

ECSE 6650 Final Project: Uncalibrated Photometric Stereo

Hayleigh Sanders RIN# 661195735

1. Introduction and Problem Statement

Photometric stereo is a shape from illumination technique for estimating the shape of an object under different lighting conditions. This project will explore an uncalibrated approach to the photometric stereo problem, where only a lambertian surface and single illuminator are assumed.

This project will implement the uncalibrated photometric stereo algorithm outlined in Hayakawa's paper *Photometric stereo under a light-source with arbitrary motion* [1] to obtain the surface normals and light source directions for a sequence of images of an object under different lighting conditions up to a generalized bas-relief ambiguity. That is, the surface normals and light source matrix are represented in an arbitrary coordinate system and the unique orientation of the object is not known. A depth map of the image will also be computed using Forsyth and Ponce's [3] integration method.

2. Literature Review

The original photometric stereo model proposed by Woodham [2] found that for a Lambertian surface and uniform albedo, surface orientation can be computed by varying the direction of a light source onto an object in a number of successive images where the camera and object position do not change. The image intensity is equal to $I = Ln$, where I is a vector of pixel intensities for each pixel in the image, n is the unknown surface normal and L is a $3 \times p$ matrix of known light directions, where p is equal to the number of pixels in the image. This means that at least three known light source directions are needed to solve for the surface normals of an object, which is accomplished with the use of a calibration object such as a chrome sphere to estimate the light directions.

However, a newer method proposed by Hayakawa [1] is capable of estimating the surface normal and reflectance of an object without a *priori* knowledge of the light source direction or light source intensity, and therefore does not need a calibration object. The only assumption made is that the object has a Lambertian surface. The surface normals and light directions can be obtained up to an unknown factor by the factorization of image data.

For surface reconstruction, Forsyth and Ponce [3] propose a depth reconstruction method using surface normal integration.

3. Proposed Approach

3.1 Recover surface normals and light directions by factorization

In the following section, Hayakawa's method for image data factorization will be outlined with equations (1) to (14).

Given a series of intensity images of an object taken from the same position and illuminated from different directions at a constant distance, form the image data matrix I :

$$I = \begin{bmatrix} i_{11} & \cdots & i_{1f} \\ \vdots & \cdots & \vdots \\ i_{p1} & \cdots & i_{pf} \end{bmatrix} \quad (1)$$

Here, each image with a total number of pixels p forms a column in the image data matrix, with a total number of columns f equal to the number of frames in the image sequence.

The image data matrix can be expressed as $I = SL$, where

$$S = \begin{bmatrix} s_{1x} & s_{1y} & s_{1z} \\ \vdots & \vdots & \vdots \\ s_{px} & s_{py} & s_{pz} \end{bmatrix}, \quad L = \begin{bmatrix} l_{x1} & \cdots & l_{xf} \\ l_{y1} & \cdots & l_{yf} \\ l_{z1} & \cdots & l_{zf} \end{bmatrix} \quad (2)$$

Here, S is the surface matrix and L is the light source matrix. The rank of both S and L is assumed to be three, assuming three surface normals and three light source directions do not lie in a plane. The image data matrix will therefore also have a rank of three in the absence of image noise.

The image data matrix can be decomposed with SVD to the following:

$$I = U \Sigma V \quad (3)$$

However, due to the presence of noise in the image data matrix, the ideal image matrix can be estimated through rank-three approximation.

$$\hat{I} = U' \Sigma' V' \quad (4)$$

Where U' corresponds to the first three columns of U , Σ' corresponds to the first 3x3 submatrix of Σ and V' corresponds to the first three rows of V . \hat{I} can be represented as the product

$$\hat{I} = \hat{S} \hat{L} \quad (5)$$

Where the pseudo surface matrix \hat{S} and pseudo light source matrix \hat{L} are defined as

$$\hat{S} = U' \left(\pm [\Sigma']^{\frac{1}{2}} \right) \quad (6)$$

$$\hat{L} = \left(\pm [\Sigma']^{\frac{1}{2}} \right) V' \quad (7)$$

The sign of Σ chosen is the one that corresponds to the right-handed coordinate system, or one that produces a surface normal with a positive z-axis component. This decomposition is not unique because for a given arbitrary invertible 3x3 matrix A ,

$$\hat{S} A A^{-1} \hat{L} = \hat{S} (A A^{-1}) \hat{L} = \hat{S} \hat{L} = \hat{I} \quad (8)$$

Therefore, the ambiguity matrix A must be found such that,

$$S = \hat{S} A \quad (9)$$

$$L = A^{-1} \hat{L} \quad (10)$$

This is achieved with a constraint that makes an assumption about the surface or light characteristics. Hayakawa identifies two such constraints,

1. Identify six pixels with a constant or known surface reflectance
2. Identify at least fix image frames with a constant or known light source intensity

For this project, the second constraint was chosen because it is easier to implement in an experiment with real world data with a single illuminator. Per-pixel surface reflectance may be susceptible to image noise with real data.

Next, a system of equations can be formed from ($k=1,2,...p$) pseudo light source matrix vectors, where $p>6$.

$$\hat{l}_k^T A A^T \hat{l}_k^T = 1 \quad (11)$$

Substituting $B = A A^T$, where B is a 3x3 symmetric matrix, the equation is now:

$$\hat{l}_k^T B \hat{l}_k^T = 1 \quad (12)$$

Once B is solved via a system of linear equations, $B = AA^T$ can be found by taking the singular value decomposition of B, which will produce

$$B = W \Pi W^T \quad (13)$$

Where W is an orthonormal matrix and Π is a diagonal matrix. From these components of the decomposition, A can be found with

$$A = W[\Pi]^{\frac{1}{2}} \quad (14)$$

3.2 Solving for B with a system of equations

The following section will outline my approach for solving for B using equations (11) and (12).

From equation (11), AA^T is substituted with B. Therefore B is equivalent to,

$$B = AA^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \quad (15)$$

Performing matrix multiplication, B can be expressed in terms of A as,

$$B = \begin{bmatrix} a_{11}^2 + a_{21}^2 + a_{31}^2 & a_{11}a_{12} + a_{21}a_{22} + a_{31}a_{32} & a_{11}a_{13} + a_{21}a_{23} + a_{31}a_{33} \\ a_{11}a_{12} + a_{21}a_{22} + a_{31}a_{32} & a_{12}^2 + a_{22}^2 + a_{32}^2 & a_{12}a_{13} + a_{22}a_{23} + a_{32}a_{33} \\ a_{11}a_{13} + a_{21}a_{23} + a_{31}a_{33} & a_{12}a_{13} + a_{22}a_{23} + a_{32}a_{33} & a_{13}^2 + a_{23}^2 + a_{33}^2 \end{bmatrix} \quad (16)$$

Because B is a symmetric matrix, it can be expressed as,

$$B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_2 & b_4 & b_5 \\ b_3 & b_5 & b_6 \end{bmatrix} \quad (17)$$

Therefore each equation in the system of at least six equations will have the form,

$$\hat{l}_{x,x}^2 b_1 + 2 \hat{l}_{x,y} \hat{l}_{y,y} b_2 + 2 \hat{l}_{x,z} \hat{l}_{z,z} b_3 + \hat{l}_{y,y}^2 b_4 + 2 \hat{l}_{y,z} \hat{l}_{z,z} b_5 + \hat{l}_{z,z}^2 b_6 = 1 \quad (18)$$

So the system of equations to solve for each element of B using components of the pseudo light source matrix will be,

$$\begin{bmatrix} \hat{l}_{1,x}^2 & 2 \hat{l}_{1,x} \hat{l}_{1,y} & 2 \hat{l}_{1,x} \hat{l}_{1,z} & \hat{l}_{1,y}^2 & 2 \hat{l}_{1,y} \hat{l}_{1,z} & \hat{l}_{1,z}^2 \\ \hat{l}_{2,x}^2 & 2 \hat{l}_{2,x} \hat{l}_{2,y} & 2 \hat{l}_{2,x} \hat{l}_{2,z} & \hat{l}_{2,y}^2 & 2 \hat{l}_{2,y} \hat{l}_{2,z} & \hat{l}_{2,z}^2 \\ \hat{l}_{3,x}^2 & 2 \hat{l}_{3,x} \hat{l}_{3,y} & 2 \hat{l}_{3,x} \hat{l}_{3,z} & \hat{l}_{3,y}^2 & 2 \hat{l}_{3,y} \hat{l}_{3,z} & \hat{l}_{3,z}^2 \\ \hat{l}_{4,x}^2 & 2 \hat{l}_{4,x} \hat{l}_{4,y} & 2 \hat{l}_{4,x} \hat{l}_{4,z} & \hat{l}_{4,y}^2 & 2 \hat{l}_{4,y} \hat{l}_{4,z} & \hat{l}_{4,z}^2 \\ \hat{l}_{5,x}^2 & 2 \hat{l}_{5,x} \hat{l}_{5,y} & 2 \hat{l}_{5,x} \hat{l}_{5,z} & \hat{l}_{5,y}^2 & 2 \hat{l}_{5,y} \hat{l}_{5,z} & \hat{l}_{5,z}^2 \\ \hat{l}_{6,x}^2 & 2 \hat{l}_{6,x} \hat{l}_{6,y} & 2 \hat{l}_{6,x} \hat{l}_{6,z} & \hat{l}_{6,y}^2 & 2 \hat{l}_{6,y} \hat{l}_{6,z} & \hat{l}_{6,z}^2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (19)$$

The equations are set to one because the light intensity is constant. When B has been found, it can be decomposed with SVD via equation (13) to yield A. With the ambiguity matrix A identified, the surface normal matrix S and light source direction matrix L can be found with equations (9) and (10).

3.3 Dealing with shadows in the image data

Shadows in the image data cause inaccuracies because photometric stereo methods assume a surface point cannot be in the shadow of any light source. I first considered two methods, the first being the procedure outlined in Hayakawa's paper [1] and the second being the method described in Forsyth and Ponce's textbook [3], but found them too computationally expensive to implement with real data given my laptop specifications. Hayakawa's method involves iteratively extracting submatrices of the image data matrix containing illuminated data and decomposing them to obtain more reliable estimates of the pseudo vectors. Forsyth and Ponce advocate forming a diagonal matrix composed of values of each of the intensities of the image data matrix and using it to zero out the contribution of shadowed data, which I found would become too large.

As a result, I formulated my own simple method to reject shadowed data in the images. Note that the image data matrix is formed from columns of each pixel intensity for each image in a sequence. Therefore, each row of n intensity values correspond to the pixel intensity subjected to n different illumination positions. If a row was found to contain illuminated data (intensity > 0), any data with an intensity equal to zero (shadowed) was rejected.

3.4 Depth map recovery

To recover the depths at each (x,y) coordinate in the image, the shape from integration method detailed by Forsyth and Ponce [3] was utilized with equations (20) through (24).

The surface normal can be expressed as a function of its (x,y) position in the image:

$$N(x, y) = \frac{1}{\sqrt{1 + \frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}}} \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & 1 \end{bmatrix}^T \quad (20)$$

The surface can be represented as (x, y, f(x,y)). The depth map can be determined by f(x,y) from the surface normal values, but first it must pass a test of integrability. This means that the surface normal change in the x and y directions is small for each pixel, which corresponds to a smooth gradually changing surface that can be integrated. For each pixel, define the following partial derivatives, where $N=(n_x, n_y, n_z)$ is the surface normal at that pixel:

$$p = \frac{\partial f}{\partial x} = \frac{n_x}{n_z}, \quad q = \frac{\partial f}{\partial y} = \frac{n_y}{n_z} \quad (21)$$

To test for integrability, the following relationship can be evaluated at each pixel:

$$\left(\frac{\partial p}{\partial y} - \frac{\partial q}{\partial x} \right)^2 \quad (22)$$

Equation (21) must be small at each (x,y) point in the image. Next, the depth can be estimated up to a constant depth error by integrating the partial derivatives along a path, as shown in equation (23).

$$f(x, y) = \oint \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \cdot dl + c \quad (23)$$

Depth recovery for each point (u,v) is shown in equation (24). Beginning at (0,0), sum the y-derivative along the line x=0 to point (0,v), then sum the x derivative along the line y=v to point (u,v).

$$f(u, v) = \int_0^v \frac{\partial f}{\partial y}(0, y) dy + \int_0^u \frac{\partial f}{\partial x}(x, v) dx + c \quad (24)$$

Equation (24) is the basis of the algorithm for per-pixel depth recovery utilized in this project after the per-pixel surface normals have been recovered using the algorithm outlined in section 3.1.

4. Experiment and Results

4.1 Hardware and Physical Setup

Image data was obtained with a Raspberry Pi NoIR V2 camera board mounted to a Raspberry Pi 2 Model B V1.1 microcontroller. The NoIR camera is sensitive to both infrared (850nm) and visible light, so an 850nm optical bandpass filter was fixed over the camera sensor to cut out visible light noise from the image data. This is important because the constraint I selected to resolve the ambiguity matrix depends on the assumption that the illumination intensity is constant. A test object was placed 30cm from the camera sensor and illuminated with an 850nm infrared illuminator positioned at a 70cm radius away from the object center. Black carbon filter fabric was used as a background to absorb excess light, so the images only contain reflected light from the test object.

4.2 Software

All code was written in Python 3, and data was processed using the Numpy matrix library and its associated linear algebra packages and displayed using the Matplotlib library. To obtain image data, a test object was placed in a fixed position in the center of the camera's field of view and illuminated from an arbitrary angle. The image capture software would wait two seconds to initialize the camera, then capture a 600x500 pixel image and save it to a data folder on the Raspberry Pi. This process would be repeated for a number of iterations equal to the number of images needed for each test object. During the two seconds before each image capture, the illuminator was moved to a different position. 64 images were captured for each test object, and two test objects with lambertian surfaces (brick, tennis ball) were selected. The image number was selected due to Hayakawa's recommendation that 64 images should be analysed in the presence of 10% intensity error. While I made sure to keep the light source at a constant distance from the object, this amount of images should reduce the error from any variation in the distance.

This data folder was then moved to a T490s laptop for photometric stereo analysis due to the computationally intensive nature of the algorithms and large dataset. Each image in a sequence was converted to a grayscale numpy array and used to form the image data matrix, I . SVD was then performed on the image matrix as shown in equation (4) and using the resulting components, equations (6) and (7) were computed for the pseudo surface matrix and pseudo light source matrix by taking the largest 3x3 submatrices. A system of equations was formed using equation (12) and B was solved using the algorithm detailed in subsection 3.2. While at least six equations from six different pseudo light source vectors were needed for this system of equations, I performed least-squares analysis on all 64 pseudo light source vectors to obtain the

best estimate of the six B matrix values. Using B, A was found with equations (13) and (14). Then with the resulting A matrix, S and L were found with equations (9) and (10).

Using the results for S and L, the depth map was computed by finding the height for each (x,y) pixel using equation (24). An integrability check was first performed using equation (22) at each pixel before it was added to the depth map. If the gradient difference was too high, the height data at that point was rejected. For each test object, a surface normal quiver map was generated, RGB normal map (n_x =red, n_y =green, n_z =blue) was found for each pixel, a 2D depth map of intensities corresponding to the height at each (x,y) pixel was found, and a 3D depth map based on the z-axis heights of each point was generated.

The results were verified by comparing the dimensions of the physical test object with their reconstructed counterparts. The coordinate system of the reconstructions is scaled by an unknown factor, but the ratios of the selected lengths should be the same, so this metric will be used to judge how accurate the dimensions of the reconstructed objects are to their physical counterparts.

4.3 Brick

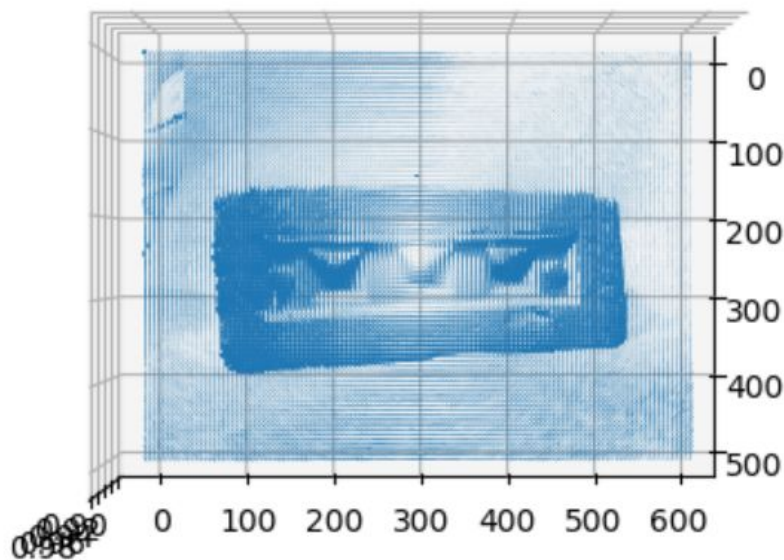


Figure 1: 3D quiver map of (n_x, n_y, n_z) normal vectors

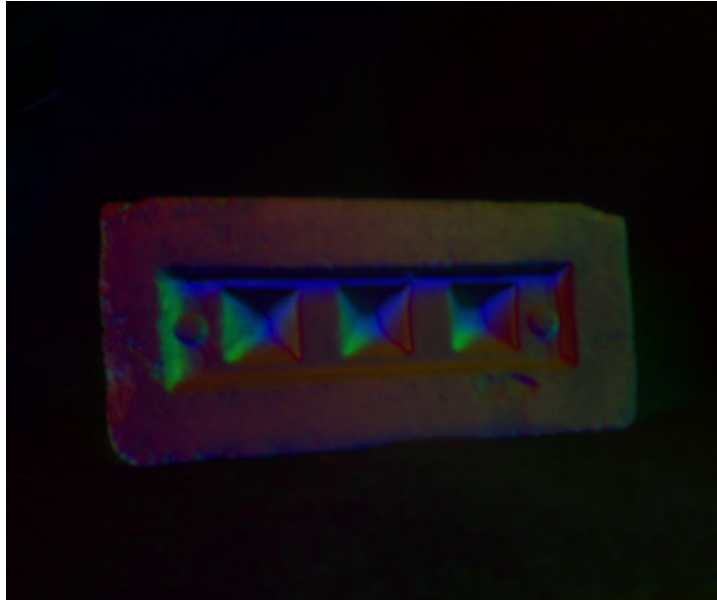


Figure 2: RGB representation of normals

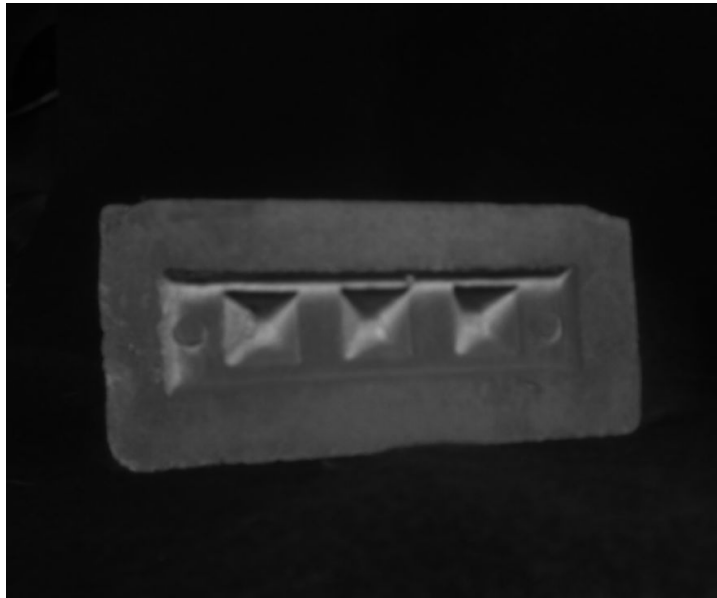


Figure 3: Albedo (magnitude of the surface normals)



Figure 4: 2D Depth map

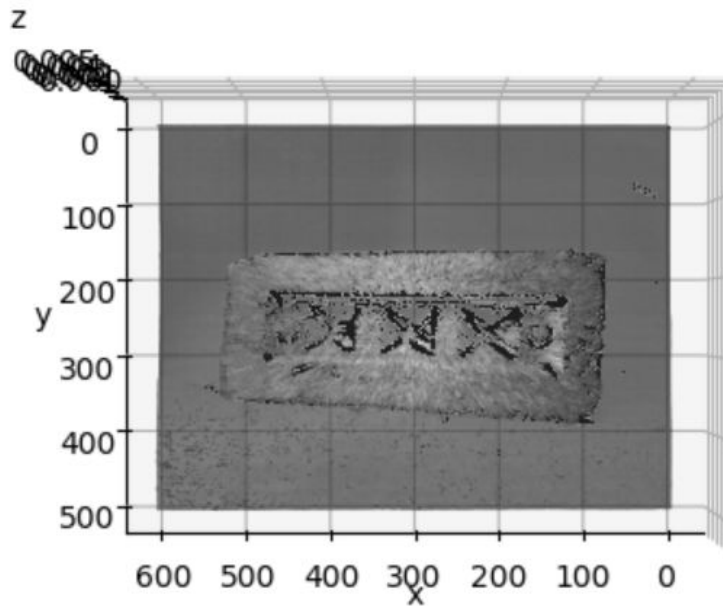


Figure 5: 3D surface map of depth. Black areas are where the height data was rejected due to the integrability constraint. These areas are where I would expect them to be (angles, edges, etc where the surface is not smooth)

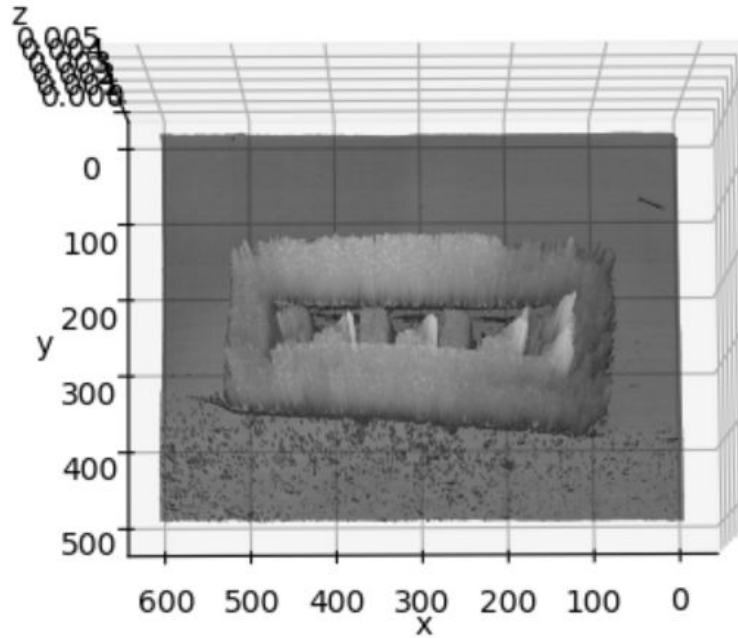


Figure 6: 3D surface map of depth, tilted to show height

The light source directions were found to be:

```
[[ 1.05792254e+00 3.07908449e-01 3.19539634e-01]
 [ 9.42482848e-01 6.19844407e-01 3.80916759e-01]
 [ 1.00719271e+00 4.69710864e-01 -4.35588370e-02]
 [ 1.06365468e+00 1.64532139e-01 -1.16494812e-01]
 [ 1.02043442e+00 -9.47038381e-02 -2.26535697e-02]
 [ 1.21305492e+00 -4.82235831e-02 3.08521659e-01]
 [ 1.11468951e+00 3.73255271e-01 4.13398773e-01]
 [ 9.53927550e-01 5.04218782e-01 6.06214944e-02]
 [ 9.18800802e-01 3.62274157e-01 -1.15652836e-01]
 [ 8.77174165e-01 2.13746512e-01 -1.30839985e-01]
 [ 1.03751393e+00 -3.66851849e-02 8.73766603e-04]
 [ 1.09111641e+00 -7.10817602e-02 3.11831023e-01]
 [ 1.09814686e+00 2.61528742e-01 3.75312216e-01]
 [ 1.04466513e+00 4.47196849e-01 4.74661545e-01]
 [ 8.86543573e-01 4.00422991e-01 1.81562563e-01]
 [ 9.07498646e-01 3.32820276e-01 -8.50752153e-02]
 [ 1.12526702e+00 5.02200387e-02 2.26880441e-02]
 [ 1.11985182e+00 -1.14237823e-01 2.88514926e-01]
 [ 1.12396411e+00 4.84834730e-01 5.10949535e-01]
 [ 8.63407455e-01 6.46797029e-01 4.54523060e-01]
 [ 8.32285084e-01 5.96796893e-01 2.81796804e-01]
 [ 9.10899188e-01 4.17958579e-01 2.76790796e-02]
 [ 9.26065511e-01 3.46891223e-01 3.14047485e-01]
 [ 9.80628074e-01 1.90609877e-01 1.30624767e-01]
 [ 9.82833649e-01 -4.14128839e-03 1.66463274e-02]
 [ 1.01759709e+00 -1.19157401e-01 3.40395776e-01]
 [ 1.01536886e+00 3.54609087e-02 1.20275181e-01]
 [ 1.01866885e+00 3.38031502e-01 1.08120183e-01]
 [ 9.36195549e-01 4.49177041e-01 3.90649720e-01]
 [ 9.55446065e-01 2.93841487e-01 5.28457935e-02]]
```

[9.50505393e-01 -4.21560223e-03 9.91584495e-03]
 [9.75896604e-01 -2.91510858e-03 3.48117857e-01]
 [9.58924354e-01 2.95277282e-01 3.08894582e-01]
 [8.75923016e-01 4.75583834e-01 3.27226440e-01]
 [9.61512916e-01 4.31485730e-01 4.85947196e-02]
 [9.68003033e-01 1.70446223e-01 -9.43586428e-02]
 [1.02824916e+00 -3.11606698e-02 3.92098365e-02]
 [1.05209565e+00 -6.18648133e-02 3.57222979e-01]
 [9.32216771e-01 3.77051011e-01 2.67982341e-01]
 [8.01686396e-01 5.24199479e-01 2.73193047e-01]
 [9.70977482e-01 2.43428503e-01 -2.17858100e-02]
 [8.97074198e-01 1.36941852e-01 2.74359220e-01]
 [9.58566375e-01 -6.61853778e-02 9.77851081e-02]
 [9.30589436e-01 5.11377607e-02 -2.97343532e-02]
 [9.01757376e-01 1.52944762e-01 2.92551727e-01]
 [9.15973608e-01 4.16782846e-01 3.20603215e-01]
 [8.70215725e-01 4.02816252e-01 2.79176211e-02]
 [9.37960867e-01 1.41414151e-01 -8.35406624e-02]
 [1.02473599e+00 -6.89080394e-02 1.39928402e-01]
 [9.94999393e-01 2.26544871e-01 2.61008478e-01]
 [8.75321174e-01 4.61363930e-01 1.95121926e-01]
 [8.83939315e-01 2.70606365e-01 -4.78138640e-02]
 [9.75392289e-01 4.61990621e-02 9.89456764e-02]
 [9.63719037e-01 1.10650794e-01 2.52950311e-01]
 [9.03456200e-01 -1.15073740e-01 1.56091812e-01]
 [9.57907342e-01 -1.23083538e-01 3.08362859e-01]
 [8.85731068e-01 7.22083959e-02 1.08093744e-01]
 [9.23579936e-01 3.70386711e-01 3.64975901e-01]
 [8.26575086e-01 4.93362038e-01 1.59480508e-01]
 [9.18614729e-01 3.56403452e-01 -5.95743014e-02]
 [1.03421791e+00 1.34097480e-01 6.51622139e-02]
 [9.47237091e-01 -1.12211654e-01 4.74690789e-02]
 [9.38127631e-01 2.59454724e-02 2.43375619e-01]]

Dimension accuracy verification

Brick	Physical object measurement (centimeters)	Reconstructed object measurement (distance up to unknown scale factor)
Long edge length	19.2	431
Short edge length	8.89	179
Long edge/Short edge	2.15	2.4
Depth of highest face	2.25	.0017
Depth of center inset face	1.45	.000815
Highest face depth/center inset face depth	1.55	1.31

The ratios of interest are highlighted. The reconstructed result was close to the original dimensions, with 11% error in the edge lengths and 15% error in the depths.

4.4 Tennis Ball

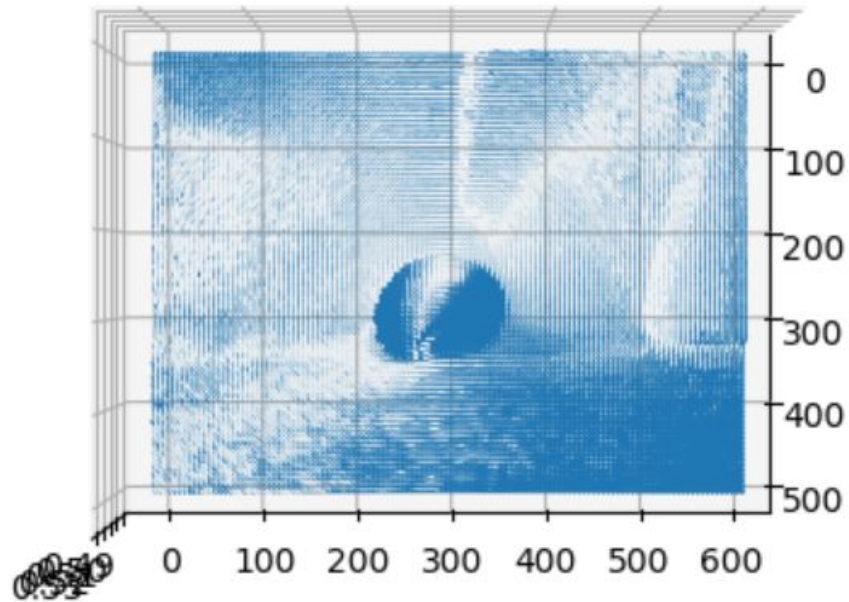


Figure 7: 3D quiver map of (n_x, n_y, n_z) normal vectors



Figure 8: RGB representation of normals



Figure 9: Surface albedo (magnitude of surface normals)



Figure 10: 2D depth map

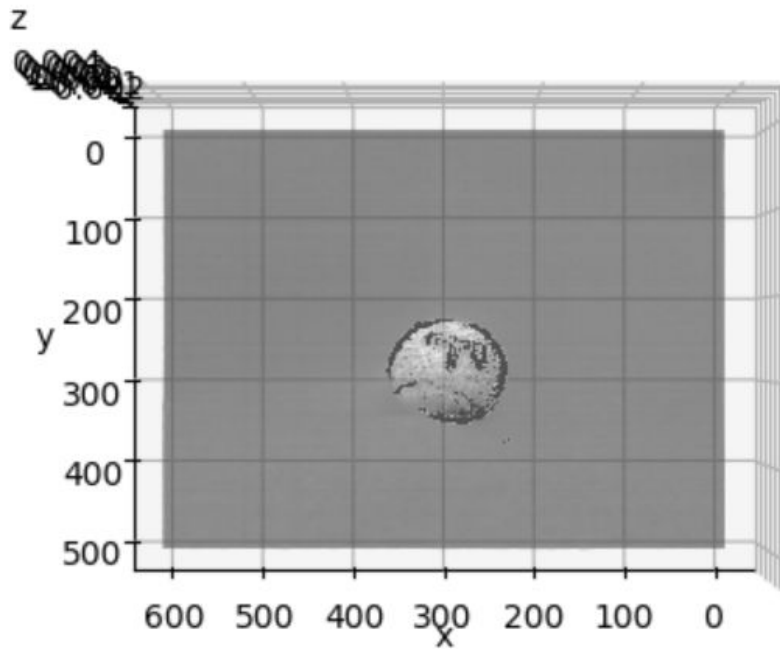


Figure 11: 3D depth map. Depth data was rejected around the edges of the ball due to the integrability constraint.

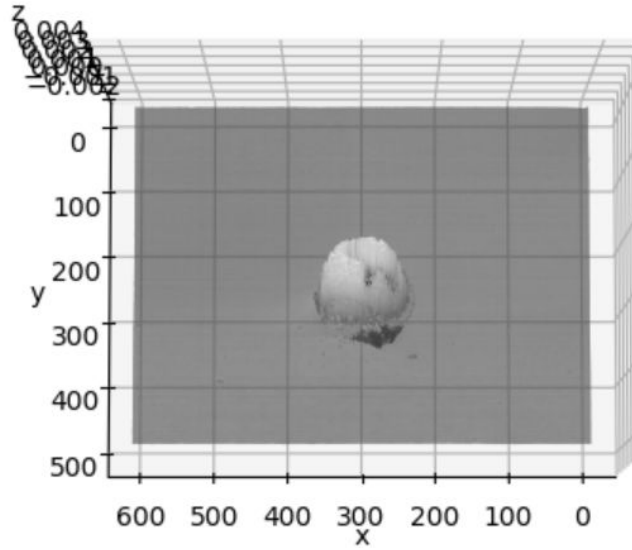


Figure 12: 3D depth map, tilted to show height.

The light source directions were found to be:

```
[[ 1.01219226e+00 -3.38720871e-02 -2.62531422e-02]
 [ 1.19904746e+00 -4.82775185e-02 -1.60090127e-01]
 [ 1.04043987e+00 -2.02010994e-01 -6.67053373e-02]
 [ 1.08200471e+00  1.96148947e-01 -4.40225167e-02]
 [ 1.08968158e+00 -2.91185616e-01  2.26146896e-02]
 [ 1.02368426e+00 -2.72041249e-01 -4.66444257e-02]
 [ 1.01336991e+00  8.05201177e-02 -2.11448505e-01]
 [ 9.24006196e-01  1.31487843e-01 -6.85693507e-02]
 [ 9.26947632e-01  5.26227830e-02 -2.54763277e-03]
 [ 9.82564814e-01 -2.52784871e-01  2.54630935e-02]
 [ 9.37526267e-01 -3.07165625e-01 -3.69011098e-02]
 [ 9.68543909e-01  2.65573972e-02 -1.75089196e-01]
 [ 9.29738383e-01  1.94607764e-01 -7.28722892e-02]
 [ 9.23721308e-01 -9.29978594e-02 -9.32904972e-02]
 [ 9.49442539e-01 -3.17543527e-01 -1.27593219e-02]
 [ 1.04202971e+00 -3.35828487e-01  4.93275194e-02]
 [ 1.00916227e+00 -3.12862350e-01  8.68449111e-02]
 [ 1.17106795e+00  1.94680341e-02  7.67142117e-02]
 [ 1.03978435e+00  1.04216792e-01 -1.47541449e-02]
 [ 1.12928897e+00  7.13238686e-02 -1.37475835e-01]
 [ 9.30422492e-01 -4.99625679e-03 -2.10351035e-01]
 [ 1.03123436e+00 -7.69591743e-02 -9.61230964e-02]
 [ 9.78860713e-01 -2.12880400e-01 -2.89406238e-02]
 [ 9.66225509e-01 -2.99093381e-01  3.32123548e-02]
 [ 1.03076390e+00  1.89155599e-02  6.65439078e-02]
 [ 9.83807484e-01  1.35110919e-01 -2.37028103e-02]
 [ 9.60688175e-01  1.47916622e-01 -1.33516346e-01]
 [ 9.91916715e-01  1.12377576e-01 -1.88632615e-02]
 [ 9.08585540e-01 -1.97047683e-01 -5.35225968e-02]
 [ 9.37252998e-01 -4.10320789e-01  2.41784284e-02]
 [ 1.02159694e+00 -1.19969552e-01 -5.94766983e-02]
 [ 9.64429341e-01 -1.76347632e-02 -1.51104000e-01]
 [ 9.36244158e-01  1.00184640e-01 -5.87965290e-02]]
```


[9.52396793e-01 2.02861085e-02 1.16852223e-02]
 [9.28975759e-01 -1.68891529e-01 6.26526243e-02]
 [9.55188997e-01 -1.93629176e-01 1.01859245e-01]
 [9.95488731e-01 -8.10723445e-02 9.83970151e-02]
 [1.12074585e+00 1.60502449e-01 4.06183724e-02]
 [1.17577094e+00 1.66519607e-01 -1.28798472e-01]
 [1.00366142e+00 8.70332248e-02 -2.11780926e-01]
 [9.72069246e-01 8.94795401e-02 -7.32350719e-02]
 [8.67055295e-01 8.21614502e-03 -5.62028095e-04]
 [8.64933529e-01 -3.14648711e-01 -2.55358565e-02]
 [8.61239075e-01 -2.68868759e-01 4.71137606e-02]
 [8.37786314e-01 -6.16467612e-02 6.94290002e-02]
 [8.88548382e-01 5.18536440e-02 2.26526754e-02]
 [9.68182493e-01 3.08690606e-03 -1.04839860e-01]
 [8.58764388e-01 9.52261126e-03 -1.74028251e-01]
 [9.45783492e-01 9.99412703e-02 -4.03517060e-02]
 [9.00502726e-01 -3.25655438e-01 -4.89061234e-02]
 [8.48950830e-01 -3.61944722e-01 3.80581543e-02]
 [8.47501321e-01 1.69807497e-02 6.05905441e-02]
 [8.70072068e-01 3.86428255e-02 -1.91800677e-02]
 [7.31532852e-01 6.35403060e-02 -1.06723643e-01]
 [8.27931914e-01 -9.93626539e-02 -4.07753292e-02]
 [8.92261394e-01 -2.80389437e-01 2.20660668e-02]
 [9.24114763e-01 8.81156109e-02 6.42809126e-03]
 [8.18786758e-01 1.41134448e-01 -8.77918307e-02]
 [9.85462498e-01 -5.82618843e-02 2.23148496e-02]
 [9.13412495e-01 -3.05743245e-01 -3.30106246e-02]
 [8.23118935e-01 9.97975735e-03 -1.48919575e-01]
 [8.45397822e-01 1.84730078e-01 -6.32735378e-02]
 [8.40301146e-01 3.41738640e-02 9.39258852e-03]]

Tennis ball	Physical object measurement (centimeters)	Reconstructed object measurement (distance up to unknown scale factor)
Diameter on x-axis	6.7	122
Diameter on y-axis	6.7	118
X-axis diameter/y-axis diameter	1	1.03
Ball radius	3.35	.00261208
Surface height at ½ ball radius	3.17	.002326
Ball radius/surface height at ½ ball radius	1.05	1.12

The ratios of interest are highlighted. The dimensions of this reconstruction are also proportionally accurate with their physical test object counterparts, with 3% error in the diameter ratios and 6% error in the height ratios.

5. Discussion and Conclusion

The algorithm was able to accurately compute the generalized bas relief of each object up to a quantifiable dimensional accuracy, with some error. The RGB map of the surface normals of the tennis ball showed some error along the rubber furrow because it likely did not exhibit lambertian reflectance in that specific spot (ie the rubber was glossy). The center inset squares of the brick appeared to have the same issue, possibly because they are more smooth than the rest of the brick surface. This non-lambertian behavior is seen in the albedo images and RGB normal representations, with very white areas in the albedo images showing strong reflectance and very blue areas showing regions with large surface normal z-components. The furrows and label on the ball also appear to have some depth distortion, possibly being confused with shadow. Likewise, the 3D depth maps of each object showed some distortion due to error accumulation in the derivative estimates of the shape from integration method. This is likely because one integration path was used for each partial during the per-pixel depth estimation, when in reality Forsyth and Ponce [3] recommend multiple integration paths per partial to spread the error around. This approach was avoided due to the computational constraints inherent in processing large amounts of data. A more accurate surface-fitting method such as the Frankot-Chellappa method should be used in the future. Additionally, sources of error most likely also came from rank three approximation, least-squares analysis and the camera distortion factor which was not considered. Overall, uncalibrated photometric stereo is a difficult model to adapt accurately to real data without a number of advanced techniques, but this approach with its basic assumptions appeared to be successful.

6. References

- [1] H. Hayakawa. *Photometric stereo under a light-source with arbitrary motion*. Journal of the Optical Society of America, 11(11):3079– 3089, 1994.
- [2] R. Woodham. *Photometric method for determining surface orientation from multiple images*. Optical Engineering, 19(1):139–144, 1980.
- [3] David Forsyth, Jean Ponce. *Computer Vision: A Modern Approach*, pg 34-40. Prentice Hall, 2003.