Homework 1

Peiyi Zheng

Written Assignment:

1.

a. Points in a circular disk that lies in a plane parallel to the image plane satisfy the following equations:

$$z = z_0 \qquad (x - x_0)^2 + (y - y_0)^2 = r^2$$

Where (x_0, y_0, z_0) is the center of the disk and r is the radius of the disk.

As we know that the mapping between a point in disk (x, y, z_0) to the image point (x_i, y_i) is:

$$\frac{x_{i'}}{f} = \frac{x}{z_0} \quad \text{and} \quad \frac{y_{i'}}{f} = \frac{y}{z_0}$$

Therefore $x = z_0 \frac{x_i}{f}$, $y = z_0 \frac{y_i}{f}$. Put them into the original equation we can get:

$$(z_0 \frac{x_i}{f} - x_0)^2 + (z_0 \frac{y_i}{f} - y_0)^2 = r^2$$

Which equals to:

$$(x_i - \frac{x_0 f}{z_0})^2 + (y_i - \frac{y_0 f}{z_0})^2 = (\frac{rf}{z_0})^2$$

When $z_0 \neq 0$, the shape of the image of the disk is a circle.

b. Since $\frac{Area_i}{Area_o} = \frac{f^2}{{z_0}^2}$, hence $Area_i = \frac{f^2}{{z_0}^2} Area_o$. If the distance z_0 is double, then $\frac{Area_{i2}}{Area_i} = \frac{\frac{f^2}{(2z_0)^2}}{\frac{f^2}{{z_0}^2}} = \frac{1}{4}$.

As a result the area of the image of the circular disk becomes $0.25mm^2$.

c. We can represent a sphere in the space with the following equation:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$$

Points in the image plane satisfy:

$$\frac{x_{i'}}{f} = \frac{x}{z_0} \;, \quad \frac{y_{i'}}{f} = \frac{y}{z_0} \text{ and } z_0 = f$$

So we put them into the original equation and get:

$$(x_i - \frac{x_0 f}{z_0})^2 + (y_i - \frac{y_0 f}{z_0})^2 = (\frac{f}{z_0})^2 [r^2 - (f - z_0)^2]$$

When $r^2 - (f - z_0)^2 < 0$, there is no shape in the image.

When $r^2 - (f - z_0)^2 = 0$, there is only one point in the image.

When $r^2 - (f - z_0)^2 > 0$, the shape is a circle.

2. Obviously the lines connect to the vanishing points must pass the pinhole, which is $(0,0,0)$. Therefore D = 0. Suppose the coordinate of the vanishing points are (x_v, y_v, f), the direction vector from vanishing points to pinhole is $(-x_v, -y_v, -f)$. This vector is perpendicular to the normal vector of the plane, which is (A, B, C). Therefore $(-x_v A) + (-y_v B) + (-fC) = 0$. So the vanishing points lie on the line $Ax + By + Cf = 0$.

Programming Assignment:

How to run:

There are two Python scripts, driver1.py is for problem one, it will run all the functions in sequence and generate result for each functions. Similarly, driver2.py run the functions with all images. The result will be written as jpg file with corresponding name.

1.

a. For this problem, I check every pixels in the image, if it is lower than threshold then I set it to 0, otherwise it becomes 1.

b. In the first pass, the program will iterate all the pixels, if it is not a background point then check the three pixels near it to determine. When the program finds a pixel can be assigned to different labels, set it to one of them and put this two labels in a same set using union-find algorithm. In the second pass, merge equivalence set. At the end the program will map all the labels in a new order from 1 to n, n is the number of different labels.

c. The ways to compute attributes in this problem have been mentioned in the slides. The only thing we need to figure out is how to compute a from a', and also b and c. Taking a as an example:

$$a = \sum\sum(i-\bar{y})^2 b_{ij} = \sum\sum(i^2 - 2i\bar{y} + \bar{y}^2) b_{ij} = \sum\sum i^2\, b_{ij} - 2\sum\sum i\bar{y}\, b_{ij} + \bar{y}^2 \sum\sum b_{ij}$$

Which equals to:

$$a' - 2\bar{y}^2 A + \bar{y}^2 A = a' - \bar{y}^2 A$$

Since all the values are known, we can generate a from a'. b and c is also computed with similar strategy. With a, b and c we can compute the rest of the attributes.

Last but not least, I add an attribute “area”.

d. In the attributes generated from p3, I consider the area and the roundness to be important in determining whether two objects are similar or not. So I compare two objects with the following comparison method:

$$0.7 \cdot abs\left(\frac{query(area) - data_i(area)}{data_i(area)}\right) + 0.3 \cdot abs\left(\frac{query(roundness) - data_i(roundness)}{data_i(roundness)}\right)$$

The formula consider the difference between area and roundness, and also give them different weight to reflect their priority. If the above formula return a value less than 0.3, than I consider two objects is likely to matched. To be honest, in two test image, one can detect two objects and another detects a correct object and a wrong one.

2.

a. In problem a, I use the following laplacian operator to detect edges because it is easy to compute and has a good result:

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

Before convolution, I smooth the image to denoise.

b. For problem b, I implement the hough transform with resolution 2L * 180, L is the diagonal of image. This resolution corresponds to possible value range [-L, L] * [-90, 90]. For each point on the edge, add one to all possible lines that pass it. I use 25 for simple images and 55 for the complex one. Larger threshold helps me remove background noise in complex image.

c. In problem c, I just filter the possible lines whose voting number are lower than threshold. I also use different threshold for simple and complex images, but the value is just a little different.

d. When the program gets the array of strong lines from p7, it will iterate the edge image and check whether the points on the edges are also on the line, if they do, then add them to another array. Finally the program draws all the points on the edges which will form the line segments.