Computer Vision hw1: Surface reconstruction

I. Photometric Stereo $I(x, y) = \rho(x, y)SN(x, y)$

II. Normal Estimation

Read images of different light source, if there are salt and pepper noise in the picture, use median filter first. (For bunny and star of special case.)

Solve $Sb = I$ to get the normal with minimum loss.

(My S and N are unit vectors.)

However, there are some bad points such as specular or shadow from images.

Use Weighted Least Squares to avoid them.

Now, solve $WSb = WI$

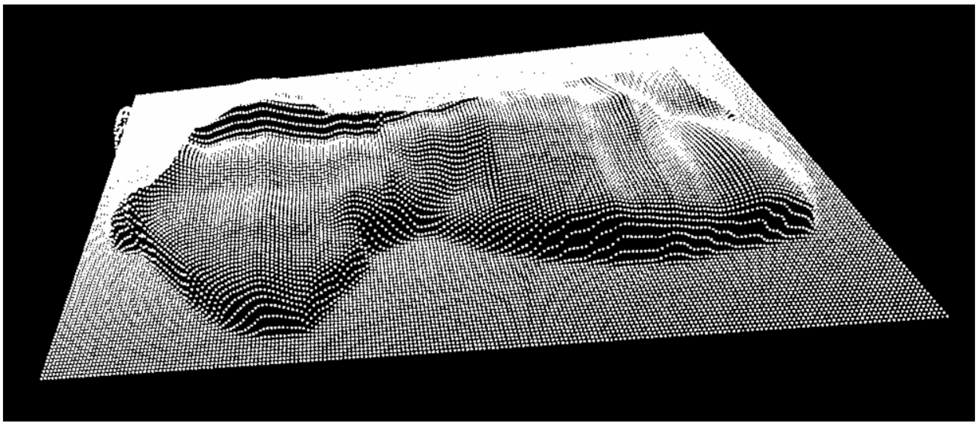
I set the weight = 0.001 if it's too bright or too dark, else 1.

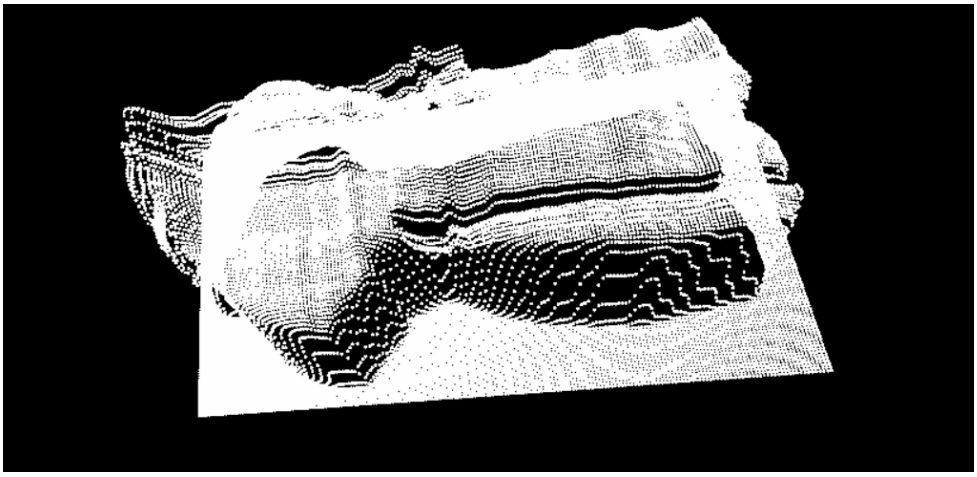
Then, calculate the gradient of X and Y direction.

<u>Attention!</u>

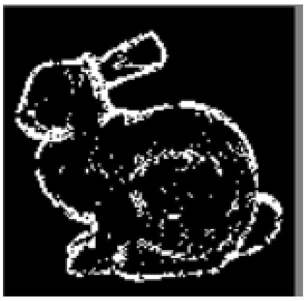
Remember filter the gradients before integration to avoid some noise in gradient.

Using median filter will get the best result, and the difference is obvious in Venus.





After getting the X, Y gradient, do Sanity Check. Record which pixels are bad points. The result of $\frac{\partial^2 z}{\partial x \partial y} = \frac{\partial^2 z}{\partial y \partial x}$ can be used to find edges and corners.

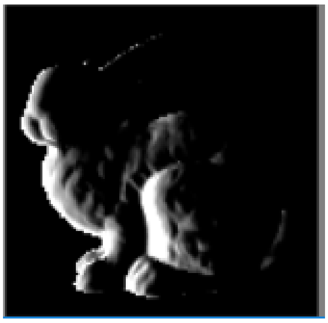


III. Surface Reconstruction

A. Integration

Start from four corners, each corner has two ways to go. (first go x then y or first go y then x) Then, there will be eight paths for integral, but get only four kinds of result because the gradients on four sides are 0s.

Attention!

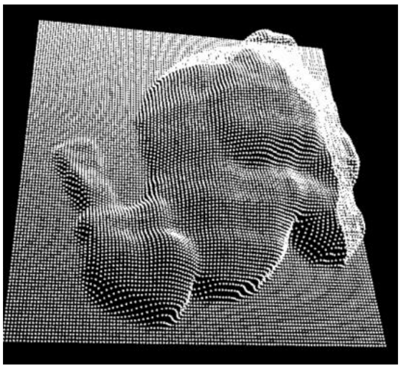


Check the direction of gradients first, so that I can know when to add or when to subtract during integration.

(If the gradient go →, then I should add when → and subtract when ←)

Now I have four integrations, discard the bad values which is too far from mean. (I use 2σ as threshold.) Then Z could be evaluated.

There are many minus values of Z, so add the minimum of them and use Gaussian filter to smooth the result picture.



B. Recovering the shape (appendix)

Concept:

Laplacian on Z can be calculated from gradient of gradient.

$$Lz = \nabla^2 f(x,y) = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} = \frac{\partial\left(\frac{\partial f}{\partial x}\right)}{\partial x} + \frac{\partial\left(\frac{\partial f}{\partial y}\right)}{\partial y} = B$$ with

$\frac{\partial f}{\partial x} = -\frac{n_a}{n_c}$ and $\frac{\partial f}{\partial y} = -\frac{n_b}{n_c}$

Then solve $Lz = b$

$$\text{L:} \begin{bmatrix} -4 & 1 & 0 & \cdots & 1 & & & & & \\ 1 & -4 & 1 & & & 1 & & & & \\ 0 & 1 & -4 & 1 & & & \ddots & & & \\ \vdots & & 1 & -4 & 1 & & \vdots & 1 & & \\ 1 & & & 1 & -4 & 1 & & & 1 & \\ & 1 & & & 1 & \ddots & \ddots & & \vdots & \\ & & 1 & & & \ddots & -4 & 1 & 0 & \\ & & & \ddots & & & 1 & -4 & 1 & \\ & & & & 1 & \cdots & 0 & 1 & -4 & \end{bmatrix}_{v*v}$$

$$(v = rows * cols)$$

$$z: \begin{bmatrix} \vdots \end{bmatrix}_{v*1} \quad , unknown$$ (what we want)

$$b: \begin{bmatrix} \vdots \end{bmatrix}_{v*1}$$, make the gradient of gradient into *v*1* vector

Therefore, $z = L^{-1}b$.

But *L* may be too big, use Jacobi method to approximate *Z*

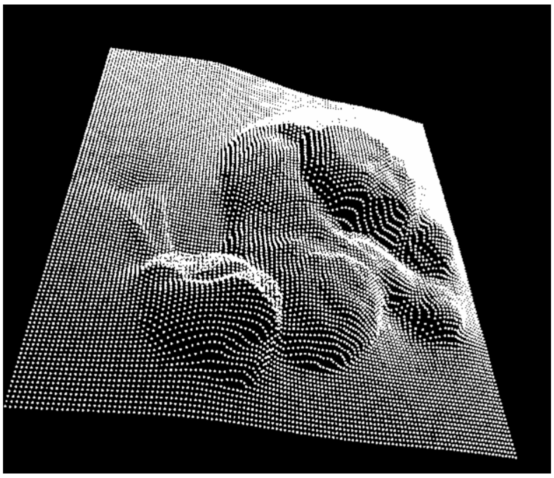
$$\mathrm{L = D + R} \quad \mathrm{D: diagonal\ \ component, R: the\ \ remainder}$$

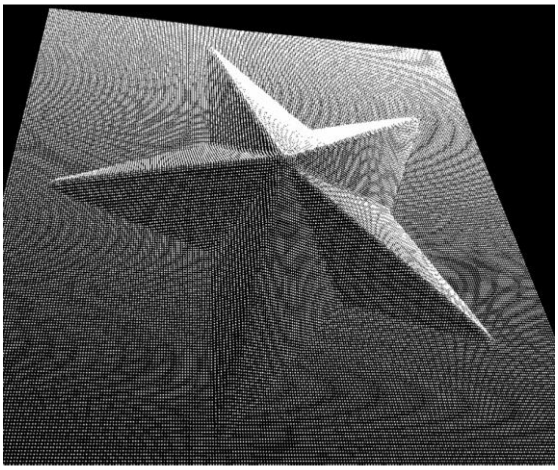
$$z^{(k+1)} = D^{-1}(b - \mathrm{R}z^{(k)})$$

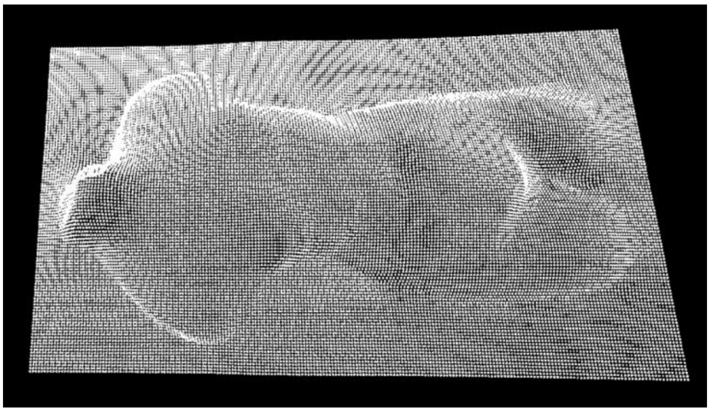
My error function of kth approximation: $\frac{\left\|Lz^{(k)} - b\right\|^2}{\left\|Lz^{(0)} - b\right\|^2}$

(Jacobi method needs some time because iterations until the threshold.)

After getting the z I want, use unsharp masking and the results from sanity check to make the results sharper. Finally, apply median filter again to filter out noise!







P.S. (for myself)

1. X: row, Y: col
2. To compute gradient, just use a -1 0 1 mask