Photometric Stereo

Yifan Qiao

Introduction

- Implement the algorithm of photometric stereo and focus on the use of point light source to reconstruct the geometry of an object. Multiple images are needed.
- The task is to reconstruct the shape of an object with Lambertian surface using multiple images where the light source position is changed and according to the object's reflecting properties, the albedo, the values of the source vectors are estimated.

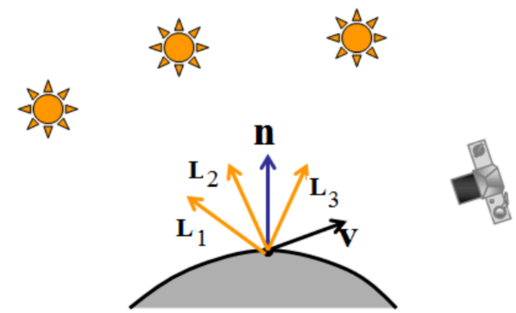
Main factors

- **ILLUMINATION**-This is defined by the position, direction and spectral energy distribution of the light rays.
- **SURFACE REFLECTIVITY OF THE OBJECT**- This is known in shape from shading literature as albedo, the proportion of the incident light or radiation that is reflected by a surface, typically that of a planet or moon.
- **LAMBERTIAN SURFACES**- I am concerned with objects which have Lambertian surfaces.
- **SURFACE GEOMETRY OF AN OBJECT**-I want to recover given the object's 2D gray-scale images.

Estimate illumination direction

- Obtain a ball of known radius, with uniform shading or (color) and hence constant albedo.
- Place a single illuminator through a small hole and place the camera on the same side of the light
- Hang the ball in front of the light and the camera, turn the light on, and take a picture of the ball. Minimize the light interference from other sources.
- Use the image of the ball as well as its radius, recover the center of the ball, hence the equation of the ball w.r.t camera frame..

Theory



- $I = \rho n s$
- $I_k(c, r) = \rho n^t(x, y, z) L_k$
- $A = \rho \begin{pmatrix} L_{1x} & L_{1y} & L_{1z} \\ L_{2x} & L_{2y} & L_{2z} \\ \vdots & & \\ L_{Nx} & L_{Ny} & L_{Nz} \end{pmatrix} \qquad b = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{pmatrix}$
- So n can be solved by minimizing $||An - b||^2$, which leads to
- $n = \frac{(A^t A)^{-1} A^t b}{\rho}$, where $\rho = \|(A^t A)^{-1} A^t b\|$

Theory

- Assume the surface depth z may be thought as a function of (x, y), i.e., z=f(x,y). A 3D point P can therefore be represented as P = (x, y, f(x, y)). Hence the normal vector at (x,y,z) is
- $n(x, y, z) = \frac{\frac{\partial P}{\partial x} X \frac{\partial P}{\partial y}}{\left\| \frac{\partial P}{\partial x} X \frac{\partial P}{\partial y} \right\|} = \frac{[-p(x,y), -q(x,y), 1]^t}{\sqrt{1+p^2(x,y) + q^2(x,y)}}$
- where p = $\frac{\partial f}{\partial x}$ and q = $\frac{\partial f}{\partial y}$. Note (x, y, z) is the coordinates of 3D point relative to the camera frame. As a result, p and q are specified with respect to the camera coordinate frame.
- Then, p and q can be easily computed when n is known.

Theory

- The problem here is to reconstruct the surface height z.
- $\boldsymbol{Z_x = p, Z_y = q}$.
- Define an error function on proposed depth Z with respect to given gradient (p, q), written as E(Z; p, q).
- $\boldsymbol{E(Z; p, q) = (Z_x - p)^2 + (Z_y - q)^2}$
- So the problem of finding optimal Z(x, y) can be formulated as a minimization of cost function
- $\boldsymbol{Cost(Z) = \iint E(Z; p, q)\, dx\, dy}$

Theory

- First, I did some tricky work here.
- I use $Z(x-1,y-1) + q(x,y) + p(x,y)$ to estimate $Z(x,y)$
- And I found that the result is not so good when the object has many detailed information like nose and fingers. And the reconstruction is not smooth.

Theory

- Then I use Frankot Chellappa Algorithm.
- Assuming the depth map can be written as a linear combination of basis function φ(x, y; ω), where ω = (ω_x, ω_y) = (u, v) is the a 2D index
- $Z(x, y) = \sum_{\omega} C(\omega)\, \varphi(x, y; \omega)$
- And define

$$P_x(\boldsymbol{\omega}) = \iint \left|\phi_x\left(x, y; \boldsymbol{\omega}\right)\right|^2 dxdy, \quad P_y(\boldsymbol{\omega}) = \iint \left|\phi_y\left(x, y; \boldsymbol{\omega}\right)\right|^2 dxdy.$$

- And I use discrete Fourier basis for its computational efficiency

$$\phi(x, y; \boldsymbol{\omega}) = \exp\left(j2\pi\left(\frac{xu}{N} + \frac{yv}{M}\right)\right),$$ where M and N are the dimensions.

Theory

- So we know the partial derivatives and power of the basis.
- The expansion coefficients $\widehat{C_1}(\omega)$and $\widehat{C_2}(\omega)$ can be calculated from the Discrete Fourier Transform (DFT) of p and q, written as $\widehat{C_p}(\omega)$ and $\widehat{C_q}(\omega)$,

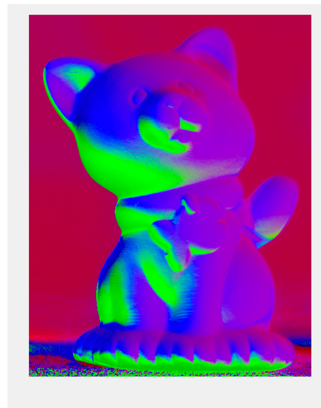
$$\widehat{C}_1(\omega) = -\frac{jN}{2\pi u}\frac{\widehat{C}_p(\omega)}{MN}, \quad \widehat{C}_2(\omega) = -\frac{jM}{2\pi v}\frac{\widehat{C}_q(\omega)}{MN}.$$

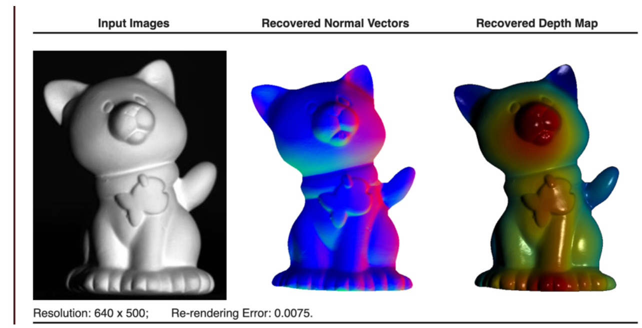
- And the final Z with respect to q and p can be written as:

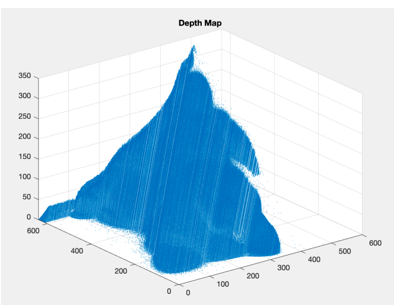
$$Z = \mathcal{F}^{-1}\left\{-j\frac{\frac{2\pi u}{N}\mathcal{F}\{p\} + \frac{2\pi v}{M}\mathcal{F}\{q\}}{\left(\frac{2\pi u}{N}\right)^2 + \left(\frac{2\pi v}{M}\right)^2}\right\} = \mathcal{F}^{-1}\left\{-\frac{j}{2\pi}\cdot\frac{\frac{u}{N}\mathcal{F}\{p\} + \frac{v}{M}\mathcal{F}\{q\}}{\left(\frac{u}{N}\right)^2 + \left(\frac{v}{M}\right)^2}\right\},$$

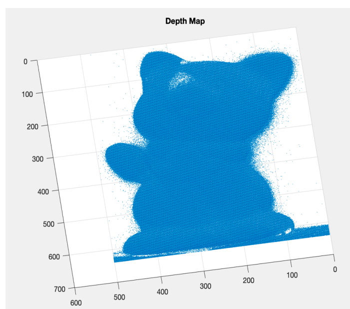
- Where $\mathcal{F}\{\cdot\} and\ \mathcal{F}^{-1}\{\cdot\}$ are DFT and inverse DFT operations, respectively.

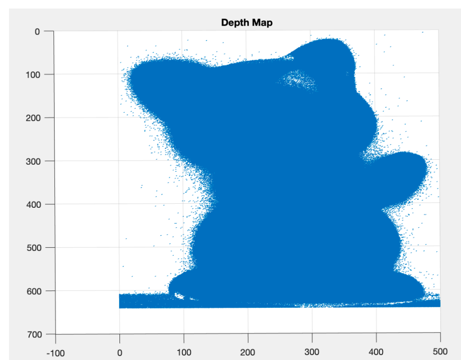
Results -- cat

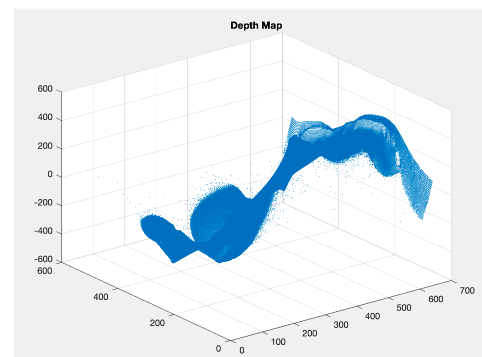



Input Images
Recovered Normal Vectors
Recovered Depth Map
Resolution: 640 x 500;
Re-rendering Error: 0.0075.

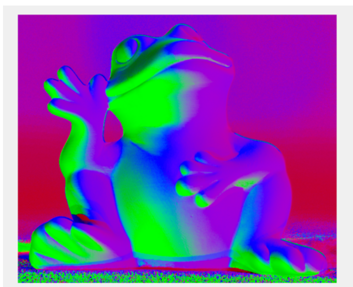

Depth Map

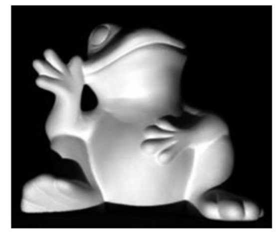

Depth Map

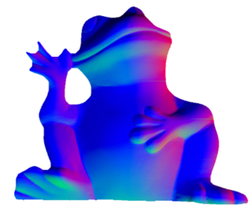

Depth Map

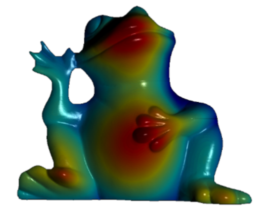

Depth Map

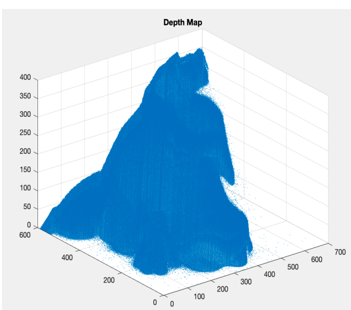
Results -- frog

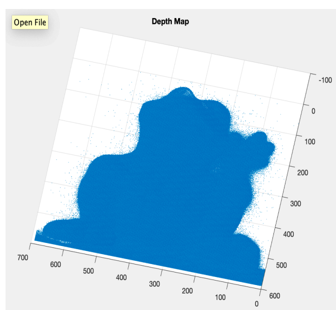


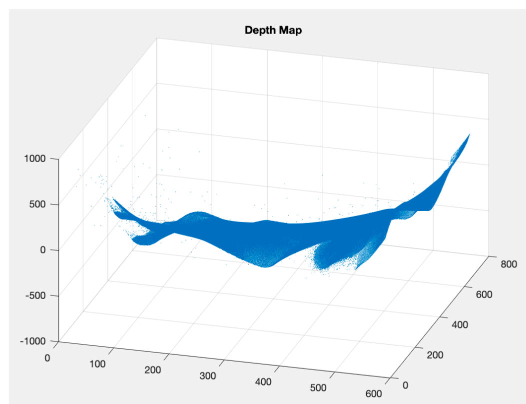


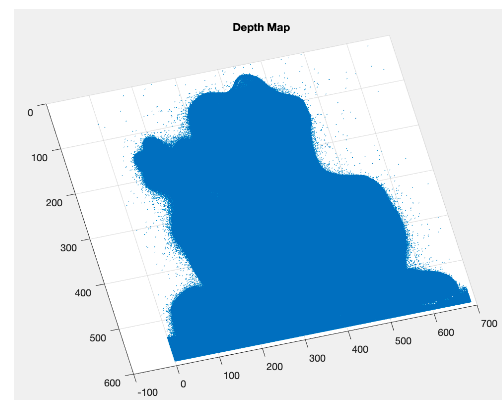












Conclusion

- The author's method can provide more clearly detailed information. My depth map can't be seen clearly, but only shows the frame of the object, and some detailed information like nose and fingers can only be seen in the specific angle.
- This can be due to the algorithm of normal vectors and shading regions where the intensity of light falling on the surface is 0. When errors get summed up due to integration of the gradients p, q to form the surface, the errors get added up giving a less perfect solution.
- What's more, the second method is more smooth than the first method.

Reference

- [1] Xiong Y, Chakrabarti A, Basri R, et al. From shading to local shape[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 37(1): 67-79.
- [2] Robert Frankot and Rama Chellappa. "A method for enforcing integrability in shape from shading algorithms." Pattern Analysis and Machine Intelligence, IEEE Transactions on 10.4 (1988): 439-451.

Thank you!