Jesse Adams

ECE 579

Homework 1

Image Processing

**Goal:** To develop a better understanding of image processing and to create an algorithm that will generate an image that highlights important image features for navigation of MCECS bot.

**Methodology:** I used MATLAB and its image-processing library for this assignment. I familiarized myself with the capabilities of MATLAB and the image processing resources available. Using images captures from the Fourth Avenue Building I went through many iterations of trial and error before stumbling upon an algorithm that would generally produce a reliable image.

**Algorithm Description:**

The first step is creating structural elements that can be used to dilate or erode images. I began with a structural element in the shape of a rectangle that was 40 pixels by 20 pixels. I created another square element that was 50x50 pixels. The image that I wanted to process was first read into the program. I then converted that image to grayscale. Then I processed that image twice creating two separate erosions of the image. These were then subtracted from each other. The image produced is an inverted picture that I complement to produce the resulting image.

**The code:**

BW1 = imread(‘fab1.jpg’);

SE = strel(‘rectangle’, [40, 20]);

BoxSE = strel(‘square’, 50);

BW2 = imerode(BW1, SE);

BW3 = imerode(BW2, BoxSE);

BW4 = BW2 - BW3;

BW5 = imcomplement(BW4);

imshow(BW5)

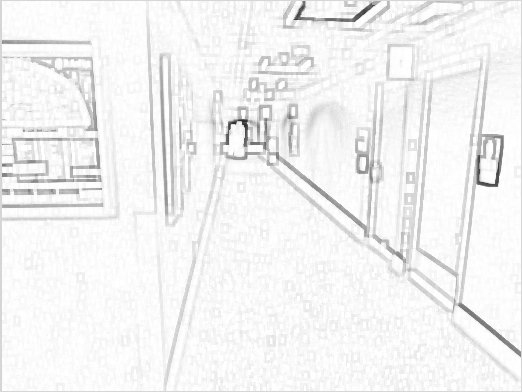
Image 1



Original Black and White Image Rectangle erosion



square erosion rectangle erosion – square erosion



Final image

Image 2



Original Black and White Image Rectangle erosion



square erosion rectangle erosion – square erosion

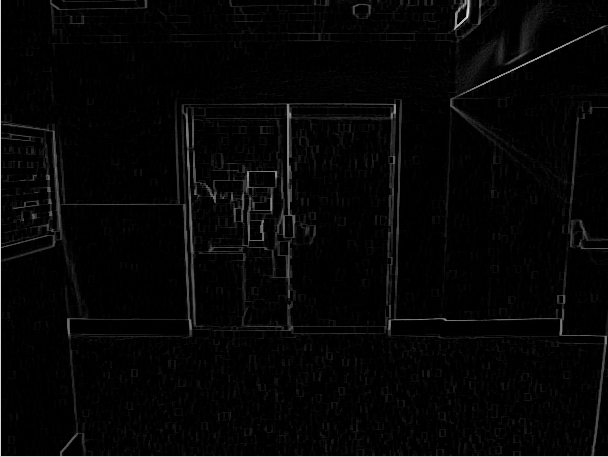


Final image

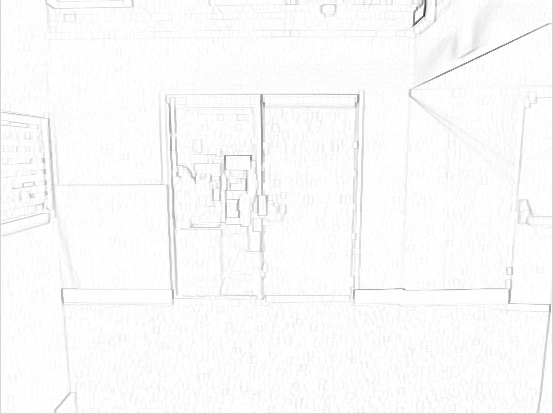
Image 3



Original Black and White Image Rectangle erosion



square erosion rectangle erosion – square erosion



Final image

Tuning: After finding the above algorithm I decided to attempt some fine-tuning. Reducing the size of the structural elements seemed to improve the images sharpness, which makes sense. The smaller the tool is that “combs” through the image the finer detailed the results will be. This did have a threshold however that below a certain point the results actually proved to be worse. I also added dilation step to the final image to help pronounce the final lines.

Final algothrithm:

SE = strel('rectangle', [20 10]);

BoxSE = strel('square', 20);

BoxSE2 = strel('square', 2);

B1 = imread('fab3.jpg');

B1 = rgb2gray(B1);

B2 = imerode(B1, SE);

B3 = imerode(B2, BoxSE);

B4 = B2 - B3;

B5 = imcomplement(B4);

B6 = imdilate(B5, BoxSE2);

imshow(B6);

Final images are on the pages that follow.

**Conclusion:** This assignment gave me a brief introduction into image processing. I can see how this is a powerful tool for robot navigation. This algorithm gives, generally, good results from images within the Fourth Avenue building. It is obvious to see from some of the images that with various lighting this algorithm may prove to not always work. Also, it does seem that the important features are highlighted in the image, but there is some noise. Perhaps finer tuning could get rid of this. For a robot like MCECS one algorithm may work. For a robot that experiences many different environments and lighting conditions I would imagine it would need various image processing techniques. Picking between which algorithm is the most appropriate for a particular lighting scenario would be necessary, but add an element of complexity.

