187

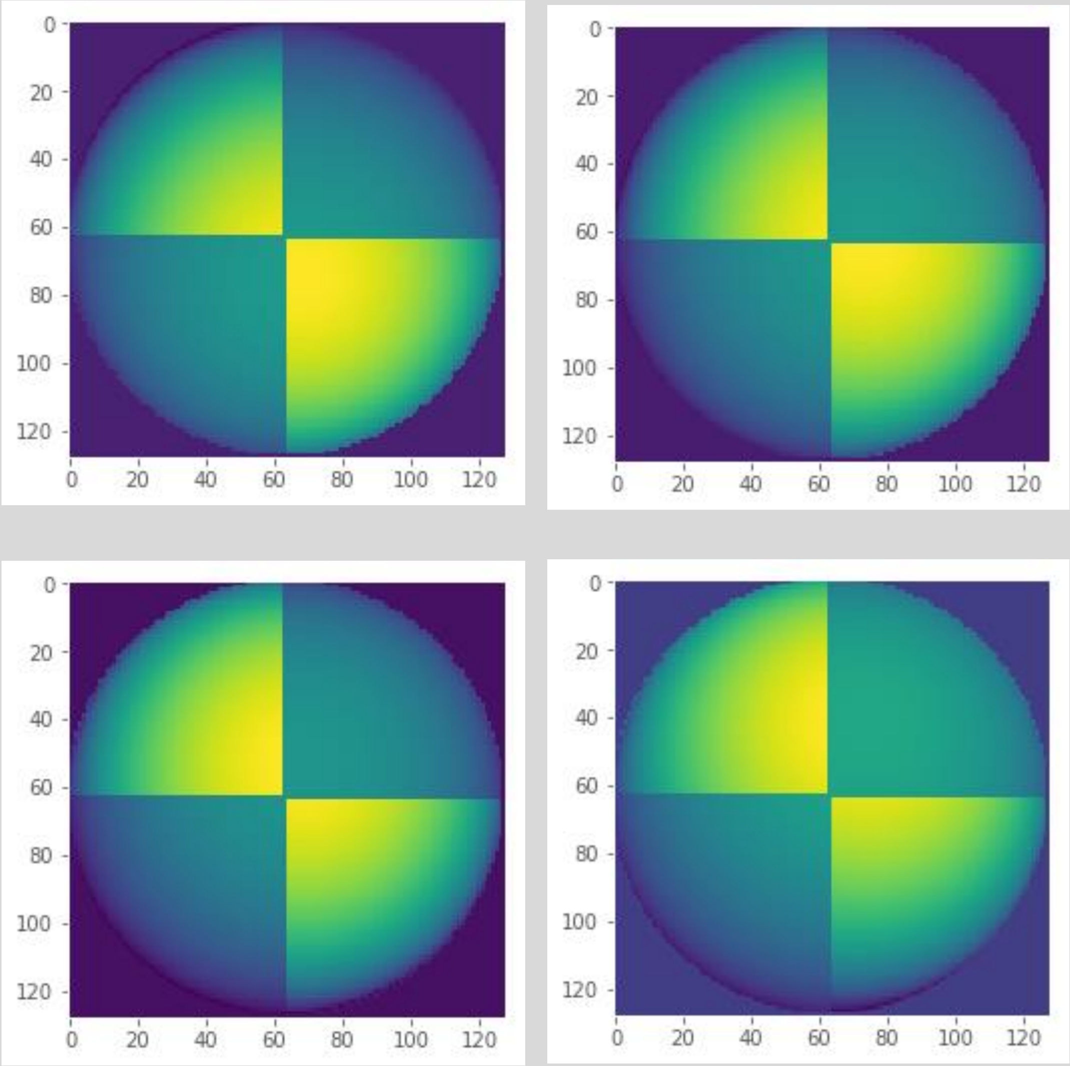
# Activity 8: Photometric Stereo

Mary Chris Go / 2014 11122

# Goal

Mary Chris Go / 2014 11122

To be able to construct a 3D shape recovered as a mesh




# Photos.mat

The files was already given to us. These photos were opened through scipy: spio.loadmat

Mary Chris Go / 2014 11122

Since I and V are known, we can solve for g in the least squares sense by

$$\mathbf{g}=(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{I} \tag{10}$$

To get the normal vector we simplify normalize g by its length

$$\hat{n}=\frac{\mathbf{g}}{|\mathbf{g}|} . \tag{11}$$

Once the surface normals (nx, ny, nz) are estimated using photometric stereo, they are related to the partial derivative of f as

$$\frac{\partial f}{\partial x}=\frac{-nx}{nz};\frac{\partial f}{\partial y}=\frac{-ny}{nz} \tag{14}$$

```
from autograd.numpy import linalg as la
#Equation 10
g = la.inv(V.T.dot(V)).dot(V.T).dot(I.swapaxes(0, 1))
#Equation 11
n = g/g.size #this is a vector
#Equation 14
dfdx = np.nan_to_num(-n[0]/n[2])
dfdy = np.nan_to_num(-n[1]/n[2])
```

# Computing the surface normal Using Equation 10, 11, 14

Now, we try to establish these equations using python which will then be used for the chosen technique.

Mary Chris Go / 2014 11122

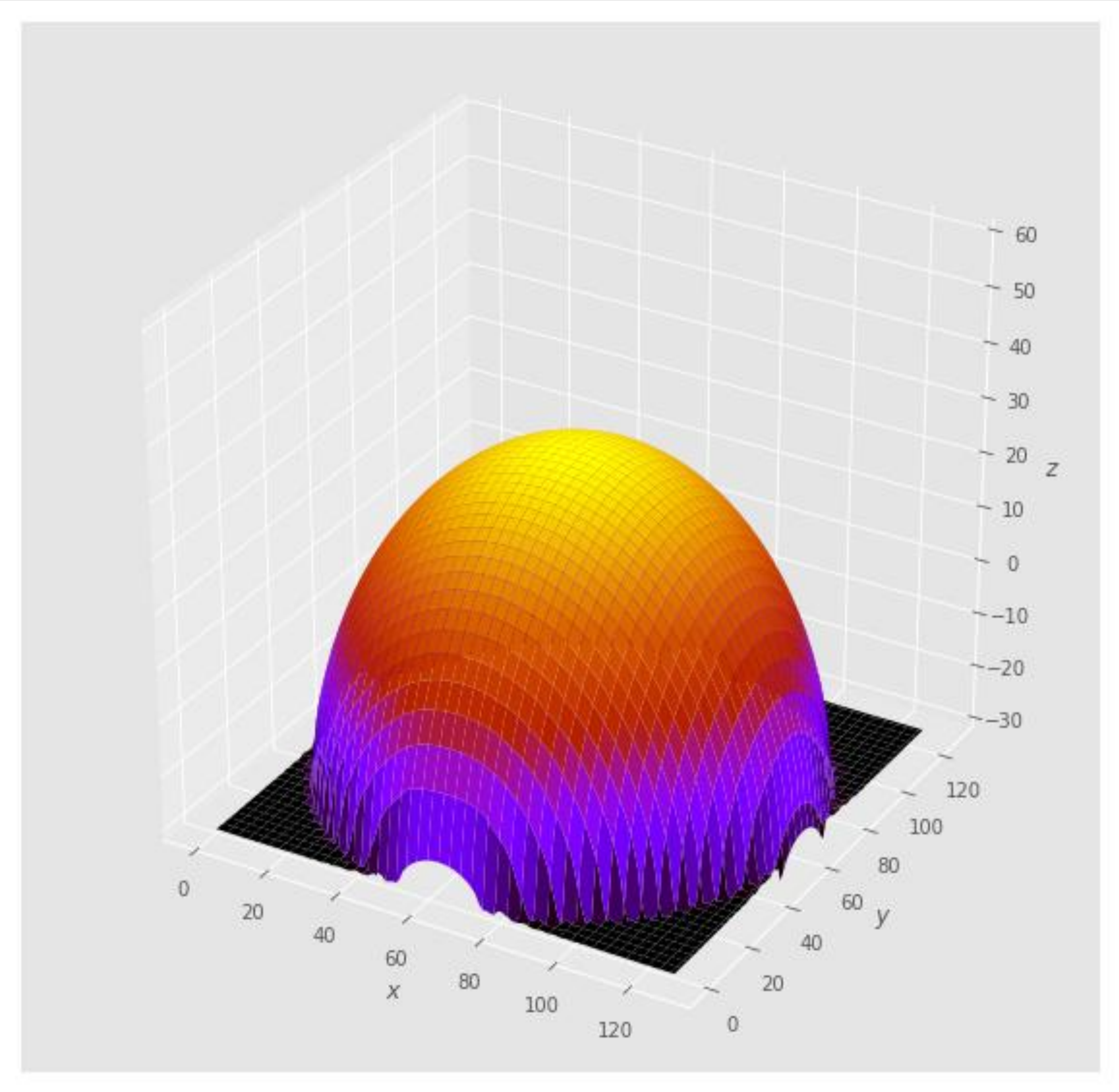
## Computing elevation, displaying 3D plot

```
import scipy.fftpack as fft
#getting the shape
shape1, shape2 = dfdx.shape #128x128

#Get the meshgrid of the shape of dfdx for x-axis and y-axis
x, y = np.meshgrid(np.arange(shape2), np.arange(shape1))

#elevation
w_x, w_y = np.meshgrid(fft.fftfreq(shape2)*2*np.pi,
                       fft.fftfreq(shape1)*2*np.pi)
upper = -1j*w_x*fft.fft2(dfdx) - 1j*w_y*fft.fft2(dfdy)
lower = w_x**2 + w_y**2 + np.finfo(float).eps
res = fft.ifft2(upper/lower)
z = res.real
```

Now this is the hardest part of the activity. Out of the three choices, I chose the Frankot-Chellappa algorithm to achieve the goals of this activity.

The algorithm involves getting the shape of the partial derivative of f with respect to x. These values will then be used for x and y respectively. This will serve as the the values for x axis and y axis of the plot.

For the elevation, we get the real part of the 2D inverse fast fourier transform of a fraction. This fraction comes from the equation from the Frankot-Chellappa algorithm which was shown from the snippet of the code.




# Plotting the 3D surface

Through plotting x, y, and z (see code before), I was able to plot the surface of the given .mat file.

Mary Chris Go / 2014 11122

# Summary

I was able to plot the surface. The challenge is to find the most efficient algorithm among the three and what works comfortably for you.

It was a short activity but it was challenging.

# References


- Soriano, M. Photometric Stereo, Applied Physics 187
- Frankot, R. T., & Chellappa, R. (1988). A method for enforcing integrability in shape from shading algorithms. IEEE Transactions on pattern analysis and machine intelligence, 10(4), 439- 451.