CHAPTER 20

Light and Surfaces

20.1 SIMPLE RADIOMETRY

TODO: Radiance; Irradiance; Radiosity; BRDF

20.2 LIGHT AND SURFACES

Three major phenomena determine the brightness of a pixel: the response of the camera to light, the fraction of light reflected from the surface to the camera, and the amount of light falling on the surface. Each can be dealt with quite straightforwardly.

Camera response: Modern camera sensors respond linearly to light. This linear response is adjusted in software, because humans find linear images confusing (such images tend to be too dark in most places, and too light in others). The *camera response function* or *CRF* determines what value is reported at each location. Typical CRF's are close to linear in mid-ranges, but have pronounced nonlinearities for darker and brighter illumination. This allows the camera to reproduce the very wide dynamic range of natural light without saturating.

Write $\mathbf{X}$ for a point in space that projects to $\mathbf{x}$ in the image, $I_{patch}(\mathbf{X})$ for the intensity of the surface patch at $\mathbf{X}$, $C(\cdot)$ for the camera response function, and $I_{camera}(\mathbf{x})$ for the camera response at $\mathbf{x}$. Then our model is:

$$I_{camera}(\mathbf{x}) = C(I_{patch}(\mathbf{x})).$$

It is quite usual to assume that the camera response is linearly related to the intensity of the surface patch. In this case, $C(I_{patch}(\mathbf{x})) = kI_{patch}(\mathbf{x})$, and it is common to assume that k is known if needed. A CRF can be recovered from enough image data, if required (Section 20.2.1).

Surface reflection: Different points on a surface may reflect more or less of the light that is arriving. Darker surfaces reflect less light, and lighter surfaces reflect more. There is a rich set of possible physical effects, but most can be ignored. Section 20.1.1 describes the relatively simple model that is sufficient for almost all purposes in computer vision.

Illumination: The amount of light a patch receives depends on the overall intensity of the light, and on the geometry. The overall intensity could change because some *luminaires* (the formal term for light sources) might be shadowed, or might have strong directional components. Geometry affects the amount of light arriving at a patch because surface patches facing the light collect more radiation and so are brighter than surface patches tilted away from the light, an effect known as *shading*. Section 20.1.2 describes the most important model used in computer vision; Section 20.1.4 describes a much more complex model that is necessary to explain some important practical difficulties in shading inference.

20.2.1 Reflection at Surfaces

Most surfaces reflect light by a process of *diffuse reflection*. Diffuse reflection scatters light evenly across the directions leaving a surface, so the brightness of a diffuse surface doesn't depend on the viewing direction. Examples are easy to identify with this test: most cloth has this property, as do most paints, rough wooden surfaces, most vegetation, and rough stone or concrete. The only parameter required to describe a surface of this type is its *albedo*, the fraction of the light arriving at the surface that is reflected. This does not depend on the direction in which the light arrives or the direction in which the light leaves. Surfaces with very high or very low albedo are difficult to make. For practical surfaces, albedo lies in the range 0.05 – 0.90 (see **?**, who argue the dynamic range is closer to 10 than the 18 implied by these numbers). Mirrors are not diffuse, because what you see depends on the direction in which you look at the mirror. The behavior of a perfect mirror is known as *specular reflection*. For an ideal mirror, light arriving along a particular direction can leave only along the *specular direction*, obtained by reflecting the direction of incoming radiation about the surface normal (Figure 20.1). Usually some fraction of incoming radiation is absorbed; on an ideal specular surface, this fraction does not depend on the incident direction.

If a surface behaves like an ideal specular reflector, you could use it as a mirror, and based on this test, relatively few surfaces actually behave like ideal specular reflectors. Imagine a near perfect mirror made of polished metal; if this surface suffers slight damage at a small scale, then around each point there will be a set of small facets, pointing in a range of directions. In turn, this means that light arriving in one direction will leave in several different directions because it strikes several facets, and so the specular reflections will be blurred. As the surface becomes less flat, these distortions will become more pronounced; eventually, the only specular reflection that is bright enough to see will come from the light source. This mechanism means that, in most shiny paint, plastic, wet, or brushed metal surfaces, one sees a bright blob—often called a *specularity*—along the specular direction from light sources, but few other specular effects. Specularities are easy to identify, because they are small and very bright (Figure 20.1; **?**). Most surfaces reflect only some of the incoming light in a specular component, and we can represent the percentage of light that is specularly reflected with a *specular albedo*. Although the diffuse albedo is an important material property that we will try to estimate from images, the specular albedo is largely seen as a nuisance and usually is not estimated.

For almost all purposes, it is enough to model all surfaces as being diffuse with specularities. This is the *lambertian+specular model*. Specularities are relatively seldom used in inference, and so there is no need for a formal model of their structure. Because specularities are small and bright, they are relatively easy to identify and remove with straightforward methods (find small bright spots, and replace them by smoothing the local pixel values). More sophisticated specularity finders use color information []. Thus, to apply the lambertian+specular model, we find and remove specularities, and then use Lambert's law (Section 20.1.2) to model image intensity.

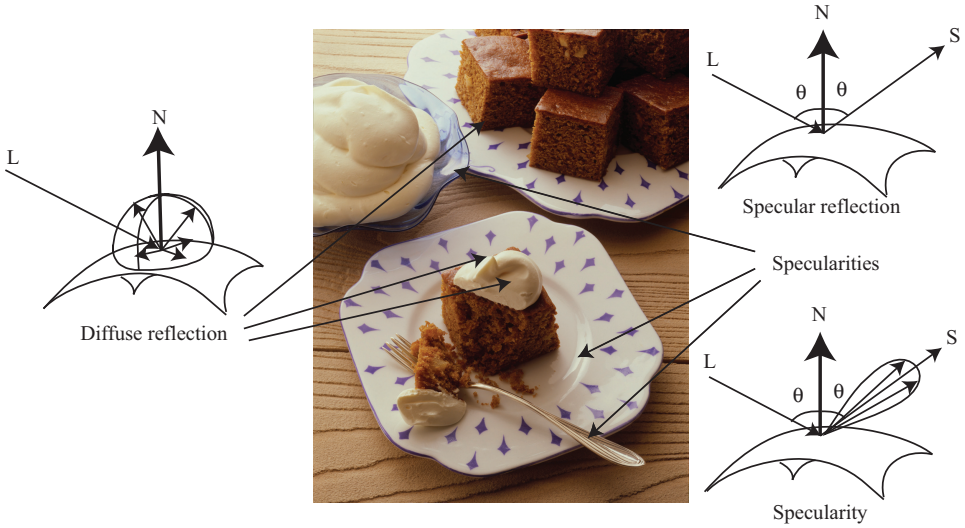

FIGURE 20.1: *The two most important reflection modes for computer vision are diffuse reflection (***left***), where incident light is spread evenly over the whole hemisphere of outgoing directions, and specular reflection (***right***), where reflected light is concentrated in a single direction. The specular direction* **S** *is coplanar with the normal and the source direction (***L***), and has the same angle to the normal that the source direction does. Most surfaces display both diffuse and specular reflection components. In most cases, the specular component is not precisely mirror like, but is concentrated around a range of directions close to the specular direction (***lower right***). This causes specularities, where one sees a mirror like reflection of the light source. Specularities, when they occur, tend to be small and bright. In the photograph, they appear on the metal spoon and on the plate. Large specularities can appear on flat metal surfaces (arrows). Most curved surfaces (such as the plate) show smaller specularities. Most of the reflection here is diffuse; some cases are indicated by arrows.* Martin Brigdale © Dorling Kindersley, used with permission.

20.2.2 Sources and Their Effects

The main source of illumination outdoors is the sun, whose rays all travel parallel to one another in a known direction because it is so far away. We model this behavior with a *distant point light source*. This is the most important model of lighting (because it is like the sun and because it is easy to use), and can be quite effective for indoor scenes as well as outdoor scenes. Because the rays are parallel to one another, a surface that faces the source cuts more rays (and so collects more light) than one oriented along the direction in which the rays travel. The amount of light collected by a surface patch in this model is proportional to the cosine of the angle θ between the illumination direction and the normal (Figure 20.2). The figure yields *Lambert's cosine law*, which states the brightness of a diffuse patch

illuminated by a distant point light source is given by

$$I = \rho I_0 \cos\theta,$$

where I_0 is the intensity of the light source, θ is the angle between the light source direction and the surface normal, and ρ is the diffuse albedo. This law predicts that bright image pixels come from surface patches that face the light directly and dark pixels come from patches that see the light only tangentially, so that the shading on a surface provides some shape information. We explore this cue in Section **??**.

If the surface cannot see the source, then it is in *shadow*. Since we assume that light arrives at our patch only from the distant point light source, our model suggests that shadows are deep black; in practice, they very seldom are, because the shadowed surface usually receives light from other sources. Outdoors, the most important such source is the sky, which is quite bright. Indoors, light reflected from other surfaces illuminates shadowed patches. This means that, for example, we tend to see few shadows in rooms with white walls, because any shadowed patch receives a lot of light from the walls. These effects are sometimes modelled by adding a constant *ambient illumination* term to the predicted intensity. The ambient term ensures that shadows are not too dark, but this is not a particularly good model of the spatial properties of interreflections. We have sketched the effects to be aware of in Section 20.1.4.

20.2.3 The Local Shading Model for Distant Luminaires

Surfaces reflect light onto one another (*interreflections*), meaning that the light arriving at a surface could have come directly from a luminaire, but it could also have been reflected from some other surface. Really accurate physical models of how light is distributed on scenes are now very well known [**?**] and are extremely useful in computer graphics. These models are very hard to use for inference, because every variable affects every other variable. For example, changes in the orientation of one surface element affect how much light it reflects onto every other surface element.

This means we must simplify the model, and so we must be using a model that isn't exact, meaning we need to keep track of what that model will do well and what it will do badly. The usual simplification is a *local shading model*, where we assume that shading is caused only by light that comes from the luminaire (i.e., that there are no interreflections).

Now assume that the luminaire is an infinitely distant source. For this case, write $\mathbf{N}(x)$ for the unit surface normal at $\mathbf{x}$, $\mathbf{S}$ for a vector pointing from $\mathbf{x}$ toward the source with length I_o (the source intensity), $\rho(\mathbf{x})$ for the albedo at $\mathbf{x}$, and $Vis(\mathbf{S}, \mathbf{x})$ for a function that is 1 when $\mathbf{x}$ can see the source and zero otherwise. Then, the intensity at $\mathbf{x}$ is

$$\begin{array}{ccccccc} I(\mathbf{x}) & = & \rho(\mathbf{x})\,(\mathbf{N}\cdot\mathbf{S})\,\mathrm{Vis}(\mathbf{S},\,\mathbf{x}) & + & \rho(\mathbf{x})A & + & M \\ \text{Image intensity} & = & \text{Diffuse term} & + & \text{Ambient term} & + & \text{Specular (mirror-like) term} \end{array}$$

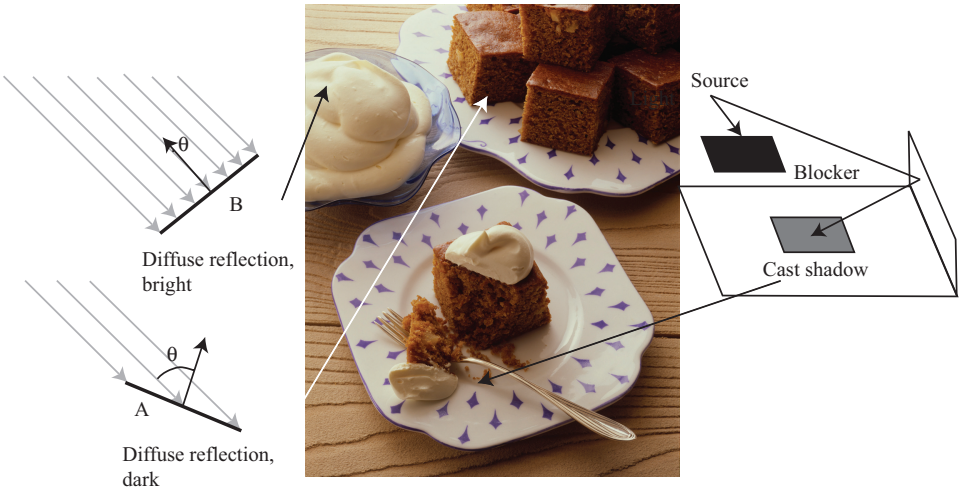

FIGURE 20.2: *The orientation of a surface patch with respect to the light affects how much light the patch gathers. We model surface patches as illuminated by a distant point source, whose rays are shown as light arrowheads. Patch A is tilted away from the source (θ is close to 90^0) and collects less energy, because it cuts fewer light rays per unit surface area. Patch B, facing the source (θ is close to 0^0), collects more energy, and so is brighter. Shadows occur when a patch cannot see a source. The shadows are not dead black, because the surface can see inter-reflected light from other surfaces. These effects are shown in the photograph. The darker surfaces are turned away from the illumination direction.* Martin Brigdale ⓒ Dorling Kindersley, used with permission.

20.2.4 Qualitative Effects of Area Sources

The local shading model is a good rough and ready model, but it isn't right. It predicts dark shadows with sharp boundaries. These are quite common outdoors where the sun is the most important light source, but are uncommon indoors. To understand why, we must look at area sources.

TODO: More material on radiometry of area sources

An *area source* is an area that radiates light. Area sources occur quite commonly in natural scenes—an overcast sky is a good example—and in synthetic environments—for example, the fluorescent light boxes found in many industrial ceilings. Area sources are common in illumination engineering, because they tend not to cast strong shadows and because the illumination due to the source does not fall off significantly as a function of the distance to the source. Detailed models of area sources are complex, but a simple model is useful to understand shadows. Shadows from area sources are very different from shadows cast by point sources. One seldom sees dark shadows with crisp boundaries indoors. Instead, one could see no visible shadows, or shadows that are rather fuzzy diffuse blobs, or sometimes fuzzy blobs with a dark core (Figure 20.3). These effects occur indoors because rooms tend to have light walls and diffuse ceiling fixtures, which act as area sources.

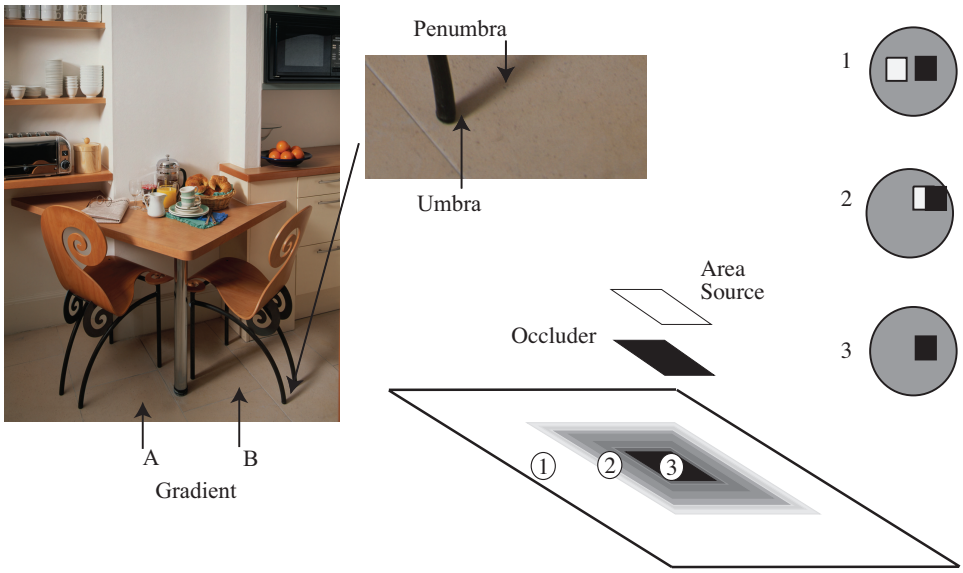

FIGURE 20.3: *Area sources generate complex shadows with smooth boundaries, because from the point of view of a surface patch, the source disappears slowly behind the occluder.* **Left:** *a photograph, showing characteristic area source shadow effects. Notice that A is much darker than B; there must be some shadowing effect here, but there is no clear shadow boundary. Instead, there is a fairly smooth gradient. The chair leg casts a complex shadow, with two distinct regions. There is a core of darkness (the* umbra*—where the source cannot be seen at all) surrounded by a partial shadow (*penumbra*— where the source can be seen partially). A good model of the geometry, illustrated* **right***, is to imagine lying with your back to the surface looking at the world above. At point 1, you can see all of the source; at point 2, you can see some of it; and at point 3, you can see none of it.* Peter Anderson ⓒ Dorling Kindersley, used with permission.

As a result, the shadows one sees are area source shadows.

To compute the intensity at a surface patch illuminated by an area source, we can break the source up into infinitesimal source elements, then sum effects from each element. If there is an occluder, then some surface patches may see none of the source elements. Such patches will be dark, and lie in the *umbra* (a Latin word meaning "shadow"). Other surface patches may see some, but not all, of the source elements. Such patches may be quite bright (if they see most of the elements), or relatively dark (if they see few elements), and lie in the *penumbra* (a compound of Latin words meaning "almost shadow"). One way to build intuition is to think of a tiny observer looking up from the surface patch. At umbral points, this observer will not see the area source at all whereas at penumbral points, the observer will see some, but not all, of the area source. An observer moving from outside the shadow, through the penumbra and into the umbra will see something that looks like an eclipse of the moon (Figure 20.3). The penumbra can be large, and can change quite

FIGURE 20.4: *The photograph on the* **left** *shows a room interior. Notice the lighting has some directional component (the vertical face indicated by the arrow is dark, because it does not face the main direction of lighting), but there are few visible shadows (for example, the chairs do not cast a shadow on the floor). On the* **right**, *a drawing to show why; here there is a small occluder and a large area source. The occluder is some way away from the shaded surface. Generally, at points on the shaded surface the incoming hemisphere looks like that at point 1. The occluder blocks out some small percentage of the area source, but the amount of light lost is too small to notice (compare figure 20.3).* Jake Fitzjones © Dorling Kindersley, used with permission.

slowly from light to dark. There might even be no umbral points at all, and, if the occluder is sufficiently far away from the surface, the penumbra could be very large and almost indistinguishable in brightness from the unshadowed patches. This is why many objects in rooms appear to cast no shadow at all (Figure 20.4).

20.3 INFERENCE FROM SIMPLE SHADING MODELS

20.3.1 Radiometric Calibration and High Dynamic Range Images

The intensity of light travelling through a point in space in some direction is represented with a unit known as *radiance*. The intensity of light arriving at a point on a surface averaged over some range of directions is known as *irradiance*. Sensors average the irradiance over the area of a pixel to obtain incoming power E. This power is summed for some time period Δt to obtain the amount of energy the pixel receives. In turn, the energy determines the pixel intensity value reported by the imaging system. A property called *reciprocity* means that the response is a function of $E\Delta t$ alone. In particular, we will get the same outcome if we image one patch of intensity E for time Δt and another patch of intensity E/k for time $k\Delta t$. The actual response that the sensor produces is a function of $E\Delta t$. Determining this function from data is known as *radiometric calibration*.

Radiometric calibration has a number of applications. For example, we might want to compare renderings of a scene with pictures of the scene, and to do that we need to work in real radiometric units and so must calibrate the camera radiomet-

rically. We might want to use pictures of a scene to estimate the lighting in that scene so we can postrender new objects into the scene, which would need to be lit correctly. Again, we would need to use radiometric units and so need to calibrate the camera.

Likely the most important application is *high dynamic range imaging* or *HDR imaging*. Many scenes have bright spots that are very much brighter than the dark spots. The dynamic range is the ratio of brightest to darkest spot. The camera response function (CRF) is typically somewhat linear over some range, and sharply non-linear near the top and bottom of this range, so that the camera can capture very dark and very light patches without saturation. However, it is quite easy to find scenes where the dynamic range is so big that images in a reasonable camera loses information. Either the brightest points are saturated or the darkest points are very close to zero. In either case, color and relative intensity information is lost. However, if we have multiple images of the scene, obtained with different values of Δt, then we can recover information that would otherwise be lost. Using a small Δt will allow very bright locations to be measured accurately (though mid range locations will be dark, and dark locations will be lost). Using a large Δt will allow very dark locations to be measured accurately (though mid range locations will be bright, and bright locations will be lost). If the CRF is known, then for each location at each Δt_i we can compute the value of $E\Delta t_i$ and so recover E for each location exactly.

Now assume we have multiple registered images, each obtained using a different exposure time. At the i, j'th pixel, we know the image intensity value $I_{ij}^{(k)}$ for the k'th exposure time, we know the value of the k'th exposure time Δt_k, and we know that the intensity of the corresponding surface patch E_{ij} is the same for each exposure, but we do not know the value of E_{ij}. Write the camera response function f, so that

$$I_{ij}^{(k)} = f(E_{ij}\Delta t_k).$$

There are now several possible approaches to solve for f. We could assume a parametric form—say, polynomial—then solve using least squares. Notice that we must solve not only for the parameters of f, but also for E_{ij}. For a color camera, we solve for calibration of each channel separately. **?** have studied the polynomial case in detail. Though the solution is not unique, ambiguous solutions are strongly different from one another, and most cases are easily ruled out. Furthermore, one does not need to know exposure times with exact accuracy to estimate a solution, as long as there are sufficient pixel values; instead, one estimates f from a fixed set of exposure times, then estimates the exposure times from f, and then re-estimates. This procedure is stable.

Alternatively, because the camera response is monotonic, we can work with its inverse $g = f^{-1}$, take logs, and write

$$\log g(I_{ij}^{(k)}) = \log E_{ij} + \log \Delta t_k.$$

We can now estimate the values that g takes at each point and the E_{ij} by placing a smoothness penalty on g. In particular, we minimize

$$\sum_{i,j,k} (\log g(I_{ij}^{(k)}) - (\log E_{ij} + \log \Delta t_k))^2 + \text{smoothness penalty on } g$$

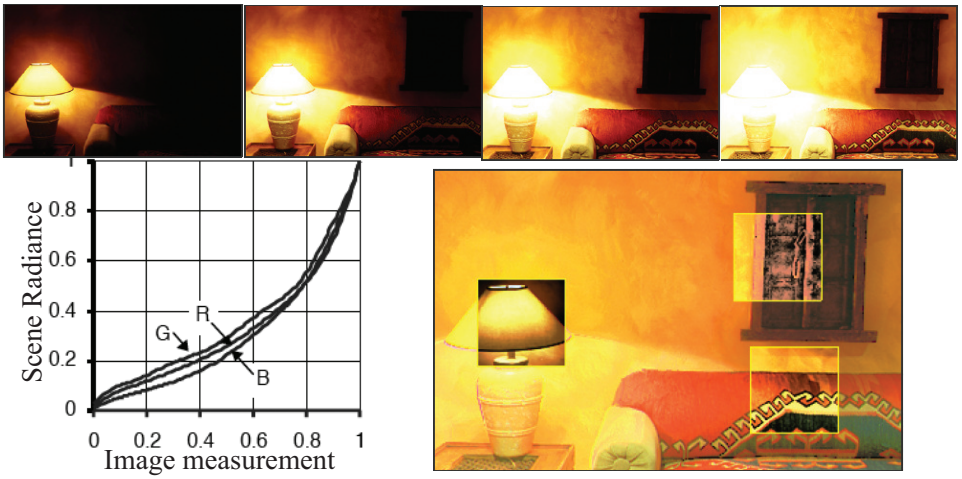

FIGURE 20.5: *It is possible to calibrate the radiometric response of a camera from multiple images obtained at different exposures. The* **top** *row shows four different exposures of the same scene, ranging from darker (shorter shutter time) to lighter (longer shutter time). Note how, in the dark frames, the lighter part of the image shows detail, and in the light frames, the darker part of the image shows detail; this is the result of non-linearities in the camera response. On the* **bottom left**, *we show the inferred calibration curves for each of the R, G, and B camera channels. On the* **bottom right**, *a composite image illustrates the results. The dynamic range of this image is far too large to print; instead, the main image is normalized to the print range. Overlaid on this image are boxes where the radiances in the box have also been normalized to the print range; these show how much information is packed into the high dynamic range image.*

by choice of g. **?** penalize the second derivative of g. Once we have a radiometrically calibrated camera, estimating a high dynamic range image is relatively straightforward. We have a set of registered images, and at each pixel location, we seek the estimate of radiance that predicts the registered image values best. In particular, we assume we know f. We seek an E_{ij} such that

$$\sum_k w(I_{ij})(I_{ij}^{(k)} - f(E_{ij}\Delta t_k))^2$$

is minimized. Notice the weights because our estimate of f is more reliable when I_{ij} is in the middle of the available range of values than when it is at larger or smaller values.

20.3.2 Inferring Lightness and Illumination

If we could estimate the albedo of a surface from an image, then we would know a property of the surface itself, rather than a property of a picture of the surface. Such properties are often called *intrinsic representations*. They are worth estimating, because they do not change when the imaging circumstances change. It might seem

that albedo is difficult to estimate, because there is an ambiguity linking albedo and illumination; for example, a high albedo viewed under middling illumination will give the same brightness as a low albedo viewed under bright light. However, humans can report whether a surface is white, gray, or black (the *lightness* of the surface), despite changes in the intensity of illumination (the *brightness*). This skill is known as *lightness constancy*. There is a lot of evidence that human lightness constancy involves two processes: one process compares the brightness of various image patches and uses this comparison to determine which patches are lighter and which darker; the second establishes some form of absolute standard to which these comparisons can be referred (e.g. **?**).

It is worth reviewing early algorithms for estimating lightness briefly, because the underlying principles remain useful. These algorithms were developed in the context of simple scenes. In particular, we assume that the scene is flat and frontal; that surfaces are diffuse, or that specularities have been removed; and that the camera responds linearly. In this case, the camera response C at a point $\mathbf{x}$ is the product of an illumination term, an albedo term, and a constant that comes from the camera gain:

$$C(\mathbf{x}) = k_c I(\mathbf{x})\rho(\mathbf{x}).$$

If we take logarithms, we get

$$\log C(\mathbf{x}) = \log k_c + \log I(\mathbf{x}) + \log \rho(\mathbf{x}).$$

We now make a second set of assumptions:

- First, we assume that albedoes change only quickly over space. This means that a typical set of albedoes will look like a collage of papers of different grays. This assumption is quite easily justified: There are relatively few continuous changes of albedo in the world (the best example occurs in ripening fruit), and changes of albedo often occur when one object occludes another (so we would expect the change to be fast). This means that spatial derivatives of the term $\log \rho(\mathbf{x})$ are either zero (where the albedo is constant) or large (at a change of albedo).

- Second, illumination changes only slowly over space. This assumption is somewhat realistic. For example, the illumination due to a point source will change relatively slowly unless the source is very close, so the sun is a particularly good source for this method, as long as there are no shadows. As another example, illumination inside rooms tends to change very slowly because the white walls of the room act as area sources. This assumption fails dramatically at shadow boundaries, however. We have to see these as a special case and assume that either there are no shadow boundaries or that we know where they are.

These assumptions are sometimes called *Mondrian world* assumptions.

The earliest algorithm is the Retinex algorithm of **?**; this took several forms, most of which have fallen into disuse. The key insight of Retinex is that small gradients are changes in illumination, and large gradients are changes in lightness. We can use this by differentiating the log transform, throwing away small gradients,

and integrating the results [**?**]. Doing this, or something like it, is widely known as Retinex. There is a constant of integration missing, so lightness ratios are available, but absolute lightness measurements are not. Figure **??** illustrates the process for a one-dimensional example, where differentiation and integration are easy.

This approach can be extended to two dimensions as well. Differentiating and thresholding is easy: at each point, we estimate the magnitude of the gradient; if the magnitude is less than some threshold, we set the gradient vector to zero; otherwise, we leave it alone. The difficulty is in integrating these gradients to get the log albedo map. The thresholded gradients may not be the gradients of an image because the mixed second partials may not be equal (integrability again; compare with Section 20.2.3).

Form the gradient of the log of the image
At each pixel, if the gradient magnitude is below
  a threshold, replace that gradient with zero
Reconstruct the log-albedo by solving the minimization
  problem described in the text
Obtain a constant of integration
Add the constant to the log-albedo, and exponentiate

Algorithm 20.1: *Determining the Lightness of Image Patches.*

The problem can be rephrased as a minimization problem: choose the log albedo map whose gradient is most like the thresholded gradient. This is a relatively simple problem because computing the gradient of an image is a linear operation. The x-component of the thresholded gradient is scanned into a vector $\mathbf{p}$, and the y-component is scanned into a vector $\mathbf{q}$. We write the vector representing log-albedo as $\mathbf{l}$. Now the process of forming the x derivative is linear, and so there is some matrix $\mathcal{M}_x$, such that $\mathcal{M}_x\mathbf{l}$ is the x derivative; for the y derivative, we write the corresponding matrix $\mathcal{M}_y$.

The problem becomes finding the vector $\mathbf{l}$ that minimizes

$$| \mathcal{M}_x\mathbf{l} - \mathbf{p} |^2 + | \mathcal{M}_y\mathbf{l} - \mathbf{q} |^2 .$$

This is a quadratic minimization problem, and the answer can be found by a linear process. Some special tricks are required because adding a constant vector to $\mathbf{l}$ cannot change the derivatives, so the problem does not have a unique solution. We explore the minimization problem in the exercises.

The constant of integration needs to be obtained from some other assumption. There are two obvious possibilities:

- we can assume that the *brightest patch is white*;
- we can assume that the *average lightness is constant*.

We explore the consequences of these models in the exercises.

More sophisticated algorithms are now available, but there were no quantitative studies of performance until recently. Grosse *et al.* built a dataset for

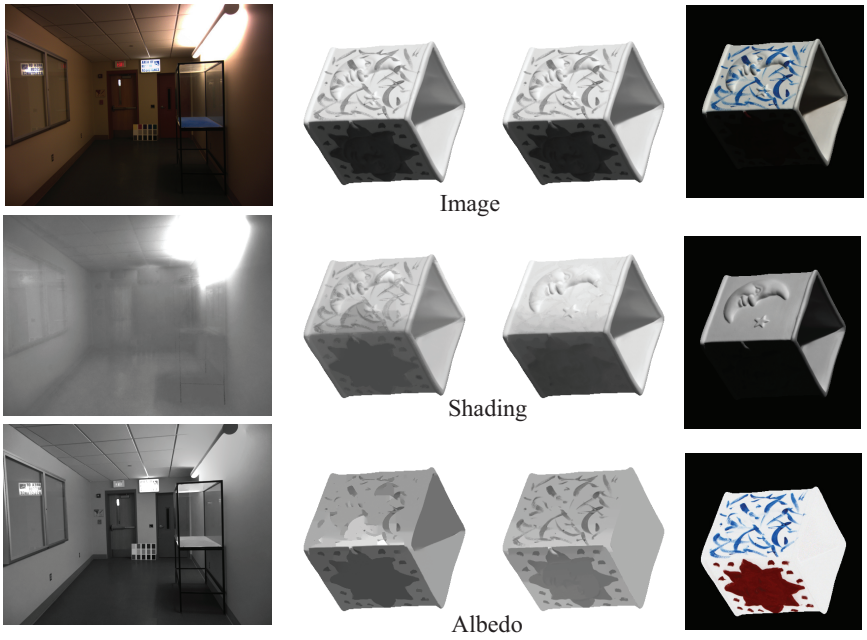

FIGURE 20.6: *Retinex remains a strong algorithm for recovering albedo from images. Here we show results from the version of Retinex described in the text applied to an image of a room (***left***) and an image from a collection of test images due to* ***?****. The* **center-left** *column shows results from Retinex for this image, and the* **center-right** *column shows results from a variant of the algorithm that uses color reasoning to improve the classification of edges into albedo versus shading. Finally, the* **right** *column shows the correct answer, known by clever experimental methods used when taking the pictures. This problem is very hard; you can see that the albedo images still contain some illumination signal.* Part of this figure courtesy Kevin Karsch, U. Illinois.

evaluating lightness algorithms, and show that a version of the procedure we describe performs extremely well compared to more sophisticated algorithms [**?**]. The major difficulty with all these approaches is caused by shadow boundaries, which we discuss in Section 22.2.1.

20.3.3 Photometric Stereo: Shape from Multiple Shaded Images

It is possible to reconstruct a patch of surface from a series of pictures of that surface taken under different illuminants. First, we need a camera model. For simplicity, we choose an orthographic camera situated so that the point (x, y, z) in space is imaged to the point (x, y) in the camera (the method can be extended to the other camera models described in Chapter **??**).

In this case, to measure the shape of the surface, we need to obtain the depth to the surface. This suggests representing the surface as $(x, y, f(x, y))$—a representation known as a *Monge patch* after the French military engineer who first

used it (Figure 20.7). This representation is attractive because we can determine a unique point on the surface by giving the image coordinates. Notice that to obtain a measurement of a solid object, we would need to reconstruct more than one patch because we need to observe the back of the object.

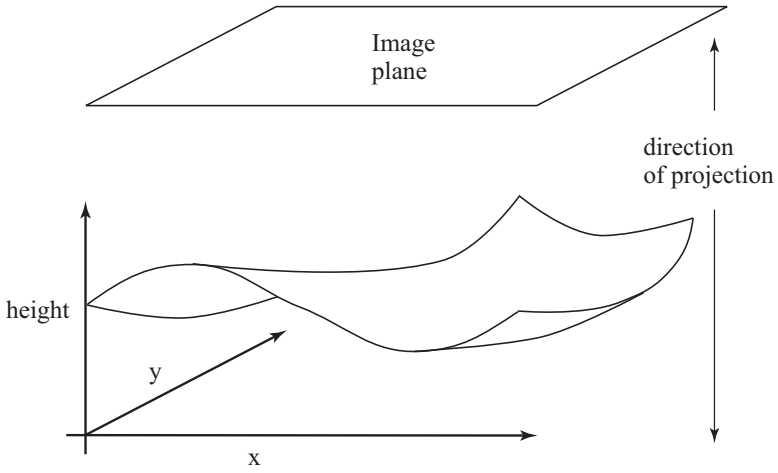

FIGURE 20.7: *A Monge patch is a representation of a piece of surface as a height function. For the photometric stereo example, we assume that an orthographic camera—one that maps (x, y, z) in space to (x, y) in the camera—is viewing a Monge patch. This means that the shape of the surface can be represented as a function of position in the image.*

Photometric stereo is a method for recovering a representation of the Monge patch from image data. The method involves reasoning about the image intensity values for several different images of a surface in a fixed view illuminated by different sources. This method recovers the height of the surface at points corresponding to each pixel; in computer vision circles, the resulting representation is often known as a *height map*, *depth map*, or *dense depth map*.

Fix the camera and the surface in position, and illuminate the surface using a point source that is far away compared with the size of the surface. We adopt a local shading model and assume that there is no ambient illumination (more about this later) so that the brightness at a point $\mathbf{x}$ on the surface is

$$B(\mathbf{x}) = \rho(\mathbf{x})\mathbf{N}(\mathbf{x}) \cdot \mathbf{S}_1,$$

where $\mathbf{N}$ is the unit surface normal and $\mathbf{S}_1$ is the source vector. We can write $B(x, y)$ for the radiosity of a point on the surface because there is only one point on the surface corresponding to the point (x, y) in the camera. Now we assume that the response of the camera is linear in the surface radiosity, and so have that the

value of a pixel at (x, y) is

$$\begin{aligned} I(x,y) &= kB(\mathbf{x}) \\ &= kB(x,y) \\ &= k\rho(x,y)\mathbf{N}(x,y) \cdot \mathbf{S}_1 \\ &= \mathbf{g}(x,y) \cdot \mathbf{V}_1, \end{aligned}$$

where $\mathbf{g}(x, y) = \rho(x, y)\mathbf{N}(x, y)$ and $\mathbf{V}_1 = k\mathbf{S}_1$, where k is the constant connecting the camera response to the input radiance.

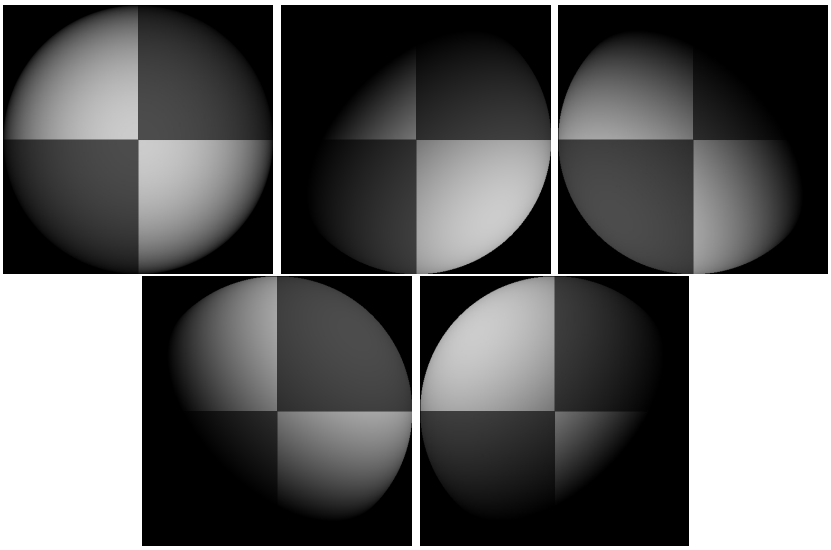

FIGURE 20.8: *Five synthetic images of a sphere, all obtained in an orthographic view from the same viewing position. These images are shaded using a local shading model and a distant point source. This is a convex object, so the only view where there is no visible shadow occurs when the source direction is parallel to the viewing direction. The variations in brightness occuring under different sources code the shape of the surface.*

In these equations, $\mathbf{g}(x, y)$ describes the surface, and $\mathbf{V}_1$ is a property of the illumination and of the camera. We have a dot product between a vector field $\mathbf{g}(x, y)$ and a vector $\mathbf{V}_1$, which could be measured; with enough of these dot products, we could reconstruct $\mathbf{g}$ and so the surface.

Now if we have n sources, for each of which $\mathbf{V}_i$ is known, we stack each of these $\mathbf{V}_i$ into a known matrix $\mathcal{V}$, where

$$\mathcal{V} = \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \cdots \\ \mathbf{V}_n^T \end{pmatrix}.$$

For each image point, we stack the measurements into a vector

$$\mathbf{i}(x,y) = \{I_1(x,y), I_2(x,y), \ldots, I_n(x,y)\}^T.$$

Notice that we have one vector per image point; each vector contains all the image brightnesses observed at that point for different sources. Now we have

$$\mathbf{i}(x,y) = \mathcal{V}\mathbf{g}(x,y),$$

and $\mathbf{g}$ is obtained by solving this linear system—or rather, one linear system per point in the image. Typically, $n > 3$, so that a least-squares solution is appropriate. This has the advantage that the residual error in the solution provides a check on our measurements.

Substantial regions of the surface might be in shadow for one or the other light (see Figure 20.8). We assume that all shadowed regions are known, and deal only with points that are not in shadow for any illuminant. More sophisticated strategies can infer shadowing because shadowed points are darker than the local geometry predicts.

We can extract the albedo from a measurement of $\mathbf{g}$ because $\mathbf{N}$ is the unit normal. This means that $|\mathbf{g}(x,y)| = \rho(x,y)$. This provides a check on our measurements as well. Because the albedo is in the range zero to one, any pixels where $|\mathbf{g}|$ is greater than one are suspect—either the pixel is not working or $\mathcal{V}$ is incorrect. Figure 20.9 shows albedo recovered using this method for the images shown in Figure 20.8.

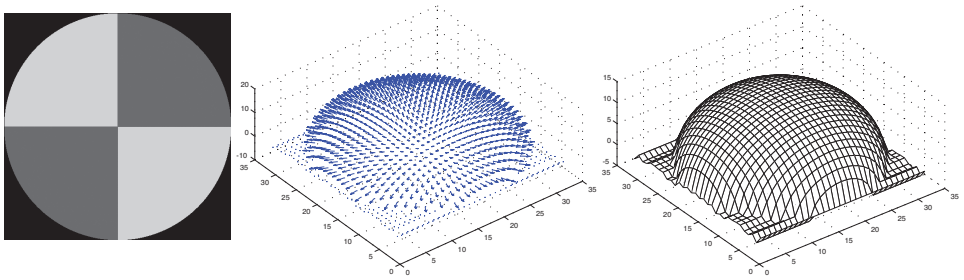

FIGURE 20.9: *The image on the* **left** *shows the magnitude of the vector field* $\mathbf{g}(x,y)$ *recovered from the input data of Figure 20.8 represented as an image—this is the reflectance of the surface. The* **center** *figure shows the normal field, and the* **right** *figure shows the height field.*

We can extract the surface normal from $\mathbf{g}$ because the normal is a unit vector

$$\mathbf{N}(x,y) = \frac{\mathbf{g}(x,y)}{|\mathbf{g}(x,y)|}.$$

Figure 20.9 shows normal values recovered for the images of Figure 20.8.

The surface is $(x, y, f(x,y))$, so the normal as a function of (x,y) is

$$\mathbf{N}(x,y) = \frac{1}{\sqrt{1 + \frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}} \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, 1 \right\}^T.$$

To recover the depth map, we need to determine $f(x, y)$ from measured values of the unit normal.

Obtain many images in a fixed view under different illuminants
Determine the matrix $\mathcal{V}$ from source and camera information

Inferring albedo and normal:
For each point in the image array that is not shadowed
  Stack image values into a vector $\mathbf{i}$
  Solve $\mathcal{V}\mathbf{g} = \mathbf{i}$ to obtain $\mathbf{g}$ for this point
  Albedo at this point is $|\mathbf{g}|$
  Normal at this point is $\frac{\mathbf{g}}{|\mathbf{g}|}$
  p at this point is $\frac{N_1}{N_3}$
  q at this point is $\frac{N_2}{N_3}$
end
Check: is $(\frac{\partial p}{\partial y} - \frac{\partial q}{\partial x})^2$ small everywhere?

Integration:
Top left corner of height map is zero
For each pixel in the left column of height map
  height value = previous height value + corresponding q value
end
For each row
  For each element of the row except for leftmost
    height value = previous height value + corresponding p value
  end
end

Algorithm 20.2: *Photometric Stereo.*

Assume that the measured value of the unit normal at some point (x, y) is $(a(x, y), b(x, y), c(x, y))$. Then

$$\frac{\partial f}{\partial x} = \frac{a(x, y)}{c(x, y)} \text{ and } \frac{\partial f}{\partial y} = \frac{b(x, y)}{c(x, y)}.$$

We have another check on our data set, because

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x},$$

so we expect that

$$\frac{\partial \left(\frac{a(x,y)}{c(x,y)}\right)}{\partial y} - \frac{\partial \left(\frac{b(x,y)}{c(x,y)}\right)}{\partial x}$$

should be small at each point. In principle it should be zero, but we would have to estimate these partial derivatives numerically and so should be willing to accept

FIGURE 20.10: *Photometric stereo could become the method of choice to capture complex deformable surfaces. On the* **top**, *three images of a garment, lit from different directions, which produce the reconstruction shown on the* **top right**. *A natural way to obtain three different images at the same time is to use a color camera; if one has a red light, a green light, and a blue light, then a single color image frame can be treated as three images under three separate lights. On the* **bottom**, *an image of the garment captured in this way, which results in the photometric stereo reconstruction on the* **bottom right**.

small values. This test is known as a test of *integrability*, which in vision applications always boils down to checking that mixed second partials are equal.

Assuming that the partial derivatives pass this sanity test, we can reconstruct the surface up to some constant depth error. The partial derivative gives the change in surface height with a small step in either the x or the y direction. This means we can get the surface by summing these changes in height along some path. In particular, we have

$$f(x,y) = \oint_C \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \cdot d\mathbf{l} + c,$$

where C is a curve starting at some fixed point and ending at (x,y), and c is a constant of integration, which represents the (unknown) height of the surface at the start point. The recovered surface does not depend on the choice of curve (exercises). Another approach to recovering shape is to choose the function $f(x,y)$ whose partial derivatives most look like the measured partial derivatives. Figure 20.9 shows the reconstruction obtained for the data shown in Figure 20.8.

Current reconstruction work tends to emphasize geometric methods that reconstruct from multiple views. These methods are very important, but often require feature matching, as we shall see in Chapters **??** and **??**. This tends to mean that it is hard to get very high spatial resolution, because some pixels are consumed in resolving features. Recall that resolution (which corresponds roughly to the

spatial frequencies that can be reconstructed accurately) is not the same as accuracy (which involves a method providing the right answers for the properties it estimates). Feature-based methods are capable of spectacularly accurate reconstructions. Because photometric cues have such spatial high resolution, they are a topic of considerable current interest. One way to use photometric cues is to try and match pixels with the same brightness across different cameras; this is difficult, but produces impressive reconstructions. Another is to use photometric stereo ideas. For some applications, photometric stereo is particularly atractive because one can get reconstructions from a single view direction—this is important, because we cannot always set up multiple cameras. In fact, with a trick, it is possible to get reconstructions from a single frame. A natural way to obtain three different images at the same time is to use a color camera; if one has a red light, a green light and a blue light, then a single color image frame can be treated as three images under three separate lights, and photometric stereo methods apply. In turn, this means that photometric stereo methods could be used to recover high-resolution reconstructions of deforming surfaces in a relatively straightforward way. This is particularly useful when it is difficult to get many cameras to view the object. Figure 20.10 shows one application to reconstructing cloth in video (from **?**), where multiple view reconstruction is complicated by the need to synchronize frames (alternatives are explored in, for example, **?** or **?**).