6.801/6.866 Machine Vision

Problem set 1

Assigned: 09/12/02
Due: 09/24/02 at 3pm

Cameras and Images formation

Problem 1

a) Show that the image of an ellipse in a plane, not necessarily one parallel to the image plane, is also an ellipse. Assume perspective projection.
b) Show that the image of a line in space is a line in the image. Assume perspective projection.

Problem 2

Exercise **2-6** from Robot Vision (Horn).

Problem 3

Exercise **1.4** from Computer Vision, A Modern Approach (Forsyth and Ponce)

Camera Geometry and Calibration

Problem 4

Exercise **2.10** from Computer Vision, A Modern Approach (Forsyth and Ponce)

Problem 5

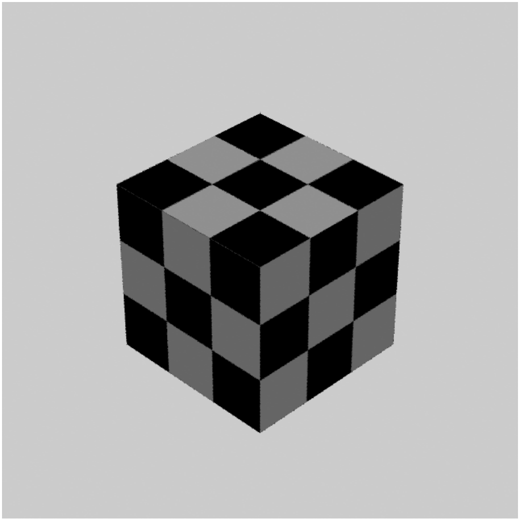
Exercise **2.12** from Computer Vision, A Modern Approach (Forsyth and Ponce)

Problem 6 – MATLAB

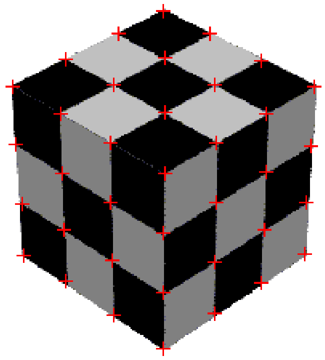
The goal of this problem is to implement a linear calibration algorithm in Matlab based on the method described in section 3.2 (Forsyth). From the following image, we want to extract the intrinsic (focal length and center of image) and extrinsic (rotation and translation) parameters of the camera used to grab this image. We assume no radial distortion.

A usual way to calibrate a camera is to take a picture of a calibration object, find 2D features in the picture and derive the calibration from the 2D features and the corresponding position in 3D. In our case, we use as calibration object a cube 5m wide with a checkerboard pattern on it (see figure 1a). We search in the image (of size 640x640) for the 2D features corresponding to corners of the checkerboard. Figure 1b shows these features.

Since we know its exact size (5m), we can find the exact 3D position of each 2D feature relative to the bottom corner of the cube. This process of finding correspondences is simple but time consuming. We did this part of the work for you. `Features2D.mat` and `Features3D.mat` contain the 2D corner features and the corresponding 3D positions.



1(a) Original image



1(b) Detected features

Your task is to write a Matlab function which takes these two lists as input and returns the calibration parameters as output. Your function should have the following syntax:

```
function [f,u0,v0,R,t] = calibrate(f2D, f3D)
```

where f is the focal length (in meters), u0 and v0 are the center of the image (in pixels), and [R t] is the relative rigid transformation between the bottom corner of the cube and the camera. The size of camera CCD is 1 inch by 1 inch and its pixels are square (i.e. $\alpha=\beta$).

You should submit your file `calibration-[last_name].m` by email to 6.801-submit@ai.mit.edu . In your problem set solution, write the values (f, u0, v0, R and t) you found using the cube data set.

Radiometry and Reflectance Map

Problem 7

Exercise **10-11** from Robot Vision (Horn)

Problem 8 – MATLAB

Based on the photometric stereo method described in section 5.4 (Forsyth), implement a Matlab function which takes images as input and returns the depth, normal and albedo for each point. Your function should have the following structure:

```
function [Nx, Ny, Nz, a] =
                      photostereo(I1, I2, I3, I4, I5, I6)
```

where Nx, Ny and Nz are the components of the normal vector at each pixel, and a is the albedo for each pixel.

To test your algorithm, use the test images `Ellipse1.bmp`, `Ellipse2.bmp`, `Ellipse3.bmp`, `Ellipse4.bmp`, `Ellipse5.bmp` and `Ellipse6.bmp`, which are also shown on the following page. The position of the spotlight for each image is described in the following table:

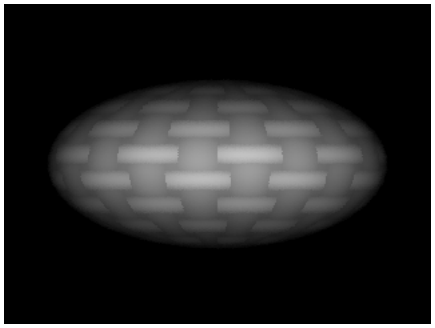
	X	Y	Z
`Ellipse1.bmp`	0	0	40
`Ellipse2.bmp`	20	5	40
`Ellipse3.bmp`	-5	15	40
`Ellipse4.bmp`	-10	-10	40
`Ellipse5.bmp`	5	-10	40
`Ellipse6.bmp`	5	20	40

In this scene the origin is at the center of the ellipsoid, the Y axis points upward, X to the right, and Z out from the screen; orthographic projection is used. The ellipsoid is 2 x 2 x 1 meter high. Spotlight positions in the table above are given in meters.

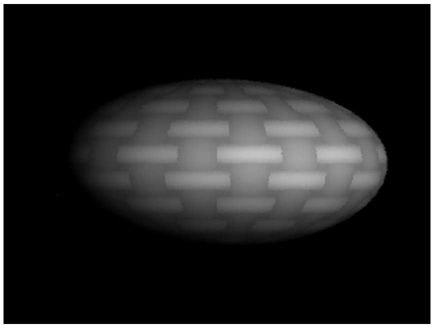
You should submit your file `photostereo-[last_name].m` by email to 6.801-submit@ai.mit.edu . In your problem set solution, insert a printed copy of the normal vectors graph (using `quiver()`) and the albedo image (using `imshow()`) you found using the ellipse data set.

Problem 9 (optional)

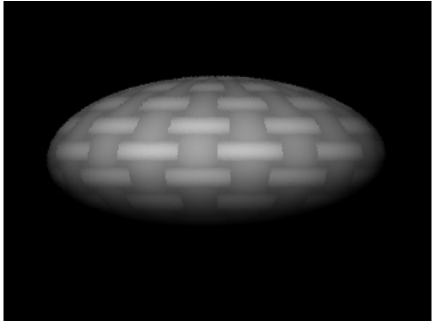
Exercise **4.4** from Computer Vision, A Modern Approach (Forsyth and Ponce)



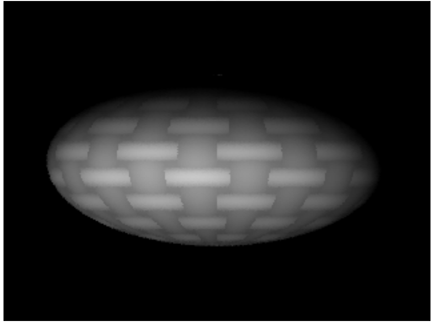
Ellipse1



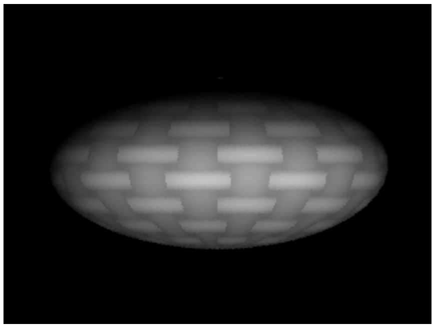
Ellipse2



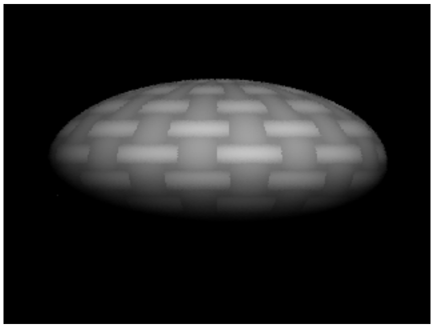
Ellipse3



Ellipse4



Ellipse5



Ellipse6