Daneshvarian Maryam Sadat 2016-12631

#1

quiver plot displays velocity vectors as arrows with components (U,V) at the points (X,Y) by componets U(1),V(1) and is displayed at the point X(1),Y(1).

quiver(U,V)

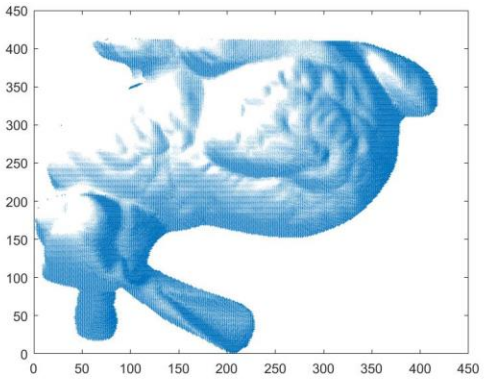
In this case,

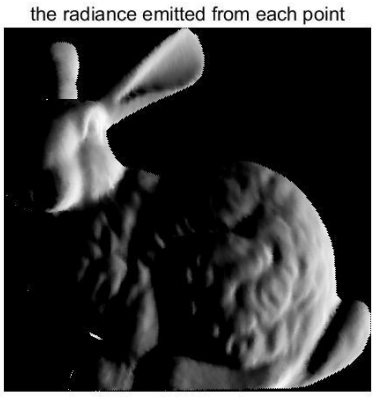
length(X) = n and length(Y) = m, where [m,n] = size(U) = size(V)

Bunny is loaded in B so :

I did used q=quiver(B.N(:,:,1),B.N(:,:,2));
this figure shows normal vectors of surface as arrows on every pixel.
In second part, to calculate the surface which is equal to the radiance emitted from each point as z is 1 in this surface and s is (0, 0 ,1) I did only considered B.N(X,Y,1)

Here are the figures obtained first by quiver and then radiance :

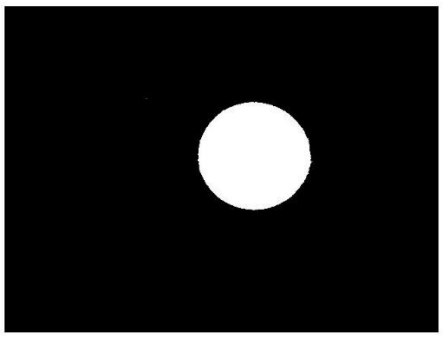




By changing the scale of quiver can obtimize the size of arrows. By comparing these two images can find out that quiver makes pixels in shadow visible so it is useful to use for detecting shape or surface.

#2

In this experiment for choosing right value for dark threshold, a level value of 0.5 corresponds to an intensity value halfway between the minimum and maximum value of the class. (0-127-255), Mid 127, so threshold value 0.5 but after using the image with circle threshold, the out put was not really desired so I did increased it to 150/255=0.58 , then afterall could obtain such image in finding circle :



For Circle threshold to get the best standard I did calculate the value by first summing image light intensity and then dividing the calculated sum by total number of pixels

total number of pixels = x*y

threshold=sum/(x*y);

In finding circles module, first did calculate the binary image of input by using threshold then calculated center and radius of circle by finding the most top point and the most left point on circle.

The results was cx=215 ,cy=368, r=79 and after comparing by matlab already implemented function saw that there is a error of +-1 in radius and +- 3 in cx, cy .

In finding light, according to guide and lecture notes, did used the intensity of the light source that is proportional to the magnitude (brightness) of the brightest pixel on the sphere.

In compute Normals part, according to the link provided in the lecture note , as shown below did calculate the normal.

First illuminate the intensities that were less than threshold then obtained normal by below formula.

$$
\begin{aligned}
I_1 &= k_d(N_x * L_{1x} + N_y L_{1y} + N_z L_{1y}) \\
I_2 &= k_d(N_x * L_{2x} + N_y L_{2y} + N_z L_{2y}) \\
I_3 &= k_d(N_x * L_{3x} + N_y L_{3y} + N_z L_{3y}) \\
. &= . \\
. &= . \\
. &= . \\
I_n &= k_d(N_x * L_{nx} + N_y L_{ny} + N_z L_{ny})
\end{aligned}
$$

$$[G] = k_d[N]$$

iquare methods as

$$
\begin{aligned}
[L]G &= [I] \\
L^T LG &= L^T I \\
G &= (L^T L)^{-1} L^T I \\
k_d &= \|G\| \\
N &= G/k_d
\end{aligned}
$$

I did brief up above equation as below : L : light source

```
L_inv=inv(L);
n=L_inv .*I;
R=norm(n);
n=n/R;
normals(i,j,1)=n(1);
normals(i,j,2)=n(2);
normals(i,j,3)=n(3);
albedo(i,j)=R;
```

but unfortunately, I could not find out the reason why I get an error while reshaping the normal in scripts provided.

#3

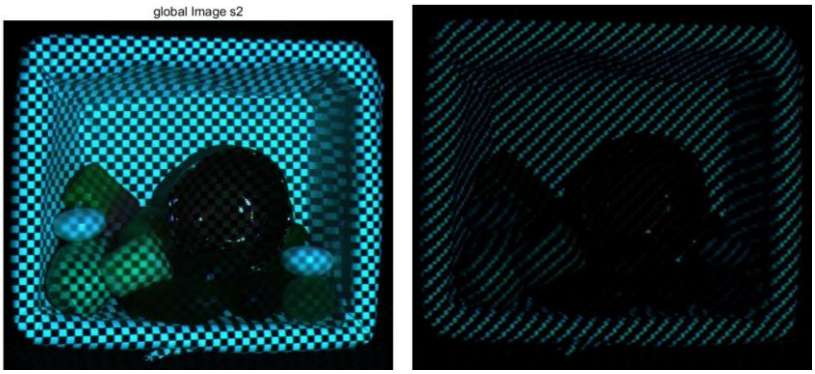
In this part I did write 2 modules : separate global direct and q3 script.

In the script first assigned the data path folders and passed to the calculation module.

In separate Global Direct, first the problem was loading the multiple images so I did used struct to first obtain the number of images in the folder.

Next obtained the min Image and max Image that have minimum and maximm intensity over all the input images by comparing their intensity's sum.

```
globalImg=2.*minImg;
directImg=maxImg-minImg;
```



The left image shows the global image , by this experiment can guess this method will be useful in object selecting in image as edges and regions are separated.