Assignment 7

CS283, Computer Vision
Harvard University

Due Monday, Oct 30, at 5:00pm

The purpose of this homework is to reinforce the topics of radiometry, reflectance, and photometric stereo. Forsyth and Ponce's chapter on Radiometry will be helpful for this assignment.

1. (*10 points*) For Lambertian surfaces, the BRDF is a constant function of the input and output directions. For such a material, we often describe the reflectance in terms of its *albedo*, which is given the symbol ρ. For a Lambertian surface, the BRDF and albedo are related by $f_r(\hat{\mathbf{v}}_i, \hat{\mathbf{v}}_o) = \rho/\pi$. Using conservation of energy, prove that $0 \leq \rho \leq 1$.

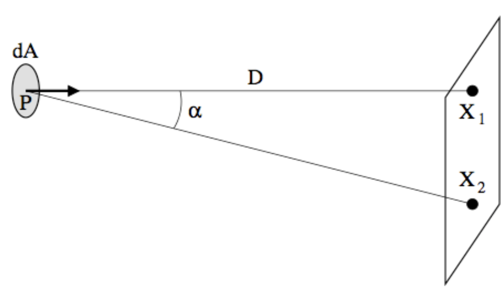
2. (*10 points*) Obtain the file `bunny.mat` from the data folder of this assignment and load it into Matlab. There is a single variable in this file; the variable `N` is an $h \times w \times 3$ array of surface normals. `N(i,j,1)`, `N(i,j,2)`, and `N(i,j,3)` are the x, y, and z components of the surface normal at the ij^{th} surface point, as observed by an orthographic camera with view direction $(0, 0, 1)$. *Submit all code for this question.*

   (a) Use `quiver` to display the x-y components of the normal field and print the result. (See the example in *Hints and Information* below.)

   (b) Compute and display the radiance emitted from each point assuming Lambertian reflectance and constant albedo (equal to one), with a distant point light source in direction $\hat{\mathbf{s}} = (0,\ 0,\ 1)$, by typing `imshow(N(:,:,3))`. Explain why this works.

   (c) Compute, display and print the emitted radiance for three different light source directions which are rotated i) 45° up, ii) 45° right, and iii) 75° right from the frontal direction $\hat{\mathbf{s}} = (0,\ 0,\ 1)$. Can you spot errors in the field of surface normals? What are the illumination effects being ignored in this calculation of scene radiance?

3. (*10 points*) A small Lambertian source dA is centered at P and emits radiance L. The orientation of this patch is the same as that of a plane containing two points, X_1 and X_2. The point X_1 is the point on this plane that is closest to P, and the distance from P to X_1 is D as shown.



   (a) Calculate the solid angle subtended by dA at points X_1 and X_2.

   (b) Calculate the irradiance E incident on the plane at points X_1 and X_2, and calculate the ratio $E(X_1)/E(X_2)$.

4. (*20 points*) The data folder of this assignment contains a set of seven photometric stereo images along with a MAT file containing the light source vectors. You can load the images into Matlab and convert them to double, grayscale format using `rgb2gray` and `im2double`.

(a) Estimate the surface normal and grayscale albedo for each pixel. Submit your Matlab code as well as the results (using `imshow` for the albedo values and `quiver` for the surface normals.)

(b) Note the poor estimates of the albedo (and surface normal) in the area surrounding the nostrils. What is the source of this error? Describe one method for finding a better estimate of this information from these seven images.

(c) Use the recovered surface information to predict what the person would look like (in grayscale) if illuminated from direction $\hat{\mathbf{s}} = (0.58, -0.58, -0.58)$ and from direction $\hat{\mathbf{s}} = (-0.58, -0.58, -0.58)$. Submit your results and your code.

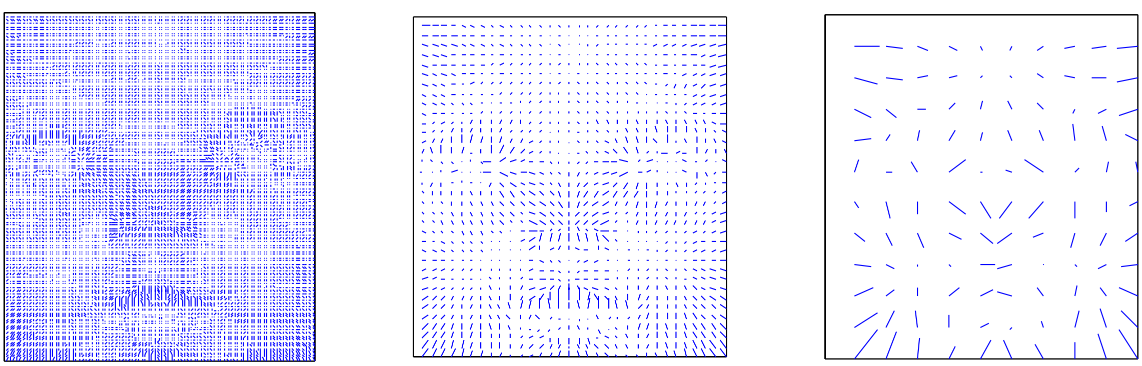
(d) The function `integrate_frankot.m` in the .MLX file can be used to recover a surface $z(x, y)$ from your surface normals $\hat{\mathbf{n}}(x, y)$, and the surface can be displayed in Matlab. Display and submit two views of the recovered shape.

Hints and Information

- The following code demonstrates one way to use the `quiver` function.

```
% vect_x is an HxW array of x-components
% vect_y is an HxW array of y-components
figure;            % create a new figure
quiver(vect_x,vect_y,'.');
axis image;  % set unit aspect ratio between x and y axes
axis ij;     % place origin in top-left corner (like an image)
```

Note that the vector components should be displayed at a suitable resolution. For example, the figure below shows the x-y components of a field of surface normals at three different resolutions. The plot in the middle provides the best summary of the surface shape. The resolution can be adjusted by plotting only every n^{th} element as in `quiver(vect_x(1:n:end,1:n:end),vect_y(1:n:end,1:n:end),'.')`.



- When computing the output radiance from a field of surface normals $\hat{\mathbf{n}}(x, y)$ illuminated from direction $\hat{\mathbf{s}}$, clip negative values using $L = \max(0, \hat{\mathbf{n}} \cdot \hat{\mathbf{s}})$.

- We are commonly faced with a situation in which we need to evaluate the integral over the hemisphere of the function $\cos\theta$, where $0 \leq \theta\pi$ is the angular distance between the radial line through a point on the hemisphere and the radial line through its apex.

$$\int_{\Omega} \cos\theta d\omega$$

The first step is to define a parameterization of the hemisphere using spherical coordinates, and the second is to write an element of the solid angle $d\omega$ in terms of this parameterization. As shown in almost any introductory calculus book (and Forsyth & Ponce), we can write this integral as

$$\int_0^{2\pi}\int_0^{\pi/2} \cos\theta \sin\theta d\theta d\phi,$$

which is straightforward to evaluate using the appropriate trigonometric identity.

- The following code demonstrates one way to display a surface in Matlab.

```
% Z is an HxW array of surface depths
figure;
s=surf(-Z);  % use the '-' sign since our Z-axis points down
             % output s is a 'handle' for the surface plot

% adjust axis
axis image; axis off;

% change surface properties using SET
set(s,'facecolor',[.5 .5 .5],'edgecolor','none');

% add a light to the axis for visual effect
l=camlight;

% enable mouse-guided rotation
rotate3d on
```

When making a figure in Matlab, each object (figure, axe, line, image, surface, etc.) is given a numerical ID or *handle* that can be used to adjust its properties. You can use these, for example, to turn grid lines on and off, change the color of the axis, change the color of a line in a graph, change the shininess of a plotted surface, and so on. The handle `s` in the previous example can be used to change the properties of the surface. To see a list of its current properties type `get(s)`, and to change these properties use `set(s)` as in the example above. Check the Matlab online documentation to find out what these properties mean. (Type `helpdesk` on your local machine, or search for "Matlab helpdesk" in a web browser and navigate to Handle Graphics Object Properties.)