CSE252a – Computer Vision – Assignment 2
Instructor: Ben Ochoa
November 11, 2014

Instructions:

- Submit your assignment electronically by email to iskwak+252a@cs.ucsd.edu with the subject line *CSE252 Assignment 2*. The email should have one file attached. Name this file: CSE_252_hw02_lastname_studentid.zip. The contents of the file should be:
  1. A pdf file with your writeup. This should have all code attached in the appendix. Name this file: CSE_252_hw02_lastname.pdf.
  2. All of your source code in a folder called code.
- No physical hand-in for this assignment.
- In general, code does not have to be efficient. Focus on clarity, correctness and function here, and we can worry about speed in another course.

1 Steradians [2 pts]

Let θ span 0 to $\pi/3$ radians and ϕ span 0 to π radians. How many steradians are in the section of the sphere covered by θ and ϕ?

2 Irradiance [3 pts]

a. Consider a cylinder with radius r and height h whose base is centered at $z = 0$ along the xy-plane. If the walls of the cylinder have constant radiance L and the top of the cylinder has constant radiance $4L$, what is the irradiance E at the point $(0, 0, 0)$ assuming that the surface at $(0, 0, 0)$ has a normal vector of $(0, 0, 1)$?

b. What is the irradiance if the radiance of the top is now $2Ld^2$ where d is the distance to the center of the top ($d \in [0, r]$)?

Express your answers in terms of the given quantities (e.g. convert any angles you might have introduced to the given quantities).

3 Lambertian surfaces [2 pts]

A Lambertian surface is one that appears equally bright from all viewing direction. In other words, the emitted radiance from a Lambertian surface is not a function of outgoing direction. Assume that we have an ideal Lambertian surface which also reflects all incident lights (absorbing none), the BRDF $\rho(\theta_{in}, \psi_{in}, \theta_{out}, \psi_{out})$ of such a surface will be a constant. What is that constant?

Hint: If we integrate the radiance of the surface over all directions, the total emitted energy is equal to the incoming energy.

4 Photometric Stereo and Specularity Removal [10pts]

The goal of this part of the assignment is to implement a couple of different algorithms that reconstruct a surface using the concept of photometric stereo. Additionally, you will implement the specular removal technique of Mallick et al., which enables photometric stereo reconstruction of certain non-Lambertian materials. You can assume a Lambertian reflectance function once speculartiesare removed, but the albedo is unknown and non-constant in the images. Your program will take in multiple images as input along with the light source direction (and color when necessary) for each image. You will also implement a second example-based photometric stereo algorithm which is based on simultaneously imaging two objects of the same material, one of which has known structure.

Part 1

Implement the photometric stereo technique described in section 2.2 of Forsyth and Ponce 2nd edition (or 5.4 in the 1st edition) and the lecture notes. Your program should have two parts:

a) Read in the images and corresponding light source directions, and estimate the surface normals and albedo map.

b) Reconstruct the depth map from the normals. You can first try the naive scanline-based shape by integration method described in the book.

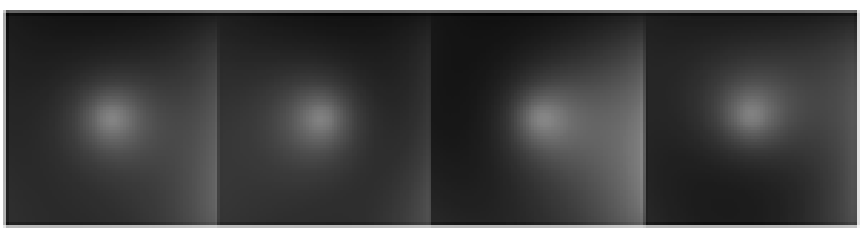

Figure 1: Synthetic Data

Try this out on the synthetic dataset (synthetic_data.mat) with three subsets of images:

a) im1, im2, im4

b) all four images (Most accurate)

What to include in your report:

a) Estimated albedo map

b) Estimated surface normals by either showing

- Needle map (you will need to subsample the image to get a needle map which can be displayed. You can use the matlab functions meshgrid() and quiver3() or for python, meshgrid() of numpy and quiver() of mplot3d, which is part of matplotlib). See Fig 2.
- Three images showing three components of surface normal

c) A wireframe of a depth map (you can use surf() in matlab or for python, plot_surface from mplot3d, which is part of matplotlib). See Fig 2.

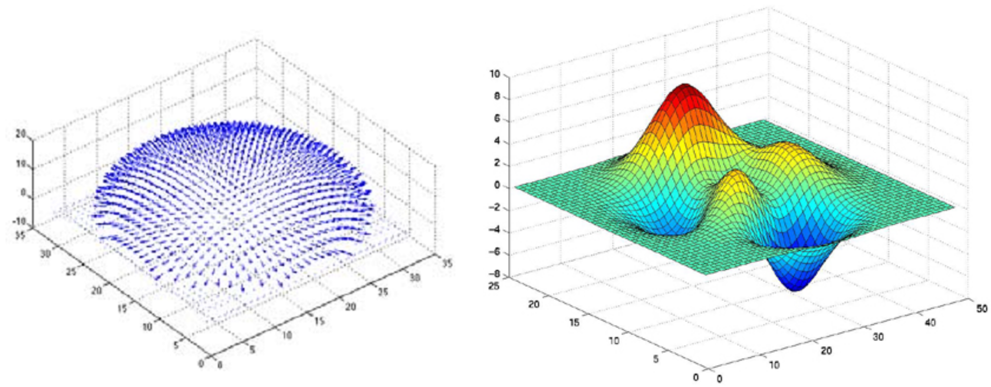

Figure 2: Example outputs for problem 4.

Part 2

Implement the specularity removal technique described in Beyond Lambert: Reconstructing Specular Surfaces Using Color (by Mallick, Zickler, Kriegman, and Belhumeur; CVPR 2005). Your program should input an RGB image and light source color and output the corresponding SUV image. Try this out first with the specular sphere images and then with the pear images. What to include in your report: For each specular sphere and pear images.

a) The original image (in RGB colorspace).

b) The recovered S channel of the image.

c) The recovered diffuse part of the image - Use $G = \sqrt{U^2 + V^2}$ to represent the diffuse part.

Part 3

Combine parts 1 and 2 by running your photometric stereo code on the diffuse components of the specular sphere and pear images. For comparison, run your photometric stereo code on the original images (converted to grayscale) as well. You should notice erroneous "bumps" in the resulting reconstructions - the result of violating the Lambertian assumption. What to include in your report: For each specular sphere and pear images.

a) The recovered diffuse images ($G = \sqrt{U^2 + V^2}$)

b) Estimated albedo map (original and diffuse images)

c) Estimated surface normals (original and diffuse images) by either showing

- Needle map (you will need to subsample the image to get a needle map which can be displayed. You can use the matlab functions meshgrid() and quiver3() or for python, meshgrid() of numpy and quiver() of mplot3d, which is part of matplotlib)
- Three images showing three components of surface normal

d) A wireframe (original and diffuse images) of a depth map (you can use surf() in matlab or for python, plot_surface from mplot3d, which is part of matplotlib).

Resources

Data

Synthetic Images, Specular Sphere Images, Pear Images : Available in a Matlab *.mat file (graciously provided by Satya Mallick) which contains

a) im1, im2, im3, im4 ... images.

b) l1, l2, l3 , l4 ... light source directions

c) c (when required) color of light source.

If you are using python, scipy.io.loadmat will load mat files for you.
All files are on the course webpage.

Good luck!