

CPS843 Assignment 2

Udbhav Prasad
500909034

Abstract—Second Assignment for Intro to Computer Vision (CPS843).

PART 1 - PROBLEM 1

Figure 1 contains the original gray scale image.



Fig. 1. Original Gray scale Image

The more details in the detections there are, the more noise is also present, in prewitt we see that there is no false detections in the sky but other detectors capture that false edges (clouds). However, all the detections have similar edges. Noise is having alot of effect on the edges.

A. Robert Edge Detection

Figure 2 contains Robert edge detected image.
Here are the vertical/horizontal masks for Robert:

-1	1
----	---

-1
1

Here are the diagonal masks for Robert:

-1	0
0	1

0	-1
1	0

B. Prewitt Edge Detection

Figure 3 contains Prewitt edge detected image.
Here are the vertical/horizontal masks for Prewitt:

-1	0	1
-1	0	1
-1	0	1



Fig. 2. Robert Edge Detected Image

-1	-1	-1
0	0	0
1	1	1

Here are the diagonal masks for Prewitt:

0	1	1
-1	0	1
-1	-1	0

-1	-1	0
-1	0	1
0	1	1



Fig. 3. Prewitt Edge Detection Image

C. Sobel Edge Detection

Figure 4 contains Sobel edge detected image.
Here are the vertical/horizontal masks for Sobel:

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Here are the diagonal masks for Sobel:

0	1	2
-1	0	1
-2	-1	0

-2	-1	0
-1	0	1
0	1	2



Fig. 4. Sobel Edge Detection Image

f(x)	f'(x)	f''(x)
0	0	6
0	6	-6
6	0	0
6	0	0
6	0	-1
6	-1	0
5	-1	0
4	-1	0
3	-1	0
2	-1	1
1	0	0
1	0	0
1	0	0
1	0	0
1	0	5
1	5	-5
6	0	0
6	0	0
6	0	0
6	0	-6
6	-6	6
0	0	N/A
0	N/A	N/A

Fig. 5. 1D Image 1st and 2nd order derivatives with zero padding

PART 1 - PROBLEM 2

In figure 5, we show the 1D image and the first and second order derivatives of it. We used zero padding twice to preserve the last two pixel values.

PART 1 - PROBLEM 3

- **Step 1:** You blur the image with Gaussian filter.
- **Step 2:** Subtract the original image with the blurred image and obtain the mask.
- **Step 3:** Add the mask to the original image with the mask you obtained multiplied by a constant k value. What you obtain is the sharpened image.

With k=1, we obtain this image:



With k=5, we obtain this image:



With k=1, the image appears to have more sharpened edges but with k=5, it has very large differences in the edges. k=5, the images appears very bold in the image, this is because we add higher intensity of the mask.

PART 1 - PROBLEM 4

In figure 6 and figure 7, we add and remove Gaussian noise, the results are very nice. Average filter applies a smoothing/blurring effect on the image. Gaussian filter removes the noise from the image and creates a more even image.

PART 1 - PROBLEM 5

Equations for First order derivative of f(x, y):

$$\frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y) \quad (1)$$



(a) Gaussian Noise
mu=0 var=0.005



(b) Gaussian Noise
mu=0.2 var=0.001



(c) Medium Filter 5x5



(c) Medium Filter 5x5



(e) Gaussian Filter 10x10



(e) Gaussian Filter 10x10

Fig. 6. Problem 4, First Image Noise Removal

Fig. 7. Problem 4, Second Image Noise Removal

$$\frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y) \quad (2)$$

Equation for Laplacian of a 2D image:

$$\nabla^2 f = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (3)$$

Image is:

0	2	5	7
2	5	7	3
5	6	3	1
5	2	1	0

First order with respect to x:

2	3	2	-7
3	2	-4	-3
1	-3	-2	-1
-3	-1	-1	0

First order with respect to y:

2	3	2	-4
3	1	-4	-2
0	-4	-2	-1
-5	-2	-1	0

Mask for Laplacian:

0	1	0
1	-4	1
0	1	0

Image after Laplacian and Normalization:

4/9	2/9	-4/9	-20/9
2/9	-3/9	-12/9	3/9
-7/9	-9/9	3/9	2/9
-13/9	4/9	1/9	2/9

Laplacian is done with mask by element wise multiplication then summation then normalization with the neighbourhood of each pixel:

Example for top left pixel: $((0)(0) + (0)(1) + (0)(0) + (0)(1) + (0)(-4) + (2)(1) + (0)(0) + (2)(1) + (5)(0))/9 = 4/9$

PART 2

Description of image enhancement algorithm:

An overview is that we invert the image then perform the haze removal algorithm then invert the image again to get the output.

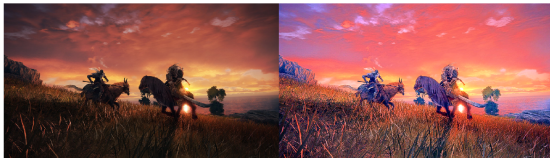
The steps:

- **Step 1:** Enhance Low Light Image using Dehazing Algorithm:

In this step, we invert the image, apply haze removal algorithm (imreadhaze) and invert it again to obtain the enhanced low light image.

- **Step 2:** Improve Results Further Using imreducehaze Optional Parameters:

In this step, we use same method as step 1 but improve on the method with better parameters: with contrast enhancement and boost per pixel.



- **Step 3:** Another Example of Improving Poorly Lit Image: In this step, we perform haze reduction algorithm on another image with just improvement of contrast enhancement.



- **Step 4:** Reduce Color Distortion by Using Different Color Space:

We perform the same steps as step 3 but we convert to L*a*b colorspace before the process and convert back to rgb after the process. Here we get better results.

- **Step 5:** Improve Results Using Denoising:

Low light images have a lot of noise hence enhancing them increases noise. After the process we use imguidfilter to remove noise from the image.

- **Step 6:** Estimate Illumination Map:

In this step we compare the image after enhancing with the color map.

CODE FOR ASSIGNMENT 2

```
% CPS 843 Assignment 2
% Udbhav Prasad - 500909034

% -----Part 1 - Problem 1-----

img_color = imread('./images/eldenring.jpg');
img_gray = rgb2gray(img_color);
imwrite(img_gray, './outputs/eldenring_gray.jpg');

robert_image = edge(img_gray, 'Roberts');
prewitt_image = edge(img_gray, 'Prewitt');
sobel_image = edge(img_gray, 'Sobel');

imwrite(robert_image, './outputs/eldenring_robert.jpg');
imwrite(prewitt_image, './outputs/eldenring_prewitt.jpg');
imwrite(sobel_image, './outputs/eldenring_sobel.jpg');

% -----Part 1 - Problem 3-----

h = fspecial('gaussian',5,2.5);

k = 1;
diff_img = img_gray - imfilter(img_gray, h);
highboost_img_1 = img_gray + k * diff_img;

k = 5;
diff_img = img_gray - imfilter(img_gray, h);
highboost_img_5 = img_gray + k * diff_img;

imwrite(highboost_img_1, './outputs/eldenring_sharpen_1.jpg');
imwrite(highboost_img_5, './outputs/eldenring_sharpen_5.jpg');

% -----Part 1 - Problem 4-----

noised_img_1 = imnoise(img_gray, 'gaussian', 0.005);
noised_img_2 = imnoise(img_gray, 'gaussian', 0.2, 0.001);

g_img_1 = filter2(fspecial('gaussian', 10), noised_img_1) / 255;
average_img_1 = filter2(fspecial('average', 5), noised_img_1) / 255;

g_img_2 = filter2(fspecial('gaussian', 10), noised_img_2) / 255;
average_img_2 = filter2(fspecial('average', 5), noised_img_2) / 255;

imwrite(noised_img_1, './outputs/eldenring_noise_1.jpg');
imwrite(noised_img_2, './outputs/eldenring_noise_2.jpg');

imwrite(g_img_1, './outputs/eldenring_g_1.jpg');
imwrite(average_img_1, './outputs/eldenring_average_1.jpg');
imwrite(g_img_2, './outputs/eldenring_g_2.jpg');
imwrite(average_img_2, './outputs/eldenring_average_2.jpg');
```

REFERENCES

- [1] Code for this assignment at ./a2.m
- [2] Assignment source code: <https://github.com/UdbhavPrasad072300/CPS843-Assignments>