Mingxuan Wang

University of California, San Diego

November 3, 2014

ASSIGNMENT 2

Problem 1. Steradians [2 pts]

Let θ span 0 to $\pi/3$ radians and ϕ span 0 to π radians. How many steradians are in the section of the sphere covered by θ and ϕ?

Solutions:

$$\int_0^{\pi}\int_0^{\frac{\pi}{3}} sin\theta \mathrm{d}\theta \mathrm{d}\phi = \frac{1}{2}\pi(steradians) \tag{1}$$

Problem 2. Irradiance [3 pts]

(a) a.Consider a cylinder with radius r and height h whose base is centered at $z = 0$ along the xy-plane. If the walls of the cylinder have constant radiance L and the top of the cylinder has constant radiance $4L$, what is the irradiance E at the point (0, 0, 0) assuming that the surface at (0, 0, 0) has a normal vector of (0, 0, 1)?

(b) b.What is the irradiance if the radiance of the top is now $2Ld^2$ where d is the distance to the center of the top?

Solutions:

(a) a.Note that $cos\theta_1 = h/(h^2 + r^2)^{1/2}$, $sin\theta_1 = r/(h^2 + r^2)^{1/2}$.

part 1: irradiance from the top

$$\int_0^{2\pi}\int_0^{\theta_1} 4Lcos\theta sin\theta \mathrm{d}\theta \mathrm{d}\phi = L\int_0^{2\pi}\int_0^{\theta_1} sin2\theta \mathrm{d}2\theta \mathrm{d}\phi = \frac{4\pi r^2 L}{h^2 + r^2} \tag{2}$$

part 2: irradiance from the wall

$$\int_0^{2\pi}\int_{\theta_1}^{\frac{\pi}{2}} Lcos\theta sin\theta \mathrm{d}\theta \mathrm{d}\phi = \frac{L}{4}\int_0^{2\pi}\int_{\theta_1}^{\frac{\pi}{2}} sin2\theta \mathrm{d}2\theta \mathrm{d}\phi = \frac{\pi h^2 L}{h^2 + r^2} \tag{3}$$

Total irradiance:

$$part1 + part2 = \frac{4\pi r^2 L + \pi h^2 L}{h^2 + r^2} \tag{4}$$

(b) b.Note that $tan\theta = d/h$

part 1: irradiance from the top

$$\int_0^{2\pi}\int_0^{\theta_1} 2Ld^2 cos\theta sin\theta \mathrm{d}\theta \mathrm{d}\phi = 2Lh^2 \int_0^{2\pi}\int_0^{\theta_1} (\frac{sin\theta}{cos\theta} - sin\theta cos\theta)\mathrm{d}\theta \mathrm{d}\phi = 4\pi Lh^2(-ln\frac{h}{(h^2+r^2)^{\frac{1}{2}}} - \frac{r^2}{2h^2+2r^2}) \tag{5}$$

part 2: irradiance from the wall

$$\int_0^{2\pi}\int_{\theta_1}^{\frac{\pi}{2}} Lcos\theta sin\theta \mathrm{d}\theta \mathrm{d}\phi = \frac{L}{4}\int_0^{2\pi}\int_{\theta_1}^{\frac{\pi}{2}} sin2\theta \mathrm{d}2\theta \mathrm{d}\phi = \frac{\pi h^2 L}{h^2 + r^2} \tag{6}$$

Total irradiance:

$$part1 + part2 = 4\pi Lh^2(-ln\frac{h}{(h^2+r^2)^{\frac{1}{2}}} - \frac{r^2}{2h^2+2r^2}) + \frac{\pi h^2 L}{h^2+r^2} = -4\pi Lh^2 ln\frac{h}{(h^2+r^2)^{\frac{1}{2}}} - \frac{\pi Lh^2 r^2}{h^2+r^2} \tag{7}$$

Problem 3. Lambertian surfaces [2pts]

A Lambertian surface is one that appears equally bright from all viewing direction. In other words, the emitted radiance from a Lambertian surface is not a function of outgoing direction. Assume that we have an ideal Lambertian surface which also re ects all incident lights (absorbing none), the BRDF$\rho(\theta_{in}, \phi_{in}, \theta_{out}, \phi_{out})$of such a surface will be a constant. What is that constant?

Solution:

Assume that BRDF of Lambertian surface reflects a fraction of ρ of the light.

We have:

$$\int_0^{2\pi}\int_0^{\frac{\pi}{2}} fcos\theta sin\theta \mathrm{d}\theta \mathrm{d}\phi = \rho \tag{8}$$

$$f = \frac{\rho}{\pi} \tag{9}$$

Then the constant is $f = \frac{\rho}{\pi}$

Problem 4. Photometric Stereo and Specularity Removal [10 points]

The goal of this part of the assignment is to implement a couple of different algorithms that reconstruct a surface using the concept of photometric stereo. Additionally, you will implement the specular removal technique of Mallick et al., which enables photometric stereo reconstruction of certain non-Lambertian materials. You can assume a Lambertian reflectance function once specularties are removed, but the albedo is unknown and non-constant in the images. Your program will take in multiple images as input along with the light source direction (and color when necessary) for each image. You will also implement a second example-based photometric stereo algorithm which is based on simultaneously imaging two objects of the same material, one of which has known structure.

(a) Part 1

Implement the photometric stereo technique described in section 2.2 of Forsyth and Ponce 2nd edition (or 5.4 in the 1st edition) and the lecture notes. Your program should have two parts:

- a)Read in the images and corresponding light source directions, and estimate the surface normals and albedo map.
- b)Reconstruct the depth map from the normals. You can first try the naive scanline-based shape by integration method described in the book.

Try this out on the synthetic dataset (synthetic data.mat) with three subsets of images:

- a)im1, im2, im4
- b)all four images (Most accurate)

(b) Part 2

Implement the specularity removal technique described in Beyond Lambert: Reconstructing Specular Surfaces Using Color (by Mallick, Zickler, Kriegman, and Belhumeur; CVPR 2005). Your program should input an RGB image and light source color and output the corresponding SUV image. Try this out first with the specular sphere images and then with the pear images. What to include in your report: For each specular sphere and pear images.

- a) The original image (in RGB colorspace).
- b) The recovered S channel of the image.
- c) The recovered diffuse part of the image-Use$G = (U^2 + V^2)^{\frac{1}{2}}$ to represent the diffuse part.

(c) Part 3

Combine parts 1 and 2 by running your photometric stereo code on the diffuse components of the specular sphere and pear images. For comparison, run your photometric stereo code on the original images (converted to grayscale) as well. You should notice erroneous "bumps" in the resulting reconstructions - the result of violating the Lambertian assumption. What to include in your report: For each specular sphere and pear images.

a) The recovered diffuse images

b) Estimated albedo map (original and diffuse images)

c) Estimated surface normals (original and diffuse images) by either showing:

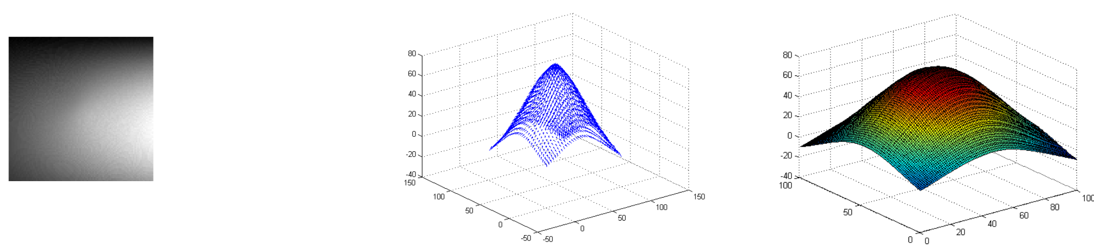
- Needle map (you will need to subsample the image to get a needle map which can be displayed. You can use the matlab functions meshgrid() and quiver3() or for python, meshgrid() of numpy and quiver() of mplot3d, which is part of matplotlib).
- Three images showing three components of surface normal.

d) A wireframe (original and diffuse images) of a depth map (you can use surf() in matlab or for python, plot surface from mplot3d, which is part of matplotlib).

Solutions:

(a) Part 1

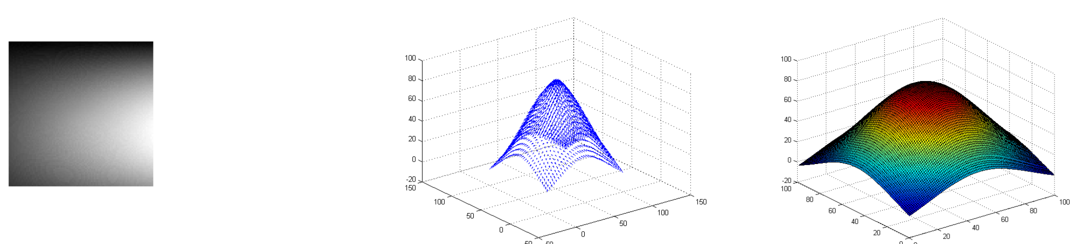
- im1, im2, im4



a b c

Figure 1: (a) albedo, (b) needle map (c) depthmap

- all 4 images



a b c

Figure 2: (a) albedo, (b) needle map (c) depth map

(b) Part 2

Pear:

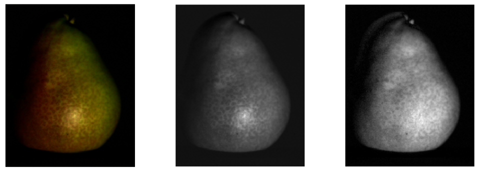


Figure 3: im 1: original, S channel, diffuse part

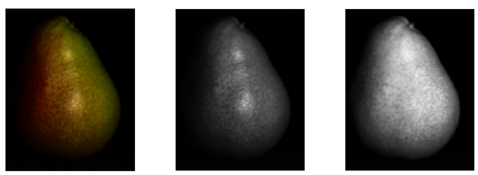


Figure 4: im 2: original, S channel, diffuse part

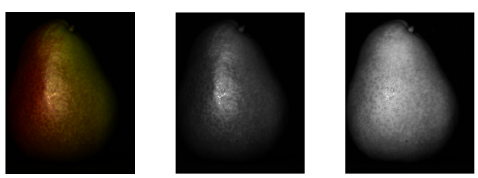


Figure 5: im 3: original, S channel, diffuse part

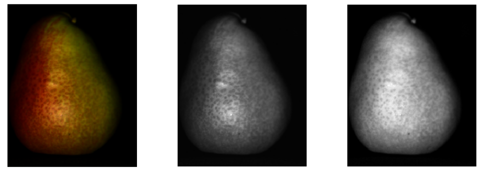


Figure 6: im 4: original, S channel, diffuse part

Sphere:

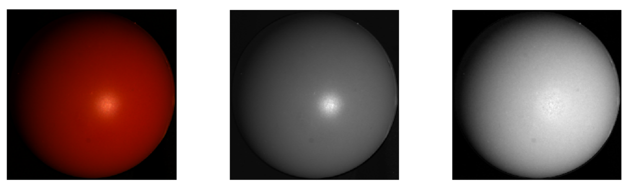


Figure 7: im 1: original, S channel, diffuse part

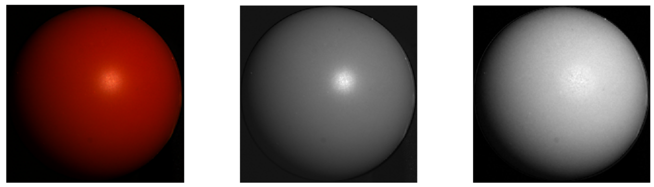


Figure 8: im 2: original, S channel, diffuse part

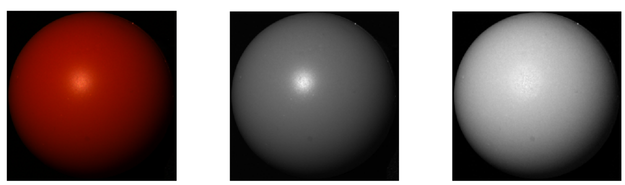


Figure 9: im 3: original, S channel, diffuse part

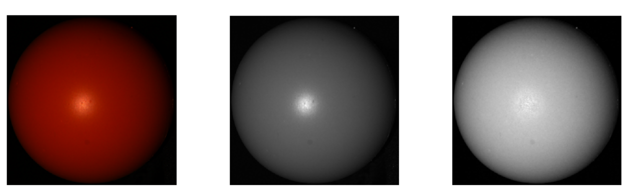


Figure 10: im 4: original, S channel, diffuse part

(c) Part 3

Pear:

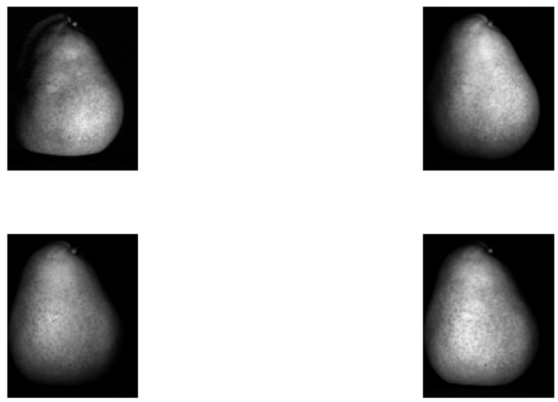
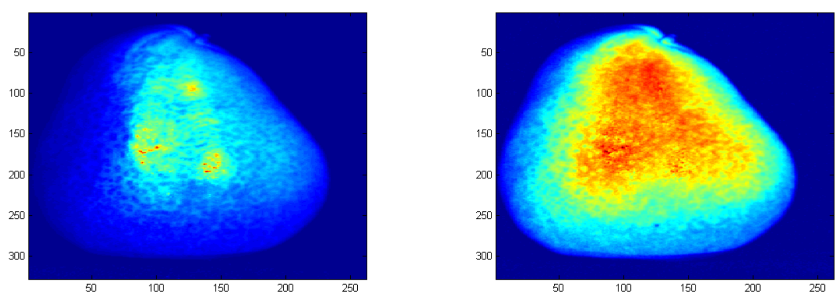
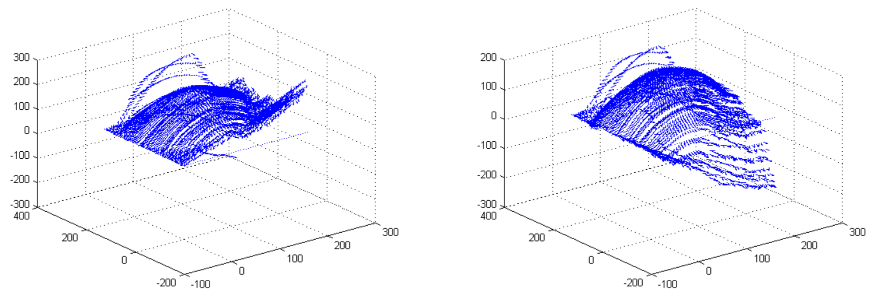


Figure 11: The recovered diffuse images



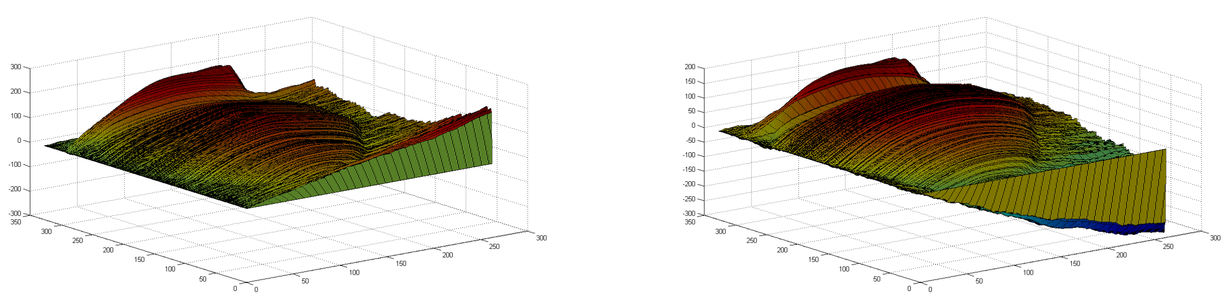
a b

Figure 12: (a) albedo original, (b) albedo diffuse image



a b

Figure 13: (a) needle map original, (b) needle map diffuse image



a b

Figure 14: (a) depth map original, (b) depth map diffuse image

Sphere:

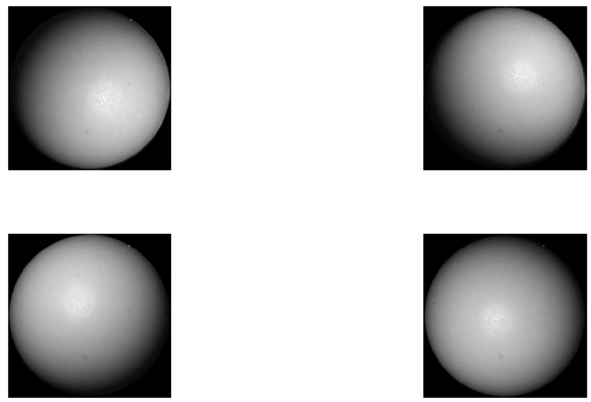
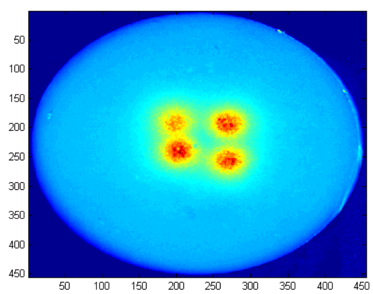
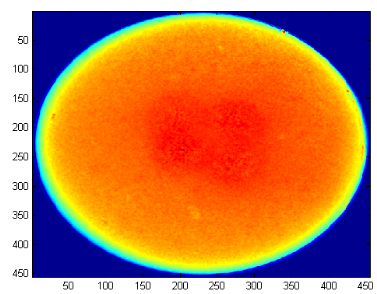


Figure 15: The recovered diffuse images

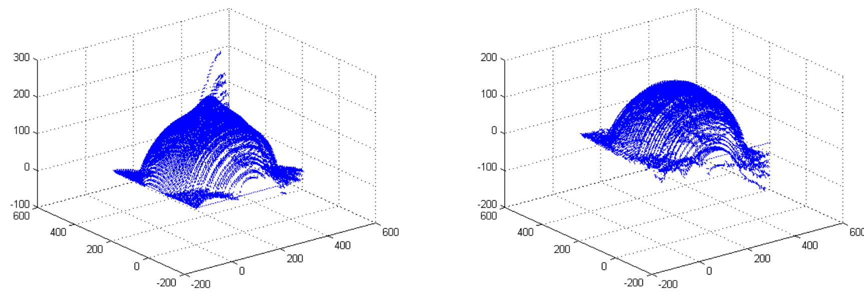


a



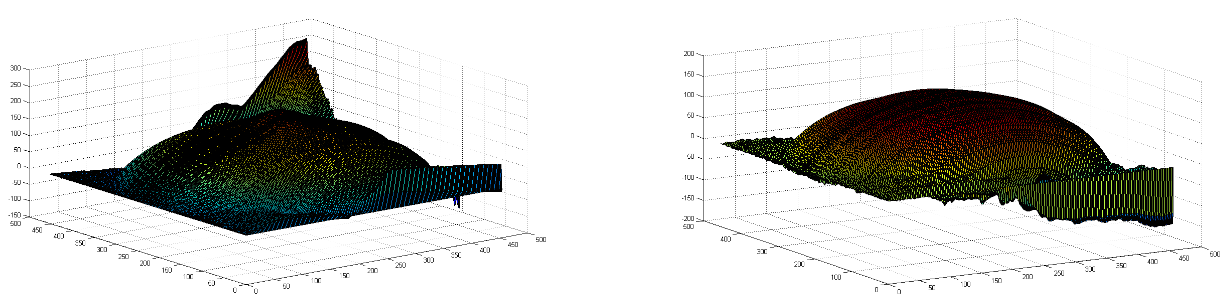
b

Figure 16: (a) albedo original, (b) albedo diffuse image



a b

Figure 17: (a) needle map original, (b) needle map diffuse image



a b

Figure 18: (a) depth map original, (b) depth map diffuse image

Listing 1: Codes of hw 2 pb 4 part1

```
clear;clc;
load('F:\synthetic_data');
[h1,w1]=size(im1);[h2,w2]=size(im2);[h3,w3]=size(im3);[h4,w4]=size(im4);

for i=1:h1
   for j=1:w1
        A=-[l1;l2;l4];
        Aplus=(inv(A'*A))*A';
        e=double([im1(i,j);im2(i,j);im4(i,j)]); %can be changed
        b=Aplus*e;
        albedo(i,j)=sqrt(dot(b,b'));
        normv(i,j,:,:,:)=b/albedo(i,j);
        p(i,j)=normv(i,j,1)/normv(i,j,3);          %calculate p&q
        q(i,j)=normv(i,j,2)/normv(i,j,3);

        u(i,j)=normv(i,j,1);
        v(i,j)=normv(i,j,2);
        w(i,j)=normv(i,j,3);


   end
end

height=zeros(h1,w1);
%estimate height map
height(1,1)=p(1,1);
for i=2:h1
      height(i,1)=height(i-1,1)+p(i,1);          % notice that the coordinate is not
end

for i=1:h1
      for j=2:w1
      height(i,j)=height(i,j-1)+q(i,j);
```

```
    end
end

xa=1:1:w1;ya=1:1:h1;
[x,y]=meshgrid(xa,ya);
z=-height;

%make quiver3 easier to be done
endd=100;
stride=3;
x1 = x(1:stride:endd,1:stride:endd);y1 = y(1:stride:endd,1:stride:endd);z1 = z(1:
u1 = u(1:stride:endd,1:stride:endd);v1 = v(1:stride:endd,1:stride:endd);w1 = w(1:

figure(1);
imshow(albedo,[]);
figure(2);
quiver3(x,y,z,u,v,w);
figure(3);
surf(x,y,z);
```

Listing 2: Codes of hw 2 pb 4 part2

```
clear;clc;
load('F:\specular-pear');
img1=rgb2gray(im1);img2=rgb2gray(im2);img3=rgb2gray(im3);img4=rgb2gray(im4);
[h1,w1]=size(img1);[h2,w2]=size(img2);[h3,w3]=size(img3);[h4,w4]=size(img4);
a=1

%calculate R
b=[1 0 0]';
b1=1;b2=0;b3=0;
cz=dot(c,c');
ct=c/cz;
a1=ct(1,:);a2=ct(2,:);a3=ct(3,:);
axis1=a2*b3-a3*b2;
axis2=a3*b1-a1*b3;
axis3=a1*b2-a2*b1;
theta=subspace(b,c);
o1=axis1/sqrt(axis1^2+axis2^2+axis3^2);
o2=axis2/sqrt(axis1^2+axis2^2+axis3^2);
o3=axis3/sqrt(axis1^2+axis2^2+axis3^2);
R1=[cos(theta)+o1^2*(1-cos(theta)) o1*o2*(1-cos(theta))-o3*sin(theta) o2*sin(theta)
R2=[o3*sin(theta)+o1*o2*(1-cos(theta)) cos(theta)+o2^2*(1-cos(theta)) -o1*sin(theta
R3=[-o2*sin(theta)+o1*o3*(1-cos(theta)) o1*sin(theta)+o2*o3*(1-cos(theta)) cos(thet
R=[R1;R2;R3];

%transfer into SUV
for i=1:h1
    for j=1:w1
        s1(i,j)=R1*[im1(i,j,1);im1(i,j,2);im1(i,j,3)];
        u1(i,j)=R2*[im1(i,j,1);im1(i,j,2);im1(i,j,3)];
        v1(i,j)=R3*[im1(i,j,1);im1(i,j,2);im1(i,j,3)];
        G1(i,j)=sqrt(u1(i,j)^2+v1(i,j)^2);
    end
end
```

```
for i=1:h1
    for j=1:w1
        s2(i,j)=R1*[im2(i,j,1);im2(i,j,2);im2(i,j,3)];
        u2(i,j)=R2*[im2(i,j,1);im2(i,j,2);im2(i,j,3)];
        v2(i,j)=R3*[im2(i,j,1);im2(i,j,2);im2(i,j,3)];
        G2(i,j)=sqrt(u2(i,j)^2+v2(i,j)^2);
    end
end
for i=1:h1
    for j=1:w1
        s3(i,j)=R1*[im3(i,j,1);im3(i,j,2);im3(i,j,3)];
        u3(i,j)=R2*[im3(i,j,1);im3(i,j,2);im3(i,j,3)];
        v3(i,j)=R3*[im3(i,j,1);im3(i,j,2);im3(i,j,3)];
        G3(i,j)=sqrt(u3(i,j)^2+v3(i,j)^2);
    end
end
for i=1:h1
    for j=1:w1
        s4(i,j)=R1*[im4(i,j,1);im4(i,j,2);im4(i,j,3)];
        u4(i,j)=R2*[im4(i,j,1);im4(i,j,2);im4(i,j,3)];
        v4(i,j)=R3*[im4(i,j,1);im4(i,j,2);im4(i,j,3)];
        G4(i,j)=sqrt(u4(i,j)^2+v4(i,j)^2);
    end
end

figure(1)
subplot(1,3,1);
imshow(im1/max(im1(:)));
subplot(1,3,2);
imshow(s1,[]);
subplot(1,3,3);
imshow(G1,[]);
```

Listing 3: Codes of hw 2 pb 4 part3

```
  clear;clc;
load('F:\specular-pear');
img1=rgb2gray(im1);img2=rgb2gray(im2);img3=rgb2gray(im3);img4=rgb2gray(im4);
[h1,w1]=size(img1);[h2,w2]=size(img2);[h3,w3]=size(img3);[h4,w4]=size(img4);


b=[1 0 0]';
b1=1;b2=0;b3=0;
a1=c(1,:);a2=c(2,:);a3=c(3,:);
axis1=a2*b3-a3*b2;
axis2=a3*b1-a1*b3;
axis3=a1*b2-a2*b1;
theta=subspace(b,c);
o1=axis1/sqrt(axis1^2+axis2^2+axis3^2);
o2=axis2/sqrt(axis1^2+axis2^2+axis3^2);
o3=axis3/sqrt(axis1^2+axis2^2+axis3^2);
R1=[cos(theta)+o1^2*(1-cos(theta)) o1*o2*(1-cos(theta))-o3*sin(theta) o2*sin(theta)
R2=[o3*sin(theta)+o1*o2*(1-cos(theta)) cos(theta)+o2^2*(1-cos(theta)) -o1*sin(theta
R3=[-o2*sin(theta)+o1*o3*(1-cos(theta)) o1*sin(theta)+o2*o3*(1-cos(theta)) cos(thet
R=[R1;R2;R3];

for i=1:h1
    for j=1:w1
        s(i,j)=R1*[im1(i,j,1);im1(i,j,2);im1(i,j,3)];
        u(i,j)=R2*[im1(i,j,1);im1(i,j,2);im1(i,j,3)];
        v(i,j)=R3*[im1(i,j,1);im1(i,j,2);im1(i,j,3)];
        G1(i,j)=sqrt(u(i,j)^2+v(i,j)^2);
    end
end

for i=1:h1
    for j=1:w1
        s(i,j)=R1*[im2(i,j,1);im2(i,j,2);im2(i,j,3)];
```

```
            u(i,j)=R2*[im2(i,j,1);im2(i,j,2);im2(i,j,3)];
            v(i,j)=R3*[im2(i,j,1);im2(i,j,2);im2(i,j,3)];
            G2(i,j)=sqrt(u(i,j)^2+v(i,j)^2);
      end
end
for i=1:h1
      for j=1:w1
            s(i,j)=R1*[im3(i,j,1);im3(i,j,2);im3(i,j,3)];
            u(i,j)=R2*[im3(i,j,1);im3(i,j,2);im3(i,j,3)];
            v(i,j)=R3*[im3(i,j,1);im3(i,j,2);im3(i,j,3)];
            G3(i,j)=sqrt(u(i,j)^2+v(i,j)^2);
      end
end
for i=1:h1
      for j=1:w1
            s(i,j)=R1*[im4(i,j,1);im4(i,j,2);im4(i,j,3)];
            u(i,j)=R2*[im4(i,j,1);im4(i,j,2);im4(i,j,3)];
            v(i,j)=R3*[im4(i,j,1);im4(i,j,2);im4(i,j,3)];
            G4(i,j)=sqrt(u(i,j)^2+v(i,j)^2);
      end
end

for i=1:h1
      for j=1:w1
            A=-[l1;l2;l3;l4];
            Aplus=(inv(A'*A))*A';
            e=[G1(i,j);G2(i,j);G3(i,j);G4(i,j)];
            b=Aplus*e;
            albedo(i,j)=sqrt(dot(b,b'));
            normv(i,j,:,:,:)=b/albedo(i,j);
            p(i,j)=normv(i,j,1)/normv(i,j,3);
            q(i,j)=normv(i,j,2)/normv(i,j,3);

            u(i,j)=normv(i,j,1);
```

```
            v(i,j)=normv(i,j,2);
            w(i,j)=normv(i,j,3);
        end
end

%set marginal norm to straight up
for i=1:h1
    normv(i,1,:,:,:)=[0;0;0.0001];
end

for i=1:w1
    normv(1,i,:,:,:)=[0;0;0.0001];
end

for i=1:h1
    for j=1:w1
        p(i,j)=normv(i,j,1)/normv(i,j,3);
        q(i,j)=normv(i,j,2)/normv(i,j,3);
        u(i,j)=normv(i,j,1);
        v(i,j)=normv(i,j,2);
        w(i,j)=normv(i,j,3);
    end
end


height=zeros(h1,w1);


height(1,1)=q(1,1);
for i=2:h1
    height(i,1)=height(i-1,1)+q(i,1);        % notice that the coordinate is not re
end

for i=1:h1
```

```
        for j=2:w1
        height(i,j)=height(i,j-1)+p(i,j);
        end
end

xa=1:1:w1;ya=1:1:h1;
[x,y]=meshgrid(xa,ya);
z=height;

enddx=h2;
enddy=w2;
stride=5;
x1 = x(1:stride:enddx,1:stride:enddy);y1 = y(1:stride:enddx,1:stride:enddy);z1 = z(
u1 = u(1:stride:enddx,1:stride:enddy);v1 = v(1:stride:enddx,1:stride:enddy);w11 = w

figure(1);
subplot(2,2,1);
imshow(G1,[]);
subplot(2,2,2);
imshow(G2,[]);
subplot(2,2,3);
imshow(G3,[]);
subplot(2,2,4);
imshow(G4,[]);

figure(2);
imagesc(albedo);
figure(3);
quiver3(x1,y1,z1,u1,v1,w11);
figure(4);
surf(x,y,z);
```

Submitted by Mingxuan Wang on November 3, 2014.