Polina Berezina

Laboratory 1 report

10/04/2018

1. PHOTOMETRIC STEREO ALGORITHM

The problem of recovering shape of objects from images is overviewed in “Physics‐Based Vision: Principles and Practice, Shape Recovery” by Wolff, Shafer, Healey, and Peters (1992), and “Chapter 1 Photometric Stereo: An Overview” by Argyriou and Petrou (2009). Overall, given 3+ images taken from different angles, we can reconstruct the shape of a 3D object using the photometric stereo approach.

First, light direction of each image can be obtained from the specular sphere. Afterwards, on a Lambertian sphere at image location (x,y) the intensity of a pixel I(x,y) is given by



where: 𝜌(𝑥, 𝑦) is the albedo, 𝑠 is a vector of a light source (the same for the whole image), 𝑛 (𝑥, 𝑦) is a vector representing the direction of the surface normal at each point. Note that 𝑠 includes the direction and the intensity of the light source. That is why we need to calculate light intensity from the brightest point on a Lambertian sphere to calibrate re-rendered pixel intensity.

This equation can be solved with a least squares method, where light source vector and intensity values are known, and normals and albedo are unknown. Resulting normal vectors and albedo allows to recover shape and surface of the object, according to Lambertian reflectance model.

1. RESULTS

*pstereo.py* implements the above-mentioned algorithm. It can be run from the command line with an argument “input\_path”, where all of the pictures for the stereo are stored.

python .\lab1\pstereo.py .\lab1\Apple

The algorithm was tested on 3 sets of images. Fig. 1-3 show resulting images of surface normal, gray-scale albedo, color albedo, and re-rendered object under the default viewing direction (0, 0, 1).

|  |  |
| --- | --- |
| Normal map | Albedo gray-scale map |
| Albedo RGB map | Re-rendered image |
| Fig. 1 Apple image set | |
| Normal map | Albedo gray-scale map |
| Albedo RGB map | Re-rendered image |
| Fig. 2 Elephant image set | |
| Normal map | Albedo gray-scale map |
| Albedo RGB map | Re-rendered image |
| Fig. 3 Pear image set | |

1. DISCUSSION

Generally, the formulas behind standard PS technique make many assumptions that allow to create a linear equation system *I*. In practice, it leads to errors. For example, the formula  implies that all points of an object are located approximately within similar distance to the light source, and that illumination at those points is the same. Another cause for shape distortions is instrumental errors.

My method can be improved with more sophisticated reflectance models, such as Blinn-Phong. Current model does not recover surface for shiny objects well, like the apple on Fig. 1.

Robustness of my method is reduced by shadows. It can be solved by disregarding darkest and brightest pixels according to a chosen threshold, for example, 10%. Then, only images that have correct pixels can be used for computing. In fact, we could test how many images from different angles are actually required to solve equation system for normals and albedo. Perhaps, computing all 22 images does not lead to best results, and noise can be reduced with less images and discarding darkest/brightest pixels.

Finally, illumination configuration can greatly influence the result as well. Optimal illumination angles can minimize noise, and Drbohlav and Chantler in “On optimal light configurations in photometric stereo” (2005) found these angles for *n* light sources.

The computation time of my algorithm is less than 40 seconds for each set of images. I think it is appropriate. The Numpy library has certainly helped to speed up the process. However, if image resolution increases, the computation time might become too long. That is why parallel processing and elimination of Python inefficient loops and lists could be the next step to improve my code.