RBE 549: Homework Assignment 1

Aditya Mehrotra

Problem 1

Let us consider a thin lens. It has a focal point f at point C. Considering $\vec{x_0}$ as the object of height x_0, let us flip the ΔAFG to the other side of the origin (O). That gives us a ΔAFB.

Now if we consider ΔACB and ΔFCE,

$\angle ACB = \angle FCE$

Also since AB is parallel to FE,

$\angle ABC = \angle EFC$ and

$\angle BAC = \angle CEF$

Hence, the two triangles are similar.

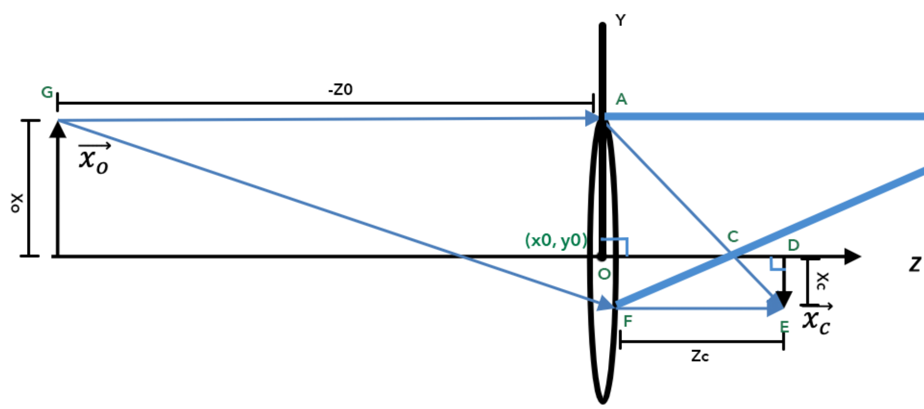

Figure 1: Two cartesian reference frames.

Similarly when we consider ΔOAC and ΔCED,

$\angle AOC = \angle ECD = 90°$,

$\angle ACO = \angle DCE$,

and furhter, $\angle COA = \angle CED$.

Thus, $\Delta OAC \simeq \Delta CED$.

Now, we know that,

AB = AG = $-z_0$

FE $= z_c$

And, $AB/FE = AC/CE$

$\Rightarrow -z_0/z_c = AC/CE$

Also,

$OC/CD = AC/CE$

$\Rightarrow f/z_c - f = AC/CE$

$\Rightarrow \frac{-z_0}{z_c} = \frac{f}{z_c - f}$

$\Rightarrow -z_0.(z_c - f) = f.z_c$

$\Rightarrow -z_0.z_c + z_0.f = f.z_c$

$\Rightarrow -z_0.z_c = f.(-z_0 + z_c)$

$\Rightarrow \frac{1}{f} = \frac{-z_0 + z_c}{-z_0.z_c}$

$\therefore \frac{1}{f} = \frac{1}{-z_0} + \frac{1}{z_c}$

Problem 2

a. Diameter of the eye $(f) = 0.024\,m = 2.4\,cm = 24\,mm$

Number of receptors $(N) = 150{,}000{,}000$

Considering the receptors cover an entire hemisphere, Surface area of a hemisphere $(A) = \pi f^2/2 = 904.8\ mm^2$

Number of receptors$/mm^2 = \frac{N}{A} = \frac{150{,}000{,}000}{904.8} = 165{,}786.4/mm^2$

b. Diameter of Mars $(y_0) = 8{,}000 km = 8 \times 10^6\ m$

Distance from Earth $(z_0) = 225{,}000{,}000\ km = 225 \times 10^9\ m$

We know that,

$\frac{1}{f} = \frac{1}{-z_0} + \frac{1}{z_c}$

On substituting the values,

$\frac{1}{0.024} = \frac{1}{-(225 \times 10^9)} + \frac{1}{z_c}$

$\Rightarrow z_c = 0.0239\,m$

Magnification factor $M = \frac{y_0}{y_c} = \frac{-z_0}{z_c}$

$\Rightarrow \frac{8 \times 10^6}{y_c} = \frac{-225 \times 10^9}{0.024}$

$\Rightarrow y_c = \frac{8 \times 10^6 \times 0.024}{-225 \times 10^9}$

$\Rightarrow y_c = 8.53 \times 10^{-7}\ m\ (inverted) = 8.53 \times 10^{-4}\ mm$

Now, Area of the image $(A_c) = \pi {y_c}^2/4 = 5.71 \times 10^{-7}\ mm^2$

Number of receptors on which the image falls $= \frac{N}{A} \times A_c$

$\Rightarrow 165{,}786.4 \times 5.71 \times 10^{-7} = 0.0948$

Thus, the image of Mars falls only on $0.095th$ of a receptor.

Problem 3

Since at the brightness values of the object and background follow a Gaussian Distribution, they might appear to overlap as shown in Fig. 2. In this overlap region, the probabilities of a selected pixel to be of the object or the background can be equated.

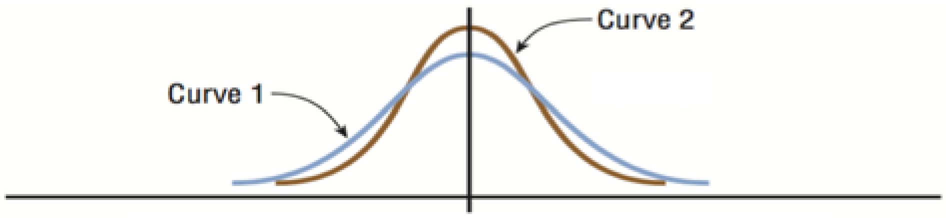

Figure 2: The brightness of the background and objects will be widely separated. In the ideal case, the threshold T can be anywhere between the two peaks. In most images, the intensities will overlap, resulting in histograms.

The value of

Given $\sigma_o < \sigma_b$,

For any value of x such that,

$P_o(x) < P_b(x)$

$\Rightarrow \frac{1}{\sqrt{2\pi}\sigma_o} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma_o^2}} > \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma_b^2}}$

$\Rightarrow \frac{\sigma_b}{\sigma_o} > \frac{e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma_b^2}}}{e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma_o^2}}}$

$\Rightarrow \frac{\sigma_b}{\sigma_o} > e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma_b^2} + \frac{1}{2}\frac{(x-\mu)^2}{\sigma_o^2}}$

$\Rightarrow \frac{\sigma_b}{\sigma_o} > e^{-\frac{1}{2}\frac{(x-\mu)^2(\sigma_b^2 - \sigma_o^2)}{\sigma_b^2\sigma_o^2}}$

On taking a natural logarithm on both sides,

$\Rightarrow \ln\frac{\sigma_b}{\sigma_o} > \frac{1}{2}\frac{(x-\mu)^2(\sigma_b^2 - \sigma_o^2)}{\sigma_b^2\sigma_o^2}$

$\Rightarrow 2(\ln\sigma_b - \ln\sigma_o) > \frac{(x-\mu)^2(\sigma_b^2 - \sigma_o^2)}{\sigma_b^2\sigma_o^2}$

$\Rightarrow (x-\mu)^2 < \sigma_b^2\sigma_o^2 \frac{2(\ln\sigma_b - \ln\sigma_o)}{\sigma_b^2 - \sigma_o^2}$

Taking square root on both sides gives us,

$\Rightarrow |x-\mu| < \sigma_o\sigma_b\sqrt{\frac{2(\ln\sigma_b - \ln\sigma_o)}{\sigma_b^2 - \sigma_o^2}}$

Now for any threshold values T $\geq |x - \mu|$ will be considered as the background whereas, for any value of T $< |x - \mu|$, the pixel will be considered as the object. For $x = \mu$ as well, the pixel will be considered to belong to the object.

Problem 4

In Cartesian coordinates, a homogeneous transformation matrix is given as, $\begin{bmatrix} \mathrm{R} & \vec{T} \\ 0 & 1 \end{bmatrix}$

Rotation of a vector can be inverted by simply using the transpose of the rotation matrix ($\because \mathrm{R}^{-1} = R^T$). Since the translation is preceded by a rotation, its rotation must also be inverted first when performing an inverse of a translation. Thus, inverse of the above matrix gives us = $\begin{bmatrix} \mathrm{R}^T & -R^T.\vec{T} \\ 0 & 1 \end{bmatrix}$

Problem 5

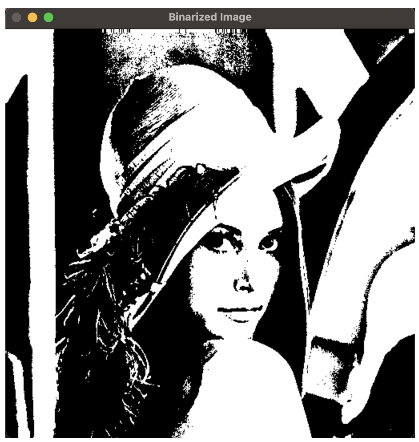

Figure 3: Thresholding Image Result

a

```
#include "opencv2/imgproc.hpp"
#include "opencv2/imgcodecs.hpp"
#include "opencv2/highgui.hpp"
#include iostream
#include vector
using namespace cv;
using namespace std;
int main(int argc, char const *argv[])
{
Mat inputImage = imread("../Data/lenna.png", IMREAD_GRAYSCALE);
Mat binaryOutputImage;
```

```
vectorint oneDImage;
oneDImage = inputImage.reshape(0,1);
cout ”1D Image size: ” oneDImage.size()endl;
sort(oneDImage.begin(), oneDImage.end());
int median = oneDImage.size()/2;
cout ”Median Pixel value: ”oneDImage[median]endl;
threshold(inputImage, binaryOutputImage, oneDImage[median], 255, THRESH_BINARY);
imshow(”Binarized Image”, binaryOutputImage);
waitKey();
return 0;
}
```

b Figure 4 shows the result of adaptive thresholding of an image.

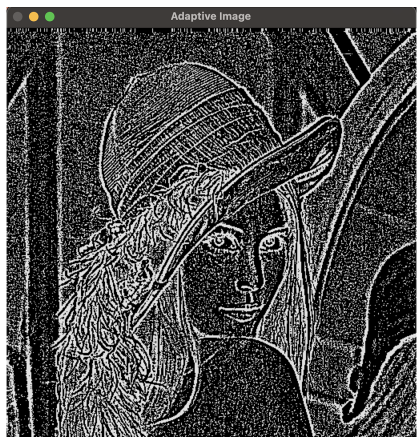

Figure 4: Adaptive Thresholding Image Result

```
#include ”opencv2/imgproc.hpp”
#include ”opencv2/imgcodecs.hpp”
#include ”opencv2/highgui.hpp”
#include ¡iostream¿
using namespace cv;
using namespace std;
int main(int argc, char const *argv[])
{
Mat inputImage = imread(”../Data/lenna.png”, IMREAD_GRAYSCALE);
Mat outputImage;
adaptiveThreshold(inputImage, outputImage, 225, ADAPTIVE_THRESH_MEAN_C,
THRESH_BINARY_INV, 5, 2);
```

```
imshow("Display", outputImage);
waitKey(0);
return 0;
}
```