

Figure 3.8. Ambient illumination is a term added to the radiosity predictions of local shading models to model the effects of radiosity from distant, reflecting surfaces. In a world like the interior of a sphere or of a cube (the case on the left), where a patch sees roughly the same thing from each point, a constant ambient illumination term is often acceptable. In more complex worlds, some surface patches see much less of the surrounding world than others. For example, the patch at the base of the groove on the right sees relatively little of the outside world, which we model as an infinite polygon of constant exitance; its input hemisphere is shown below.

of a groove), this can be taken into account. To do so, we need a model of the world *from the perspective of the patch under consideration*. A natural strategy is to model the world as a large, distant polygon of constant radiosity, where the view of this polygon is occluded at some patches (see Figure 3.8). The result is that the ambient term is smaller for patches that see less of the world. This model is often more accurate than adding a constant ambient term. Unfortunately, it is much more difficult to extract information from this model, possibly as difficult as for a global shading model.

3.5 Application: Photometric Stereo

We will reconstruct a patch of surface from a series of pictures of the surface, taken under different illuminants. First, we need a camera model. For simplicity, we choose a camera situated so that the point (x, y, z) in space is imaged to the point

(x, y) in the camera (the method we describe will work for 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 a French military engineer who first used it (figure 3.9). 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.

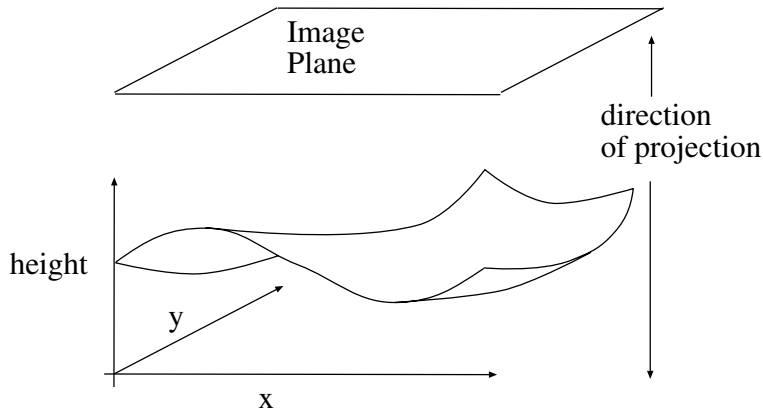


Figure 3.9. A Monge patch is a representation of a piece of surface as a height function. For the photometric stereo example, we assume that an orthographic camera — one that maps (x, y, z) in space to (x, y) in the camera — is viewing a Monge patch. This means that the shape of the surface can be represented 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 will recover 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 to 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 radiosity 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.

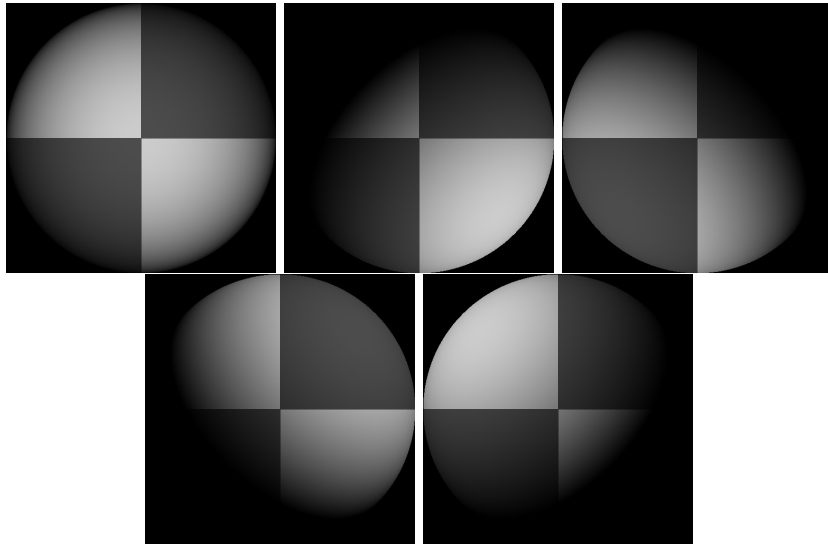


Figure 3.10. Five synthetic images of a sphere, all obtained in an orthographic view from the same viewing position. These images are shaded using a local shading model and a distant point source. This is a convex object, so the only view where there is no visible shadow occurs when the source direction is parallel to the viewing direction. The variations in brightness occurring 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.

3.5.1 Normal and Albedo from Many Views

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

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

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

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

Notice 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.

The difficulty with this approach is that substantial regions of the surface may be in shadow for one or the other light (see figure 3.10). There is a simple trick that deals with shadows. If there really is no ambient illumination, then we can form a matrix from the image vector and multiply both sides by this matrix; this will zero out any equations from points that are in shadow. We form

$$\mathcal{I}(x, y) = \begin{pmatrix} I_1(x, y) & \dots & 0 & 0 \\ 0 & I_2(x, y) & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & I_n(x, y) \end{pmatrix}$$

and

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

and \mathcal{I} has the effect of zeroing the contributions from shadowed regions, because the relevant elements of the matrix are zero at points that are in shadow. Again, there is one linear system per point in the image; at each point, we solve this linear system to recover the \mathbf{g} vector at that point. Figure 3.11 shows the vector field $\mathbf{g}(x, y)$ recovered from the images of figure 3.10.

Measuring Albedo

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 3.12 shows albedo recovered using this method for the images of figure 3.10.

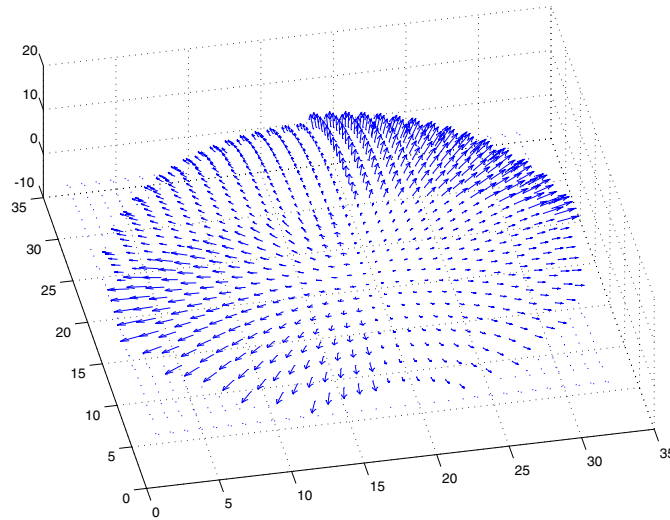


Figure 3.11. The vector field $\mathbf{g}(x, y)$ recovered from the input data of 3.10, mapped onto the recovered surface (this just makes the vectors more visible. The vector field is shown for every 16'th pixel in each direction, to reduce the complexity of the plot and make the structure clear.

Recovering Normals

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 3.13 shows normal values recovered for the images of figure 3.10.

3.5.2 Shape from Normals

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

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

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

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 we have that

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

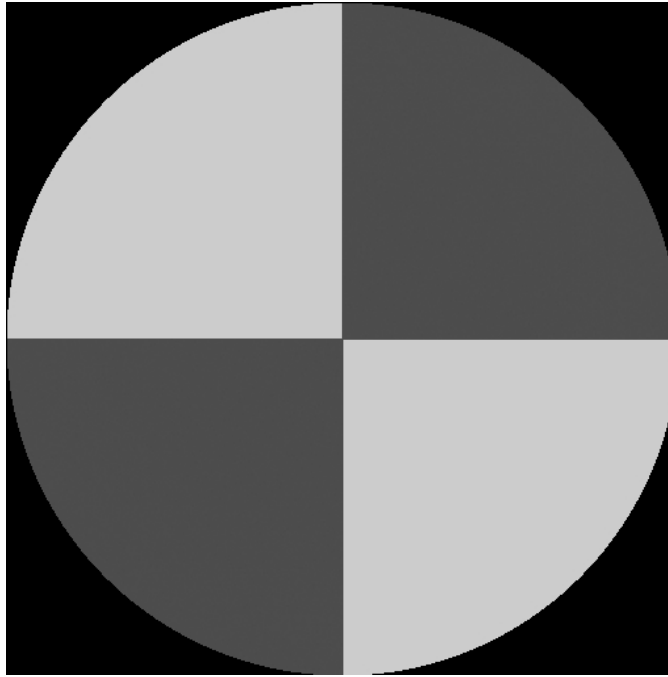


Figure 3.12. The magnitude of the vector field $\mathbf{g}(x, y)$ recovered from the input data of 3.10 represented as an image — this is the reflectance of the surface.

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

Shape by Integration

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

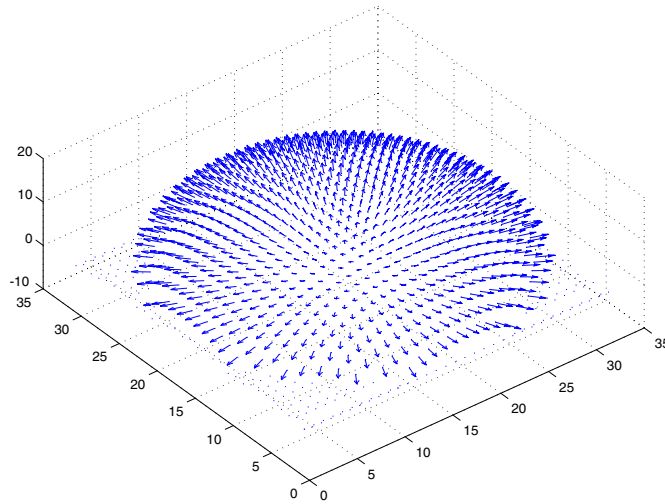


Figure 3.13. The normal field recovered from the surface of figure 3.10.

we can get the surface by summing these changes in height along some path. In particular, we have that

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

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. Exercise ?? asks you to show that the recovered surface does not depend on the choice of curve.

For example, we can reconstruct the surface at (u, v) by starting at $(0, 0)$, summing the y -derivative along the line $x = 0$ to the point $(0, v)$, and then summing the x -derivative along the line $y = v$ to the point (u, v)

$$f(u, v) = \int_0^v \frac{\partial f}{\partial y}(0, y) dy + \int_0^u \frac{\partial f}{\partial x}(x, v) dx + c$$

This is the integration path given in algorithm 1. Any other set of paths would work as well, though it is probably best to use many different paths and average, so as to spread around the error in the derivative estimates. Figure 3.14 shows the reconstruction obtained for the data of figure 3.10 and figure 3.11.

Another approach to recovering shape is to choose the function $f(x, y)$ whose partial derivatives most look like the measured partial derivatives. We explore this approach for a similar problem in section 4.5.2.

```

Obtain many images in a fixed view under different illuminants

Determine the matrix  $\mathcal{V}$  from source and camera information

Create arrays for albedo, normal (3 components),
  p (measured value of  $\frac{\partial f}{\partial x}$ ) and
  q (measured value of  $\frac{\partial f}{\partial y}$ )

For each point in the image array
  Stack image values into a vector  $i$ 
  Construct the diagonal matrix  $\mathcal{I}$ 
  Solve  $\mathcal{I}\mathcal{V}g = \mathcal{I}i$ 
    to obtain  $g$  for this point

  albedo at this point is  $|g|$ 
  normal at this point is  $\frac{g}{|g|}$ 
  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?

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 3.1: *Photometric Stereo*

3.6 Interreflections: Global Shading Models

As we indicated above, local shading models can be quite misleading. In the real world, each surface patch is illuminated not only by sources, but also by other

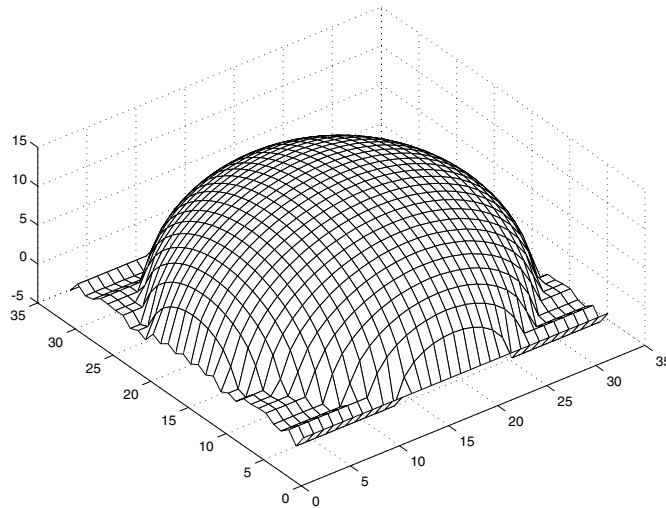


Figure 3.14. The height field obtained by integrating this normal field using the method described in the text.

surface patches. This leads to a variety of complex shading effects, which are still quite poorly understood. Unfortunately, these effects occur widely, and it is still not yet known how to simplify interreflection models without losing essential qualitative properties.

For example, Figure 3.15 shows views of the interior of two rooms. One room has black walls and contains black objects. The other has white walls, and contains white objects. Each is illuminated (approximately!) by a distant point source. Given that the intensity of the source is adjusted appropriately, the local shading model predicts that these pictures would be indistinguishable. In fact, the black room has much darker shadows and much more crisp boundaries at the creases of the polyhedra than the white room. This is because surfaces in the black room reflect less light onto other surfaces (they are darker) whereas in the white room, other surfaces are significant sources of radiation. The sections of the camera response to the radiosity (these are proportional to radiosity for diffuse surfaces) shown in the figure are hugely different qualitatively. In the black room, the radiosity is constant in patches as a local shading model would predict, whereas in the white room slow image gradients are quite common — these occur in concave corners, where object faces reflect light onto one another.

This effect also explains why a room illuminated by a point light source does not show the very sharp illumination gradients that a local shading model predicts (recall section 3.3.1). The walls and floor of the room reflect illumination back, and this tends to light up the corners, which would otherwise be dark.

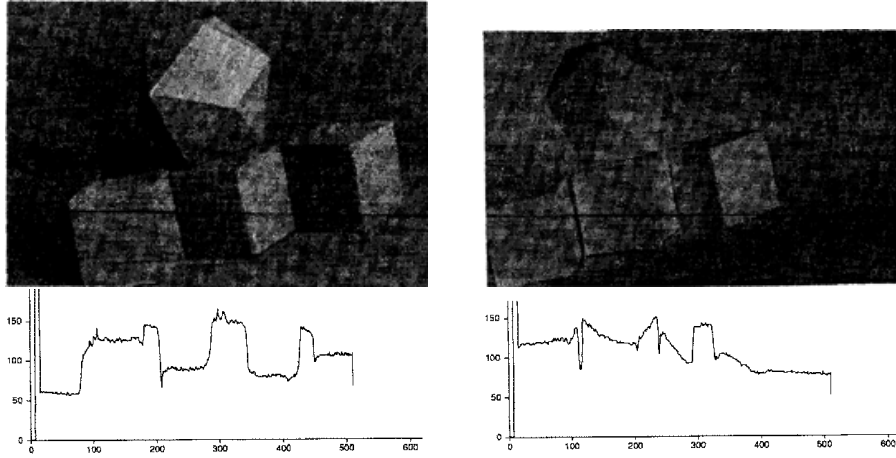


Figure 3.15. The column on the left shows data from a room with matte black walls and containing a collection of matte black polyhedral objects; that on the right shows data from a white room containing white objects. The images are qualitatively different, with darker shadows and crisper boundaries in the black room, and bright reflexes in the concave corners in the white room. The graphs show sections of the image intensity along the corresponding lines in the images. *data obtained from the paper “Mutual Illumination,” by D.A. Forsyth and A.P. Zisserman, page 473 in the fervent hope that permission will be granted*

3.6.1 An Interreflection Model

It is quite well understood how to predict the radiosity on a set of diffuse surface patches. The total radiosity of a patch will be its exitance — which will be zero for all but sources — *plus* all the radiosity due to all the other patches it can see:

$$B(\mathbf{u}) = E(\mathbf{u}) + B_{\text{incoming}}(\mathbf{u})$$

From the point of view of our patch, there is no distinction between energy leaving another patch due to exitance and that due to reflection. This means we can take the expression for an area source, and use it to obtain an expression for $B_{\text{incoming}}(\mathbf{u})$. In particular, from the perspective of our patch, every other patch in the world that it can see is an area source, with exitance $B(\mathbf{v})$. This means that we can rework equation 3.5 to get

$$B_{\text{incoming}}(\mathbf{u}) = \rho_d(\mathbf{u}) \int_{\text{world}} \text{visible}(\mathbf{u}, \mathbf{v}) B(\mathbf{v}) \frac{\cos \theta_u \cos \theta_v}{\pi d_{uv}^2} dA_{\mathbf{v}} \quad (3.6.1)$$

$$= \rho_d(\mathbf{u}) \int_{\text{world}} \text{visible}(\mathbf{u}, \mathbf{v}) K(\mathbf{u}, \mathbf{v}) B(\mathbf{v}) dA_{\mathbf{v}} \quad (3.6.2)$$