CHAPTER 27

Modelling Pixel Intensity

A camera collects power P at a pixel for time Δt and produces an output $C(P\Delta t)$. The power that it collects is determined by the average intensity of light leaving a small patch on a single surface in the direction of the pixel. In turn, this is determined by two factors.

- The amount of light reflected from the surface patch to the pixel (Section 27.1).
- The amount of light collected by the surface, which is affected by the orientation of the surface patch (Section 27.2), and by which light sources the patch can see (Section 27.3).

This yields quite simple models. These models offer useful insights, though many of their predictions are not consistent with easy observations. However, the models are strong enough to allow some inferences from shading (Section 27.4).

27.1 REFLECTION AT SURFACES

When light arrives at a surface, some fraction of the light is absorbed by the surface and the rest leaves again. There is a hemisphere of possible directions that the light can leave by. Most cases are dealt with by a combination of two models. In diffuse reflection, the light leaving the surface is evenly distributed across the outgoing directions. In specular reflection, there are tight geometric relationships between the direction in which the light arrives and the direction in which it leaves. Models of other effects are seldom used in vision, but are sketched below.

27.1.1 Diffuse Reflection

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. Albedo can vary with wavelength, which is one reason surfaces appear colored (Chapter 21.3).

27.1.2 Specular Reflection

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*.

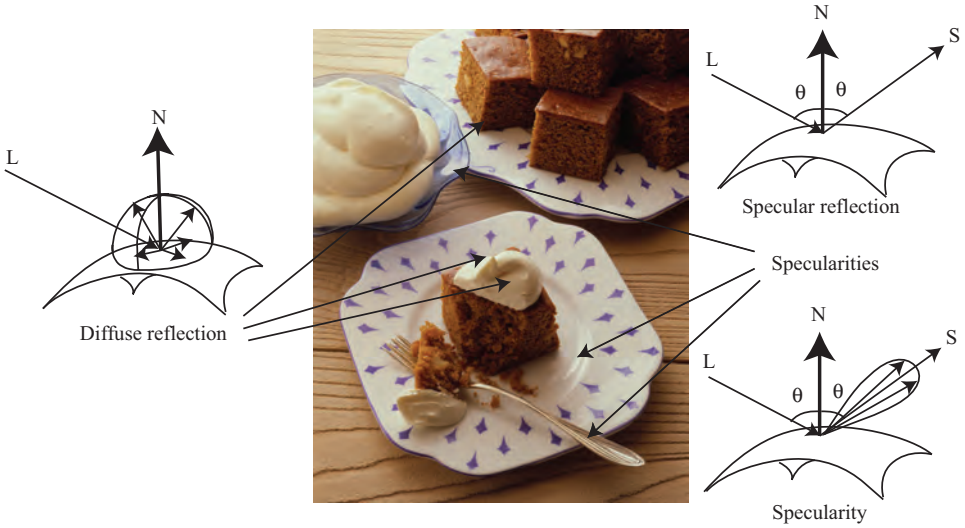

FIGURE 27.1: *The two most important reflection modes for computer vision are diffuse reflection (***left***), where incident light is spread evenly over the whole hemisphere of outgoing directions, and specular reflection (***right***), where reflected light is concentrated in a single direction. The 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.

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 27.1 and exercises). Usually some fraction of incoming radiation is absorbed; on an ideal specular surface, this fraction does not depend on the incident direction.

27.1.3 Specularities

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 27.1). 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*. For surfaces that conduct electricity (mostly, metal surfaces), specularities take on a color characteristic of the material. For example, gold has yellow specularities; copper has orange specularities; and osmium has blue-gray specularities (or so I'm told!). For surfaces that do not conduct electricity, specularities are typically the color of the light source.

27.1.4 Other Phenomena

The vast majority of vision work ignores anything other than diffuse and specular effects. This isn't a major source of problems, but what actually happens to light at surfaces is often complicated. There are two major classes of model. In the simplest class, all light leaving a point on a surface arrives at that point, and energy doesn't change wavelength. For this class of model, a complete description of the effects is a table of how incoming light at various different angles of incidence causes outgoing light at different exit angles, ideally as a function of wavelength. This table – a *bidirectional reflectance distribution function* or *BRDF* – can be measured for a given surface (though measurement is onerous). One can build various parametric models of a BRDF as well. In fact, both the diffuse and the specular models are parametric BRDF models. Common effects that can be encoded in this way are *specular backscatter* (where a surface scatters light back in the direction of the illumination), *off-specular glint* (where one sees effects like specularities in unexpected directions) and *gloss* (where a surface appears to shine).

Even more complex interactions occur quite commonly. Light arriving at a surface could penetrate the surface, wander around the volume, and emerge somewhere else (this is why you can see the veins of a person with sufficiently pale skin). It could be trapped in a layer of oil, travel inside this layer for a bit, then emerge (something that occurs on skin quite often). It could be absorbed by the surface, then re-emitted at a different wavelength (an effect known as *fluorescence*).

27.2 POINT SOURCES AND LOCAL SHADING

The amount and color of light falling on a surface depends on the *luminaires* (the formal term for light sources), on the relative geometry of luminaire and surface, and on other surfaces in the scene. The amount of light a luminaire produces usually varies across wavelength, which is why different luminaires appear to have different colors – for example, sunlight is relatively yellow, and fluorescent light is relatively blue (Chapter 21.3). In the simplest model, surfaces that can see a luminaire collect light from it, and from it alone. This is the *local shading model*. It is quite easy to understand and work with, but it is not physically correct, and

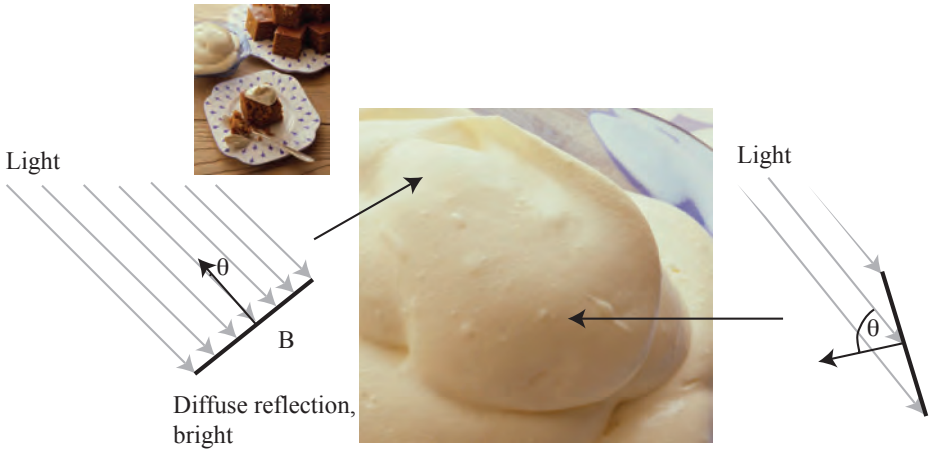

FIGURE 27.2: *The orientation of a surface patch with respect to the light affects how much light the patch gathers. A surface patch is illuminated by a distant point source, whose rays are shown as light gray arrows. The small inset image shows the whole scene, and the image at the center is a detail of the bowl of cream. The two patches identified by the arrowheads are both on the cream, and so have the same albedo – any difference in brightness is due to source effects. On the* **left**, *a patch is facing the source (θ is close to 0^0), collects more energy, and so is brighter. On the* **right**, *a patch 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.* Martin Brigdale © Dorling Kindersley, used with permission.

should be used with caution.

27.2.1 Distant Point Sources and Lambert's Law

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 27.2). Further, the effects of illumination are linear, meaning that the amount of light collected by a surface patch is proportional to the intensity of the source.

The figure yields *Lambert's cosine law* or, more usually, *Lambert's law*. Lambert's law yields the diffuse intensity $I(\mathbf{x})$ at a diffuse surface patch $\mathbf{x}$. Write S for the source intensity, $\rho(\mathbf{x})$ for the albedo of the patch, and $\theta(\mathbf{x})$ for the angle

between the surface normal at $\mathbf{x}$ and the direction to the source. Then

$$I(\mathbf{x}) = \rho(\mathbf{x})S\cos\theta(\mathbf{x})$$

Now represent the light source with a source vector $\mathbf{S}$ that points *toward* the source. If the magnitude of $\mathbf{S}$ is S, this becomes

$$I(\mathbf{x}) = \rho(\mathbf{x})\mathbf{S}^T\mathbf{N}(\mathbf{x})$$

where $\mathbf{N}(\mathbf{x})$ is the outward pointing unit normal to the surface at $\mathbf{x}$. This law predicts the shading on a surface provides some shape information, because bright image pixels come from surface patches that face the light directly and dark pixels come from patches that see the light only tangentially. Section 21.3 shows one procedure for exploiting this information.

27.2.2 Nearby Point Sources

A much less common shading model is a *nearby point source*, where the luminaire is a glowing point that is nearby. Lambert's law works in this case as well. For a distant point source, the direction to the source is the same everywhere, but for a nearby point source the direction changes from point to point, so $\mathbf{S}$ becomes $\mathbf{S}(\mathbf{x})$ yielding

$$I(\mathbf{x}) = \rho(\mathbf{x})\mathbf{S}^T(\mathbf{x})\mathbf{N}(\mathbf{x}).$$

It is quite easy to see that a nearby point source model is not a particularly good model. Place a point source at the center of a room that is a cube. It is an exercise to show the model predicts that the corners of the room are about 60% as bright as the center of each wall. But rooms aren't like that. One strategy used to deal with this is to add what is known as an *ambient illumination* term to the prediction made by Lambert's law. This term is usually a constant everywhere. It is an exercise to check this strategy is not a good predictor of observations.

27.2.3 Local Shading and the Diffuse + Specular Model

The assumption that all reflection effects are either diffuse or specular is widespread and useful. Typically, one assumes that specular effects are actually specularities (so, no mirrors). These specularities are identified (small bright points), and are removed. What remains is diffuse reflection which is then modelled with Lambert's law.

27.3 SHADOWS, AREA SOURCES AND INTERREFLECTIONS

Most people know what a *shadow* is, but defining shadows accurately is surprisingly hard. Different points on a surface may receive quite different amounts of light, because the light arriving at the *receiver* (point on the surface receiving light) changes. Drawing the hemisphere of directions arriving at the receiver shows how these changes work. In the local shading model, a receiver can either see a source or can't. When it can't, it is in shadow. This yields an acceptable model of some outdoor shadows, but more complex shadows require a more sophisticated explanation.

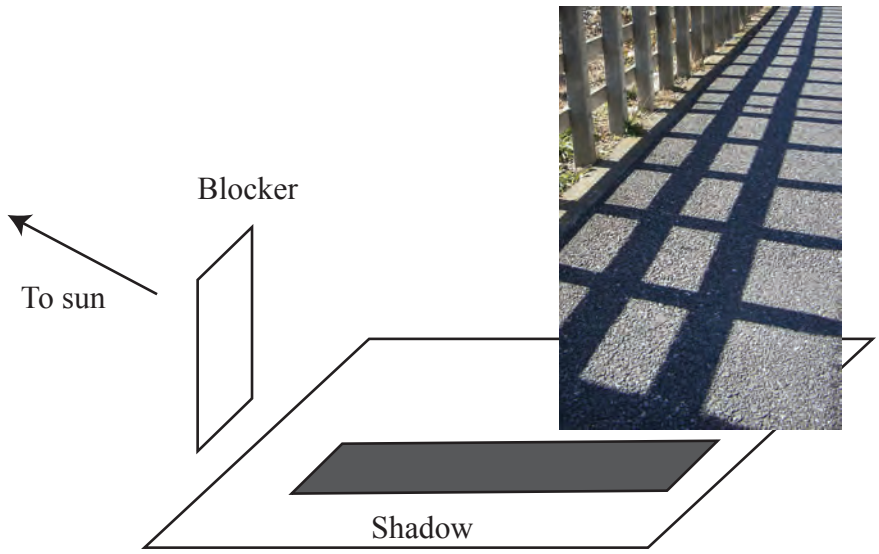

FIGURE 27.3: *The simplest shadows occur when a surface cannot see a source. The image is a picture by Ishikawa Ken, published on Flickr with a creative commons license (`https://creativecommons.org/licenses/by-sa/2.0/`). The drawing shows a blocker stopping the ground from seeing the sun. In the drawing, the sun is at a low angle, meaning the shadow is stretched. The shadowed sections of the ground in the image are darker than the unshadowed regions, but are not deep black, because the ground can see the sky. The sky is not as bright as the sun, but contributes a significant amount of light.*

27.3.1 Outdoor Shadows

In the local shading model, if the surface cannot see a source, then it is in *shadow*. The object that prevents the surface from seeing the source is sometimes called a *blocker*. Outdoors – where light arrives at a point only from the sun – the local shading model predicts that shadows are deep black and that that there are sharp boundaries at shadows. In fact, shadows are seldom very dark outdoors, because the shadowed surface usually receives light from the sky as well as the sun (Figure 27.3). Adding an ambient term ensures that shadows are not too dark, and is often an acceptable model for outrdoor images.

The sun – like the moon, which is safer to look at than the sun – is a small bright circle in the sky, rather than a point. A point on a surface receives light from all directions in an incoming hemisphere, and so receives light from the sun in a small set of directions rather than from just one direction. This means that if the blocker is in the right position, it might block only part of the sun. In turn, an outdoor shadow can be blurry (Figure 27.4). The same blocker can block a larger or smaller range of directions depending on how close to the ground it is. For example, consider an airplane on the ground with the sun high in the sky. The

FIGURE 27.4: *Some outdoor shadows have blurred outlines, because the sun occupies a small range of directions in the sky – when you look up, it looks like a small circle not a point. On the* **left**, *a point on a diffuse surface receives light from the sun (***yellow arrows***) and from the sky (***blue arrows***) and reflects this light evenly spread around all outgoing directions. The image is a picture by Pamela V White, published on Flickr with a creative commons license (*`https://creativecommons.org/licenses/by-sa/2.0/`*). On the* **center right**, *a blocker stopping the ground from seeing the sun. The sun is quite high in the sky, and the blocker is high off the ground, meaning it blocks a small range of directions. At* **1**, *the surface can see the whole of the sun. At* **2** *and* **4**, *the surface can see only part of the sun because the blocker covers part of it. At* **3**, *the blocker covers all of the sun.*

shadow below the airplane is dark, because directly below the airplane the ground can see only a very small part of the sky and cannot see the sun at all. But if the airplane is high in the sky, the shadow it casts on the ground is very difficult for an observer on the ground to see, because the airplane blocks only a small fraction of the sun and so the shadow is very large and not very dark. A passenger who looks in the right direction from the airplane can often see the shadow on the ground, however.

27.3.2 Area Sources and Indoor Shadows

An *area source* is an area that radiates light. Outdoors, the sun is an area source, though there is little real reason to use this model apart from explaining blurry shadows. The sky is a very good example of an area source. Furthermore, area sources are common indoors. An uncommon, but easy, example is a light fitting in a ceiling with a diffuser.

A simple model explains the qualitative behavior of an area source. Break the source up into a large number of small elements (say, rectangles). Any receiver – surface lit by the source – is lit by each element, and adds up the effects of each element to get the total effect of the source. This model predicts that area sources cast shadows that tend not to be dark, and that have smooth boundaries. At some points, the receiver will see all of the elements of the area source. At other points, the observer will only some of the elements on the area source (these points are very occasionally referred to as being in a *penumbra*). It is possible that, at some other points, the observer sees none of the elements on the area source (and so be in what is sometimes called *umbra*). This means one sees shadows that are rather

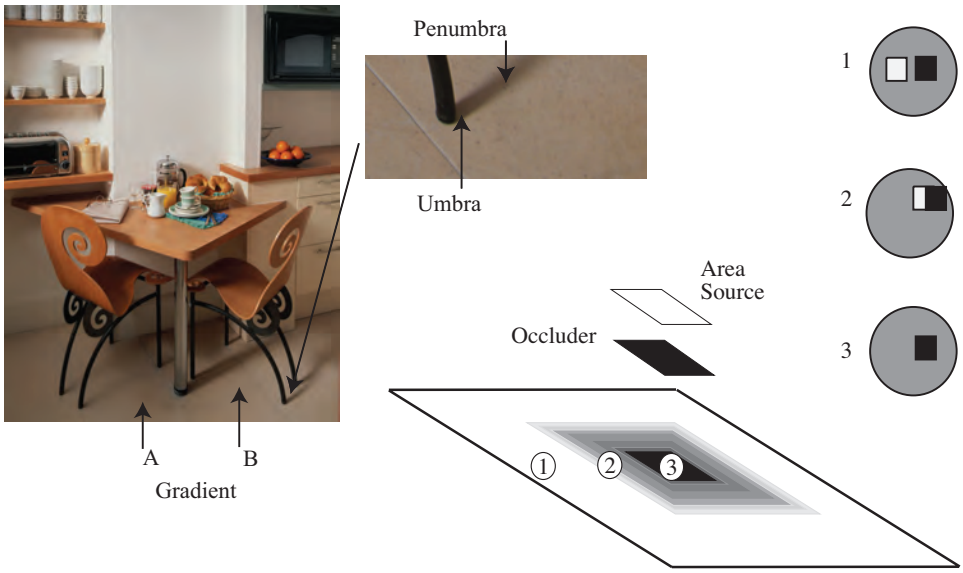

FIGURE 27.5: *Area sources generate complex shadows with smooth boundaries, because from the point of view of a surface patch, the source disappears slowly behind the occluder.* **Left:** *a photograph, showing characteristic area source shadow effects. Notice that A is much darker than B; there must 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.

fuzzy diffuse blobs, or sometimes fuzzy blobs with a dark core (Figure 27.5).

Much more common examples of area sources indoors are walls, ceilings and floors. These are not luminaires (they do not create light) but they do radiate light. A receiver cannot distinguish between light that comes directly from a luminaire, and light that was reflected from (say) a wall. A small blocker may not be able to block enough of a big area source to create a shadow that is dark enough to be visible. Rooms tend to have large white walls and ceilings, and rather smaller objects, so there are often few visible shadows (Figure 27.6).

27.3.3 Area Sources and Interreflections

The local shading model is not a physically accurate model of light transport, because a receiver cannot tell whether incoming light came directly from a source or was reflected from a surface. A better model is that light leaves a luminaire and may then be reflected from surface to camera; from surface to surface to camera;

FIGURE 27.6: *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 27.5).* Jake Fitzjones © Dorling Kindersley, used with permission.

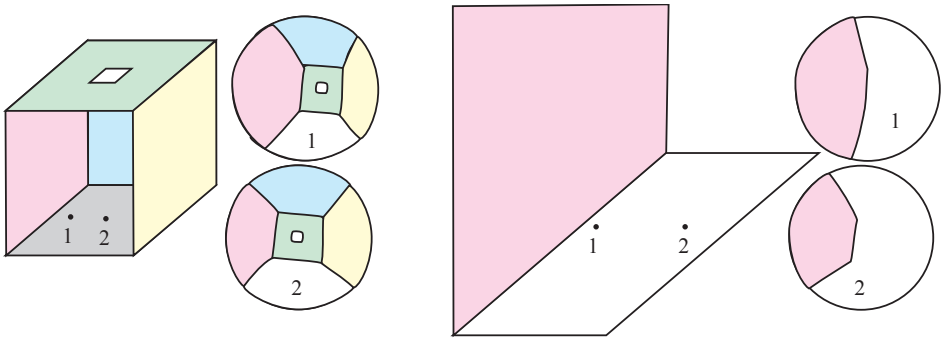

FIGURE 27.7: *On the* **left**, *a highly simplified room. Colors identify the walls (the missing wall is white), and there is a small light fitting in the ceiling (which is also white). On the* **center left**, *drawings of the hemisphere of incoming directions at points 1 and 2, flattened out for easy drawing. Notice how, as the receiver moves from 1 to 2, the hemisphere changes – it can see less of the pink wall and more of the green wall – but not by much. This effect explains why indoor shading is often quite uniform. On the* **center right**, *a version showing what happens when 1 is close to the edge. The pink wall occupies almost half of the incoming hemisphere of directions (***right***), so the receiver must collect a lot of light from it. This means that interreflection effects can cause points close to edges to be a lot brighter than predicted by a local shading model.*

Missing Figure

FIGURE 27.8: *Different lightings of the same scene result in different pictures (MIT dataset) and this is an important nuisance.*

and so on. Some of this light is absorbed. All this can be formulated into a detailed and accurate physical model of how light is distributed on scenes. Such models are now very well understood [**?**] and are extremely useful in computer graphics. These models are very hard to use directly in computer vision, because every variable affects every other variable.

Effects explained by this more accurate model are sometimes known as *interreflections*. The effects are particularly pronounced indoors. Section 27.3.2 used an interreflection argument to account for the qualitative properties of shadows indoors. Indoor scenes tend to appear quite uniform in shading, and this is also easily explained by an interreflection argument. At most points, the receiver sees about the same surfaces, but at slightly different angles, so illumination indoors is quite uniform (Figure 27.7). A version of this model explains why edges in rooms are not dark, too. As the observer approaches the edge along one wall, the other wall looks bigger (Figure 27.7), and the observer will receive more light from it. Finally, an interreflection model suggests (correctly, as Adrian Mole discovered to his cost) that a room with walls painted black looks very different to a room with walls painted white. It is known that people can tell whether a picture shows a room with black walls and black objects (shown in a bright light) or a room with white walls and white objects (shown in low light) []

27.4 INFERENCE FROM SIMPLE SHADING MODELS

27.4.1 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. **?**).

Early algorithms for estimating lightness are mostly variants of an idea referred to as *Retinex*. They remain useful and quite competitive. These algorithms 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 assume that:

- albedoes are piecewise constant over space;
- and shading changes only slowly over space.

These assumptions are sometimes called *Mondrian world* assumptions. Versions of these assumptions are built into modern vision systems.

The albedo assumption means that a typical set of albedoes will look like a collage of papers of different colors. 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).

The illumination 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 the shadow boundaries are.

Now differentiate the log transform and use some method to separate small and large gradients. It is enough to test the magnitude against a threshold, but there are a wide range of variant tests. By our assumptions, large gradients will occur where the albedo changes and small gradients are due to shading.

For images, differentiating and thresholding is easy: at each point, test the gradient of the log image; if it is a shading gradient, pass it into a shading gradient map, and if it is an albedo gradient, pass it into an albedo gradient map. There is a mild technical complication because the resulting maps will not be actual gradient maps. A gradient map meets a constraint sometimes called *integrability*.

Follow the long tradition of writing

$$\frac{\partial f}{\partial x} \text{ as } f_x \text{ or as } p \text{ and } \frac{\partial f}{\partial y} \text{ as } f_y \text{ or as } q.$$

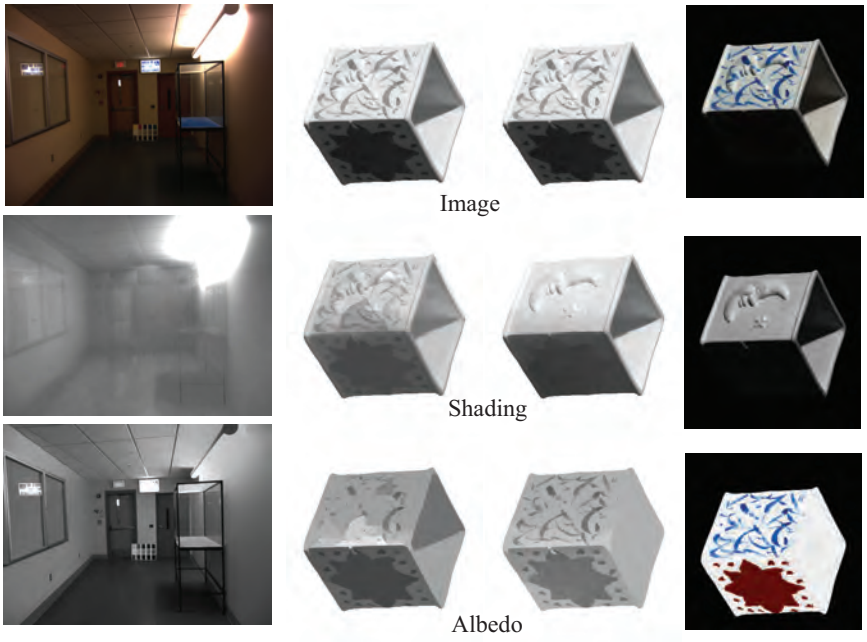

FIGURE 27.9: *Retinex remains a strong algorithm for recovering albedo from images. Here we show results from the version of Retinex described in the text applied to an image of a room (***left***) and an image from a collection of test images due to* ***?****. The* **center-left** *column shows results 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.

If (p, q) is a gradient map, then $p = f_x$ and $q = f_y$ for some function f, meaning that

$$p_y = q_x = f_{xy} = f_{yx} = \frac{\partial^2 f}{\partial y \partial x}$$

(in jargon, the mixed second partials must be equal). This will not be true of the shadow or albedo gradient maps.

Reconstructing log albedo from the albedo gradient maps can be phrased as a minimization problem: choose the log albedo map whose gradient is most like the albedo gradient map. This is a relatively simple problem because computing the gradient of an image is a linear operation. The x-component of the albedo gradient map is scanned into a vector $\hat{\mathbf{p}}$, and the y-component is scanned into a vector $\hat{\mathbf{q}}$. 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

$$\underset{\mathbf{I}}{\text{argmin}} \left\{ [\mathcal{M}_x\mathbf{I} - \hat{\mathbf{p}}]^T [\mathcal{M}_x\mathbf{I} - \hat{\mathbf{p}}] + [\mathcal{M}_y\mathbf{I} - \hat{\mathbf{q}}]^T [\mathcal{M}_y\mathbf{I} - \hat{\mathbf{q}}] \right\}$$

This is a quadratic minimization problem, and the answer can be found using linear algebra. 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.

You should think of this constant as a constant of integration. The constant of integration needs to be obtained from some other assumption. There are two obvious possibilities, explored in the exercises:

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

*** a paragraph on consequences ***

27.4.2 Photometric Stereo: Shape from Multiple Shaded Images

Photometric stereo is a method for recovering a representation of a surface from multiple images of the same thing under different lightings. 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.*

Substantial regions of the surface might be in shadow for one or the other light (see Figure 27.10). 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.

Fix a camera in some location, illuminate a surface with several different known distant point light sources and take then pictures. If the camera is linear and the camera gain is known, and if the local shading model is right, the surface normal at any point can be recovered from this information. Recall the local shading model gives that the intensity of the surface at a point $\mathbf{x}$ under the k'th light source will be

$$I_k(\mathbf{x}) = \rho(\mathbf{x}) S_k \cos\theta_k(\mathbf{x}),$$

where $\theta_k(\mathbf{x})$ is the angle between the normal at $\mathbf{x}$ and the vector to the light source and S_k is the intensity of the source. Represent the k'th light source with some vector $\mathbf{S}_k$ that points *toward* the source. If the magnitude of $\mathbf{S}_k$ is the same as S_k, this becomes

$$I_k(\mathbf{x}) = \rho(\mathbf{x}) \mathbf{S}_k^T \mathbf{N}(\mathbf{x})$$

where $\mathbf{N}(\mathbf{x})$ is the normal to the surface at $\mathbf{x}$. If the camera is linear, and has a known gain constant k, the normal *and* albedo can be recovered from enough images. The camera response at $\mathbf{x}$ to the i'th source is

$$P_k(\mathbf{x}) = k\rho(\mathbf{x}) \mathbf{S}_k^T \mathbf{N}(\mathbf{x}).$$

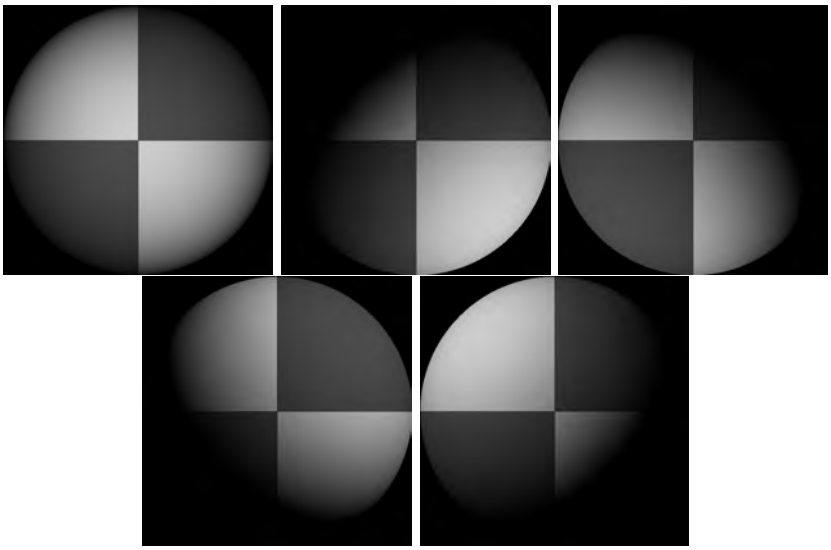

FIGURE 27.10: *Five synthetic images of a sphere, all obtained in an orthographic view from the same viewing position. These images are shaded using a local shading model and a distant point source. This is a convex object, so the only view where there is no visible shadow occurs when the source direction is parallel to the viewing direction. The variations in brightness occuring under different sources code the shape of the surface.*

Now write $\mathbf{V}(\vec{x}) = \rho(\mathbf{x})\mathbf{N}(\mathbf{x})$. There are N sources, and so N observations which give a linear system in the (unknown) $\mathbf{V}(\mathbf{x})$, given by

$$\begin{pmatrix} P_1(\mathbf{x}) \\ \dots \\ P_N(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} k\mathbf{S}_1^T \\ \dots \\ k\mathbf{S}_N^T \end{pmatrix} \mathbf{V}(\mathbf{x}).$$

Notice that k and $\mathbf{S}_k$ are known by assumption.

There is one linear system at each point on the surface, so at the i, j'th pixel (with location $\mathbf{x}_{ij}$, which we will usually compress to i, j) there is a linear system. Solve this system to obtain obtain $\mathbf{V}_{ij} = \mathbf{V}(\mathbf{x}_{ij})$. Since the normal is a unit vector, ρ_{ij} can be recovered as $\sqrt{\mathbf{V}_{ij}^T\mathbf{V}_{ij}}$. In turn, it is possible to recover normal and albedo when N is 3 or more, and normal if $N = 2$ and albedo is known.

More is possible. An estimate of the surface normal at each point can be turned into an estimate of surface height. For a point in the image at $\mathbf{x} = (x, y)$, the point on the surface is $(x, y, f(x, y))$. We will see later why this is a reasonable model, together with a number of alternative models. For the moment, notice that a point on the surface maps to a point in the image by dropping the third coordinate.

Again, follow the long tradition of writing

$$\frac{\partial f}{\partial x} \text{ as } f_x \text{ or as } p \text{ and } \frac{\partial f}{\partial y} \text{ as } f_y \text{ or as } q.$$

For a surface given by $(x, y, f(x, y))$, the normal is

$$\frac{1}{\sqrt{1 + f_x^2 + f_y^2}} \begin{pmatrix} -f_x \\ -f_y \\ 1 \end{pmatrix} = \frac{1}{\sqrt{1 + p^2 + q^2}} \begin{pmatrix} -p \\ -q \\ 1 \end{pmatrix}$$

(look at Section 21.3 if you're uncertain about this). At every pixel location i, j in the image, we have an estimate of the normal

$$\begin{pmatrix} n_{1;ij} \\ n_{2;ij} \\ n_{3;ij} \end{pmatrix}$$

so that at i, j, estimates are

$$\hat{p}_{ij} = -\frac{n_{1;ij}}{n_{3;ij}} \text{ and } \hat{p}_{ij} = -\frac{n_{2;ij}}{n_{3;ij}}.$$

The surface cannot be obtained by simply integrating these estimates, because there is an issue of integrability as in Section 21.3. Instead, rearrange the locations in the image to form vectors, and write $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ for the vectors containing estimates of $\mathbf{p}$ and $\mathbf{q}$ respectively. Recall the matrices $\mathcal{M}_x$, $\mathcal{M}_y$ from Section 21.3. Create a vector $\mathbf{f}$ which will contain the height at each pixel. We can now solve the following least squares problem to recover $\mathbf{f}$

$$\underset{\mathbf{f}}{\text{argmin}} \left\{ \left[\mathcal{M}_x\mathbf{f} - \hat{\mathbf{p}}\right]^T \left[\mathcal{M}_x\mathbf{f} - \hat{\mathbf{p}}\right] + \left[\mathcal{M}_y\mathbf{f} - \hat{\mathbf{q}}\right]^T \left[\mathcal{M}_y\mathbf{f} - \hat{\mathbf{q}}\right] \right\}$$

Just like the problem of Section 21.3 this is a quadratic minimization problem, and the answer can be found using linear algebra. Again, adding a constant vector to $\mathbf{l}$ cannot change the derivatives, so the problem does not have a unique solution.

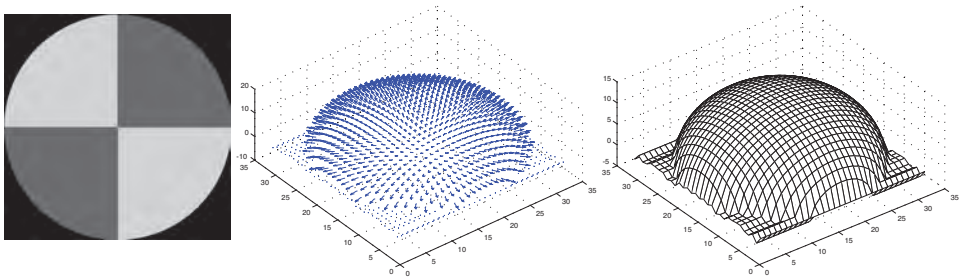

FIGURE 27.11: *The image on the* **left** *shows the magnitude of the vector field* $\mathbf{g}(x, y)$ *recovered from the input data of Figure 27.10 represented as an image—this is the reflectance of the surface. The* **center** *figure shows the normal field, and the* **right** *figure shows the height field.*

*** A paragraph on consequences ****

27.4.3 Radiometric Calibration

Recall that a camera response function turns the energy $E_{\mathbf{x}} = [\Delta t]\,[P_{\mathbf{X}->\mathbf{x}}]$ collected at a pixel $\mathbf{x}$ into the value $I_{camera}(\mathbf{x})$ that the camera reports at that pixel, so

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

Inferring the camera response function from data is referred to as *radiometric calibration*. The CRF can be estimated from multiple registered images, each obtained using a different exposure time. If you know the exposure times, estimation is simpler, but the example is less interesting (**exercises**). I will assume the exposure times are not known.

Assume you have N images of exactly the same scene under exactly the same lighting, but obtained with different exposure times. Turn each image into a vector, so there are fewer indices to bother with. Write p_i for the power arriving at the i'th location. This is the same for each image, but the energy is different because the exposure time is different. The energy arriving at the i'th location for the k'th exposure is

$$E_{ik} = p_i\,[\Delta t]_k\,.$$

If you write $\mathbf{p}$ for a vector of powers, $\mathbf{t}$ for a vector of exposure times, and arrange these unknown energies into a matrix $\mathcal{E}$, it will have the form

$$\mathcal{E} = \mathbf{p}\mathbf{t}^T$$

and so rank one (**exercises**).

You do not observe $\mathcal{E}$. Instead, you observe $\mathcal{I}$ – a matrix whose columns are the images. The key to estimating the CRF is that if you apply the inverse of the CRF to $\mathcal{I}$, you get a matrix that has rank one. This means that it is a good idea to model both the CRF and its inverse. You can do this by modelling the CRF as a piecewise linear function from $[0, 1]$ to $[0, 1]$ (**exercises**). It is important to use a model such that $C(x)$ is defined for $x > 1$ and $C'(x) > 0$ for $x > 1$ (**exercises**).

Write $F(\cdot;\theta)$ for the CRF (so $F(\mathcal{M};\theta)$ is the matrix of values obtained by applying the CRF to the elements of $\mathcal{M}$, and θ are the parameters of the CRF). Choose some loss to compare pixel predictions from the model with observations, and write $\mathcal{L}(\cdot)$ for that loss. Choose a smoothness penalty for the CRF **exercises**, and write $\mathcal{L}_s(\theta)$ for that penalty. You want to find θ, $\mathbf{e}$ and $\mathbf{t}$ to minimize

$$\mathcal{L}(\mathcal{I} - F(\mathbf{e}\mathbf{t}^T;\theta)) + \mathcal{L}_s(\theta).$$

The number of components for the piecewise linear function can be chosen by cross-validation. Use a large fraction of pixels (the training set) to estimate $\hat{\mathbf{e}}$, $\hat{\mathbf{t}}$ and $\hat{\theta}$ for some number of linear components c. Now use a test set of pixels to make a new matrix $\mathcal{I}^{\text{test}}$. Apply the inverse map to $\mathcal{I}^{\text{test}}$ to form $F(\mathcal{I}^{\text{test}};\phi(\hat{\theta}))$. Now estimate the p values for each test pixel by computing

$$\mathbf{p}^{(test)} = \underset{\mathbf{p}}{\text{argmin}}\ \| F(\mathcal{I}^{\text{test}};\phi(\hat{\theta})) - \mathbf{p}\hat{\mathbf{t}}^T \|_F$$

(which is straightforward **exercises**). Now evaluate

$$\mathcal{L}(\mathcal{I}^{\text{test}} - F(\mathbf{p}^{\text{test}}\hat{\mathbf{t}}^T;\hat{\theta}))$$

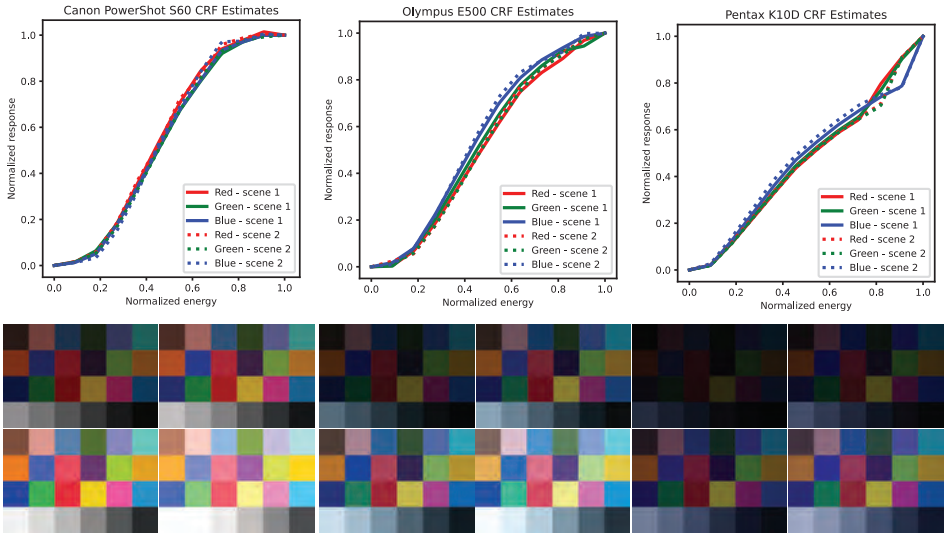

FIGURE 27.12: *The CRF of three different cameras obtained from multiple images obtained at different exposures using the algorithm described in the text. I used a piecewise linear model (exercises), and the number of linear components was chosen by cross-validation. To check the estimate, I estimated the CRF from two sets of images of two different scenes and compare them on the plots. Note that the estimates are consistent. The images used are shown on the* **bottom** *row. This figure was constructed from the dataset published at* `https://github.com/zyfccc/Representing-camera-response-function`*, described in the paper "Representing Camera Response Function by a Single Latent Variable and Fully Connected Neural Network", by Y. Zhao, S. Ferguson, H. Zhou and K. Rafferty in Signal, Image and Video Processing, 2022.*

and choose the c that gets the best value of this loss. Figure **??** shows results obtained using this procedure.

EXERCISES

27.1. Specular direction: Light arrives at a surface at a point $\mathbf{x}$ with normal $\mathbf{N}(\mathbf{x})$. The unit vector from $\mathbf{x}$ toward the source is $\mathbf{S}$, and the unit vector in the direction of specular reflection is $\mathbf{R}$, which has the properties $\mathbf{N}^T\mathbf{S} = \mathbf{N}^T\mathbf{R}$ and that $\mathbf{N}$, $\mathbf{S}$ and $\mathbf{R}$ are coplanar.

- **(a)** Show that these properties uniquely identify $\mathbf{R}$.
- **(b)** Explain why a user, viewing themselves in a mirror, is left-right reflected but not up-down reflected. How does the mirror know the difference?
- **(c)** A user standing at a distance d away from a mirror spans about the same height in the mirror as a user standing at distance $2d$ from the mirror. Explain.

27.2. Matrices that differentiate: $I(x, y)$ is a continuous function on the unit square. Represent this function by the values at a set of grid points. There are N_x points in the x direction and N_y points in the y direction. The values

are stored in a table $I_{i,j}$ where

$$I_{i,j} = I\left(\frac{(i-1)}{N_x - 1}, \frac{(j-1)}{N_y - 1}\right)$$

(note that indices start at 1 rather than 0!).

(a) Show that

$$(N_x - 1)\left(I_{(i+1),j} - I_{i,j}\right) \approx \frac{\partial I}{\partial x}$$

where the derivative is evaluated at $\left(\frac{(i-1)}{N_x-1}, \frac{(j-1)}{N_y-1}\right)$.

(b) Show that

$$(N_x - 1)\left(I_{N_x,j} - I_{N_x-1,j}\right) \approx \frac{\partial I}{\partial x}$$

where the derivative is evaluated at $\left(1, \frac{(j-1)}{N_y-1}\right)$.

(c) Show that

$$(N_y - 1)\left(I_{i,(j+1)} - I_{i,j}\right) \approx \frac{\partial I}{\partial x}$$

where the derivative is evaluated at $\left(\frac{(i-1)}{N_x-1}, \frac{(j-1)}{N_y-1}\right)$.

(d) Show that

$$(N_x - 1)\left(I_{i,N_y} - I_{i,N_y-1}\right) \approx \frac{\partial I}{\partial x}$$

where the derivative is evaluated at $\left(\frac{(i-1)}{N_x-1}, 1\right)$.

(e) Rearrange I_{ij} into a $N_x N_y \times 1$ vector $\mathbf{I}$ by stacking the columns, to obtain

$$\mathbf{I} = \begin{bmatrix} I_{1,1} \\ \dots \\ I_{N_x,1} \\ I_{1,2} \\ \dots \\ I_{N_x,N_y} \end{bmatrix}.$$

Now describe the missing rows of $\mathcal{M}_x$ (below) so that $\mathcal{M}_x\mathbf{I}$ yields an estimate of the x-derivative at each grid point, where

$$\mathcal{M}_x = (N_x - 1)\begin{bmatrix} -1 & 1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & \dots \\ & & \dots & & \\ 0 & 0 & \dots & 1 & -1 \end{bmatrix}.$$

(f) What is the form of $\mathcal{M}_y$ (below) so that $\mathcal{M}_y\mathbf{I}$ yields an estimate of the y-derivative at each grid point.

27.3. Other representations of the derivative: $I(x, y)$ is a continuous function on the unit square. Represent this function by the values at a set of grid points. There are N_x points in the x direction and N_y points in the y direction. The values are stored in a table $I_{i,j}$ where

$$I_{i,j} = I\left(\frac{(i-1)}{N_x - 1}, \frac{(j-1)}{N_y - 1}\right)$$

(note that indices start at 1 rather than 0!).

(a) Show that

$$(N_x - 1)\left(I_{(i+1),j} - I_{i-1,j}\right) \approx \frac{\partial I}{\partial x}$$

where the derivative is evaluated at $\left(\frac{(i-1)}{N_x-1}, \frac{(j-1)}{N_y-1}\right)$, for $1 < i < N_x$. This scheme for estimating a derivative is known as a symmetric first difference.

(b) If one were to use a symmetric first difference to estimate a derivative, how would one estimate the derivative at $\left(0, \frac{(j-1)}{N_y-1}\right)$ and at $\left(1, \frac{(j-1)}{N_y-1}\right)$.

(c) Describe the matrices $\mathcal{M}'_x$ and $\mathcal{M}'_y$ that would estimate derivatives using symmetric first differences.

(d) Assume N_x and N_y are odd numbers. Write

$$\mathbf{d} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ \dots \\ 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

and $\mathcal{I} = \mathbf{d}\mathbf{1}^T$. Show that $\mathcal{I}$ has zero first derivative in both x and y directions *if* you estimate the derivatives using a symmetric first difference.

27.4. Optimization and Retinex: We explore the optimization problem

$$\underset{\mathbf{I}}{\text{argmin}} \left\{ \left[\mathcal{M}_x\mathbf{I} - \hat{\mathbf{p}}\right]^T \left[\mathcal{M}_x\mathbf{I} - \hat{\mathbf{p}}\right] + \left[\mathcal{M}_y\mathbf{I} - \hat{\mathbf{q}}\right]^T \left[\mathcal{M}_y\mathbf{I} - \hat{\mathbf{q}}\right] \right\}$$

from Section 27.4.1.

(a) Write $\mathbf{c}$ for $c\mathbf{1}$ (i.e. a vector of all c's). Show that $\mathcal{M}_y\mathbf{c} = \mathcal{M}_x\mathbf{c} = \mathbf{0}$.

(b) Write

$$\mathcal{M} = \begin{bmatrix} \mathcal{M}_x \\ \mathcal{M}_y \end{bmatrix} \text{ and } \mathbf{d} = \begin{bmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{q}} \end{bmatrix}$$

and show that solving the minimization problem requires solving

$$\mathcal{M}^T\mathcal{M}\mathbf{I} = \mathcal{M}^T\mathbf{d}.$$

Does this problem have a unique solution?

(c) Now consider the constraint $\mathbf{1}^T\mathbf{I} = 1$, which is equivalent to an assumption that the average lightness is a known constant. Show that solving the minimization problem subject to this constraint requires solving

$$\begin{bmatrix} \mathcal{M}^T\mathcal{M} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathcal{M}^T\mathbf{d} \\ 1 \end{bmatrix}$$

Does this problem have a unique solution?

(d) Now write $\mathbf{e}_m$ for a vector with one component corresponding to each pixel. Every component is zero, except for the component corresponding to the brightest pixel in the image, which is one. This means the constraint $\mathbf{e}_m^T\mathbf{I} = 1$ requires that the brightest point in the image is reconstructed to have albedo 1. Show that solving the minimization problem subject to this constraint requires solving

$$\begin{bmatrix} \mathcal{M}^T\mathcal{M} & \mathbf{e}_m \\ \mathbf{e}_m^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathcal{M}^T\mathbf{d} \\ 1 \end{bmatrix}$$

Does this problem have a unique solution?

(e) Would it be a good idea to use symmetric first differences rather than first differences to solve any of these problems? What would happen?

27.5. Optimization and surface reconstruction for photometric stereo: We apply the results of the previous exercise to the optimization problem

$$\underset{\mathbf{f}}{\text{argmin}} \left\{ [\mathcal{M}_x\mathbf{f} - \hat{\mathbf{p}}]^T [\mathcal{M}_x\mathbf{f} - \hat{\mathbf{p}}] + [\mathcal{M}_y\mathbf{f} - \hat{\mathbf{q}}]^T [\mathcal{M}_y\mathbf{f} - \hat{\mathbf{q}}] \right\}$$

from Section 27.4.2.

(a) Why does this problem not have a unique solution?

(b) How can it be adjusted to have a unique solution?

27.6. Radiometric calibration

(a) Show that if f is a function from the real line to the real line, and the derivative of f exists and is greater than zero everywhere, the inverse of f exists.

(b) Write

$$h(x; k_0, k_1, k_2) = \begin{cases} 0 & \text{for } x < k_0 \\ \frac{(x-k_0)}{(k_1-k_0)} & \text{for } k_0 < x \leq k_1 \\ \frac{(k_2-x)}{(k_2-k_1)} & \text{for } k_1 < x \leq k_2 \\ 0 & \text{for } x > k_2 \end{cases}$$

Now choose a set of $N+2$ points on the x-axis, $-\inf < k_0 < k_1 < \ldots < k_{N+1}$ and a set of N coefficients a_i. Show that

$$G(x; k_0, \ldots, k_{N+1}, a_1, \ldots, a_N) = \sum_{i=1}^{N} a_i h(x; k_{i-1}, k_i, k_{i+1})$$

is a piecewise linear function with the following properties.

- $G(x; k_0, \ldots, k_{N+1}, a_1, \ldots, a_N) = 0$ for $x < k_0$.
- $G(x; k_0, \ldots, k_{N+1}, a_1, \ldots, a_N) = 0$ for $x > k_{N+1}$.
- $G(k_i; k_0, \ldots, k_{N+1}, a_1, \ldots, a_N) = a_i$ for $0 < i < N+1$ (i.e. the function interpolates (k_i, a_i)).

(c) Write $\text{ReLU}(x; k) = \max(x - k, 0)$. Show that

$$h(x; k_0, k_1, k_2) = \frac{1}{(k_1 - k_0)}\text{ReLU}(x; k_0) - 2\frac{1}{(k_2 - k_1)}\text{ReLU}(x; k_1) + \frac{1}{(k_2 - k_1)}\text{ReLU}(x; k_2).$$

(d) Write $\text{ReLU}(x; k) = \max(x - k, 0)$. Show that

$$h(x; k_0, k_1, k_2) = \frac{1}{(k_1 - k_0)}\text{ReLU}(x; k_0) - 2\frac{1}{(k_2 - k_1)}\text{ReLU}(x; k_1) + \frac{1}{(k_2 - k_1)}\text{ReLU}(x; k_2).$$

(e) Now choose a_0 and a_{N+1}. Show that

$$F(x; k_0, \ldots, k_{N+1}, a_1, \ldots, a_N) = \sum_{i=1}^{N} a_i h(x; k_{i-1}, k_i, k_{i+1}) - a_0 \text{ReLU}(-x; -k_1) + a_{N+1} \text{ReLU}(x; k_N)$$

is a piecewise linear function that interpolates $(k_1, a_1), \ldots, (k_N, a_N)$.

(f) Show that $a_0, a_{N+1} > 0$ and $0 < a_1 < \ldots < a_N$ is equivalent to the requirement that F be monotonic.

PROGRAMMING EXERCISES

27.7. Specular direction: Light arrives at a surface at a point $\mathbf{x}$ with normal $\mathbf{N}(\mathbf{x})$. The unit vector from $\mathbf{x}$ toward the source is $\mathbf{S}$, and the unit vector in the direction of specular reflection is $\mathbf{R}$.

(a) Using a plane mirror and some pins, verify that $\mathbf{N}^T\mathbf{S} = \mathbf{N}^T\mathbf{R}$ and that $\mathbf{N}$, $\mathbf{S}$ and $\mathbf{R}$ are coplanar. To do this place the mirror standing perpendicular to a surface you can put pins in. Make a line of pins, view them in the mirror, then insert more pins to mark out the specularly reflected line. This is an informative exercise, but not for everyone.

27.8. Matrices that differentiate:

27.9. CRF estimation