Chapter 04.00D

Physical Problem for Computer Engineering Simultaneous Linear Equations

Problem Statement

Human vision has the remarkable ability to infer 3D shapes from 2D images. When we look at 2D photographs or TV we do not see them as 2D shapes, rather as 3D entities with surfaces and volumes. Perception research has unraveled many of the cues that are used by us. The intriguing question is can we replicate some of these abilities on a computer? To this end, in this assignment we are going to look at one way to engineer a solution to the 3D shape from 2D images problem. Apart from the pure fun of solving a new problem, there are many practical applications of this technology such as in automated inspection of machine parts, inference of obstructing surfaces for robot navigation, and even in robot assisted surgery.

Image is a collection of gray level values at set of predetermined sites known as pixels, arranged in an array. These gray level values are also known as image intensities. The registered intensity at an image pixel is dependent on a number of factors such as the lighting conditions, surface orientation and surface material properties. The nature of lighting and its placement can drastically affect the appearance of a scene. Can we infer shape of the underlying surface given images as in the images below in Figure 1.
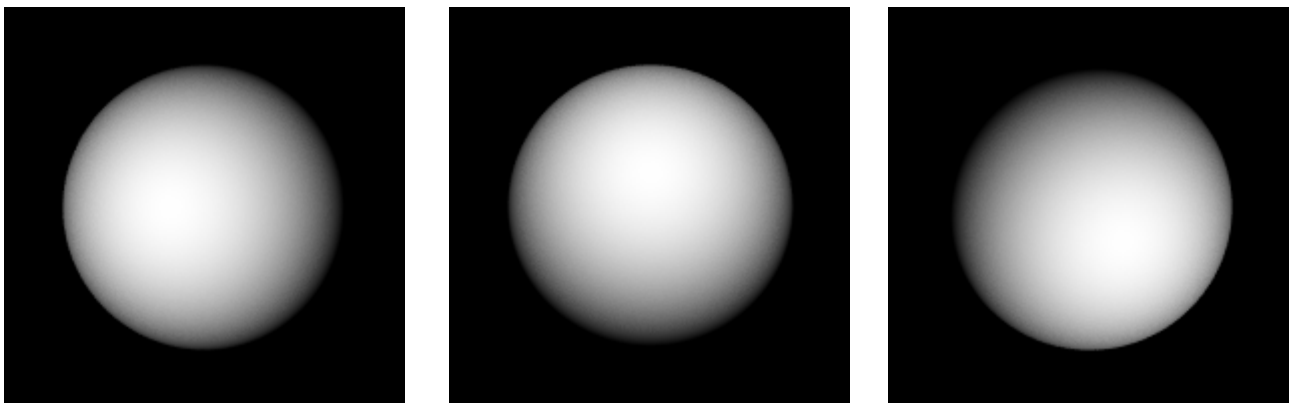

Figure 1 Images of a surface taken with three different light directions. Can you guess the shape of the underlying surface?

Physics of the Problem

To be able to reconstruct the shape of the underlying surface, we have to first understand the image formation process. The simplest image formation model assumes that the camera is far away from the scene so that we have assume that the image is a scaled version of the world. The simplest light model consists of a point light that is far away. This is not an unrealistic assumption. A good example of such a light source is the sun. We also assume that the surface is essentially matte that reflects lights uniformly in all directions, unlike specular (or mirror-like surfaces). These kinds of surfaces are called Lambertian surfaces; examples include walls, and carpet.
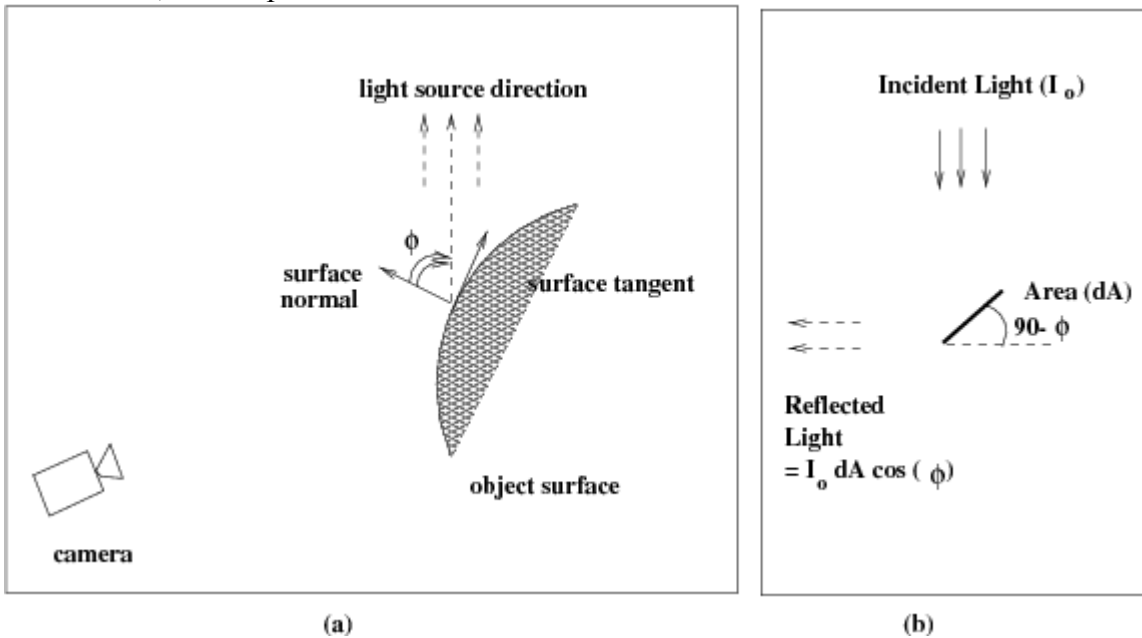

(a)   (b)

Figure 2 Relationship between the surface and the light source. (b) Amount of light reflected by an elemental area (dA) is proportional to the cosine of the angle between the light source direction and the surface normal

The image brightness of a Lambertian surface is dependent on the local surface orientation with respect to the light source. Since the point light source is far away we will assume that the incident light is locally uniform and comes from a single direction, i.e, the light rays are parallel to each other. This situation is illustrated in Figure 2. Let the incident light intensity be denoted by I_0. Also let the angle between the light source and local surface orientation be denoted by ϕ. Then the registered image intensity, $I(u,v)$, of that point is given by

$$I(u,v) = I_0 \rho \cos(\phi) = I_0 \rho (n_x s_x + n_y s_y + n_z s_z$$

where the surface normal, n, and the light source direction, s, are given by:

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, s = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}$$

and ρ is a number capturing the surface reflection property at the location (x, y), and It is referred to as the surface albedo. Black surfaces tend to have low albedo and white surfaces

tend to high albedo. Note that the registered intensity in the image does not depend on the camera location because a Lambertian surface reflects light equally in all directions. This would not be true for specular surfaces and the corresponding image formation equation would involve the viewing direction.

The variation in image intensity is essentially dependent on the local surface orientation. If the surface normal and the light source directions are aligned then we have the maximum intensity and the observed intensity of the lowest when the angle between the light source and the local surface orientation is 90^0. Thus, given the knowledge of light source and the local surface albedo, it should be possible to infer the local surface orientation from the local image intensity variations. This is what we explore next.

Solution

The mapping of surface orientation to image intensity is many to one. Thus, it is not possible to infer the surface orientation from just one intensity image in the absence of any other knowledge. We need multiple samples per point in the scene. How many samples do we need? The vector specifying the surface normal has three components, which implies that we need three. So, we engineer a setup to infer surface orientation from image intensities. Instead of just one image of a scene, let us take three images of the same scene, without moving either the camera or the scene, but with three different light sources, turned on one at a time. These three different light sources are placed at different locations in space. Let the three light source directions, relative to the camera, be specified by the vectors

$$s^1 = \begin{bmatrix} s_x^1 \\ s_y^1 \\ s_z^1 \end{bmatrix}, s^2 = \begin{bmatrix} s_x^2 \\ s_y^2 \\ s_z^2 \end{bmatrix}, s^3 = \begin{bmatrix} s_x^3 \\ s_y^3 \\ s_z^3 \end{bmatrix}$$

Corresponding pixels in the three images would have three different intensities I_1, I_2, and I_3 for three light source directions. Let the surface normal corresponding to the pixel under consideration be denoted by

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$$

Assuming Lambertian surfaces, the three intensities can be related to the surface normal and the light source directions

$$I_i = I_0\rho(n_x s_x^i + n_y s_y^i + n_z s_z^i), \forall i = 1,2,3$$

In these three equations, the known variables are the intensities, I_1, I_2, I_3, and the light source directions, s_1, s_2, s_3. The unknowns are the incident intensity, I_3, surface albedo, ρ, and the surface normal, n. These unknowns can be bundled into three unknown variables, $m_x = I_0\rho n_x$, $m_y = I_0\rho n_y$, and $m_z = I_0\rho n_z$. We will recover the surface normal by normalizing the recovered m vector, using the fact that the magnitude of the normal is one. The normalizing constant will give us the product $I_0\rho$. Thus, for each point in the image, we have three simultaneous equations in three unknowns.

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} s_x^1 & s_y^1 & s_z^1 \\ s_x^2 & s_y^2 & s_z^2 \\ s_x^3 & s_y^3 & s_z^3 \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

Worked Out Example

Consider the middle of the sphere in Figure 1. We know that the surface normal points towards the camera (i.e. towards the viewer). Assume a 3D coordinate system centered at the camera with x-direction along the horizontal direction, y-direction along the vertical direction, and z-direction is away from the camera into the scene. Then the actual surface normal of the middle of the sphere is given by [0, 0, -1] – the negative denotes that it point in the direction opposite the z-axis. Let us see how close our estimate is to the actual value.

The intensity of the middle of the sphere in the three views, $I_1 = 247$, $I_2 = 248$, and $I_3 = 239$, respectively. The light directions for the three images are along [5, 0, -20], [0, 5, -20], and [-5, -5, -20], respectively. Normalizing the three vectors we get the normal directions towards the lights and construct the 3 by 3 matrix

$$\begin{bmatrix} s_x^1 & s_y^1 & s_z^1 \\ s_x^2 & s_y^2 & s_z^2 \\ s_x^3 & s_y^3 & s_z^3 \end{bmatrix} = \begin{bmatrix} 0.2425 & 0 & -0.9701 \\ 0 & 0.2425 & -0.9701 \\ -0.2357 & -0.2357 & -0.9428 \end{bmatrix}$$

Solving the corresponding 3 simultaneous equations, we arrive the following solution for the m-vector:

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} 0.0976 \\ 4.2207 \\ -254.5774 \end{bmatrix}$$

Normalizing this vector we get the surface normal

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} 0.0004 \\ 0.0166 \\ -0.9999 \end{bmatrix}$$

The corresponding normalizing constant is 254.6124, which is the product of the intensity of the illumination and the surface (albedo) reflectance factor $(I_0\rho)$. Compare the estimate of the surface normal to the actual one. The difference is quantization effects – in images we can represent intensities as a finite sized integer – 8-bit integers in our case.

We can repeat the above computations for each point in the scene to arrive at estimates of the corresponding surface normals. Figure 3(a) is a visualization of the surface normals thus estimated as a vector field. In Figure 3(b), we see the product $I_0\rho$ visualized as image intensities. As expected, it is the same at all points on the sphere. In another problem module, how do we recover the underlying surface from these surface normals?
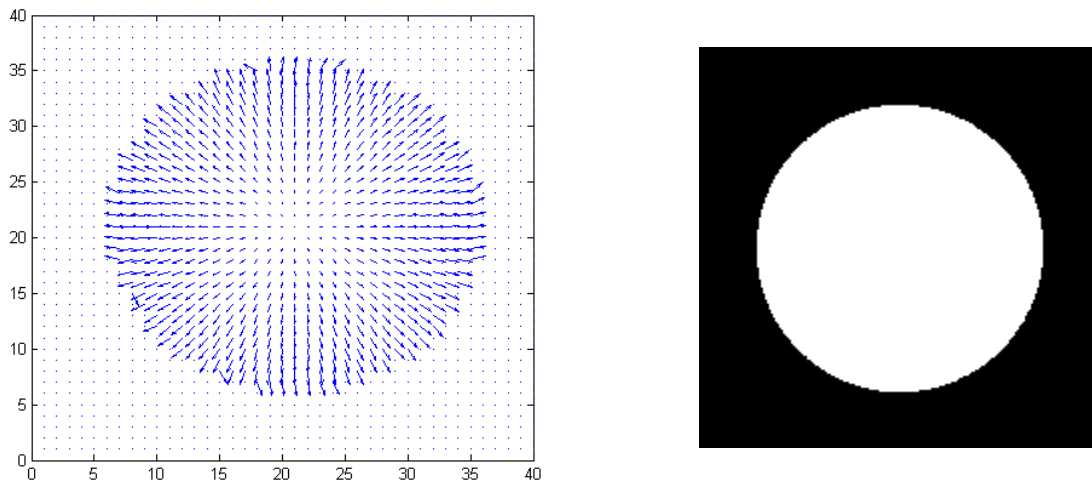
(a)   (b)

Figure 3 (a) Recovered surface normal at each point on the sphere. Just the first two components of the vectors are shown as a arrows. (b) Recovered product $I_0\rho$ for all points

QUESTIONS

1. What can you infer about the surface normal for the brightest point in the image? What about the darkest point in the scene?
2. What assumptions do you have to make to make the above inferences?

SIMULTANEOUS LINEAR EQUATIONS	
Topic	Physical Problem
Summary	To infer the surface shape from images
Major	Computer Engineering
Authors	Sudeep Sarkar
Last Revised	July 22, 2005
Web Site	http://numericalmethods.eng.usf.edu