EE569 Digital Image Processing

**HOMEWORK #2**

**HOMEWORK – Introduction to Digital Image Processing**

**Issued: 31/01/2020 Due: 16/02/2020**

# Problem 1: Edge Detection (50%)

1. **Sobel Edge Detector (10%)**

**Motivation**

The Sobel Edge Detector uses the gradient of pixel intensity value of image data to detect edge. The computation cost and procedure are low and simple. The basic theoretical foundation of the algorithm is to compute the gradient of two direction. After we get the result of gradient value in horizontal direction and vertical direction. A threshold is chosen to make the intensity value of image to be a binary value 0 or 255.

**Approach and Procedure**

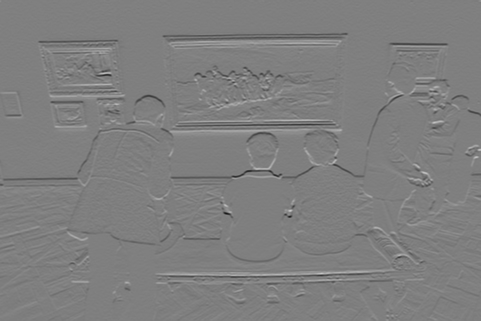
The implementation is simple. After normal procedure of allocation memory for image and intermediate image for extension or other possible modification of image, the main procedure is to determine the intensity value by threshold. Different threshold is chosen to see the trend of image vision performance. From comparison of these results, we can get a preliminary view of characteristics of possible problem.

**Results**

**A close up of text on a whiteboard

Description automatically generated**

**Figure 1.1.1**

****

**Figure 1.1.2**

**A close up of a sign

Description automatically generated**

**Figure 1.1.3**



**Figure 1.1.4**

A picture containing photo, standing, outdoor, building

Description automatically generated

**Figure 1.1.5 true magnitude edge map**

A close up of a map

Description automatically generated

A group of people in a dark room

Description automatically generatedA close up of a mans face

Description automatically generated**Figure 1.1.6 false tuned result**

**Figure 1.1.9**

**Figure 1.1.8**

**Figure 1.1.7**

**A close up of a device

Description automatically generated**

**Figure 1.1.10**

A picture containing nature

Description automatically generated

**Figure 1.1.11**



**Figure 1.1.12**

A picture containing outdoor, ground, grass, black

Description automatically generated

**A picture containing tree, outdoor

Description automatically generatedA picture containing outdoor, tree

Description automatically generatedA picture containing outdoor, tree

Description automatically generatedFigure 1.1.13**

**Figure 1.1.14 Figure 1.1.15 Figure 1.1.16**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trials** | **Percentage** | **Visual Performance** | **Reference** |
| **1** | **22** | **Much noise** | **Figure 1.1.7** |
| **2** | **52** | **Background turned to be white** | **Figure 1.1.8** |
| **3** | **202** | **Less edge detected** | **Figure 1.1.9** |
| **4** | **82** | **Pretty nice** | **Figure 1.1.10** |
|  |  |  |  |

**Table 1.1.1 parameters of Gallery.raw**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trials number** | **High threshold** | **Low threshold** | **Visual Performance** | **Reference** |
|  |  |  | **Figure 1.1.15** |  |
|  |  |  | **Figure 1.1.16** |  |
|  |  |  |  |  |

**Table 1.1.2 parameters of Dogs.raw**

The false image Figure 1.1.6 and 1.1.4 is obtained by quantization of pixel intensity too early. The quantization of computed pixel value should be in the final step. From the tuning result of the edge detector, the best value of thresholding percentage is around 80. The edge map graph shows clear and less noise. If the threshold value is high, the detail will be eliminated more. The low thresholding gives rise to the noise of the image. More errors appear when the threshold decreases. Some trade-offs are needed to show both detailed and less noisy image.

**Discussion**

According to comparison of two image result of edge map of Dogs.raw and Gallery.raw, it is hard to detect the edge of image with more texture inside. When the threshold is tuned low, much noise appears in the Dogs.raw. Hence, the contour of dog object is obscure. When the threshold is tuned high, some details of edge is ignored by mistaking selecting them as noise. Hence, the contour of the object is incomplete. The trade-off method to tune the threshold is not sufficient to ensure the clear vision performance when treating the complex image with much texture inside.

**Answers**

1. **Canny Edge Detector (10%)**

**Motivation**

**Approach and Procedure**

**Results**

**Discussion**

**Answers**

1. Flow chart for algorithm

Filter the noise

Compute the gradient magnitude

Apply non-maximum suppression to obsolete small local magnitude

Use double threshold to determine the edge

Finalize by suppressing all other weak and unconnected edges

1. The low threshold value is used to denoise the edge map. Some pixels may be mistakenly regarded as edge pixel. By using the low value edge threshold, these pixels can be filtered by the threshold.
2. **Structured Edge (15%)**

**Motivation**

The structured edge combines machine learning algorithm with traditional edge detection assumption. It extracts 32\*32 patches to construct the basic element for random forest algorithm. It gives better performance to detect more edge from the noisy image. Compared with above algorithm, it is more powerful.

**Approach and Procedure**

The setup for program needs some computer science experience.

*mex -DMX\_COMPAT\_32 private/edgeBoxesMex.cpp -outdir private*

The above command template is used to compile the C++ code for MatLab software. The difference with the old version of Matlab is the command that has additional option -DMX\_COMPAT\_32 to compile the code. Another step to run the program is to add the path of toolbox of *piotr\_toolbox*. Before addition of the folder of the toolbox, the subfolder paths of toolbox should be generated by Matlab function *genpath* to ensure all tools that will be used by Structure Edge program.

The *edgesDemo.m* file is revised to fit the requirement. The *writeraw()* function is used to get the image data of the edge image data. The edge detection file of *Dogs.raw* and *Gallery.raw* is stored as the *GalleryEdge.raw* and DogEdge.raw as a source file for performance evaluation problem. The additional transformation of edge data is implemented by formula below.

(1.3.1)

The formula ensures the *EdgeImageData* has the same graph of the comparison of the original image and edge image of Structured Edge implementation.

**Results**

**A group of people in a room

Description automatically generated**

**Figure 1.3.1 the origin image generated by Structure Edge Detector**

**A close up of text on a white background

Description automatically generated**

**Figure 3.3.2 the obtained image generated by default parameters of Structure Edge**

**A cat with its mouth open

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**Discussion**

**Answers**

1. **Performance Evaluation (15%)**

**Motivation**

There are four basic elements to construct the performance quality measurement of edge detection. They are true positive, false positive, false negative and false positive.

**Approach and Procedure**

**Results**

**Discussion**

**Answers**

**Problem 2: Digital Half-toning (50%)**

1. **Dithering (15%)**

**Motivation**

Halftoning is an important technique to quantize image for display of some devices such printer, HD display or other physical device. Fixed threshold dithering is to binarize the pixel intensity value of image by one threshold. The method is simple. The random threshold halftoning algorithm is to give each pixel a random threshold. The value is 0 when pixel value is lower than the threshold. The value is 255 when the pixel value is larger than random threshold. The dithering algorithm is used to place black dots corresponding to true distribution of black dots of original image. The threshold of each pixel to be halftoned is determined by the dithering matrix. The relative location of pixel value determines the value of dither matrix of different location.

**Approach and Procedure**

**Results**

**A picture containing sky, outdoor, tree, tower

Description automatically generated**

**Figure 2.1.1 Fixed threshold 128**

**A picture containing sky, outdoor, tree, tower

Description automatically generated**

**Figure 2.1.2 random halftoned**

**A close up of grater

Description automatically generated**

**Figure 2.1.3 Halftoned Image with Dither Matrix I2**

**A close up of an object

Description automatically generated**

**Figure 2.1.4 Halftoned Image with Dither Matrix I8**

**A close up of an object

Description automatically generated**

**Figure 2.1.5 Halftoned image Dithering Matrix I32**

**Discussion**

**The halftoned image contains much noise when the method**

**Answers**

1. **Error Diffusion (15%)**

**Motivation**

Error Diffusion method should be more advanced than the above halftoned method. The core thought of error diffusion is to diffuse the error obtained by the quantization to the neighbor pixel. Although the diffusion procedure is obliged to parsing order, all pixels in the image is adaptive to the errors of some previous pixels. The scanning order is senpertine order

**Approach and Procedure**

**Results**

**Discussion**

**Answers**

1. **Color Halftoning with Error Diffusion (20%)**

**Motivation**

Error Diffusion is applied to halftone three channels of a color image. The error diffusion is to diffuse the error caused by quantization to neighbor pixels of different weights. There are three different ways to construct the weight matrix of errors. The weight matrix influences the performance of halftoned image. We compare two methods to halftone the color image. By intuition, the method to operate channels separate should not be better than the MBVQ-based method. The reason is that the separate operation ignores the similarity of color vector in high dimension.

**Approach and Procedure**

**Results**

**Discussion**

As shown in the image, the method of separable error diffusion also can give a good vision result. Although the separable error diffusion can give a good estimation for simple channel, the luminance correlated to three channels value is influenced by separate operation of channel implementation.

**Answers**