Photometric Stereo Using Computer Screen Lighting Configurations

Richard Kim with the support of Mostafa Shuva, Hana Baker, John Lipor, David Hong, Professor Laura Balzano

LSA MICHIGAN RESEARCH COMMUNITY UNIVERSITY OF MICHIGAN

Abstract

Traditional methods of photography have been about capturing location, color, and intensity of light. However, depth information about parts of the image is not obtainable from such methods. Photometric stereo is a method of depth computation from images taken under different lighting conditions. Along with image stitching algorithms, photometric stereo constructs a 3D depth image of an object. This is fundamentally possible due to the concept found in linear algebra called “least squares”, where by looking at the trend of similarity between images, the depth information can be extracted. Currently, the applications of photometric stereo are vast in areas such as computer vision, or even the movie industry, as it can be employed to capture a better 3D computer model of an object used later for special effects.

The project attempts to create a MATLAB program that would be able to retrieve a real-time 3D model of an object seen from a webcam, as the computer screen flashes different patterns of lights. For preliminary testing, we used a pre-existing dataset and algorithm provided in EECS 453/551 through MATLAB. As photometric stereo requires at least three distinct pictures, we created an algorithm that would flash four different light patterns. The computer then processes the images captured by the webcam in order to get a 3D depth model in real time. This project aims to push the application of photometric stereo in to an uncertain and real time environments. The effort was in part to capture the interest of incoming engineering students and show them the applicability of mathematics in interesting computer vision problems.

Objective

This project attempts to perform Photometric Stereo in less certain, real-time environments. The traditional method assumes that the exact position of the light sources relative to the object are known. However, the lighting parameters are much more generalized in the case of computer screen display, as the exact distances and direction from the computer light screen and the object are not known. Figure 1a [1] shows traditional photometric stereo, with lights at different positions assumed to be a point source, while Figure 1b shows the different lightings from the screen in our set up.

Figure 1a: Traditional lighting parameter [1]

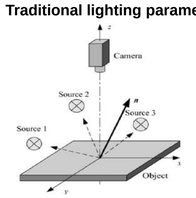
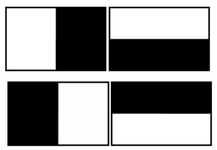


Figure 1b: Experimental lighting parameter



Methods

Photometric stereo aims to reconstruct a 3D surface of an object, given multiple pictures of the same object in the same angle under different lighting conditions. We used the data files from EECS 453 to become familiarized with photometric stereo. The data package contained 12 images of a cat under different lighting conditions along with the unit vectors describing the position of the lights relative to the object for each of the pictures, as shown in Figure 2.

Figure 2: Sample Images with unique lighting condition for Photometric Stereo through EECS 453

The 3D reconstruction is possible because the intensity of the light at every point of the object can be decomposed into an equation in which we can extract the unit-norm surface normal vector of the object, given in Equation 1 below. Highlighted in blue are the known values, and highlighted in red the unknown values.

$$I(x,y) = \alpha(x,y)(l^T n(x,y)) \quad (1)$$

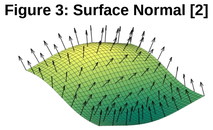
Matrix $\mathrm{L} \in R^3$, containing vectors l, represent the unit vectors of the light source from the object, $\alpha(x,y)$ the scaling constant called the surface albedo, and $n(x,y)$ the unit-norm surface normal vector of the object at (x,y). Since the inherent reflective characteristic of the object isn’t known, the equation can be further simplified to Equation 2, in matrix form.

$$I_{xy} = L^T \rho(x,y) \quad (2)$$

where $\rho(x,y) = \alpha(x,y)n(x,y)$

This assumption is valid since the albedo, or $\alpha(x,y)$, is the scaling constant, which won’t affect the direction and therefore the unit-norm surface normal vector of the object $n(x,y)$, which is found through Equation 3, and represented in Figure 3 below.

Figure 3: Surface Normal [2]

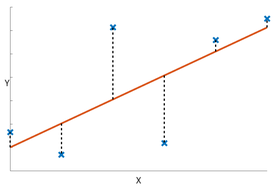


$$n(x,y) = \frac{\rho(x,y)}{\|\rho(x,y)\|_2} \quad (3)$$

Least Squares

In solving for $\rho(x,y)$, an understanding of least squares is needed, which a basic application of is shown in Figure 4 and Equation 4 below.

Figure 3: Representation of a Least Squares solution



$$\begin{aligned} y_1 &\approx mx_1 \\ y_2 &\approx mx_2 \\ y_3 &\approx mx_3 \\ y_4 &\approx mx_4 \\ y_5 &\approx mx_5 \end{aligned}$$

Given measurements $\mathrm{Y} = [y_1 \ \dots \ y_n]^T$ and their corresponding locations $\mathrm{X} = [x_1 \ \dots \ x_n]^T$, least square solves

$$m = \underset{m}{\arg min} \ \|Y - Xm\|_2^2 \quad (4)$$

This concept of least squares is the main mathematical model driving photometric stereo as it gets the most accurate $\rho(x,y)$ value representing the object. In solving for $\rho(x,y)$, Equation 5 presents itself in a similar fashion to the Equation 4 above.

Given Intensities $I_{xy} = [I_1(x,y) \ \dots \ I_d(x,y)]^T$ of each picture and their corresponding light direction vectors $\mathrm{L} = [l_1 \ \dots \ l_d]$, solves

$$\rho = \underset{\rho}{\arg min} \ \|I - L^T\rho\|_2^2 \quad (5)$$

Using this equation, we look at the same pixel location of each photographs, where by comparing the different photographs at the same pixel location, an oversolved system is presented. Each picture shows a different intensity value at the same location due to the differing light locations (i.e. different light direction vectors). However, the surface normal recovered are similar according to Equation 2. Least squares allows for the best representative surface normal.

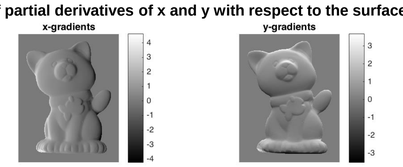
Surface Reconstruction

Unit-norm surface normal vectors that were found using least squares are needed for surface reconstruction as it helps determine the partial derivatives $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ of the surface function $f(x,y)$ characterizing the object, as shown in Equation 6.

$$\frac{\partial f}{\partial x} = -\frac{n_1(x,y)}{n_3(x,y)} \text{ and } \frac{\partial f}{\partial y} = -\frac{n_2(x,y)}{n_3(x,y)} \quad (6)$$

Having each of the x and y partial derivatives are important as it represents the magnitude of the change in depth of the object as it moves along x and y direction respectively. Graphing the partial derivatives results in Figure 5.

Figure 5: Plot of partial derivatives of x and y with respect to the surface function f(x,y)



We can also define the relationship between the object surface function $f(x,y)$ and the partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ by using a finite differences approximation of the derivative. Equation 7 displays this relationship, where the “+1” compares the difference between the values of pixels next to each other.

$$\frac{\partial f}{\partial x} \approx \frac{f(x+1,y)-f(x,y)}{(x+1)-x} \text{ and } \frac{\partial f}{\partial y} \approx \frac{f(x,y+1)-f(x,y)}{(y+1)-y} \quad (7)$$

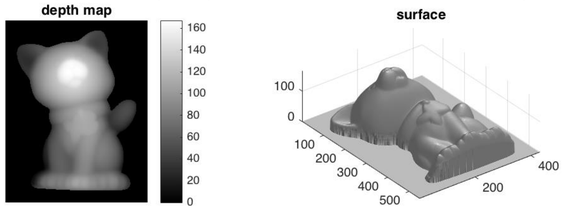
Equation 7 can be represented in matrix quantities by $\begin{bmatrix} D_n \otimes I_m \\ I_n \otimes D_m \end{bmatrix} f$. With the rest of the parameters known, the surface function f can be solved for by using the method of least squares.

$$f(x,y) = \arg\min \|b - Af\|_2^2 \quad (8)$$

$$\text{where } b = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \text{ and } \mathrm{A} = \begin{bmatrix} D \otimes I_m \\ I_n \otimes D_m \end{bmatrix}$$

After solving for $f(x,y)$, plotting the result returns the reconstructed 3D surface shown in Figure 6 below.

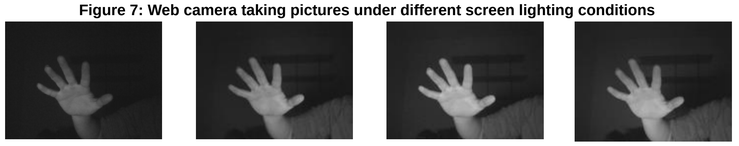
Figure 6: The depth map and 3D surface reconstructed from the object



Surface Reconstruction from Computer Screen Lighting

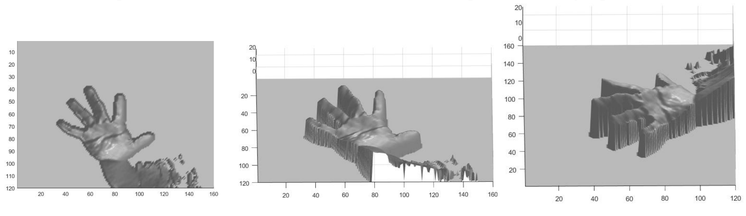
The aim here was to create a program in MATLAB that would extend photometric stereo to a more uncertain, real-time environment Since photometric stereo requires multiple lighting conditions, the computer screen would then display four different lighting conditions while the web camera connected to the computer would take the pictures, as demonstrated in Figure 7.

Figure 7: Web camera taking pictures under different screen lighting conditions



In the ideal photometric stereo set up, the L matrix containing unit-vectors from the camera to the object was provided in order to find the unit-norm surface normal vectors $n(x,y)$. However in our set up, it would not be clear what L matrix would be as the lighting from computer screen could not be assumed as point-source. Another obstacle was that the object would not always be fixed in the same location. To overcome the challenge, a paper from Georgia Institute of Technology [3] has found a relationship between matrix I_{xy} and $n(x,y)$ that is present by performing single value decomposition (SVD) on matrix I_{xy}. This relationship can be used to readily recover the surface normal vectors $n(x,y)$ without having to solve Equation 5. After being able to recover the unit-norm surface normal vectors, a surface can then be constructed in the same fashion as in traditional photometric stereo. The resulting 3D surface is shown in Figure 8 below.

Figure 8: 3D surface reconstruction of a hand using the MATLAB program



Limits and Unresolved Questions

Currently, the surface reconstruction from our MATLAB code contains much more noise compared to the traditional methods of Photometric Stereo. This may be due to lighting condition, where the object’s environment has minimal to no lighting. Also, having reflective or light objects could cause unwanted artifacts in the surface reconstruction.

Having more unique screen lighting configurations could also lessen the noise of the 3D reconstruction from the MATLAB program. Photometric Stereo requires a minimum of 3 pictures with unique lighting conditions. Using more unique patterns could lessen the noise as there are more data points to interpolate the surface-norm vectors. However, this would result in a longer delay for the reconstruction, as calculations would need to be done with more data.

Since the program was made to perform construction in real time, the program performs the reconstruction through the most recent four images captured. Therefore, if an object were to be moved while the program was running, there would be a ghosting effect between the reconstructions, as the position of the object in the four images would not align.

Conclusion

We were able to create a program that used methods of Photometric Stereo to get back the 3D surface with less known parameters. This was possible by using singular value decomposition, which relates the picture intensities matrix I_{xy} with the unit-norm surface normal matrix N. The direction matrix L, containing the unit-vectors of the position of light sources relative to the object, was not needed in our reconstruction. The limits of our program is that the surface contains a lot of noise, possibly due to a variety of factors such as the lighting environment, the minimal amount of pictures used, and the reconstruction of moving objects. Overall, the project was a success in pushing photometric stereo to real time, as the traditional computation on MATLAB takes over a minute. Also, the set-up of the scene is much less demanding, as obtaining the exact direction vectors of the light relative to the object in the traditional methods would be difficult without the right measuring tools.

In the future, we would like to explorer the exact tradeoffs of utilizing the SVD reconstruction method instead of the traditional reconstruction method. We would also like to find ways to make the program efficient. Currently the speed of the program is bottlenecked at the computation of least squares. Finding or establishing an approximation algorithm that could return values that are close to the values found through least squares may be an option in utilizing more unique lighting conditions and higher resolution photos for reconstruction.

References

[1] Photo courtesy of Wikepedia. “Least Squares”. 2016.

[2] Photo courtesy of Wikipedia. “Surface Normals”. 2016.

[3] G. Schindler, 2008. [Online]. Available: http://www.cc.gatech.edu/conferences/3DPVT08/Program/Papers/paper209.pdf [Apr. 3, 2016]