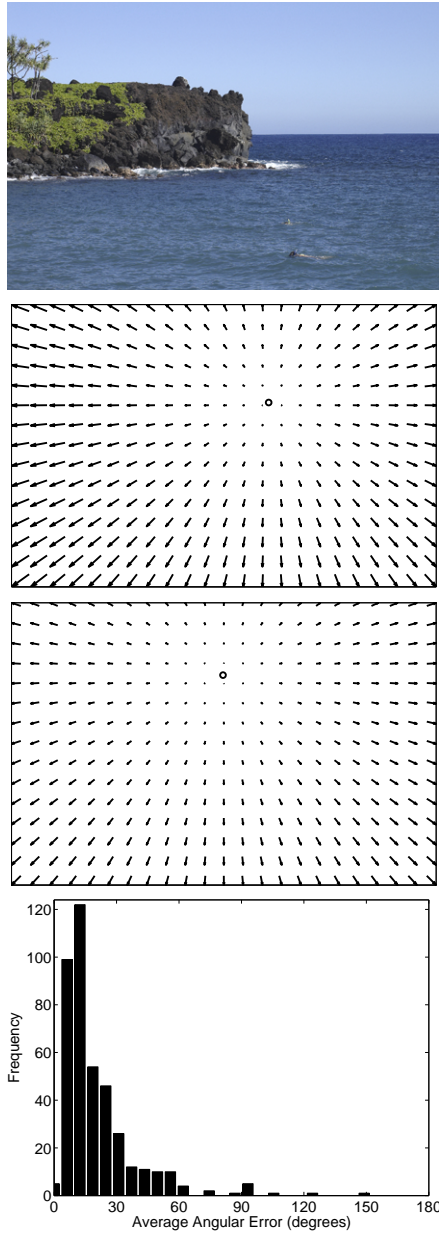
### **JPEG Compression**

The results of the previous section were based on uncompressed TIFF format images. Here we explore the effect of lossy JPEG compression on the estimation of chromatic aberration. Each of the 205 uncompressed images described in the previous section were compressed with a JPEG quality of 95, 85, and 75 (on a scale of 1 to 100). The chromatic aberration was estimated as described above, and the same error metric was computed. For a quality of 95, the average error was  $26.1^\circ$  with 93.7% of the errors less than  $60^\circ$ . For a quality of 85, the average error was  $26.7^\circ$  with 93.4% of the errors less than  $60^\circ$ . For a quality of 75, the average error was  $28.9^\circ$  with 93.2% of the errors less than  $60^\circ$ . These errors should be compared to the uncompressed images with an average error of  $20.3^\circ$  and with 96.6% of the errors less than  $60^\circ$ . While the estimation suffers a bit, it is still possible to estimate, with a reasonable amount of accuracy, chromatic aberration from JPEG compressed images

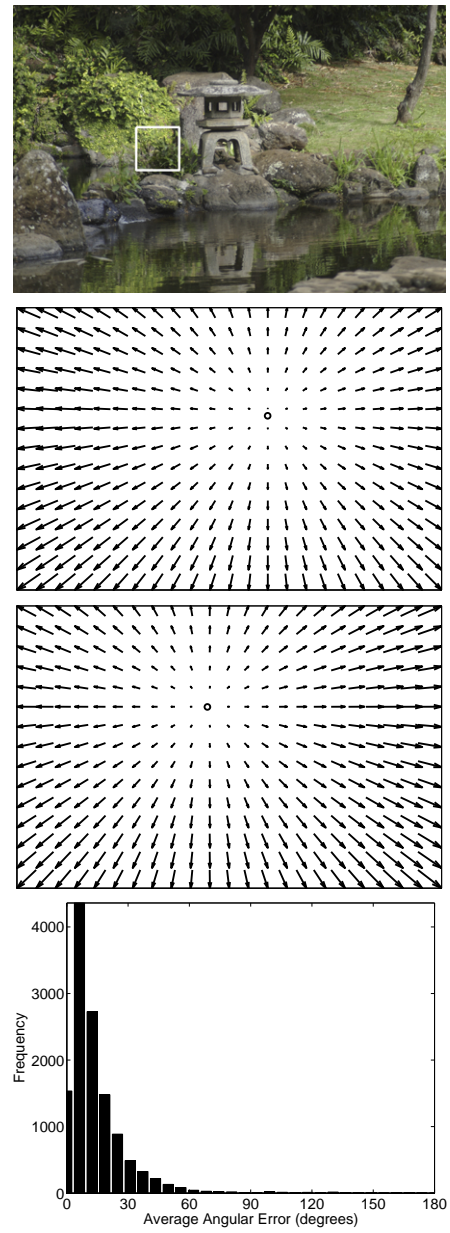
### **5.2.3 Forgeries**

When creating a forgery, it is sometimes necessary to conceal a part of an image with another part of the image or to move an object from one part of an image to another part of an image. These types of manipulations will lead to inconsistencies in the lateral chromatic aberrations, which can therefore be used as evidence of tampering.

In order to detect tampering based on inconsistent chromatic aberration, it is first assumed that only a relatively small portion of an image has been manipulated. With the additional assumption that this manipulation will not significantly affect a global estimate, the aberration is estimated from



**Figure 5.5:** Calibrated images. From top to bottom, one of the 205 images, the calibrated blue to green aberration, the estimated aberration, and a histogram of angular errors from 205 images, for the blue to green and red to green aberrations. For purposes of display, the vector fields are scaled by a factor of 150.



**Figure 5.6:** Block-based estimates. From top to bottom, one of the 205 images with one of the  $300 \times 300$  pixel blocks outlined, the estimated aberration based on the entire image, the estimated aberration based on a single block, and a histogram of 10,250 average angular errors (50 blocks from 205 images) between the image-based and block-based estimates for both the red to green and blue to green aberrations. For purposes of display, the vector fields are scaled by a factor of 150.

the entire image. This global estimate is then compared against estimates from small blocks. Any block that deviates significantly from the global estimate is suspected of having been manipulated.

The 205 calibrated images described in the previous section were each partitioned into overlapping  $300 \times 300$  pixels blocks. It is difficult to estimate chromatic aberration from a block with little or no spatial frequency content (e.g., a largely uniform patch of sky). As such, the average gradient for each image block was computed and only 50 blocks with the largest gradients were considered. The gradient,  $\nabla I(x, y)$ , is computed as follows:

$$\nabla I(x, y) = \sqrt{I_x^2(x, y) + I_y^2(x, y)}, \quad (5.16)$$

where  $I_x(\cdot)$  and  $I_y(\cdot)$  are the horizontal and vertical partial derivatives estimated as follows:

$$I_x(x, y) = I(x, y) \star d(x) \star p(y), \quad (5.17)$$

$$I_y(x, y) = I(x, y) \star d(y) \star p(x), \quad (5.18)$$

where  $\star$  denotes convolution and  $d(\cdot)$  and  $p(\cdot)$  are a pair of 1-D derivative and low-pass filters [15].

In the top panel of Figure 5.6 is one of the 205 images with an outline around one of the  $300 \times 300$  blocks. In the second and third panels are, respectively, estimated blue to green warps from the entire image and from just a single block. In the bottom panel is a histogram of angular errors, Equation (5.15), between estimates based on the entire image and those based on a single block. These errors are estimated over 50 blocks per 205 images, and over the blue to green and red to green estimates. The average angular error is  $14.8^\circ$  with 98.0% less than  $60^\circ$ . These results suggest that inconsistencies in block-based estimates significantly larger than  $60^\circ$  are indicative of tampering.

In the left column of Figure 5.7 are three original images, and in the right column are visually plausible forgeries where a small part of each image was manipulated. For each image, the blue to green and red to green aberration is estimated from the entire image. Each aberration is then estimated for all  $300 \times 300$  blocks with an average gradient above a threshold of 2.5 gray-levels/pixel. The angular error for each block-based estimate is compared with the image-based estimate. Blocks with an average error larger than  $60^\circ$ , and an average distortion larger than 0.15 pixels are considered to be inconsistent with the global estimate, and are used to indicate tampering. The red (dashed outline) blocks in Figure 5.7 reveal the traces of tampering, while the green (solid outline) blocks are consistent with the global estimate and hence authentic. For purpose of display, only a subset of all blocks are shown.

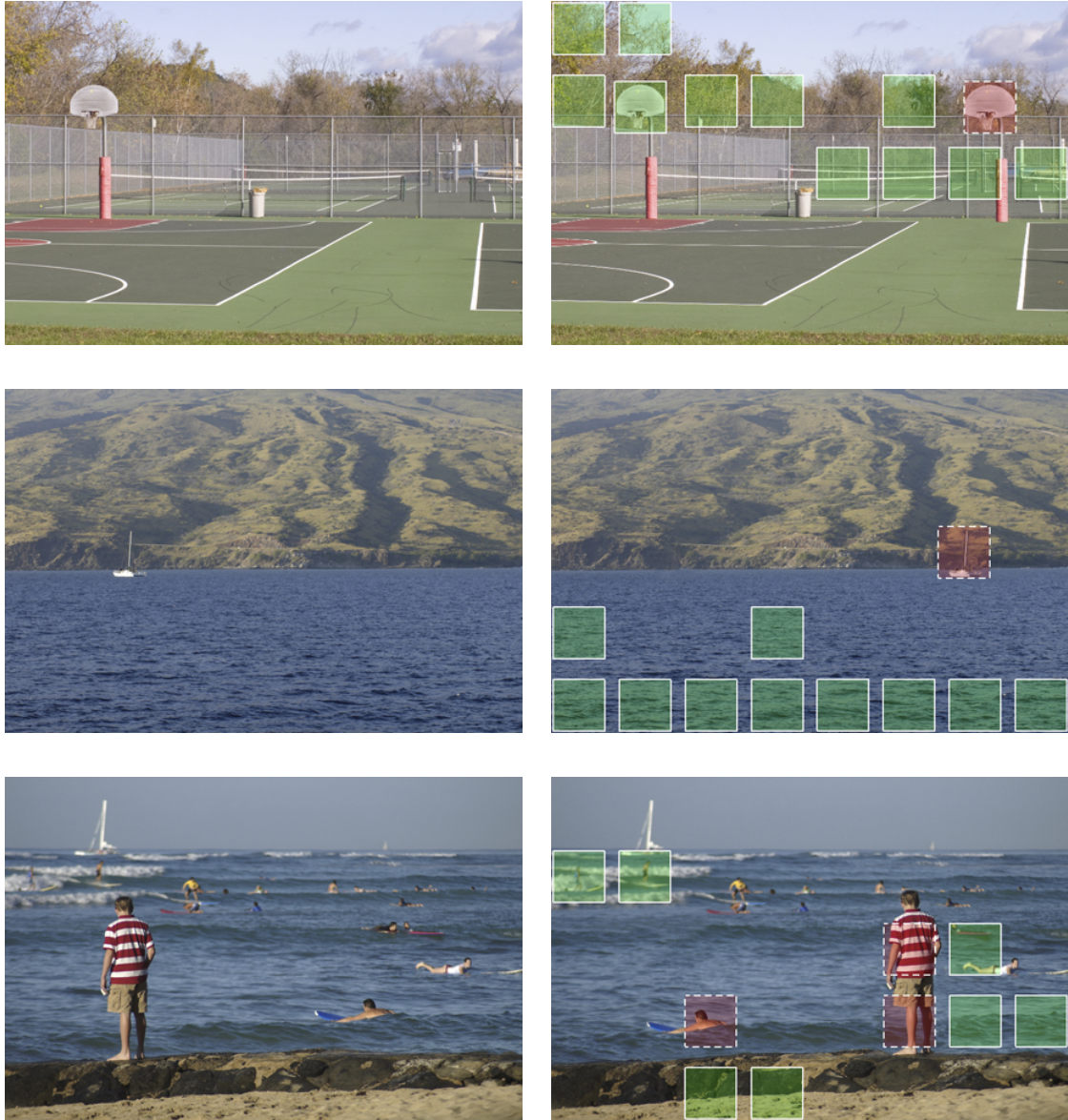
This approach for detecting tampering is effective when the manipulated region is relatively small, allowing for a reliable global estimate. In the case when the tampering may be more significant, an alternate approach may be taken. An image, as above, can be partitioned into small blocks. An estimate of the global aberration is estimated from each block. The estimates from all such blocks are then compared for global consistency. An image is considered to be authentic if the global consistency is within an expected  $60^\circ$ .

### 5.3 Discussion

We have described an image forensic tool that exploits imperfections in a camera's optical system. Our current approach only considers lateral chromatic aberration, which is well approximated by a low-parameter model. We have developed an automatic technique for estimating the model parameters that is based on maximizing the mutual information between color channels. We have also shown the efficacy of this approach in detecting tampering in synthetic and real images.

Techniques exist for estimating lens distortion from images [14] and it may be possible to use these estimates together with estimates of chromatic aberration for ballistics. In other words, a camera make or model may be able to be identified from one or more images. Future work could also consider other aberrations, such as longitudinal chromatic aberration, spherical aberration, and astigmatism. All of these lens aberrations are sources of regularities in natural images that could be useful for forensics.





**Figure 5.7:** Three original images (left) and three image forgeries (right). The red (dashed outline) blocks denote regions that are inconsistent with the global aberration estimate. The green (solid outline) blocks denote regions that are consistent with the global estimate.

# Chapter 6

## Discussion

Although tampering with images is not a new phenomenon, the availability of digital image technology and image processing software makes it easy for anyone to create a forgery. Not surprisingly, tampered images are showing up everywhere, from courtrooms to scientific journals, and these images can have a profound effect on society. There is a clear need for tools to detect forgeries, and the field of digital image forensics has emerged to address this problem without assuming specialized hardware, such as cameras with watermarking technology. Instead of watermarks, current forensic tools assume that images contain statistical regularities from a variety of sources, including the world, the lens, the camera, and the image, and that digital tampering disturbs these regularities. Image forensic tools can expose the tampering by measuring these regularities and detecting changes.

The first forensic tools have primarily exploited the digital sources of regularities: the sensor, the post-processing algorithms, or the image itself. The tools in this dissertation complement previous work by exploring regularities from the world and the lens; specifically, how lighting and optical properties of images can be used for forensics. In this context, we presented four new image forensic tools: illuminant direction, specularly, lighting environment, and chromatic aberration. Together with current statistical techniques, these tools move the problem of creating a convincing forgery out of the hands of novice users and may help to restore public confidence in published images.

When creating a digital composite using objects from different images, it is often difficult to match the lighting on the individual objects. In chapter 2, we described a technique for estimating a 2-D illuminant direction from objects in an image—strong inconsistencies in estimates across the image were evidence of tampering. We also derived two variations of the technique from the same framework: one allowed for differences in an object’s reflectance function and the other allowed for a local light source. We demonstrated results on a variety of images and showed robustness to JPEG compression. Two drawbacks of the technique are that it only allowed for a 2-D estimate and that it made strong assumptions about the lighting environment. We addressed these drawbacks by developing two other tools that also estimate properties of the light on an object: the specularly tool, to estimate a 3-D illuminant direction from specular highlights on eyes, and the lighting environment tool, to allow for more complex lighting environments.

The human eye is a glossy surface that reflects its environment. In a high-resolution image, it is not uncommon to see the camera flash, windows, or other light sources reflected from a person's eye. Therefore, differences in reflections on eyes across the image can be a telltale sign of digital tampering. The specular tool, described in chapter 3, is able to estimate 3-D illuminant directions from specular highlights on eyes using a 3-D model of an eye. We showed results on simulated images, real images, and forgeries. We also described a method for measuring consistency between estimates from different eyes.

The main drawback of the specular tool is the reliance on user-specified ellipses. The pose estimation is sensitive to the shape of the ellipses and eyes in images are often very small, making automated shape estimation a difficult problem. As digital cameras continue to increase in resolution, we expect automated shape estimation to become easier, though still difficult for eyes that are far from the camera. To better condition the estimate, it may be possible to use measurements from other sources, such as the face and body, or to combine estimates from several people in the image. Such techniques would make the tool easier to use and more robust. Nevertheless, with the current tool, it is possible to render eyes according to the estimated parameters so that the user can judge if the pose is reasonable before measuring light directions.

The lighting environment tool was developed to address the second drawback of the 2-D illuminant direction tool: the assumption of a single point light source infinitely far away. While this assumption might be reasonable outside on a clear day, it will fail to be true in many other cases, such as outside on an overcast day, or inside under mixed lighting. The lighting environment tool is able to model complex lighting using a spherical harmonic representation. In chapter 4, we describe the model, show how to estimate its parameters from an object in an image, and describe a method for comparing estimates from different objects. We give results on simulated images, real images with a known object (a sphere), real images with arbitrary objects, and forgeries.

The two limitations of the lighting environment tool, and the illuminant direction tool as well, are the Lambertian assumption and the restriction of intensity measurements to the occluding contour. While the Lambertian reflectance function is a convenient approximation for some surfaces, it is actually ill-suited for distinguishing between lighting environments at a fine level since it behaves like a low-pass filter—it reduces all lighting environments to their lowest-order terms. Specular surfaces, on the other hand, are more complex to model, but are rich with information about both the lighting environment and the shape of the object. Models of specular surfaces could therefore be used to improve lighting-based forensic tools, but the ability to estimate the shape of an object would also address the second limitation of the current tools—the restriction of intensity measurements to the occluding contour. It seems slightly peculiar that the intensity measurements for these tools come from the edges, the region of an object where the intensities can be unstable. Incorporating shape estimation techniques would allow the algorithms to choose intensities from other regions on the object and perhaps estimate 3-D lighting environment parameters as well.

The chromatic aberration tool exploits a common lens aberration. In chapter 5, we derive a simple model for this aberration and describe how to estimate the model parameters from a single

image. For reliable estimates, the algorithm requires textured regions at different locations in the image and it will typically be unable to detect tampering if the manipulated region is small. The technique is also sensitive to downsampling since the shift caused by chromatic aberration is often on the order of one to two pixels near the edges of the image. Future work could consider simple models for other lens aberrations or explore the uses of chromatic aberration estimates for ballistics.

While each of the tools in this thesis could be used as a starting point for future work, it is important to consider the types of forgeries that such tools can detect and how the tools would be used to detect the forgeries. The tools in this thesis were designed with a specific workflow in mind: an expert user analyzing individual images. This workflow is common in legal situations, for example, where an image analyst might be asked to testify about a specific image. But, these tools are not practical in situations where large quantities of images need to be analyzed. Newspapers, scientific journals, and other publications receive thousands of images per day and it is important for them to be able to screen images for tampering. Statistical techniques may be more effective in this setting, though user-driven techniques could be applied after an initial screening.

While successful on specific types of forgeries, each tool presented in this thesis has several known counterattacks, though some of the counterattacks may be difficult to realize. The easiest and most obvious attack against all three lighting tools is to choose the source images carefully—if the lighting on objects in different images is already consistent, the tools will be unable to detect the tampering. But, sometimes it is not possible to choose other source images. In this case, for the illuminant direction and lighting environment tools, new lighting gradients will need to be applied to the objects to make the estimates more consistent, and this manipulation may be challenging for arbitrary geometries. For the specularity tool, the highlights can be moved, but it may not be obvious where to place the highlight so that it is consistent with other highlights in the image. Finally, for the chromatic aberration tool, the color channels could be shifted to be consistent with the global pattern. None of these counterattacks are beyond the abilities of a sophisticated forger, but would be challenging for a novice especially considering the variety of other ways in which the forgery could be detected.

Digital image forensics is in its early stages and many sources of regularities in images are unexplored or even undiscovered. For example, real-world illumination has statistical regularities [12, 16], and statistical models of illumination and surface reflectance could lead to more powerful lighting tools. Also, known geometry has been explored in photogrammetry and robot vision, but more could be done to examine how tampering affects the geometric and projective properties of an image. Finally, the wealth of images available through a search engine or an image sharing website, such as Flickr [60], could be exploited for forensics. Many objects and people have hundreds of images available, and these images could be used as statistical priors to help determine if an object or person in an image has been modified. These ideas, and others from the computer vision and graphics communities, are rich possibilities for future forensic tools. It is our hope that this and future work on image forensics will contribute to a better understanding of images and the imaging process, while making it more difficult for the average person to make a convincing forgery.

# Appendix A

## Curve fitting

In this appendix, we describe a method for fitting implicit curves under affine and planar projective transforms where the parameters of the curve and the transform are unknown. Our method uses the orthogonal distance fitting framework proposed by Ahn [1] and extends it to handle both affine and projective transforms.

In this context, the problem we would like to solve can be described as follows. Suppose a set of points  $\vec{x}$  are the images of world points  $\vec{X}$  under an unknown affine or projective transform. In addition, suppose that the points  $\vec{X}$  all lie on an implicit curve in the world coordinate system. We would like to find the parameters of the unknown transform as well as the parameters of the implicit curve that best fit the given points  $\vec{x}$ .

This problem is known as orthogonal distance fitting and solutions have been proposed for the case when the unknown transform is a *similarity*; that is, when

$$\vec{x} = R\vec{X} + \vec{b}, \tag{A.1}$$

where the matrix  $R$  is an arbitrary rotation matrix and the vector  $\vec{b}$  is a displacement vector [1, 53]. In [1], Ahn et al. orthogonal distance fitting framework for fitting implicit curves under similarity transforms that displayed robust and rapid convergence as well as separate estimation of model and transform parameters. In this appendix, we extend this framework to handle implicit curves under affine and planar projective transforms.

In the discussion that follows, we refer the coordinate system of vector  $\vec{X}$  as the *world* coordinate system. A vector  $\vec{X}$  in the world coordinate system will be mapped to a vector  $\vec{x}$  by an unknown affine or projective transform; we refer to the coordinate system of vector  $\vec{x}$  as the *image* coordinate system.

### A.1 Minimization

At a high level, the orthogonal distance fitting approach solves a nonlinear least-squares problem. In general, a nonlinear least-squares problem finds the vector  $\vec{a}$  that best solves a system of nonlinear

equations:

$$\begin{aligned}f_1(\vec{a}) &= 0, \\f_2(\vec{a}) &= 0, \\&\vdots \\f_m(\vec{a}) &= 0,\end{aligned}$$

where the  $m$  functions  $f_1(\cdot)$  to  $f_m(\cdot)$  are nonlinear in the terms of the vector  $\vec{a}$ . The least-squares solution to this system of equations is the vector  $\vec{a}$  that minimizes the error function:

$$E(\vec{a}) = \left\| \vec{f}(\vec{a}) \right\|^2,$$

where:

$$\vec{f}(\vec{a}) = \begin{pmatrix} f_1(\vec{a}) \\ f_2(\vec{a}) \\ \vdots \\ f_m(\vec{a}) \end{pmatrix}.$$

Before we describe the details of orthogonal distance fitting, we review the Gauss-Newton iteration, an iterative scheme for solving nonlinear least-squares problems.

### Gauss-Newton iteration

Like all iterative optimization techniques, the Gauss-Newton iteration begins at an initial estimate  $\vec{a}_0$  and updates the estimate until a stopping condition is satisfied. The two important details are the update rule and the stopping condition.

In the neighborhood of the estimate  $\vec{a}_0$ , the vector-valued function  $\vec{f}(\cdot)$  can be approximated by its Taylor series:

$$\vec{f}(\vec{a}_0 + \vec{h}) \approx \vec{f}(\vec{a}_0) + J(\vec{a}_0)\vec{h},$$

where  $J(\vec{a}_0)$  is the Jacobian of the system of equations (matrix of first partial derivatives) evaluated at the vector  $\vec{a}_0$ . To find the next estimate,  $\vec{a}_1 = \vec{a}_0 + \vec{h}$ , we find the update vector  $\vec{h}$  that minimizes:

$$\left\| \vec{f}(\vec{a}_0) + J(\vec{a}_0)\vec{h} \right\|^2,$$

which is a linear least-squares problem. Differentiating with respect to  $\vec{h}$  yields the normal equations:

$$J(\vec{a}_0)^T J(\vec{a}_0)\vec{h} = -J(\vec{a}_0)^T \vec{f}(\vec{a}_0),$$

which have an analytic solution:

$$\vec{h} = -J(\vec{a}_0)^+ \vec{f}(\vec{a}_0),$$

where the ‘+’ denotes pseudo-inverse. Therefore, the update rule for the Gauss-Newton iteration is simply [50]:

$$\vec{h} = -J(\vec{a}_i)^+ \vec{f}(\vec{a}_i), \quad (\text{A.2})$$

$$\vec{a}_{i+1} = \vec{a}_i + \delta \vec{h}, \quad (\text{A.3})$$

for some scalar  $\delta$ , often equal to 1. Note that the update rule requires both the vector-valued function  $\vec{f}(\vec{a})$  and its Jacobian  $J(\vec{a})$ . A typical stopping condition for the iteration is:

$$\|\vec{h}\| < \epsilon$$

for a small value of  $\epsilon$ .

As with nonlinear optimization routines in general, the convergence of the iteration depends strongly on the initial estimate  $\vec{a}_0$  and the particular function  $\vec{f}(\cdot)$ . In addition, the update vector  $\vec{h}$  may be unstable if the matrix  $J(\vec{a}_i)$  is ill-conditioned. Other schemes, such as the Levenberg-Marquardt iteration, can provide faster and more robust convergence [50].

## A.2 Affine transforms

We first describe the orthogonal distance fitting approach for implicit curves under unknown affine transforms. The projective case is similar and will require modifications to only a few of the equations.

### Implicit curves

An implicit curve on points  $\vec{X}$  in the world coordinate system is defined by an implicit equation:

$$f(\vec{X}; \vec{a}) = 0, \quad (\text{A.4})$$

where vector  $\vec{a}$  represents the parameters of the curve. For example, the implicit equation for a circle of radius  $r$  centered at  $(C_1 \ C_2)^T$  is:

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

where the parameter vector  $\vec{a} = (C_1 \ C_2 \ r)^T$  contains parameters describing the center of the circle and the radius. Two-dimensional lines and ellipses can also be described by implicit equations. If the vector  $\vec{X}$  is three-dimensional, then Equation (A.4) describes a surface. Some examples of implicit surfaces are planes, spheres, cones, and cylinders.

### A.2.1 Error function

Suppose points  $\vec{X}$  on an implicit curve  $f$  in the world coordinate system are mapped to points  $\vec{x}$  in the image coordinate system by an unknown affine transform  $(A, \vec{b})$ :

$$\vec{x} = A\vec{X} + \vec{b}. \quad (\text{A.5})$$

Given the points  $\vec{x}$  in the image, we would like to estimate the affine transform as well as the parameters of the implicit curve  $\vec{a}$  such that the points mapped back to the world coordinate system lie on the curve:

$$f(\vec{X}; \vec{a}) = 0 \quad \text{with} \quad \vec{X} = A^{-1}(\vec{x} - \vec{b}). \quad (\text{A.6})$$

We formulate this problem in an orthogonal distance fitting framework as follows. Let vector  $\vec{a}$  represent the unknown parameters of both the implicit curve and the transform:

$$\vec{a} = \begin{pmatrix} \vec{a} \\ \vec{A} \\ \vec{b} \end{pmatrix},$$

where the notation  $\vec{A}$  is a column vector containing the elements of the matrix  $A$ . Let points  $\vec{x}_i$  be the images of points  $\vec{X}_i$  on implicit curve  $f$  with additive noise  $\vec{\epsilon}_i$  at every point:

$$\vec{x}_i = A\vec{X}_i + \vec{b} + \vec{\epsilon}_i.$$

We define an error function  $E$  on the parameter vector  $\vec{a}$  as

$$E(\vec{a}) = \sum_i \|\vec{x}_i - \vec{x}^*\|^2, \quad (\text{A.7})$$

where the point  $\vec{x}^*$  is the closest point to  $\vec{x}_i$  that is on the implicit curve under the affine transform  $(A, \vec{b})$ . In other words,

$$f(\vec{X}^*; \vec{a}) = 0 \quad \text{with} \quad \vec{X}^* = A^{-1}(\vec{x}^* - \vec{b}). \quad (\text{A.8})$$

Recall that the parameter vector  $\vec{a}$  contains the parameters of the affine transform  $(A, \vec{b})$  as well as the parameters  $\vec{a}$  of the implicit curve  $f$ . Thus it parameterizes the constraints of Equation (A.8) and the point  $\vec{x}^*$  depends on the parameter vector  $\vec{a}$ , though this dependency is not made explicit in Equation (A.7).

The error function  $E(\cdot)$  is minimized in nested iterations. The inner iteration computes the closest point  $\vec{x}^*$  on the model for each image point  $\vec{x}_i$ , where the model is specified by the current state of  $\vec{a}$ . The outer iteration then updates the parameter vector  $\vec{a}$  according to the results of the inner iteration. This process is repeated and terminates when the norm of the update to  $\vec{a}$  is below a specified threshold.



## Closest point

For a given point  $\vec{x}_i$  in image coordinates, we seek the closest point  $\vec{x}^*$  on the model. The point  $\vec{x}^*$  that satisfies this condition must, of course, be on the model, Equation (A.8). In addition, the vector between the points  $\vec{x}_i$  and  $\vec{x}^*$  must be parallel to the gradient of the model in image coordinates, Figure A.1. Note that this condition is true at the closest point  $\vec{x}^*$ , but it may be true for other points on the model as well. In the world coordinate system, the gradient of  $f$  is

$$\nabla_{\vec{x}} f = \begin{pmatrix} \frac{\partial f}{\partial X_1} \\ \frac{\partial f}{\partial X_2} \\ \vdots \\ \frac{\partial f}{\partial X_n} \end{pmatrix}.$$

We use the chain rule to find the gradient of  $f$  with respect to the image coordinate  $\vec{x}$ ,

$$\begin{aligned} \nabla_{\vec{x}} f &= \begin{pmatrix} \frac{\partial f}{\partial \vec{X}} & \frac{\partial \vec{X}}{\partial \vec{x}} \end{pmatrix}^T \\ &= \begin{pmatrix} \frac{\partial f}{\partial \vec{X}} & A^{-1} \end{pmatrix}^T, \\ &= A^{-T} \nabla_{\vec{X}} f, \end{aligned}$$

where the  $\partial \vec{X} / \partial \vec{x}$  is found by differentiating Equation (A.6) with respect to  $\vec{x}$ . For the remainder of the discussion, the gradient symbol ‘ $\nabla$ ’ will indicate the gradient with respect to the world coordinate system unless the image coordinate system is specifically referenced, e.g. ‘ $\nabla_{\vec{x}}$ .’

Figure A.1 shows the relationships between the closest point  $\vec{x}^*$  in the world and image coordinate systems as well as the gradient of the model at this point in both coordinate systems. Note that in the world coordinate system, the point  $\vec{X}^*$  is not necessarily the closest point to  $\vec{X}_i$  nor is the gradient at  $\vec{X}^*$  parallel to the displacement vector between  $\vec{X}_i$  and  $\vec{X}^*$ .

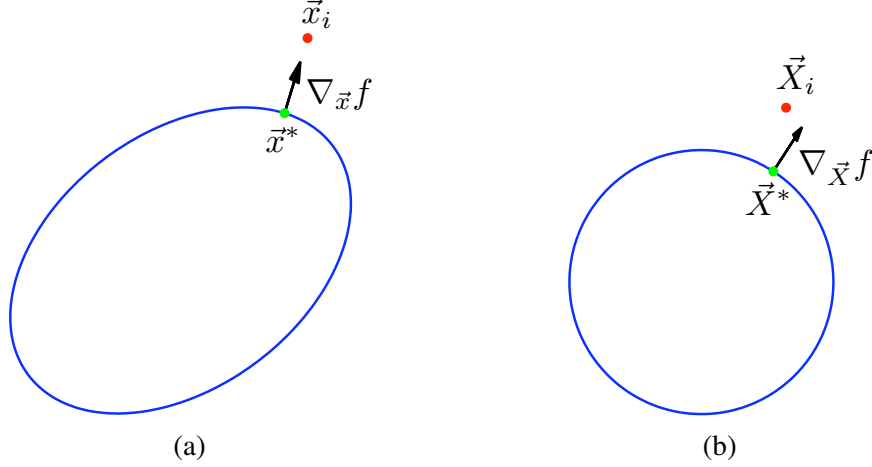
To constrain the gradient to be parallel to the vector between  $\vec{x}_i$  and  $\vec{x}^*$ , we use the cross product<sup>1</sup> since it is not affected by the scale of the individual vectors,

$$\begin{aligned} (\vec{x}_i - \vec{x}^*) \times A^{-T} \nabla f &= \vec{0}, \\ A(\vec{X}_i - \vec{X}^*) \times A^{-T} \nabla f &= \vec{0}. \end{aligned} \tag{A.9}$$

The constraints in Equations (A.6) and (A.9) are combined into a system of nonlinear equations,

$$\vec{g}(\vec{X}, \vec{X}_i, \vec{a}) = \begin{pmatrix} f(\vec{X}; \vec{a}) \\ A(\vec{X}_i - \vec{X}) \times A^{-T} \nabla f \end{pmatrix} = \vec{0}, \tag{A.10}$$

<sup>1</sup>In 2-D, we use the convention that  $\vec{u} \times \vec{v} = \det \begin{pmatrix} \vec{u} & \vec{v} \end{pmatrix}$ .



**Figure A.1:** Closest point. (a) Point  $\vec{x}^*$  is the point on the implicit curve closest to  $\vec{x}_i$ , and the gradient at  $\vec{x}^*$  is parallel to the difference vector between  $\vec{x}_i$  and  $\vec{x}^*$ . (b) In the world coordinate system, the point  $\vec{X}^*$  is not necessarily the closest point to  $\vec{X}_i$  nor is the gradient at  $\vec{X}^*$  parallel to the displacement vector between  $\vec{X}_i$  and  $\vec{X}^*$ .

which can be solved using a Gauss-Newton iteration.

Recall that the Gauss-Newton iteration requires the Jacobian of  $\vec{g}(\cdot)$  as well as a starting condition. The starting condition for the iteration is simply  $\vec{X} = \vec{X}_i$ . The Jacobian is computed by differentiating  $\vec{g}(\cdot)$  with respect to  $\vec{X}$ , which can be simplified by rewriting the cross product in Equation (A.10) as a vector of dot products. To demonstrate this, consider the cross product between vectors  $\vec{s}$  and  $\vec{t}$ , scaled by matrices  $M$  and  $N$ :

$$M\vec{s} \times N\vec{t},$$

where

$$M = \begin{pmatrix} \vec{m}_1^T \\ \vec{m}_2^T \\ \vec{m}_3^T \end{pmatrix}, \quad N = \begin{pmatrix} \vec{n}_1^T \\ \vec{n}_2^T \\ \vec{n}_3^T \end{pmatrix}.$$

Then

$$M\vec{s} \times N\vec{t} = \begin{pmatrix} \vec{m}_1^T \vec{s} \\ \vec{m}_2^T \vec{s} \\ \vec{m}_3^T \vec{s} \end{pmatrix} \times \begin{pmatrix} \vec{n}_1^T \vec{t} \\ \vec{n}_2^T \vec{t} \\ \vec{n}_3^T \vec{t} \end{pmatrix} = \begin{pmatrix} (\vec{m}_2^T \vec{s})(\vec{n}_3^T \vec{t}) - (\vec{m}_3^T \vec{s})(\vec{n}_2^T \vec{t}) \\ (\vec{m}_3^T \vec{s})(\vec{n}_1^T \vec{t}) - (\vec{m}_1^T \vec{s})(\vec{n}_3^T \vec{t}) \\ (\vec{m}_1^T \vec{s})(\vec{n}_2^T \vec{t}) - (\vec{m}_2^T \vec{s})(\vec{n}_1^T \vec{t}) \end{pmatrix}. \quad (\text{A.11})$$

A product of dot products, such as  $(\vec{m}_2^T \vec{s})(\vec{n}_3^T \vec{t})$ , can be rewritten as a single dot product with an appropriate scaling matrix:

$$(\vec{m}_2^T \vec{s})(\vec{n}_3^T \vec{t}) = (\vec{m}_2^T \vec{s})^T (\vec{n}_3^T \vec{t}) = \vec{s}^T (\vec{m}_2 \vec{n}_3^T) \vec{t} = \vec{s}^T C_{23} \vec{t},$$

where

$$C_{23} = \vec{m}_2 \vec{n}_3^T.$$

Applying this relationship to the terms in Equation (A.11) yields:

$$\begin{pmatrix} \vec{s}^T C_{23} \vec{t} - \vec{s}^T C_{32} \vec{t} \\ \vec{s}^T C_{31} \vec{t} - \vec{s}^T C_{13} \vec{t} \\ \vec{s}^T C_{12} \vec{t} - \vec{s}^T C_{21} \vec{t} \end{pmatrix} = \begin{pmatrix} \vec{s}^T (C_{23} - C_{32}) \vec{t} \\ \vec{s}^T (C_{31} - C_{13}) \vec{t} \\ \vec{s}^T (C_{12} - C_{21}) \vec{t} \end{pmatrix} = \begin{pmatrix} \vec{s}^T D_{23} \vec{t} \\ \vec{s}^T D_{31} \vec{t} \\ \vec{s}^T D_{12} \vec{t} \end{pmatrix}.$$

With the appropriate matrices  $D_{23}$ ,  $D_{31}$ , and  $D_{12}$ , the vector-valued function  $\vec{g}(\cdot)$  in Equation (A.10) becomes:

$$\vec{g}(\vec{X}, \vec{X}_i, \vec{a}) = \begin{pmatrix} f(\vec{X}; \vec{a}) \\ (\vec{X}_i - \vec{X})^T D_{23} \nabla f \\ (\vec{X}_i - \vec{X})^T D_{31} \nabla f \\ (\vec{X}_i - \vec{X})^T D_{12} \nabla f \end{pmatrix}. \quad (\text{A.12})$$

Differentiating  $\vec{g}(\cdot)$  with respect to  $\vec{X}$  yields:

$$\frac{\partial \vec{g}}{\partial \vec{X}} = \begin{pmatrix} (\nabla f)^T \\ (\vec{X}_i - \vec{X})^T D_{ij} H - (\nabla f)^T D_{ij}^T \end{pmatrix}, \quad (\text{A.13})$$

where  $H$  is the Hessian of  $f$  (the matrix of second partial derivatives) and the final row is repeated for each  $D$  matrix in Equation (A.12). With the Jacobian  $\partial \vec{g} / \partial \vec{X}$  and the initial estimate  $\vec{X} = \vec{X}_i$ , the vector-valued function  $\vec{g}(\cdot)$  can be minimized using the Gauss-Newton iteration yielding the closest point  $\vec{X}^*$  in world coordinates. In image coordinates, the closest point is:

$$\vec{x}^* = A \vec{X}^* + \vec{b}.$$

### Parameter update

Once the inner iteration completes and the closest points  $\vec{x}^*$  have been computed for each image point  $\vec{x}_i$ , the parameter vector  $\vec{a}$  is updated with the goal of minimizing the error function  $E$ , Equation (A.7). A necessary condition at the minimum of  $E$  is that the partial derivatives with respect to the parameters in  $\vec{a}$  are zero,

$$\frac{\partial E}{\partial \vec{a}} = -2 \sum_i \left( \frac{\partial \vec{x}^*}{\partial \vec{a}} \right)^T (\vec{x}_i - \vec{x}^*) = \vec{0}. \quad (\text{A.14})$$

This equation is solved using a Gauss-Newton iteration, which requires  $\partial \vec{x} / \partial \vec{a}$  evaluated at the closest point  $\vec{x}^*$ . This term is found by differentiating Equation (A.5) with respect to  $\vec{a}$ :

$$\frac{\partial \vec{x}}{\partial \vec{a}} = \frac{\partial}{\partial \vec{a}} (A \vec{X} + \vec{b}),$$

$$= \left( \frac{\partial A}{\partial \vec{a}} \right) [\vec{X}] + A \frac{\partial \vec{X}}{\partial \vec{a}} + \frac{\partial \vec{b}}{\partial \vec{a}}, \quad (\text{A.15})$$

where  $[\vec{X}]$  is a block-diagonal matrix with  $\vec{X}$  on the diagonal. The term  $\partial A / \partial \vec{a}$  is found by simply differentiating the matrix  $A$  with respect to each of its components  $a_i$ . The term  $\partial X / \partial \vec{a}$  is derived by implicitly differentiating the vector-valued function  $\vec{g}(\cdot)$ , Equation (A.10), with respect to  $\vec{a}$ :

$$\frac{\partial \vec{g}}{\partial \vec{a}} + \frac{\partial \vec{g}}{\partial \vec{X}} \frac{\partial \vec{X}}{\partial \vec{a}} + \frac{\partial \vec{g}}{\partial \vec{X}_i} \frac{\partial \vec{X}_i}{\partial \vec{a}} = 0,$$

and solving for  $\partial X / \partial \vec{a}$ :

$$\frac{\partial \vec{X}}{\partial \vec{a}} = - \left( \frac{\partial \vec{g}}{\partial \vec{X}} \right)^{-1} \left( \frac{\partial \vec{g}}{\partial \vec{a}} + \frac{\partial \vec{g}}{\partial \vec{X}_i} \frac{\partial \vec{X}_i}{\partial \vec{a}} \right). \quad (\text{A.16})$$

The individual derivatives in this expression are given as:

$$\frac{\partial \vec{g}}{\partial \vec{a}} = \left( \begin{array}{c} \frac{\partial f / \partial \vec{a}}{(\vec{X}_i - \vec{X})^T \left( \frac{\partial D_{ij}}{\partial \vec{a}} [\nabla f] + D_{ij} \frac{\partial \nabla f}{\partial \vec{a}} \right)} \end{array} \right), \quad (\text{A.17})$$

$$\frac{\partial \vec{g}}{\partial \vec{X}_i} = \left( \begin{array}{c} \vec{0} \\ (\nabla f)^T D_{ij}^T \end{array} \right), \quad (\text{A.18})$$

$$\frac{\partial \vec{X}_i}{\partial \vec{a}} = \frac{\partial A^{-1}}{\partial \vec{a}} [\vec{x}_i - \vec{b}] - A^{-1} \frac{\partial \vec{b}}{\partial \vec{a}}. \quad (\text{A.19})$$

The derivatives for all  $m$  image points,  $\vec{x}_1$  to  $\vec{x}_m$ , are then stacked into a Jacobian matrix which is used by the Gauss-Newton iteration to compute the update to the parameter vector  $\vec{a}$ . The outer iteration terminates when the update to  $\vec{a}$  is below a specified threshold. At this point, the vector  $\vec{a}$  contains the parameters of the affine transform as well as the parameters for the implicit curve that best fit the given points  $\vec{x}_i$  in a geometric sense.

### A.3 Planar projective transforms

For planar projective transforms, we let the transform between world and image coordinates be represented by a  $3 \times 3$  matrix  $P$ ,

$$\vec{x} = P\vec{X}, \quad (\text{A.20})$$

where both  $\vec{X}$  and  $\vec{x}$  are vectors in homogeneous coordinates. In order to contend with the scale ambiguity inherent to homogeneous coordinates, the model in world coordinates takes on a slightly different form: the division is accounted for within the model. For example, the implicit equation

for a circle centered at the origin becomes:

$$f(\vec{X}; r) = (X_1/X_3)^2 + (X_2/X_3)^2 - r^2.$$

### A.3.1 Error function

The error function for the projective case is the same as Equation (A.7), but in this case the parameter vector  $\vec{a}$  contains the elements of the unknown projective transform  $P$ :

$$\vec{a} = \begin{pmatrix} \vec{a} \\ \vec{P} \end{pmatrix}.$$

As before, this error function is minimized in nested iterations. The inner iteration finds the closest point  $\vec{x}^*$  to each image point  $\vec{x}_i$ , and the outer iteration updates the model parameters  $\vec{a}$ .

#### Closest point

For  $\vec{x}^*$  to be the closest point on the model to the image point  $\vec{x}_i$ , it must satisfy three constraints. As with the affine case, the point must be on the curve in the world coordinate system:

$$f(\vec{X}^*; \vec{a}) = 0 \quad \text{with} \quad \vec{X}^* = P^{-1}\vec{x}^*. \quad (\text{A.21})$$

The second constraint is that the vector between the image point  $\vec{x}_i$  and the model point  $\vec{x}^*$  (expressed in image coordinates) must be parallel to the gradient of the model in image coordinates,  $P^{-T}\nabla f$ , yielding the following constraint:

$$\vec{z}^T((\vec{x}_i - \vec{x}^*) \times P^{-T}\nabla f) = 0, \quad (\text{A.22})$$

where  $\vec{z}^T = (0 \ 0 \ 1)$  restricts this constraint to the image plane. The final constraint is that the model point  $\vec{x}^*$  must lie in the image plane (recall that the homogeneous points  $\vec{x}_i$  lie in the plane  $z = 1$ ). This constraint is expressed by making the difference vector orthogonal to the normal to the image plane:

$$\vec{z}^T(\vec{x}_i - \vec{x}^*) = 0. \quad (\text{A.23})$$

The constraints in Equations (A.22) and (A.23) can be rewritten in world coordinates as:

$$\begin{aligned} \vec{z}^T(P(\vec{X}_i - \vec{X}^*) \times P^{-T}\nabla f) &= 0, \\ \vec{z}^T P(\vec{X}_i - \vec{X}^*) &= 0. \end{aligned}$$

All three constraints form a system of nonlinear equations,

$$\vec{g}(\vec{X}, \vec{X}_i, \vec{a}) = \begin{pmatrix} f(\vec{X}; \vec{a}) \\ \vec{z}^T (P(\vec{X}_i - \vec{X}) \times P^{-T} \nabla f) \\ \vec{z}^T P(\vec{X}_i - \vec{X}) \end{pmatrix} = \vec{0}, \quad (\text{A.24})$$

which can be solved using a Gauss-Newton iteration. To simplify differentiation, the term with cross product can be expressed as a dot product:

$$\vec{z}^T (P(\vec{X}_i - \vec{X}) \times P^{-T} \nabla f) = (\vec{X}_i - \vec{X})^T D \nabla f,$$

where  $D = m_1 n_2^T - m_2 n_1^T$  and:

$$\begin{aligned} m_1^T &= \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix}, \\ m_2^T &= \begin{pmatrix} p_4 & p_5 & p_6 \end{pmatrix}, \\ n_1^T &= \begin{pmatrix} p_5 p_9 - p_6 p_8 & -p_4 p_9 + p_6 p_7 & p_4 p_8 - p_5 p_7 \end{pmatrix}, \\ n_2^T &= \begin{pmatrix} -p_2 p_9 + p_3 p_8 & p_1 p_9 - p_3 p_7 & -p_1 p_8 + p_2 p_7 \end{pmatrix}, \end{aligned}$$

where  $p_i$  are the elements of matrix  $P$  in row-major order. Differentiating  $\vec{g}(\cdot)$  with respect to  $\vec{X}$  yields:

$$\frac{\partial \vec{g}}{\partial \vec{X}} = \begin{pmatrix} (\nabla f)^T \\ (\vec{X}_i - \vec{X})^T D H - (\nabla f)^T D^T \\ -\vec{z}^T P \end{pmatrix}. \quad (\text{A.25})$$

### Parameter update

As with the affine case, the inner iteration computes the closest points  $\vec{x}^*$  to each image point  $\vec{x}_i$ . The outer iteration updates the parameter vector  $\vec{a}$  to solve the system of equations in Equation (A.14). This system requires  $\partial \vec{x} / \partial \vec{a}$  evaluated at the closest point  $\vec{x}^*$ , which is computed by differentiating Equation (A.20) with respect to  $\vec{a}$ :

$$\begin{aligned} \frac{\partial \vec{x}}{\partial \vec{a}} &= \frac{\partial}{\partial \vec{a}} P \vec{X}, \\ &= \left( \frac{\partial P}{\partial \vec{a}} \right) [\vec{X}] + P \frac{\partial \vec{X}}{\partial \vec{a}}, \end{aligned} \quad (\text{A.26})$$

where  $\begin{bmatrix} \vec{X} \end{bmatrix}$  is a block-diagonal matrix with  $\vec{X}$  on the diagonal. The equation for computing  $\partial \vec{X} / \partial \vec{a}$  is the same as in Equation (A.16), but with the following individual derivatives:

$$\frac{\partial \vec{g}}{\partial \vec{a}} = \begin{pmatrix} \frac{\partial f / \partial \vec{a}}{(\vec{X}_i - \vec{X})^T \left( \frac{\partial D}{\partial \vec{a}} [\nabla f] + D \frac{\partial \nabla f}{\partial \vec{a}} \right)} \\ \vec{z}^T \left( \frac{\partial P}{\partial \vec{a}} [\vec{X}_i - \vec{X}] \right) \end{pmatrix}, \quad (\text{A.27})$$

$$\frac{\partial \vec{g}}{\partial \vec{X}_i} = \begin{pmatrix} \vec{0} \\ (\nabla f)^T D^T \\ \vec{z}^T P \end{pmatrix}, \quad (\text{A.28})$$

$$\frac{\partial \vec{X}_i}{\partial \vec{a}} = \frac{\partial P^{-1}}{\partial \vec{a}} [\vec{x}_i]. \quad (\text{A.29})$$

For the derivatives of the inverse matrix  $P^{-1}$ , we use the identity:

$$\frac{\partial P^{-1}}{\partial p_i} = -P^{-1} \left( \frac{\partial P}{\partial p_i} \right) P^{-1},$$

and the nine derivatives of the matrix  $D$  are:

$$\frac{\partial D}{\partial p_1} = \begin{pmatrix} -p_2 p_9 + p_3 p_8 & 2p_1 p_9 - p_3 p_7 & -2p_1 p_8 + p_2 p_7 \\ 0 & p_2 p_9 & -p_2 p_8 \\ 0 & p_3 p_9 & -p_3 p_8 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_2} = \begin{pmatrix} -p_1 p_9 & 0 & p_1 p_7 \\ -2p_2 p_9 + p_3 p_8 & p_1 p_9 - p_3 p_7 & -p_1 p_8 + 2p_2 p_7 \\ -p_3 p_9 & 0 & p_3 p_7 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_3} = \begin{pmatrix} p_1 p_8 & -p_1 p_7 & 0 \\ p_2 p_8 & -p_2 p_7 & 0 \\ -p_2 p_9 + 2p_3 p_8 & p_1 p_9 - 2p_3 p_7 & -p_1 p_8 + p_2 p_7 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_4} = \begin{pmatrix} -p_5 p_9 + p_6 p_8 & 2p_4 p_9 - p_6 p_7 & -2p_4 p_8 + p_5 p_7 \\ 0 & p_5 p_9 & -p_5 p_8 \\ 0 & p_6 p_9 & -p_6 p_8 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_5} = \begin{pmatrix} -p_4 p_9 & 0 & p_4 p_7 \\ -2p_5 p_9 + p_6 p_8 & p_4 p_9 - p_6 p_7 & -p_4 p_8 + 2p_5 p_7 \\ -p_6 p_9 & 0 & p_6 p_7 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_6} = \begin{pmatrix} p_4 p_8 & -p_4 p_7 & 0 \\ p_5 p_8 & -p_5 p_7 & 0 \\ -p_5 p_9 + 2p_6 p_8 & p_4 p_9 - 2p_6 p_7 & -p_4 p_8 + p_5 p_7 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_7} = \begin{pmatrix} 0 & -p_1 p_3 - p_4 p_6 & p_1 p_2 + p_4 p_5 \\ 0 & -p_2 p_3 - p_5 p_6 & p_2^2 + p_5^2 \\ 0 & -p_3^2 - p_6^2 & p_2 p_3 + p_5 p_6 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_8} = \begin{pmatrix} p_1 p_3 + p_4 p_6 & 0 & -p_1^2 - p_4^2 \\ p_2 p_3 + p_5 p_6 & 0 & -p_1 p_2 - p_4 p_5 \\ p_3^2 + p_6^2 & 0 & -p_1 p_3 - p_4 p_6 \end{pmatrix},$$

$$\frac{\partial D}{\partial p_9} = \begin{pmatrix} -p_1 p_2 - p_4 p_5 & p_1^2 + p_4^2 & 0 \\ -p_2^2 - p_5^2 & p_1 p_2 + p_4 p_5 & 0 \\ -p_2 p_3 - p_5 p_6 & p_1 p_3 + p_4 p_6 & 0 \end{pmatrix},$$

where  $p_i$  are the elements of the matrix  $P$  in row-major order.

The outer iteration terminates when the update to  $\vec{a}$  is below a specified threshold. At this point, the vector  $\vec{a}$  contains the parameters of the projective transform as well as the parameters for the implicit curve that best fit the given points  $\vec{x}_i$  in a geometric sense.

## A.4 Constraints

In many cases, there can be ambiguities between the parameters of the curve and the parameters of the transform. For example, the scale of the transform can affect the scale of the curve (e.g., the radius of an unknown circle). To resolve these ambiguities, constraint equations on the parameters are added to the system of equations for the outer iteration. For a system of nonlinear equations  $\vec{f}(\vec{a}) = \vec{0}$  with Jacobian  $J(\vec{a})$ , the constrained system is:

$$\vec{f}^*(\vec{a}) = \begin{pmatrix} \vec{f}(\vec{a}) \\ \omega \hat{f}(\vec{a}) \end{pmatrix},$$

with Jacobian:

$$J^*(\vec{a}) = \begin{pmatrix} J(\vec{a}) \\ \omega \hat{J}(\vec{a}) \end{pmatrix},$$

for some scalar weight  $\omega$ , typically set to a large value such as  $10^6$ . In the case of circle, we often constrain the radius to be close to one:

$$\hat{f}(\vec{a}) = r - 1 = 0,$$

which has a Jacobian that is zero for all terms in  $\vec{a}$  except the radius:

$$\frac{\partial \hat{f}}{\partial r} = 1.$$



## A.5 Multiple curves

The orthogonal distance fitting approach can be extended to handle multiple curves as well. For example, consider two curves with parameter vectors  $\vec{a}_1$  and  $\vec{a}_2$ . A new error function is defined from the individual error functions for each curve as:

$$E^*(\vec{a}) = E(\vec{a}_1) + E(\vec{a}_2) + \omega \hat{E}(\vec{a}),$$

where

$$\vec{a} = \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \end{pmatrix},$$

and  $\hat{E}(\vec{a})$  is an error term for the constraints on the parameters of  $\vec{a}_1$  and  $\vec{a}_2$ . Since both  $\vec{a}_1$  and  $\vec{a}_2$  contain elements for the unknown transform, at a minimum, the constraint equations should constrain corresponding elements to be equal. There might also be relationships between the parameters of the implicit curves that could be expressed as constraints, such as identical y-coordinates for the centers of two circles.

This new error function corresponds to a linear system with the following Jacobian:

$$J^*(\vec{a}) = \begin{pmatrix} J(\vec{a}_1) & \\ & J(\vec{a}_2) \\ \hat{J}(\vec{a}_1) & \hat{J}(\vec{a}_2) \end{pmatrix},$$

where  $J(\vec{a}_1)$  is the Jacobian for the first curve,  $J(\vec{a}_2)$  is the Jacobian for the second curve, and  $\hat{J}(\vec{a}_1)$  and  $\hat{J}(\vec{a}_2)$  are the Jacobians of the constraint equations for the two curves.

## A.6 Examples

The ability to estimate affine and planar projective transforms from points in an image is useful for many computer vision algorithms, and is generally known as rectification. In contrast to previous work on rectification from simple geometries, this approach provides a unifying framework: the algorithm does not depend on the types of curves in the image. The same restrictions and necessary conditions still apply, however, so that the problem is well-posed. For example, it is not possible to uniquely fit an unknown ellipse under an unknown projective transform since all ellipses are projectively equivalent. A thorough review of necessary conditions for solving for an unknown projective transform can be found in [22].

To conclude this appendix, we show the application of the orthogonal distance fitting framework to several problems. First, we derive the necessary components for fitting a circle under affine and planar projective transforms and a line under a projective transform. Then, we show results on three images: an image with a pair of eyes, an image with four wheels, and an image of a building. The

pair of eyes formulation is used throughout chapter 3 to find the transform between the eyes and camera and solve for the light direction. The four wheels and building images are examples of metric rectification from circles and lines in the world.

### Circle: affine

As a first example, consider the implicit equation of a circle of radius  $r$  centered at the origin in the world coordinate system:

$$f(\vec{X}; r) = \frac{1}{2} (X_1^2 + X_2^2 - r^2) = 0.$$

The scale factor  $\frac{1}{2}$  simplifies the derivatives of  $f$ . In addition, suppose a set of points  $\vec{x}_i$  are the images of points on the circle under an unknown affine transform. In other words, there exists an affine transform  $(A, \vec{b})$  such that:

$$f(\vec{X}_i; r) = 0 \quad \text{with} \quad \vec{X}_i = A^{-1}(\vec{x}_i - \vec{b}).$$

Let the parameter vector  $\vec{a} = (r \ a_1 \ a_2 \ a_3 \ a_4 \ b_1 \ b_2)^T$  where  $r$  is the radius of the circle,

$$A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \quad \text{and} \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

The orthogonal distance fitting approach finds the parameters  $\vec{a}$  that best model the image points  $\vec{x}_i$  in the least-squares sense:

$$E(\vec{a}) = \sum_i \|\vec{x}_i - \vec{x}^*\|^2.$$

### Closest point

The inner iteration computes the closest point  $\vec{x}^*$  to each image point  $\vec{x}_i$ . The corresponding point in world coordinates,  $\vec{X}^*$ , is the solution to the nonlinear least-squares problem,

$$\vec{g}(\vec{X}, \vec{X}_i, \vec{a}) = \begin{pmatrix} f(\vec{X}; \vec{a}) \\ A(\vec{X}_i - \vec{X}) \times A^{-T} \nabla f \end{pmatrix} = \vec{0}. \quad (\text{A.30})$$

As described above, the cross product (in this case, the determinant of a  $2 \times 2$  matrix) can be expressed as the following dot product:

$$A(\vec{X}_i - \vec{X}) \times A^{-T} \nabla f = (\vec{X}_i - \vec{X})^T D \nabla f,$$

where

$$D = \begin{pmatrix} -a_1 a_2 - a_3 a_4 & a_1^2 + a_3^2 \\ -a_2^2 - a_4^2 & a_1 a_2 + a_3 a_4 \end{pmatrix},$$

and where the terms  $a_i$  are the elements of the matrix  $A$  in row-major order.

The nonlinear least-squares problem in Equation (A.30) can be solved with the Gauss-Newton iteration, which requires the Jacobian of  $\vec{g}(\cdot)$ :

$$\frac{\partial \vec{g}}{\partial \vec{X}} = \begin{pmatrix} (\nabla f)^T \\ (\vec{X}_i - \vec{X})^T D H - (\nabla f)^T D^T \end{pmatrix},$$

where

$$\nabla f = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

### Parameter update

The outer iteration updates the parameter vector  $\vec{a}$ . This iteration also requires the Jacobian of the system of equations which in turn requires several different derivatives outlined in Equations (A.15) to (A.19). For the case of a circle under an affine transform, the necessary components to compute these derivatives are:

$$\begin{aligned} \frac{\partial D}{\partial a_1} &= \begin{pmatrix} -a_2 & 2a_1 \\ 0 & a_2 \end{pmatrix} & \frac{\partial D}{\partial a_2} &= \begin{pmatrix} -a_1 & 0 \\ -2a_2 & a_1 \end{pmatrix} \\ \frac{\partial D}{\partial a_3} &= \begin{pmatrix} -a_4 & 2a_3 \\ 0 & a_4 \end{pmatrix} & \frac{\partial D}{\partial a_4} &= \begin{pmatrix} -a_3 & 0 \\ -2a_4 & a_3 \end{pmatrix} \\ \frac{\partial A^{-1}}{\partial a_1} &= \frac{1}{\Delta^2} \begin{pmatrix} -a_4^2 & a_2 a_4 \\ a_3 a_4 & -a_2 a_3 \end{pmatrix} & \frac{\partial A^{-1}}{\partial a_2} &= \frac{1}{\Delta^2} \begin{pmatrix} a_3 a_4 & -a_1 a_4 \\ -a_3^2 & a_1 a_3 \end{pmatrix} \\ \frac{\partial A^{-1}}{\partial a_3} &= \frac{1}{\Delta^2} \begin{pmatrix} a_2 a_4 & -a_2^2 \\ -a_1 a_4 & a_1 a_2 \end{pmatrix} & \frac{\partial A^{-1}}{\partial a_4} &= \frac{1}{\Delta^2} \begin{pmatrix} -a_2 a_3 & a_1 a_2 \\ a_1 a_3 & -a_1^2 \end{pmatrix} \end{aligned}$$

where  $\Delta = \det(A)$ .

### Circle: projective

As a second example, consider the implicit equation of a circle centered at  $(C_1 \ C_2)$  in the world coordinate system with radius  $r$ :

$$f(\vec{X}; \vec{a}) = \frac{1}{2} [(X_1/X_3 - C_1)^2 + (X_2/X_3 - C_2)^2 - r^2] = 0,$$

where  $\vec{a} = (C_1 \ C_2 \ r)^T$  and the scale factor  $\frac{1}{2}$  is introduced to simplify the derivatives. The full parameter vector  $\vec{a}$  for the error function, Equation (A.7), has twelve elements: three from the parameter vector  $\vec{a}$  and nine from the components of the  $3 \times 3$  matrix  $P$ .

### Closest point

The inner iteration computes the closest point  $\vec{x}^*$  to each image point  $\vec{x}_i$ . The corresponding point in world coordinates,  $\vec{X}^*$ , is the solution to the nonlinear least-squares problem,

$$\vec{g}(\vec{X}, \vec{X}_i, \vec{a}) = \begin{pmatrix} f(\vec{X}; \vec{a}) \\ \vec{z}^T (P(\vec{X}_i - \vec{X}) \times P^{-T} \nabla f) \\ \vec{z}^T P(\vec{X}_i - \vec{X}) \end{pmatrix} = \vec{0},$$

which can be solved using the Gauss-Newton iteration. The iteration requires the derivative of  $\vec{g}(\cdot)$  with respect to  $\vec{X}$ , Equation (A.25), which in turn requires the gradient of  $f$  and the Hessian:

$$\nabla f = \frac{1}{X_3} \begin{pmatrix} \hat{X}_1 - C_1 \\ \hat{X}_2 - C_2 \\ -(\hat{X}_1 - C_1)\hat{X}_1 - (\hat{X}_2 - C_2)\hat{X}_2 \end{pmatrix},$$

$$H = \frac{1}{X_3^2} \begin{pmatrix} 1 & 0 & -2\hat{X}_1 + C_1 \\ 0 & 1 & -2\hat{X}_2 + C_2 \\ -2\hat{X}_1 + C_1 & -2\hat{X}_2 + C_2 & \hat{X}_1^2 + 2(\hat{X}_1 - C_1)\hat{X}_1 + \hat{X}_2^2 + 2(\hat{X}_2 - C_2)\hat{X}_2 \end{pmatrix},$$

where  $\hat{X}_1 = X_1/X_3$  and  $\hat{X}_2 = X_2/X_3$ .

### Parameter update

The outer iteration updates the parameter vector  $\vec{a}$ . The necessary derivatives are given in Equations (A.27) through (A.29). For this example, these additional derivatives are needed:

$$\frac{\partial f}{\partial \vec{a}} = \begin{pmatrix} -\hat{X}_1 + C_1 & -\hat{X}_2 + C_2 & -r \end{pmatrix},$$

$$\frac{\partial \nabla f}{\partial \vec{a}} = \frac{1}{X_3} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ \hat{X}_1 & \hat{X}_2 & 0 \end{pmatrix}.$$

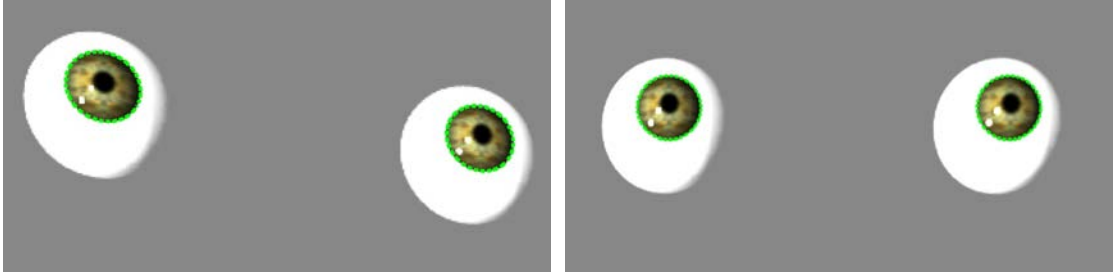
### Line

The implicit equation of a line in homogeneous coordinates is:

$$f(\vec{X}; \vec{L}) = \vec{L}^T \vec{X} = 0,$$

where  $\vec{L}$  is a 3-vector representing the line. The gradient of  $f$  is:

$$\nabla f = \vec{L},$$



**Figure A.2:** A pair of eyes. (Left) The limbus on each eye, a circle in the world coordinate system, is imaged as an ellipse. (Right) The rectified image.

and the Hessian is a  $3 \times 3$  matrix of zeros. For the outer iteration, the following derivatives are needed:

$$\begin{aligned} \frac{\partial f}{\partial \vec{L}} &= X^T \\ \frac{\partial \nabla f}{\partial \vec{L}} &= I, \end{aligned}$$

where  $I$  is the  $3 \times 3$  identity matrix.

### Pair of eyes

For a pair of eyes, we use two circles, described in example A.6, with constraints on the radii as well as constraints on the elements of the transform for each circle. These constraints add the following equations to the nonlinear system:

$$\begin{aligned} r_1 - r_2 &= 0, \\ p_1 - q_1 &= 0, \\ &\vdots \\ p_9 - q_9 &= 0, \end{aligned}$$

where  $r_1$  is the radius of the first circle,  $r_2$  is the radius of the second circle,  $p_i$  are the nine components of the projective transform for the first circle, and  $q_i$  are the nine components of the projective transform for the second circle. The known radius of the limbus, 5.8 mm is used to resolve the scale ambiguity.

Shown in Figure A.2 is a pair of eyes, rendered according to the 3-D model described in chapter 3. The eyes were positioned at  $(-15 \ 15 \ 45)$  in the world coordinate system and rotated by  $(0 \ 25^\circ \ 0)$  about the  $x$ ,  $y$ , and  $z$  axes.

The projective transform between the world and camera coordinate systems was estimated by fitting two circles under a projective transform as described above. The extrinsic components of the estimated projective transform were a position of  $(-15.0 \ 15.5 \ 45.5)$  with rotation angles of  $(-2^\circ \ 26^\circ \ 1^\circ)$  about the  $x$ ,  $y$ , and  $z$  axes, indicating a good estimate of the actual projective



**Figure A.3:** The top image was rectified using four circles (wheels). Shown below is the rectified image from which the distance between the cars can be measured.

transform.

### Four wheels

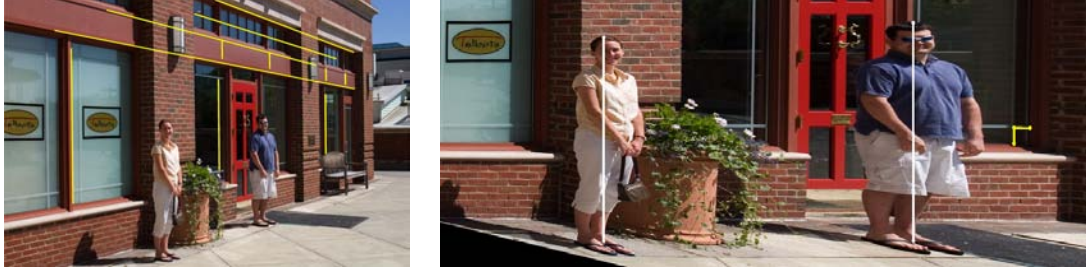
Shown in Figure A.3 is an image of two parked cars. The image can be rectified by assuming that the four wheels in the image are coplanar circles. The four circles are fit using four of the projective circle models, example A.6, with constraints on the radii and the transform components. The radii were constrained to be equal for wheels from the same car, but not between different cars. The transform components were constrained to be the same for all four circles.

The known hubcap diameter of 16.5 inches was used to determine the final scaling. The three line segments numbered 1, 2, and 3 denote the wheelbase of the first car, the distance between cars, and the wheelbase of second car. These distances were measured in the physical world to be 93.0, 121.2, and 106.6 inches, and measured in the rectified image to be 94.4, 123.0, 108.5 inches. The average error between these measurements is 1.6%.

### Building

Shown in Figure A.4 is an image of two people standing outside of a store. The lines on the building were used with the line model, given in example A.6, to rectify the image. The transform components were constrained to be the same and several constraints were imposed on each line. First, the first two components of the line were constrained to be unit length:

$$l_1^2 + l_2^2 - 1 = 0,$$



**Figure A.4:** The building wall was rectified using vanishing lines and a known angle and length ratio. Shown on the right is the rectified image, from which measurements of the two people can be made.

where  $\vec{L} = ( l_1 \ l_2 \ l_3 )^T$ . Second, an orthogonality constraint was imposed on pairs of orthogonal lines:

$$l_1 m_1 + l_2 m_2 = 0,$$

where  $\vec{M} = ( m_1 \ m_2 \ m_3 )^T$  is a line orthogonal to  $\vec{L}$ .

The final scaling of the transform was determined by assuming that the decorative border in the corners of the window was approximately a square. Since the two people are standing in a plane that is approximately parallel to the plane of the store front, their relative heights can be measured after rectification. Using the height of the person on the left as a reference (64.75 inches), the height of the person on the right was estimated to be 69.4 inches. This person's actual height is 68.75 inches, yielding an error of 0.65 inches or 0.9%.

# Bibliography

- [1] Sung Joon Ahn. *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*, volume 3151 of *Lecture Notes in Computer Science*. Springer, 2004.
- [2] İsmail Avcıbaşı, Sevinç Bayram, Nasir Memon, Bülent Sankur, and Mahalingam Ramkumar. A classifier design for detecting image manipulations. In *2004 International Conference on Image Processing, ICIP '04*, volume 4, pages 2645–2648, 2004.
- [3] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.
- [4] Terrance E. Boult and George Wolberg. Correcting chromatic aberrations using image warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 684–687, 1992.
- [5] Dino A. Brugioni. *Photo fakery: the history and techniques of photographic deception and manipulation*. Brassey's, Dulles, VA, 1999.
- [6] Roberto Brunelli. Estimation of pose and illuminant direction for face processing. *Image and Vision Computing*, 15(10):741–748, 1997.
- [7] Wojciech Chojnacki and Michael J. Brooks. Revisiting Pentland's estimator of light source direction. *Journal of the Optical Society of America*, 11(1):118–124, 1994.
- [8] Gareth Cook. Technology seen abetting manipulation of research. *The Boston Globe*, 2006.
- [9] Ingemar Cox, Matthew Miller, and Jeffrey Bloom. *Digital Watermarking: Principles & Practice*. Morgan Kaufmann, 2001.
- [10] Antonio Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer Verlag, 2001.
- [11] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 189–198. ACM Press, 1998.



- [12] Ron O. Dror, Alan S. Willsky, and Edward H. Adelson. Statistical characterization of real-world illumination. *Journal of Vision*, 4(9):821–837, 2004.
- [13] Hany Farid. Creating and detecting doctored and virtual images: Implications to the child pornography prevention act. Technical Report TR2004-518, Department of Computer Science, Dartmouth College, 2004.
- [14] Hany Farid and Alin C. Popescu. Blind removal of lens distortions. *Journal of the Optical Society of America*, 18(9):2072–2078, 2001.
- [15] Hany Farid and Eero P. Simoncelli. Differentiation of multi-dimensional signals. *IEEE Transactions on Image Processing*, 13(4):496–508, 2004.
- [16] Roland W. Fleming, Ron O. Dror, and Edward H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, 3(5):347–368, 2003.
- [17] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, Inc., 2nd edition, 1990.
- [18] Jessica Fridrich, Mo Chen, and Miroslav Goljan. Imaging sensor noise as digital x-ray for revealing forgeries. In *9th International Workshop on Information Hiding*, Saint Malo, France, 2007.
- [19] Jessica Fridrich, David Soukal, and Jan Lukáš. Detection of copy-move forgery in digital images. In *Proceedings of DFRWS*, 2003.
- [20] Jiri Fridrich. Image watermarking for tamper detection. In *International Conference on Image Processing*, volume 2, pages 404–408, 1998.
- [21] Gene H. Golub, Per Christian Hansen, and Dianne P. O’Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.
- [22] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [23] Eugene Hecht. *Optics*. Addison-Wesley Publishing Company, Inc., 4th edition, 2002.
- [24] Michael J. Hogan, Jorge A. Alvarado, and Joan Esperson Weddell. *Histology of the Human Eye*. W.B Saunders Company, 1971.
- [25] Yu-Feng Hsu and Shih-Fu Chang. Image splicing detection using camera response function consistency and automatic segmentation. In *International Conference on Multimedia and Expo*, 2007.
- [26] Woo-Suk Hwang et al. Evidence of a pluripotent human embryonic stem cell line derived from a cloned blastocyst. *Science*, 303(5664):1669–1674, 2004.

- [27] D. Robert Iskander. A parametric approach to measuring limbus corneae from digital images. *IEEE Transactions on Biomedical Engineering*, 53(6):1134–1140, June 2006.
- [28] Alain Jaubert. *Le Commissariat aux Archives*. Bernard Barrault, 1992.
- [29] Micah K. Johnson and Hany Farid. Metric measurements on a plane from a single image. Technical Report TR2006-579, Department of Computer Science, Dartmouth College, 2006.
- [30] Stefan Katzenbeisser and Fabien Petitcolas, editors. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House Books, 1999.
- [31] Donald Kennedy. Editorial retraction. *Science*, 311(5759):335, 2006.
- [32] Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. *Proceedings of the IEEE (USA)*, 87(7):1167–1180, 1999.
- [33] Aaron Lefohn, Richard Caruso, Erik Reinhard, Brian Budge, and Peter Shirley. An ocularist’s approach to human iris synthesis. *IEEE Computer Graphics and Applications*, 23(6):70–75, 2003.
- [34] Ching-Yung Lin and Shih-Fu Chang. Robust image authentication method surviving JPEG lossy compression. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 296–307, 1998.
- [35] Zhouchen Lin, Rongrong Wang, Xiaoou Tang, and Heung-Yeung Shum. Detecting doctored images using camera response normality and consistency. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.
- [36] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. Detecting digital image forgeries using sensor pattern noise. In *Proceedings of the SPIE*, volume 6072, 2006.
- [37] Tian-Tsong Ng and Shih-Fu Chang. A model for image splicing. In *IEEE International Conference on Image Processing (ICIP)*, Singapore, October 2004.
- [38] Peter Nillius and Jan-Olof Eklundh. Automatic estimation of the projected light source direction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.
- [39] Ko Nishino and Shree K. Nayar. Eyes for relighting. *ACM Transactions on Graphics*, 23(3):704–711, 2004.
- [40] Yuri Ostrovsky, Patrick Cavanagh, and Pawan Sinha. Perceiving illumination inconsistencies in scenes. Technical Report AI Memo 2001-029, Massachusetts Institute of Technology, 2001.
- [41] Alex Pentland. Finding the illuminant direction. *Journal of the Optical Society of America*, 72(4):448–455, 1982.

- [42] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2004.
- [43] Jean-Marie Pinel, Henri Nicolas, and Carole Le Bris. Estimation of 2D illuminant direction and shadow segmentation in natural video sequences. In *Proceedings of VLDB*, pages 197–202, 2001.
- [44] Alin C. Popescu and Hany Farid. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR2004-515, Department of Computer Science, Dartmouth College, 2004.
- [45] Alin C. Popescu and Hany Farid. Statistical tools for digital forensics. In *Proceedings of the 6th Information Hiding Workshop*, 2004.
- [46] Alin C. Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.
- [47] Alin C. Popescu and Hany Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.
- [48] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM Press, 2001.
- [49] Ravi Ramamoorthi and Pat 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.
- [50] Andrzej Ruszczyński. *Nonlinear Optimization*. Princeton University Press, 2006.
- [51] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, Carnegie Mellon University, 1994.
- [52] Pawan Sinha. Perceiving illumination inconsistencies. In *Investigative Ophthalmology and Visual Science*, volume 41/4:1192, 2000.
- [53] David Sourlier. *Three dimensional feature independent bestfit in coordinate metrology*. PhD thesis, Swiss Federal Institute of Technology Zürich, 1995.
- [54] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. SIAM, 1997.
- [55] Paul Viola and William M. Wells, III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.
- [56] Nicholas Wade. Korean scientist said to admit fabrication in a cloning study. *The New York Times*, 2005.

- [57] Reg G. Willson and Steven A. Shafer. What is the center of the image? *Journal of the Optical Society of America A*, 11(11):2946–2955, November 1994.
- [58] Worth1000. Worth1000 home page. <http://www.worth1000.com>.
- [59] Yihong Wu, Haijiang Zhu, Zhanyi Hu, and Fuchao Wu. Camera calibration from the quasi-affine invariance of two parallel circles. In *European Conference on Computer Vision*, pages 190–202, 2004.
- [60] Yahoo. Flickr home page. <http://www.flickr.com>.
- [61] Xiang Zhou, Xiaohui Duan, and Daoxian Wang. A semi-fragile watermark scheme for image authentication. In *Proceedings of the 10th International Multimedia Modelling Conference*, page 374, 2004.