Activity 5: Enhancement by Histogram Manipulation

DELA CRUZ, MARY NATHALIE G
2015-09114

What is the objective? We want to improve the quality of a dark image.

Why do we need to do it? Information is lost in the dark parts of the image.

How to achieve it? A histogram can show the intensity distribution of a grayscale image. To get a good contrast of the image, we manipulated the image histogram.

Figure 1: Original Image

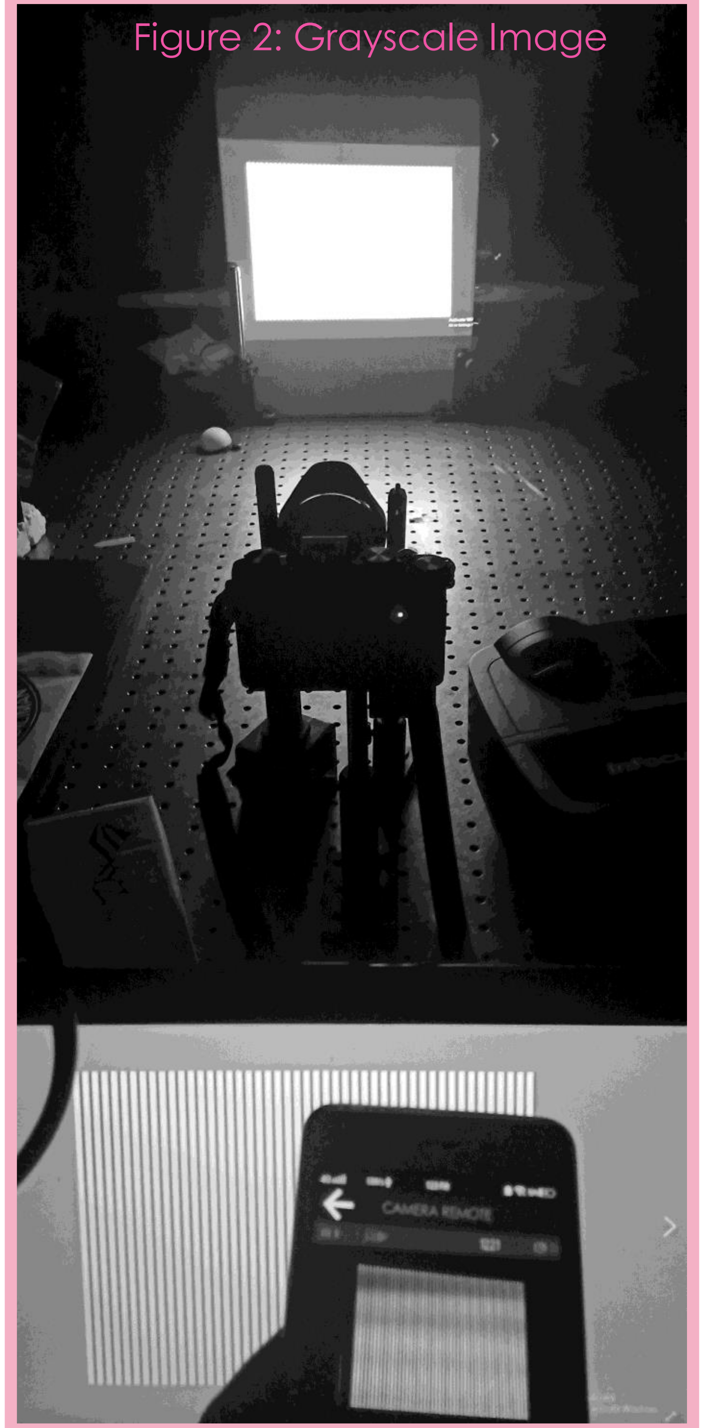


Figure 2: Grayscale Image

Image we wanted to enhance was first converted to grayscale.

```
im = imread('1.jpg');
im = rgb2gray(im);
im_min = min(im); im_max = max(im);
```

Note: Contrast sketching was not performed for this image because max and min grayscale of the image are 0 and 255, respectively.

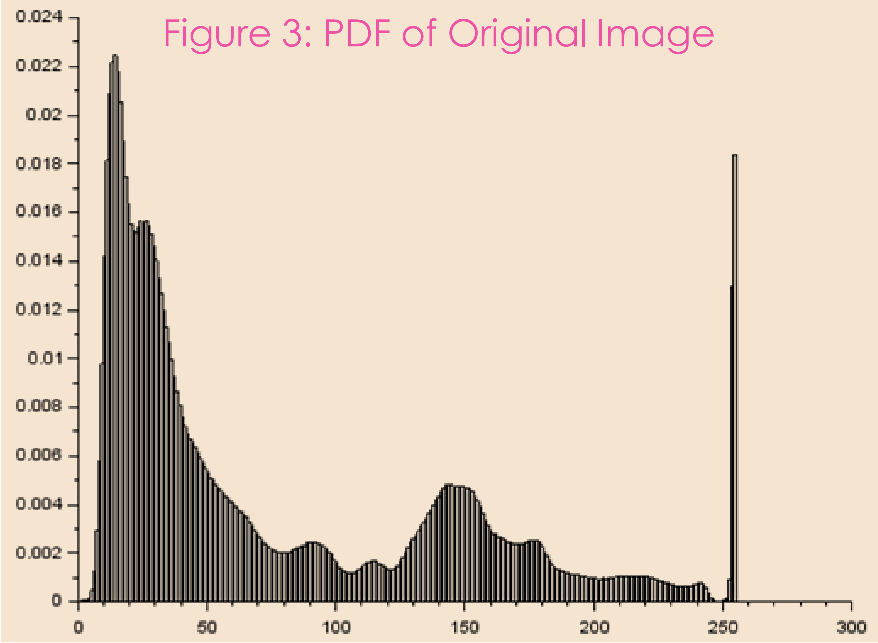
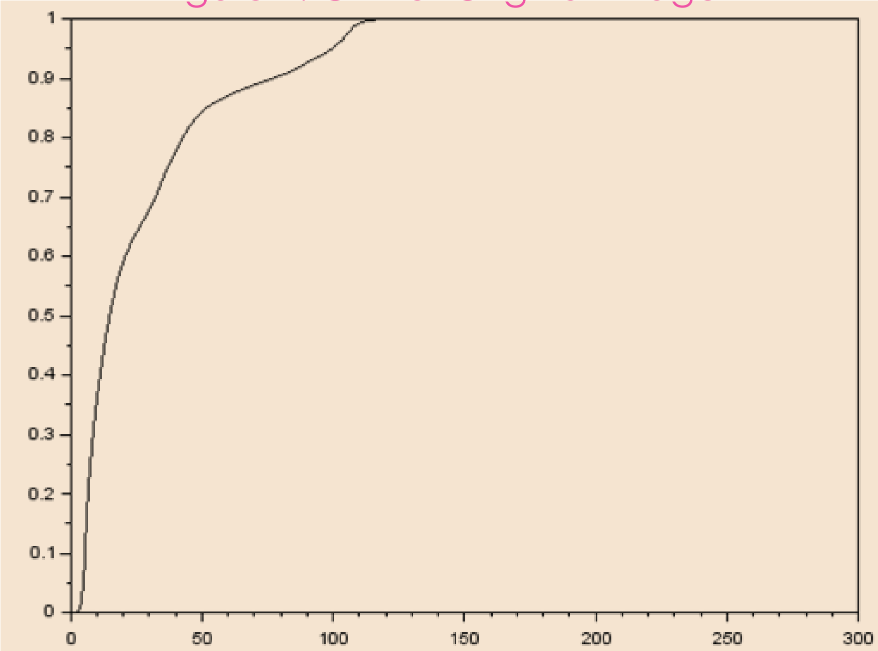


Figure 3: PDF of Original Image

Figure 4: CDF of Original Image



Probability distribution function (PDF) and cumulative distribution function(CDF) of the image were obtained. Note that image histogram is the PDF.

```
scf(); histplot(256, double(im));
xtitle("Grayscale PDF of Original Image");
```

From the PDF, CDF can be obtained because CDF is the integral of PDF.

```
[counts1, loc1] = imhist(im);
CDF1 = cumsum(counts1)/length(im);
scf(); plot(CDF);
xtitle("CDF of Original Image");
```

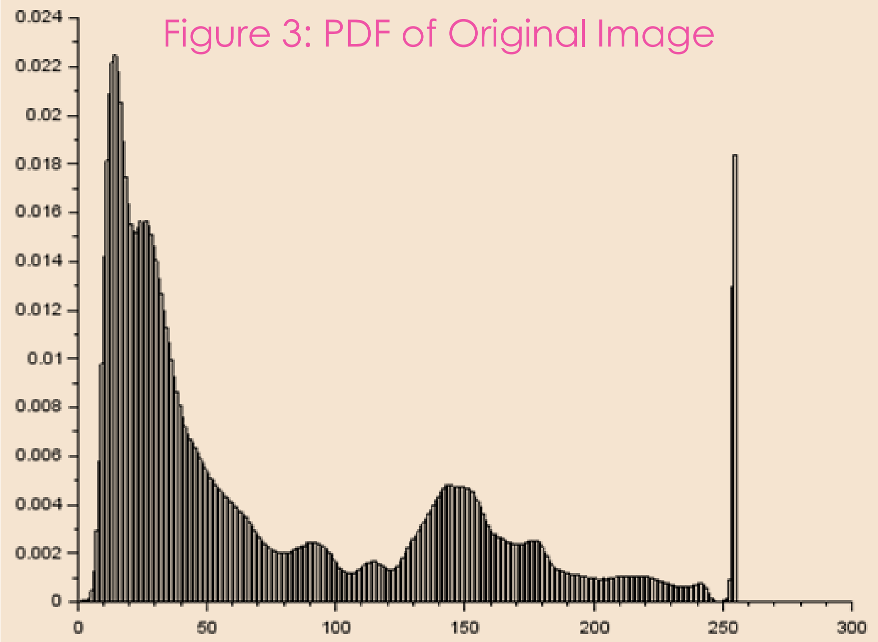
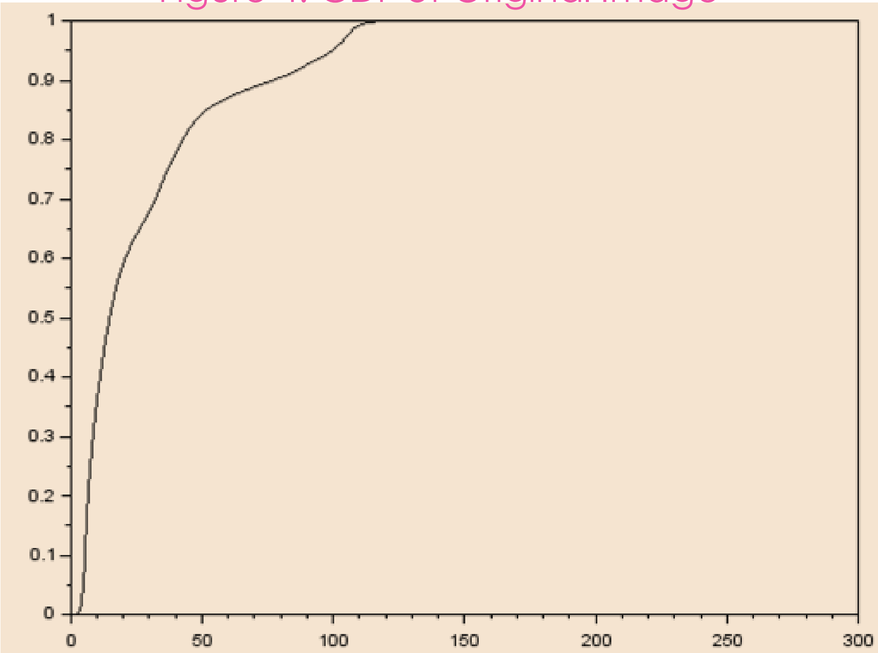


Figure 4: CDF of Original Image



Note: The x-axis of PDF corresponds to the pixel values of the image while the y-axis corresponds to the frequency of every pixel value

PDF is not evenly distributed, implying that the image has poor contrast. In fact, prominent peaks are concentrated at very low pixel values and one prominent peak is found at the highest pixel value. The peaks at the low pixel values illustrates the dark parts of the image while the sudden peak at the highest pixel value illustrates the bright background of the image.

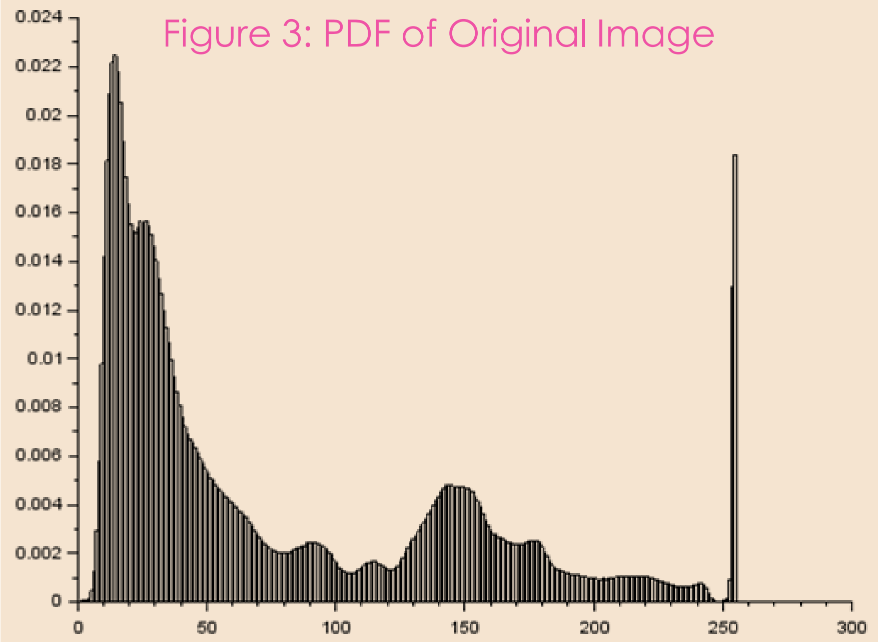
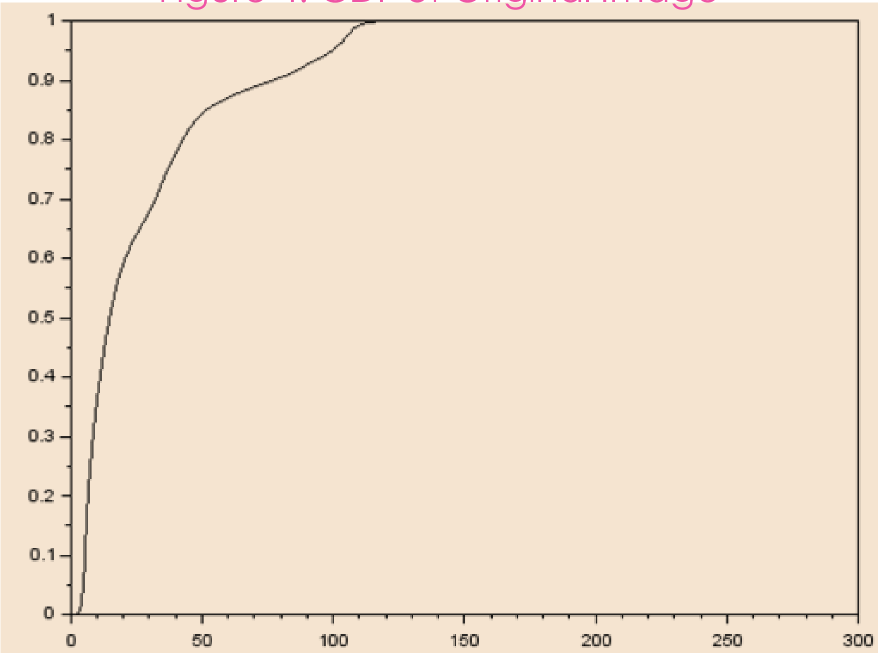


Figure 3: PDF of Original Image

Figure 4: CDF of Original Image



Note: The x-axis of CDF corresponds to the pixel values of the image while the y-axis corresponds to the percentage of the area covering up to a pixel value.

CDF is increasing with varying slopes, where the slopes are very high at very low pixel values because the prominent peaks of PDF are concentrated here. Also, CDF quickly approaches unity as pixel value increases, as expected, because the percentage of area covering up to pixel value of 255 must be 100%.

Desired CDFs were modeled: linear, quadratic, and error function.

```
x = [0:255];
xx = linspace(-2,2,256);
des_lin = x/255.0;
des_quad = x.^2/(255.0^2);
des_erf = 0.5*(erf(xx)+1);
scf(); plot(des_lin); xtitle("Desired Linear CDF");
scf(); plot(des_quad); xtitle("Desired Quadratic CDF");
scf(); plot(des_erf); xtitle("Desired Erf CDF");
```

Figure 5: Desired Linear CDF

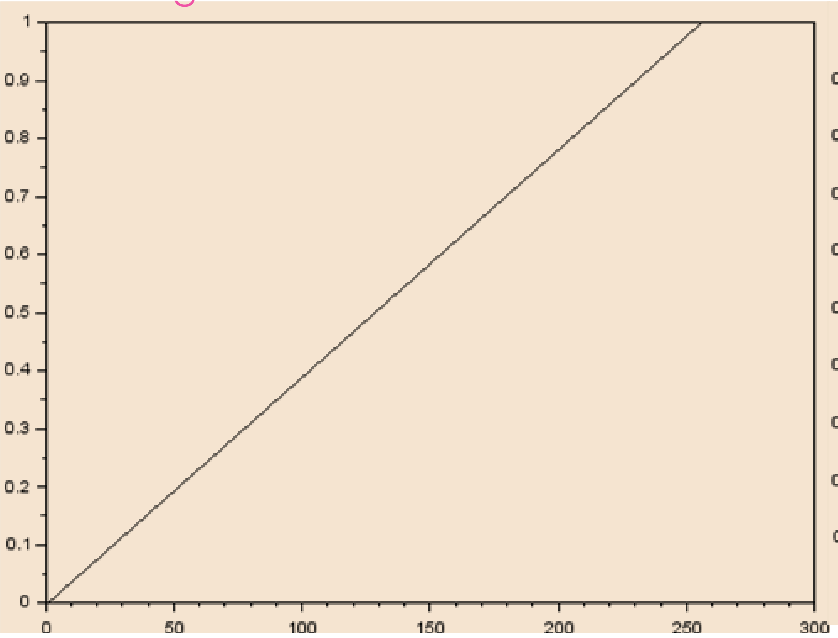


Figure 6: Desired Quadratic CDF

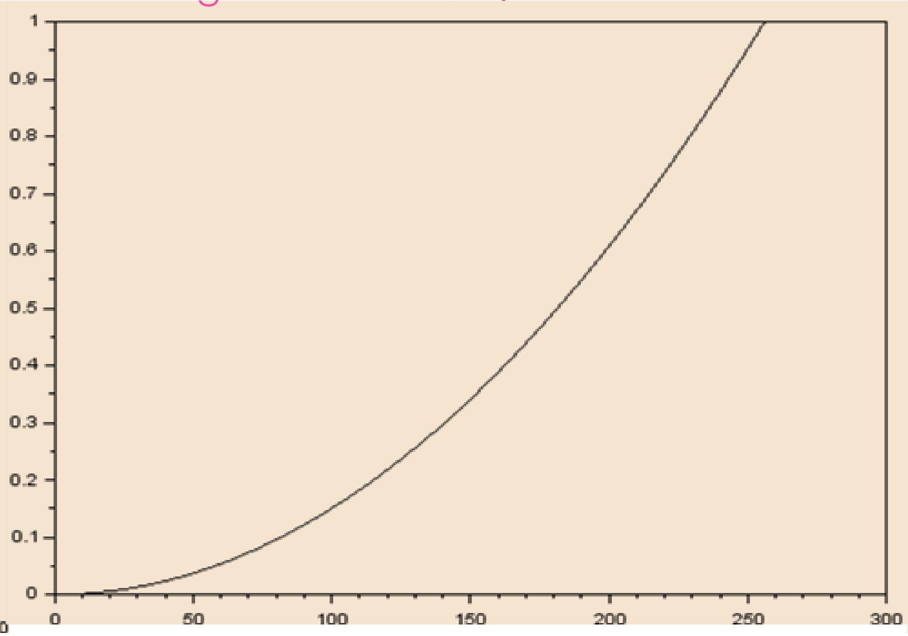
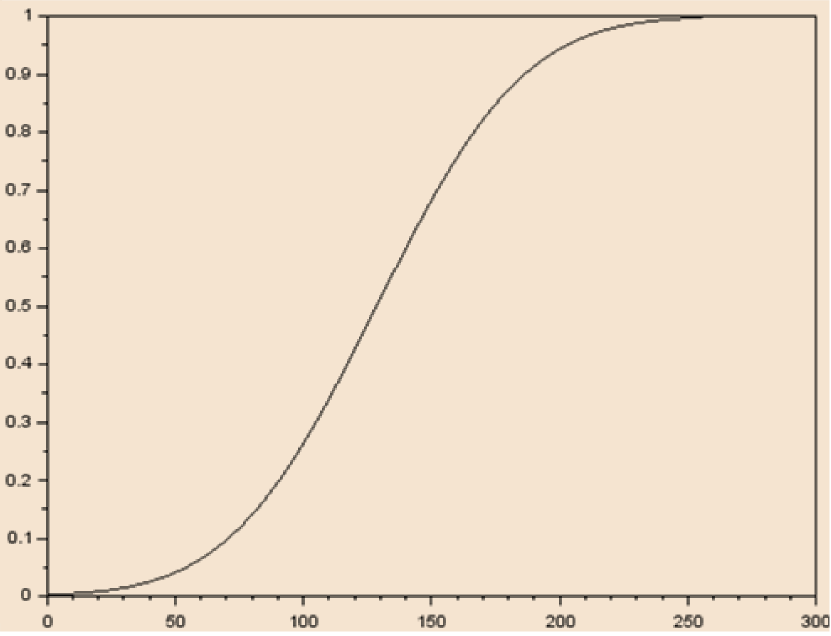


Figure 7: Desired Erf CDF



Linear CDF gives a widely-spread PDF.

Quadratic CDF has a slowly increasing slope. This shows that the gathering of low-valued pixel numbers is slow, resulting to a brighter image.

Erf CDF gives a PDF that have peaks concentrated at the center.

Figure 5: Desired Linear CDF

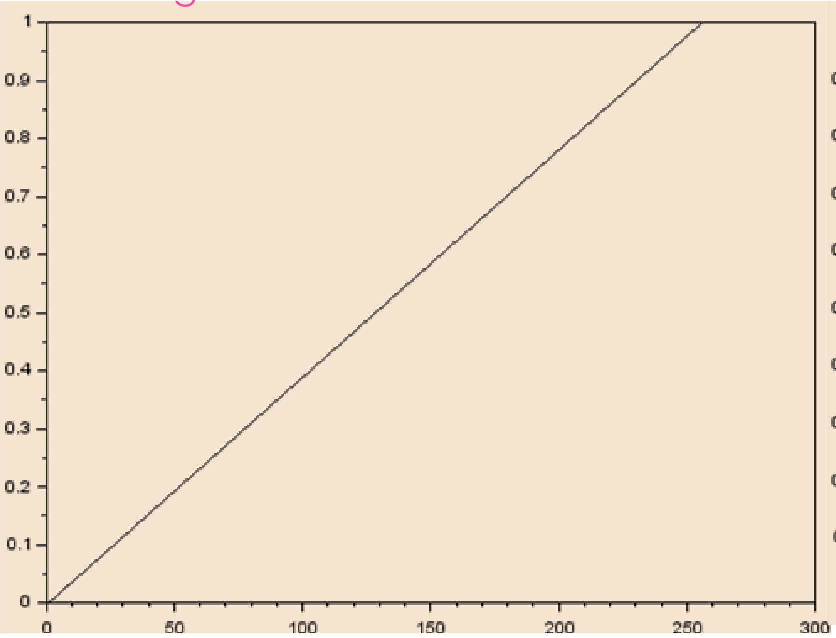


Figure 6: Desired Quadratic CDF

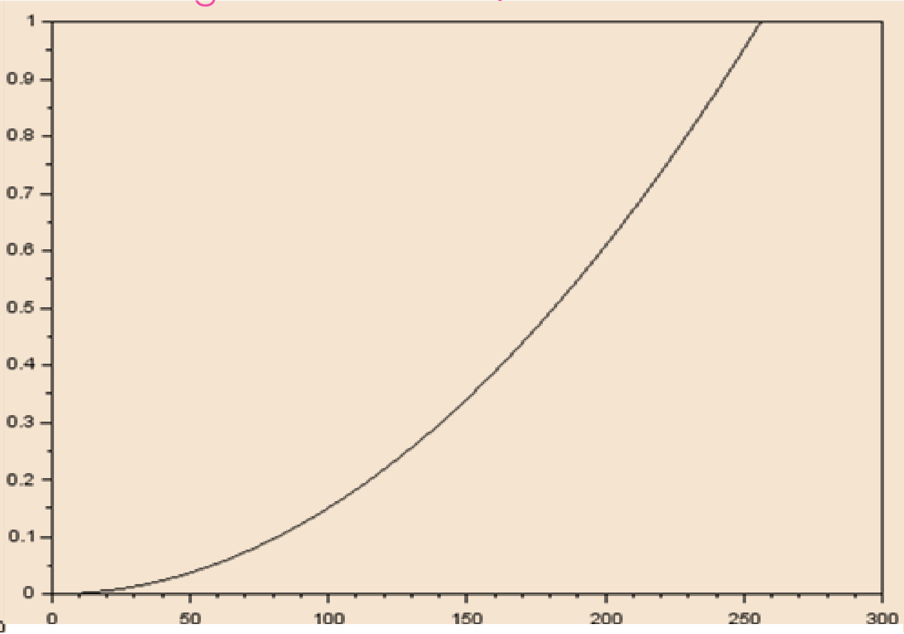
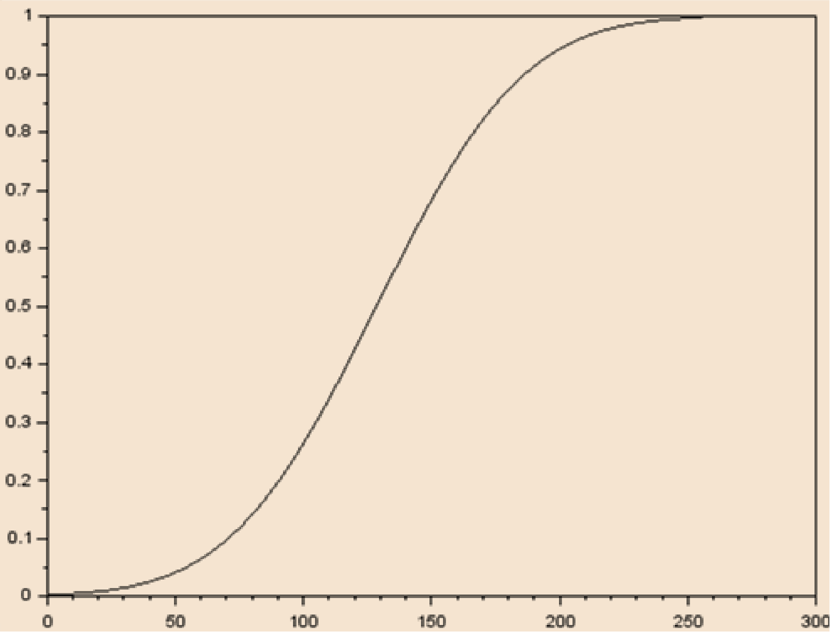


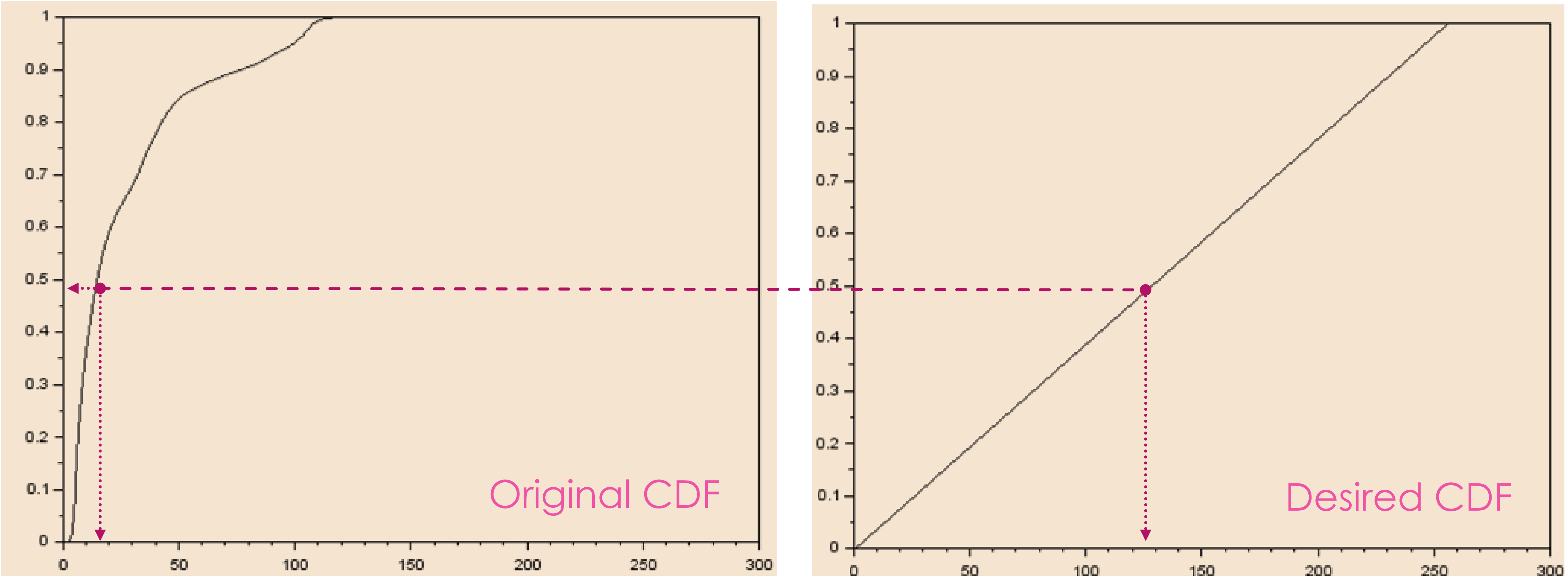
Figure 7: Desired Erf CDF



Backprojection of gray values was performed between orig. CDF and desired CDF. For each pixel value of the orig. image, the corresp. y-value in the desired CDF was found. The corresp. pixel value of the found y-value was retrieved to replace the corresp. pixel value of the actual CDF.

```
des = des_erf;
d = splin(x,des);
[y1, y2, y3] = interp(CDF1, des, x, d);
image = im;
for i = 0:255
    image(find(im == i)) = y1(i+1);
end
imwrite(image, "1_erf_enhanced.png")
```

Figure 8: Backprojection of gray values using CDF



We compared the original grayscale image to the image enhanced by using different desired CDFs.

Figure 9: Grayscale Image

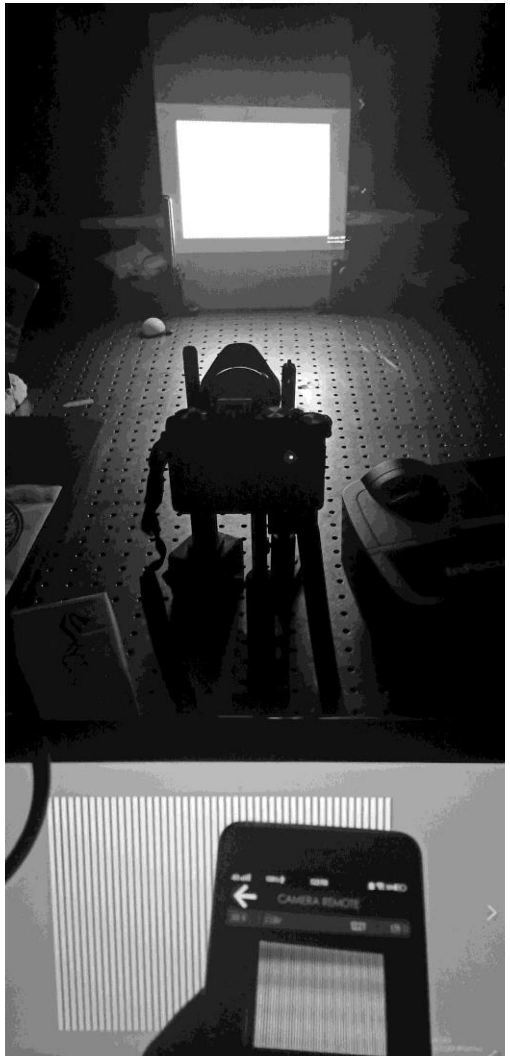


Figure 10: Image enhanced using linear CDF

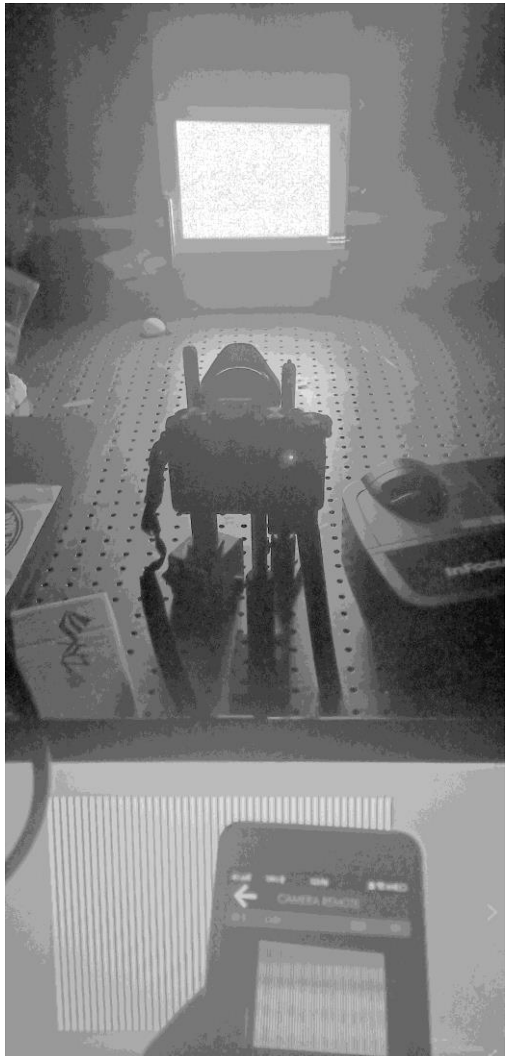


Figure 11: Image enhanced using quadratic CDF

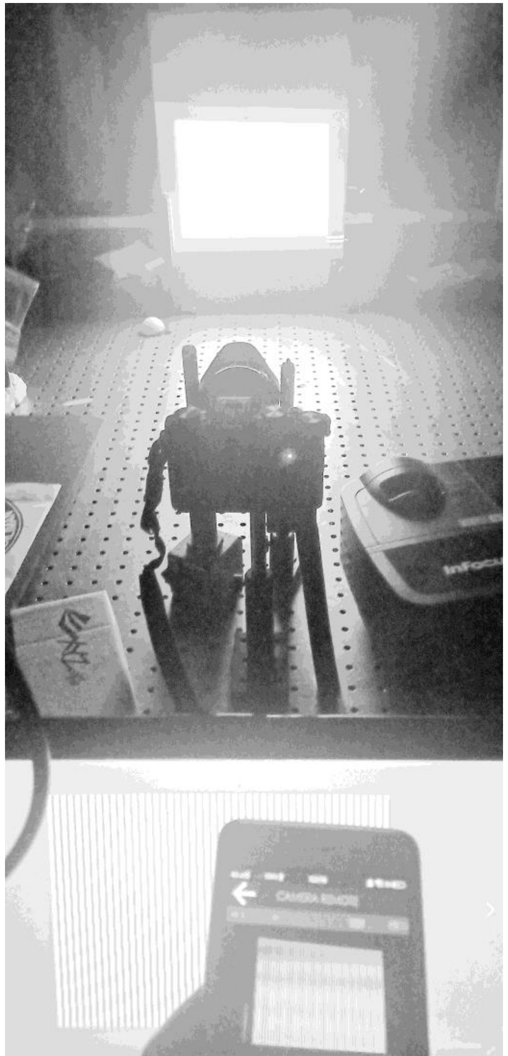
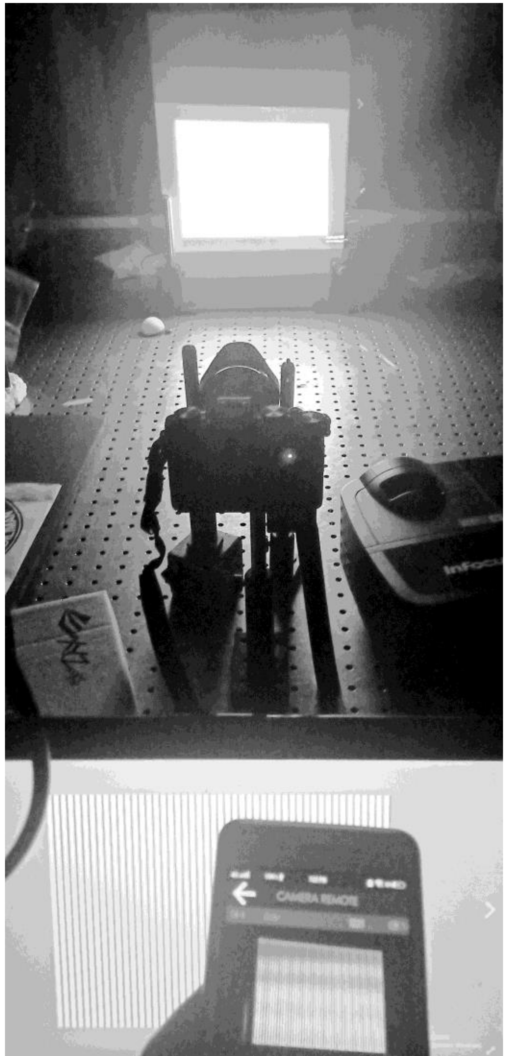


Figure 12: Image enhanced using erf CDF



Below are their corresponding PDF.

Figure 13: PDF of Grayscale Image

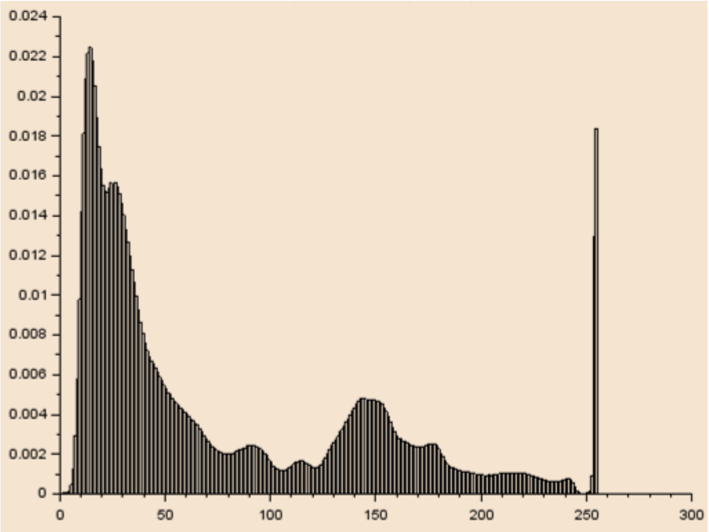


Figure 14: PDF of image enhanced using linear CDF

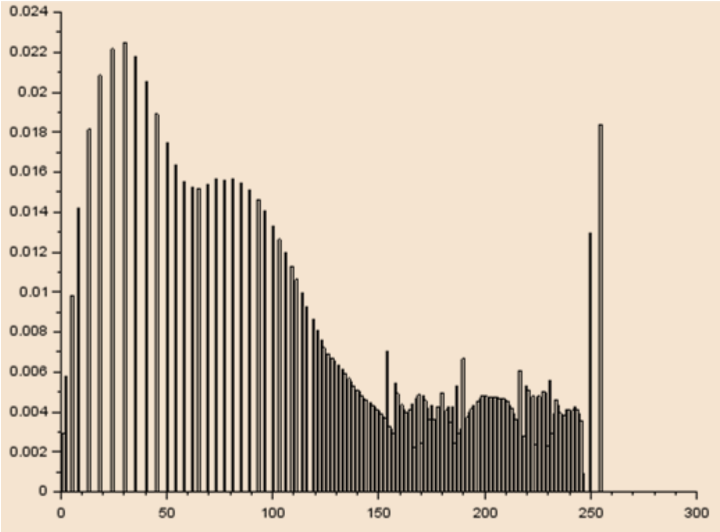


Figure 15: PDF of image enhanced using quadratic CDF

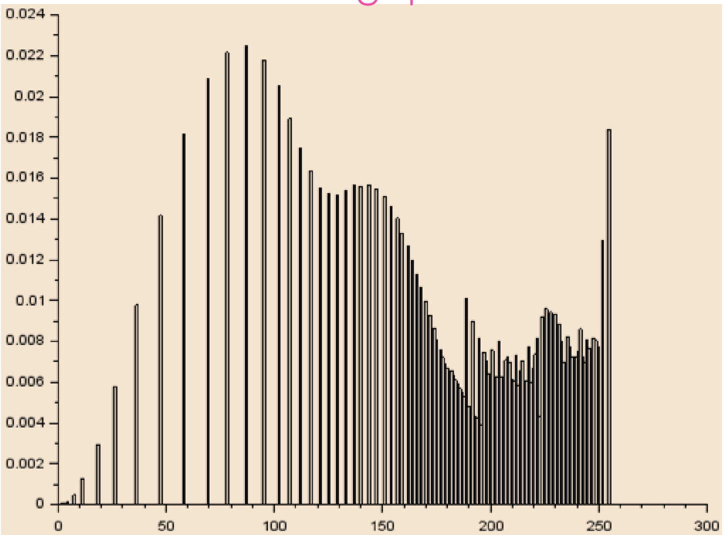
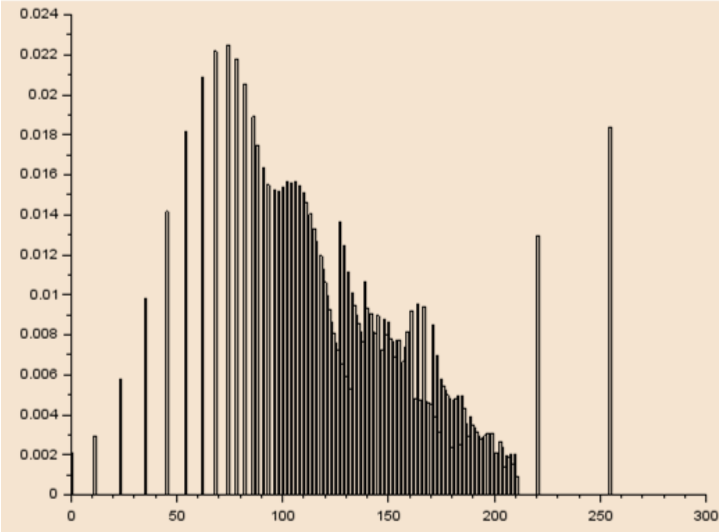


Figure 16: PDF of image enhanced using erf CDF



Below are their corresponding CDF.

Figure 17: CDF of Grayscale Image

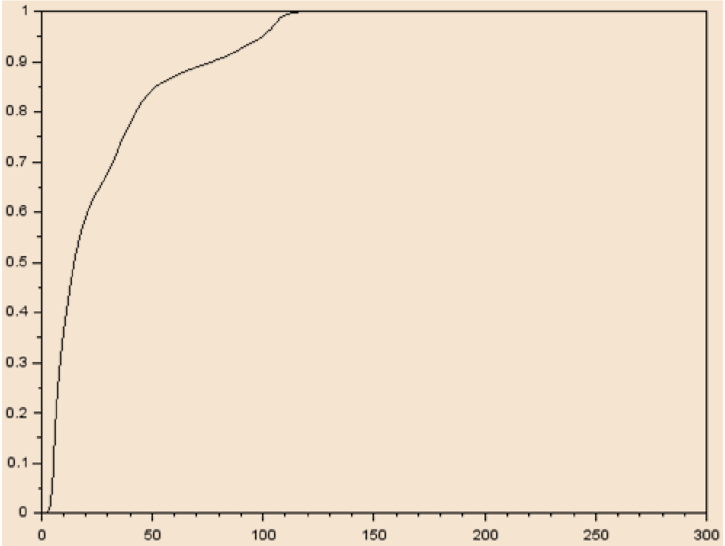


Figure 18: CDF of image enhanced using linear CDF

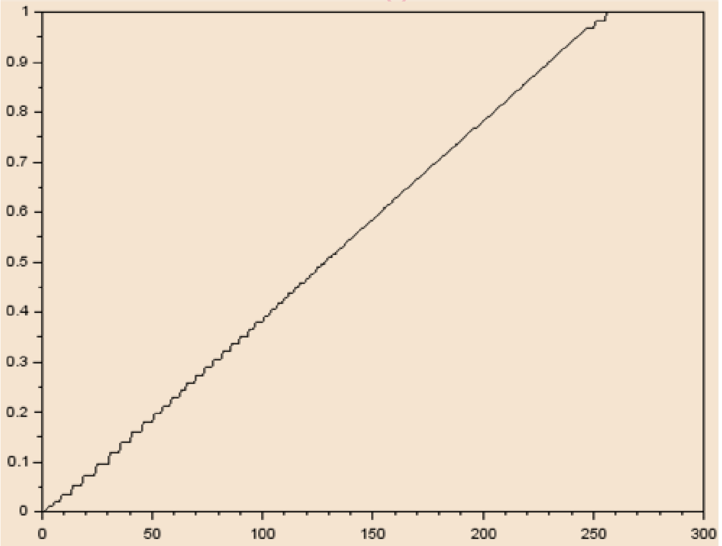


Figure 19: CDF of image enhanced using quadratic CDF

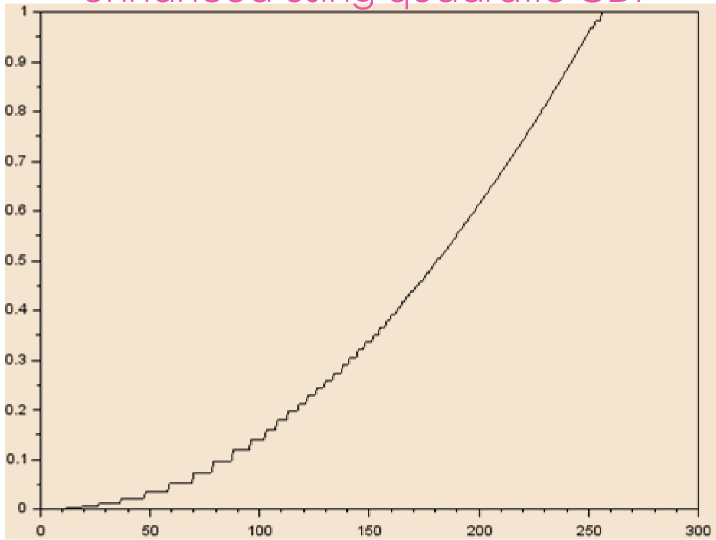
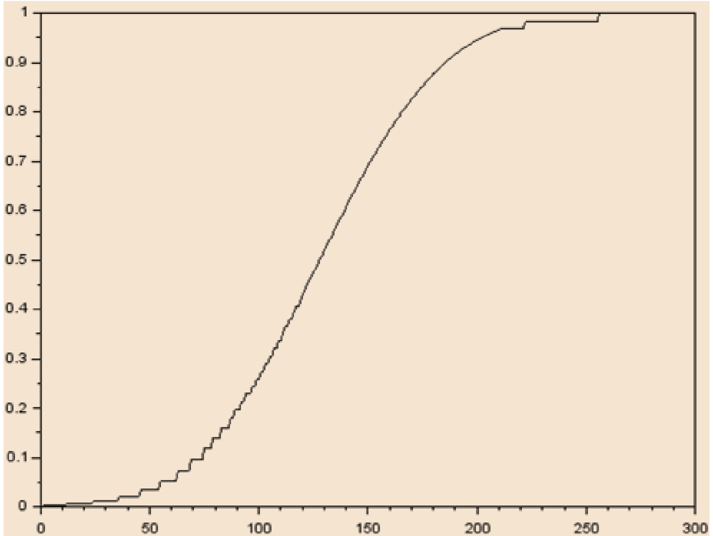


Figure 20: CDF of image enhanced using erf CDF



OBSERVATIONS

Using linear CDF in image enhancement

- Its PDF is more widely spread than the original PDF.
- The y values of its CDF is increasing to unity at a constant slope.
- The original image is enhanced such that the shades are more evident.

Using quadratic CDF in image enhancement

- It tries to concentrate the peaks of PDF towards higher pixel values.
- The slope of its CDF is slowly increasing, such that the accumulation is faster at high pixel values than at low pixel values.
- Its enhanced image is the brightest.

Using erf CDF in image enhancement

- It tries to concentrate the peaks of PDF towards the center.
- The slope of its CDF is highest in mid-range pixel values.
- Its enhanced image is the glossiest.

SciLab Code

```
xdel(winsid())

im = imread('4.jpg');
im = rgb2gray(im);
im_min = min(im); im_max = max(im);
disp("min = "); disp(im_min);
disp("max = "); disp(im_max);
imwrite(im, '4_gray.png');

//PDF of Original Image
scf(); histplot(256, double(im));
xtitle("Grayscale PDF of Original Image");

//CDF of Original Image
[counts1, loc1] = imhist(im);
CDF1 = cumsum(counts1)/length(im);
scf(); plot(CDF1);
xtitle("CDF of Original Image");

//desired CDF
x = [0:255];
xx = linspace(-2,2,256);
des_lin = x/255.0;
des_quad = x.^2/(255.0^2);
des_erf = 0.5*(erf(xx)+1);
scf(); plot(des_lin); xtitle("Desired Linear CDF");
scf(); plot(des_quad); xtitle("Desired Quadratic CDF");
scf(); plot(des_erf); xtitle("Desired Erf CDF");

//backprojection
des = des_erf;
d = splin(x,des);
[y1, y2, y3] = interp(CDF1, des, x, d);
image = im;
for i = 0:255
    image(find(im == i)) = y1(i+1);
end
imwrite(image, "4_erf_enhanced.png")

//PDF of enhanced image
scf(); histplot(256, double(image));
xtitle("Grayscale PDF of Enhanced Image");

//CDF of enhanced image
[counts2, bins2] = imhist(image);
CDF2 = cumsum(counts2)/length(image);
scf(); plot(CDF2);
xtitle("CDF of Enhanced Image");
```

Other image enhancements:

Other image enhancements:

Grayscale Image

Image enhanced using quadratic CDF

Image enhanced using linear CDF

Image enhanced using erf CDF

Contrast stretching was done with the last image because its min and max gray value are 0 and 245, respectively, and not 0 and 255.

As shown, the difference is not as obvious as the image enhancement by histogram manipulation.

```
im = imread('5.jpg');
im = double(rgb2gray(im));
min = min(im(:)); max = max(im(:));
imcont = (im-min)./(max-min);

imwrite(imcont, '5_cont.jpg');
```

Grayscale Image

Image enhanced by Contrast Stretching

Grayscale Image

Image enhanced by Contrast Stretching

We tried to capture an image with smaller difference between its min and max gray values. This image has 0 min gray value and 178 max gray value. As shown, the contrast of this image is enhanced because this technique expands the dynamic range of the image.

References:

[1] http://torezaimage.blogspot.com/2013/06/enhancement-by-histogram-manipulation.html

Self-evaluation: 12/10