16-385 Computer Vision, Spring 2020

Homework Assignment 4
Physics-based vision

Due Date: Wed March 25, 2019 23:59

Instructions

1. **Integrity and collaboration:** Students are encouraged to work in groups but each student must submit their own work. If you work as a group, include the names of your collaborators in your write up. Code should **NOT** be shared or copied. Please **DO NOT** use external code unless permitted. Plagiarism is strongly prohibited and will lead to failure of this course.

2. **Start early!** Especially if you are not familiar with Python.

3. **Questions:** If you have any question, please look at piazza first. Other students may have encountered the same problem, and is solved already. If not, post your question on the discussion board. TAs will respond as soon as possible.

4. **Write-up:** Items to be included in the writeup are mentioned in each question, and summarized in the Writeup section. Please note that we **DO NOT** accept handwritten scans for your write-up in this assignment. Please type your answers to theory questions and discussions for experiments electronically.

5. **Handout:** The handout zip file contains 3 items. `assgn4.pdf` is the assignment handout. `data` contains seven photometric stereo images of a face, and some npz files. `python` contains the scripts that you will make use of in this homework.

6. **Submission:** Your submission for this assignment should be a zip file, `<AndrewID>.zip`, composed of your write-up, and your Python implementations (including helper functions). Your final upload should have the files arranged in this layout:

`<AndrewID>.zip`

- `<AndrewID>`
  - `<AndrewID>.pdf`
  - `python`
    * `helper.py` (provided)
    * `q1.py` (provided)
    * `q2.py` (provided)
    * `q3.py` (provided)

Overview

The purpose of this homework is to go over the topics of radiometry, reflectance, and photometric stereo. The main focus is on the latter, as a standard shape acquisition technique in computer vision.

1 Simple rendering (30 points)

Obtain the file `bunny.npy` from the data folder of this assignment and load it into Python. There is a single variable in this file; the variable `N` is an $h \times w \times 3$ array of surface normals. `N(i,j,1)`, `N(i,j,2)`, and `N(i,j,3)` are the x, y, and z components of the surface normal at the ij^{th} surface point, as observed by an orthographic camera with view direction $(0, 0, 1)$.

1. Compute and display the radiance emitted from each point assuming Lambertian reflectance and constant albedo (equal to one), with a distant point light source in direction $\boldsymbol{l} = (0,\ 0,\ 1)$. How is the final image related to the surface normals?

2. Compute and display the emitted radiance for three different light source directions which are rotated i) 45° up, ii) 45° right, and iii) 75° right from the frontal direction $\boldsymbol{l} = (0,\ 0,\ 1)$. Assume uniform unit albedo once again, for all cases.Can you spot errors in the field of surface normals? What are the illumination effects being ignored in this calculation of scene radiance?

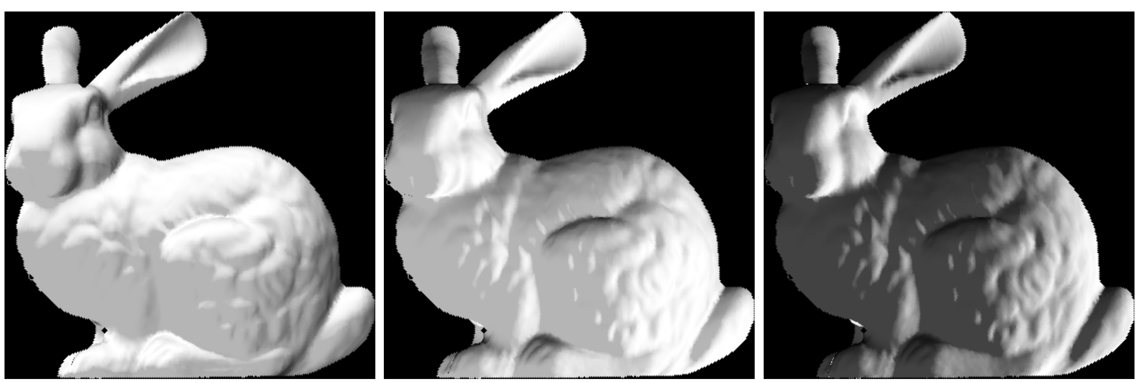

Figure 1: A few bunny renderings.

In your write-up: Please include figures for the three renderings of the bunny, as well as answers to the associated questions.

2 Lambertian photometric stereo (35 points)

The data folder of this assignment contains a set of seven photometric stereo images of a face, measured using a near-orthographic camera with fixed viewpoint, and under different illuminations. These images are available as files `./data/input_N.tif` in the homework ZIP archive, where $N = \{1, \ldots, 7\}$. These images are linear images, corresponding to RAW files that have been demosaicked and converted to the *linear* sRGB color space. Once we convert these images into grayscale, we can stack them into a matrix $\boldsymbol{I}$ of size $7 \times P$, where P is the number of pixels of each grayscale image channel.

Additionally, each of our seven input images is captured under some directional light, described by a 3×1 vector $\boldsymbol{l}_i$. These vectors are available in the file file `sources.npy`. We can stack the seven light vectors into a 3×7 matrix $\boldsymbol{L}$, which we call the *light matrix*.

Our goal is to recover a 3×1 normal vector $\boldsymbol{n}$ and a scalar albedo a at each pixel of the camera. As we did in class, it will be convenient to consider at each pixel the *pseudo-normal* $\boldsymbol{b} = a \cdot \boldsymbol{n}$. We can stack the pseudo-normals for all pixels into a $3 \times P$ matrix $\boldsymbol{B}$, which we call the *pseudo-normal matrix*.

Photometric stereo relies on the "n-dot-l" shading model we discussed in class, which is valid under directional light and Lambertian reflectance. Under this model, we can relate the matrices $\boldsymbol{I}$, $\boldsymbol{L}$ and $\boldsymbol{B}$ through a simple matrix product,

$$\boldsymbol{I} = \boldsymbol{L}^T \cdot \boldsymbol{B}. \tag{1}$$

Given that $\boldsymbol{I}$ and $\boldsymbol{L}$ are known, you can recover the unknown matrix $\boldsymbol{B}$ by simply solving the linear system of Equation (1) in the least-squares sense. Then, from $\boldsymbol{B}$, you can use normalization to recover estimates for the $1 \times P$ albedo matrix $\boldsymbol{A}$ and the $3 \times P$ normal matrix N. Finally, you can use the normal field to recover a surface, in the form of a *heightfield* $Z = f(x, y)$, through normal integration.

Implement a script that performs the following steps:

1. Load the seven images, and convert them to grayscale double format. Then, stack the seven grayscale images into the matrix $\boldsymbol{I}$.

2. Estimate the surface normal and grayscale albedo for each pixel. Display your recovered albedo and normal values.

3. Note the poor estimates of the albedo (and surface normal) in the area surrounding the nostrils. What is the source of this error? Describe one method for finding a better estimate of this information from these seven images.

4. Use the recovered surface and albedo information to predict what the person would look like (in grayscale) if illuminated from direction $\boldsymbol{l} = (0.58, -0.58, -0.58)$ and from direction $\boldsymbol{l} = (-0.58, -0.58, -0.58)$.

5. You can now use the normal field to compute a surface $Z = f(x, y)$. First, compute from the normals the derivatives $(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$. Then, integrate the derivatives using

the provided function `integrateFrankot` to recover the surface $Z(x, y)$. Display the recovered shape, both as a depth image, and as a 3D surface.

Figure 2 shows what kind of results you should expect to obtain from photometric stereo.

When visualizing the recovered normal fields, you may be tempted to visualize the three-channel "images" containing the per-pixel normals directly as an RGB image. However, you need to take into account that the coordinates of these normals will have values in the range $[-1, 1]$. Therefore, before displaying them as an RGB image, you should first transform them to the range $[0, 1]$. The standard way to do this is to map each normal $\boldsymbol{n}$ to $(\boldsymbol{n} + 1)/2$. This is how all the images in the write-up showing normals were produced, and you should do the same for your own results.

Additionally, when converting normals to heightfield derivatives for integration, you need to divide by each normal's z coordinate. This creates numerical problems when the z coordinate is (close to) zero; you can avoid these by adding a small ϵ to the z coordinate before dividing.

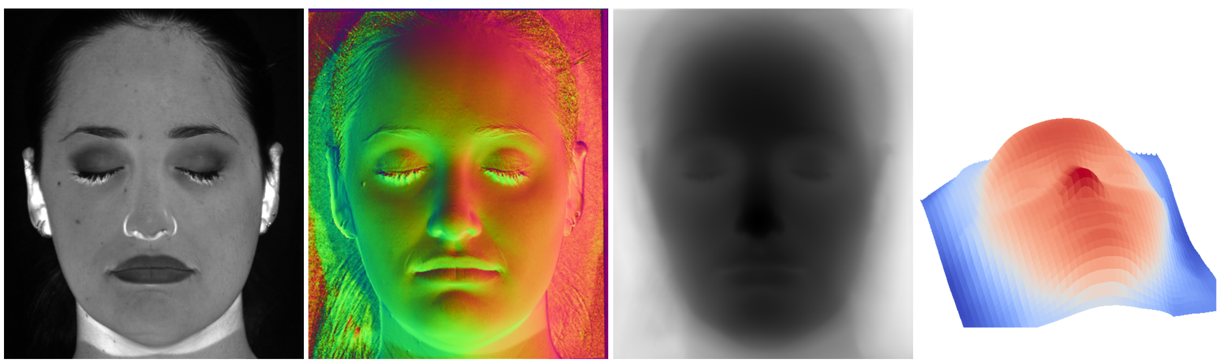

Figure 2: Photometric stereo results. From left to right: estimated albedo, normals, and depth (normalized to $[0, 1]$).

In your write-up: Include figures for the recovered normals and albedo, the re-rendered face images, and the final reconstructed surface, as well as answers to all associated questions.

3 Non-Lambertian photometric stereo (35 points)

The Blinn-Phong model is a very simple non-Lambertian reflectance model. (It is not, technically speaking, a BRDF, given that as written here it violates the convservation of energy.) Using this model, the radiance from a surface point with normal $\boldsymbol{n}$, illuminated from a unit-power directional source in direction $\boldsymbol{l}$, and viewed from direction $\boldsymbol{v}$ (all three vectors assumed to have unit norm), is given by

$$L(\boldsymbol{v}) = k_d(\boldsymbol{n} \cdot \boldsymbol{l}) + k_s(\boldsymbol{h} \cdot \boldsymbol{n})^\alpha$$

where $\boldsymbol{h}$ is the *halfway vector*, a unit-norm vector that bisects $\boldsymbol{l}$ and $\boldsymbol{v}$ (that is, $\boldsymbol{h} = (\boldsymbol{v} + \boldsymbol{l}) / \|\boldsymbol{v} + \boldsymbol{l}\|$,) and (k_d, k_s, α) are user-defined reflectance parameters that control the model.

You will use this reflectance model, together with the bunny surface from earlier, to explore the effects of non-Lambertian reflectance on photometric reconstruction results.

1. Synthesize images of the bunny for viewing direction $\boldsymbol{v} = (0,\ 0,\ 1)$; illumination directions i) 45° up, and ii) 45° right from the frontal direction; and the Blinn-Phong model with different sets of parameters. How does the appearance change as you vary the parameters k_d, k_s, and α? Explain the effect of each of the three parameters using examples, and presenting arguments based on the material we have covered in class.

2. Assuming again a viewing direction of $\boldsymbol{v} = (0,\ 0,\ 1)$, compute three images of the bunny surface using the Blinn-Phong model with parameters $(k_d,\ k_s,\ \alpha) = (.3,\ .5,\ 10)$, and three different light source directions

$$\begin{aligned} \boldsymbol{l}_1 &= (0.1,\ 0,\ 0.9) \\ \boldsymbol{l}_2 &= (0.1,\ 0.1,\ 0.9) \\ \boldsymbol{l}_3 &= (-0.1,\ -0.1,\ 0.9). \end{aligned}$$

   Using these images as input, reconstruct the Lambertian albedo and surface normals using Lambertian photometric stereo, and use `integrateFrankot` to recover the surface. How do the normals you recover compare to the original normals?

3. Repeat the same photometric stereo experiment, this time using $(k_d,\ k_s,\ \alpha) = (1,\ 0,\ 1)$. How do the normals you recover compare to the original normals? Explain why.

In your write-up: Include figures for the rendered bunny images, the recovered normals and albedo, and the final reconstructed surface, as well as answers to all associated questions. Make sure to also mention the parameters you use for the Blinn-Phong model.