A picture containing logo

Description automatically generated

**CZ4003**

**Lab 2 Report**

**Name** : Chua Zi Jian

**Matriculation No**.: U2022354J

**Date** : 1/11/2022

**Content *Page***

1. Edge Detection 3
2. Line Finding using Hough Transform 9
3. 3D Stereo 14
4. Optional: Beyond Bags of Features: Spatial

Pyramid Matching for Recognizing Natural

Scene Categories 21

1. **Edge Detection**

**Text, application

Description automatically generated with medium confidence**

1. Download `macritchie.jpg’ from NTULearn and convert the image to grayscale. Display the image.

***Code:***

**Text

Description automatically generated**

**Fig 1.1.1 Code to run**

***Result:***

***A group of people running

Description automatically generated with medium confidence***

**Fig 1.1.2 Macritchie**

1. Create 3x3 horizontal and vertical Sobel masks and filter the image using conv2. Display the edge-filtered images. What happens to edges which are not strictly vertical nor horizontal, i.e. diagonal?

***Code:***

***Graphical user interface, text, application

Description automatically generated***

***Fig 1.2.1 Code to run***

***Result:***

**A black and white image of a city at night

Description automatically generated with medium confidence**

**Fig 1.2.2 Sobel Filter Horizontal**

**A picture containing text, plant

Description automatically generated**

**Fig 1.2.3 Sobel Filter Vertical**

**A picture containing graphical user interface

Description automatically generated**

**Fig 1.2.4 Sobel Filter All**

**Answer:**

If we are using horizontal sobel filter, the vertical edges will be filtered out and if we are using vertical sobel filter, the horizontal edges will be filtered out. For both cases, **the edges which are not strictly horizontal nor vertical will become fainter** instead of filtered out like the strictly horizontal/vertical edges.

1. Generate a combined edge image by squaring (i.e. .^2) the horizontal and vertical edge images and adding the squared images. Suggest a reason why a squaring operation is carried out.

***Code:***

***Text

Description automatically generated***

***Fig 1.3.1 Code to run***

***Result:***

**A picture containing text, tree, outdoor, people

Description automatically generatedA picture containing text, black

Description automatically generated**

**Fig 1.3.2 Combined Edge Image Fig 1.3.3 Squared Combined Edge Image**

**Reason:**

Applying the horizontal sobel filter give the horizontal gradient vector and the vertical sobel filter gives the vertical gradient vector.

After the Sobel Filters have been applied, the gradient of each pixel can be negative and what we need is the resultant magnitude of the edges and the positive and negative is just an indication of direction so squaring operation will help us to obtain the resultant magnitude of the gradient.

And after that we will select the edges which has greater magnitude than the threshold.

1. Threshold the edge image E at value t by

>> Et = E>t;

This creates a binary image. Try different threshold values and display the binary edge images. What are the advantages and disadvantages of using different thresholds?

***Code:***

***A picture containing text

Description automatically generated***

***Fig 1.4.1 Code to run***

***A picture containing text

Description automatically generatedResult:***

***A black and white image of trees

Description automatically generated with low confidenceA black and white image of trees

Description automatically generated with low confidenceA picture containing text, black, flower, white

Description automatically generated Fig 1.4.2 Edge, Threshold > 100 Fig 1.4.3 Edge, Threshold > 1000***

***A picture containing shape

Description automatically generatedA picture containing background pattern

Description automatically generatedFig 1.4.4 Edge, Threshold > 5000 Fig 1.4.5 Edge, Threshold > 10000***

***Fig 1.4.6 Edge, Threshold > 50000 Fig 1.4.7 Edge, Threshold > 100000***

**Answer:**

Advantages of high threshold:

1. The higher accuracy of getting the outline of the object in image.
2. Lesser noises in the edges detected.

Disadvantages of high threshold:

1. The image will have lesser edge compared to the lower threshold.

Advantages and disadvantages of using low threshold will be the other way around. So we can see that there will be a trade-off of details of edges and noises depends on your threshold, the higher the threshold the lesser the details but lesser noises also.

1. Recompute the edge image using the more advanced Canny edge detection algorithm with tl=0.04, th=0.1, sigma=1.0

>> E = edge(I,’canny’,[tl th],sigma);

This generates a binary image without the need for thresholding.

1. Try different values of sigma ranging from 1.0 to 5.0 and determine the effect on the edge images. What do you see and can you give an explanation for why this occurs? Discuss how different sigma are suitable for (a) noisy edge removal, and (b) location accuracy of edges.
2. (ii) Try raising and lowering the value of tl. What does this do? How does this relate to your knowledge of the Canny algorithm?

***Code:***

***Graphical user interface, text, application

Description automatically generated***

***Fig 1.5.1 Code to run***

***Result:***

**Background pattern

Description automatically generated*I)***

**Background pattern

Description automatically generatedBackground pattern

Description automatically generated Fig 1.5.2 Canny Edge Detector, Default**

**A picture containing text

Description automatically generatedA picture containing text

Description automatically generatedFig 1.5.3 Canny Edge, Sigma = 1 Fig 1.5.4 Canny Edge, Sigma = 2**

**Fig 1.5.5 Canny Edge, Sigma = 3 Fig 1.5.6 Canny Edge, Sigma = 4**

**A close-up of a brain

Description automatically generated with low confidence Fig 1.5.7 Canny Edge, Sigma = 5**

**Discussion:**

Lower sigma will give better location accuracy of edges but will have more

noises. As the sigma increases, although the noises is being removed but more details of the edges is also being removed and the location accuracy will drop.

I think the reasons is because canny algorithm uses gaussian filter to smooth the edges and when the sigma is increases the details of the edges will also be removed and we will get a lower accuracy of location of edges.

**Background pattern

Description automatically generatedII)**

**Background pattern

Description automatically generated Fig 1.5.8 Canny Edge, Threshold = 0.01**

**Fig 1.5.9 Canny Edge, Threshold = 0.04**

**Background pattern

Description automatically generated**

**Fig 1.5.10 Canny Edge, Threshold = 0.08**

**Discussion:**

When the tl is small, there will be more edges and noises. As the tl increases, the noises will be lesser but the edges will also become less detailed.

Canny algorithm uses hysteresis thresholding and the tl is the lower bound threshold value, any value lesser than the tl will be set to 0 so that weak edges or noises can be removed.

1. **Line Finding using Hough Transform**

**A picture containing text, sky

Description automatically generatedText

Description automatically generated**

1. Reuse the edge image computed via the Canny algorithm with sigma=1.0.

***Code:***

***Text

Description automatically generated***

***Fig 2.1.1 Code to run***

***Result:***

***Background pattern

Description automatically generated Fig 2.1.2 Edge detected***

1. As there is no function available to compute the Hough transform in MATLAB, we will use the Radon transform, which for binary images is equivalent to the Hough transform. Read the help manual on Radon transform, and explain why the transforms are equivalent in this case. When are they different?

>> [H, xp] = radon(E);

Display H as an image. The Hough transform will have horizontal bins of angles corresponding to 0-179 degrees, and vertical bins of radial distance in pixels as captured in xp. The transform is taken with respect to a Cartesian coordinate system where the origin is located at the centre of the image, and the x-axis pointing right and the y-axis pointing up.

***Graphical user interface, application

Description automatically generatedCode:***

***Fig 2.2.1 Code to run***

***Background pattern

Description automatically generated with medium confidenceResult:***

***Fig 2.2.2 Colormap***

**Answer:**

The Hough transform and the Radon transform are indeed very similar to each other and their relation can be loosely defined as the former being a discretized form of the latter.

The Radon transform is a mathematical integral transform, defined for continuous functions on R(n) on hyperplanes in R(n). The Hough transform, on the other hand, is inherently a discrete algorithm that detects lines (extendable to other shapes) in an image by polling and binning (or voting).

In this case, Radon transform will have the same value as Hough transform because the image is applying discrete radon transform. With this both of the transform will map each pixel into equivalent sinusodial function. However the Radon transform will result differently if we consider a continuous radon transform.

In continuous radon transform is obviously different because hough transform works discretely. Hence the density of the line will have different value.

1. Find the location of the maximum pixel intensity in the Hough image in the form of [theta, radius]. These are the parameters corresponding to the line in the image with the strongest edge support.

***Code:***

***Logo

Description automatically generated with low confidence***

***Fig 2.3.1 Code to run***

***Result:***

***A picture containing text

Description automatically generated***

***Fig 2.3.2 Result of radius and theta***

1. Derive the equations to convert the [theta, radius] line representation to the normal line equation form Ax + By = C in image coordinates. Show that A and B can be obtained via

>> [A, B] = pol2cart(theta\*pi/180, radius);

>> B = -B;

B needs to be negated because the y-axis is pointing downwards for image coordinates.

Find C. Reminder: the Hough transform is done with respect to an origin at the centre of the image, and you will need to convert back to image coordinates where the origin is in the top-left corner of the image.

***Code:***

***Graphical user interface, text, application

Description automatically generated***

***Fig 2.4.1 Code to run***

***Logo

Description automatically generated with medium confidenceResult:***

***Fig 2.4.2 Result of C***

1. Based on the equation of the line Ax+By = C that you obtained, compute yl and yr values for corresponding xl = 0 and xr = width of image - 1.

***Code:***

***Text

Description automatically generated***

***Fig 2.5.1 Code to run***

***Text

Description automatically generatedResult:***

***Fig 2.5.2 Result of yl and yr***

1. Display the original ‘macritchie.jpg’ image. Superimpose your estimated line by

>> line([xl xr], [yl yr]);

Does the line match up with the edge of the running path? What are, if any, sources of errors? Can you suggest ways of improving the estimation?

***Graphical user interface, text

Description automatically generated with medium confidenceCode:***

***Fig 2.6.1 Code to run***

***Result:***

***Fig 2.6.2 Detected running track Edge***

**Answer:**

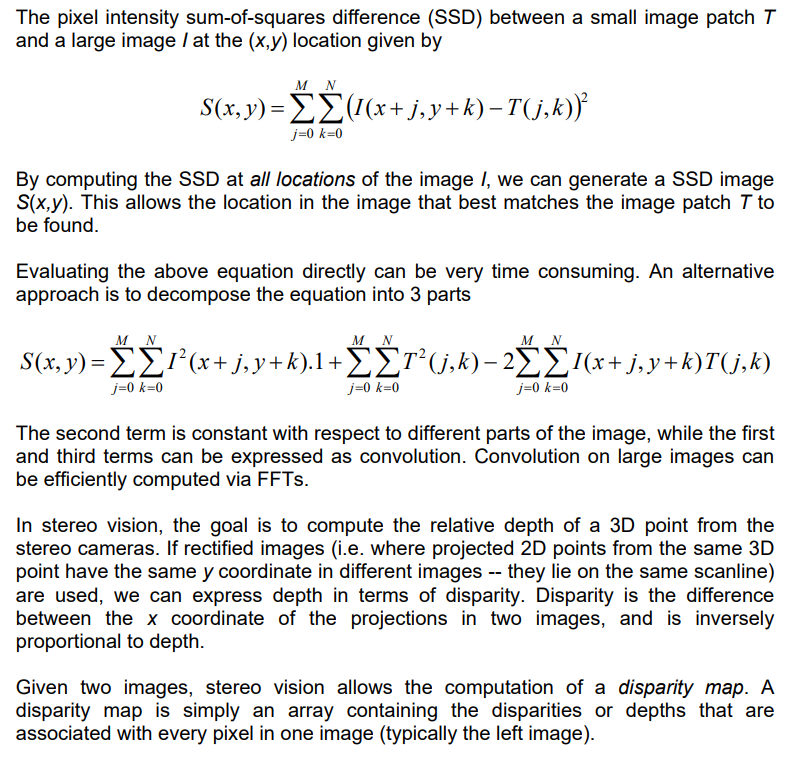
From the result, we can see that the estimated edge is aligned with the running track but if we see it closely we can found out that the edge is almost but not perfectly aligned to the running track.

These are the several possibilities that I can think of:

1. The line of the running track is not necessarily straight, maybe there will be some curve in between
2. There can be small precision error conversion from Radon transform parameters to the coordinate in image space.
3. There might be some noises that will affect to get the maximum intensity pixel.

**Suggestion:**

1. We can try to use non-linear function when the running track is not totally straight.
2. Try using a larger sigma value to reduce the noises on the picture
3. **3D Stereo**

****

**Text, letter

Description automatically generated**

1. Write the disparity map algorithm as a MATLAB function script which takes two arguments of left and right images, and 2 arguments specifying the template dimensions. It should return the disparity map. Try and minimize the use of for loops, instead relying on the vector / matrix processing functions.

***Code:***

**Text, letter

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

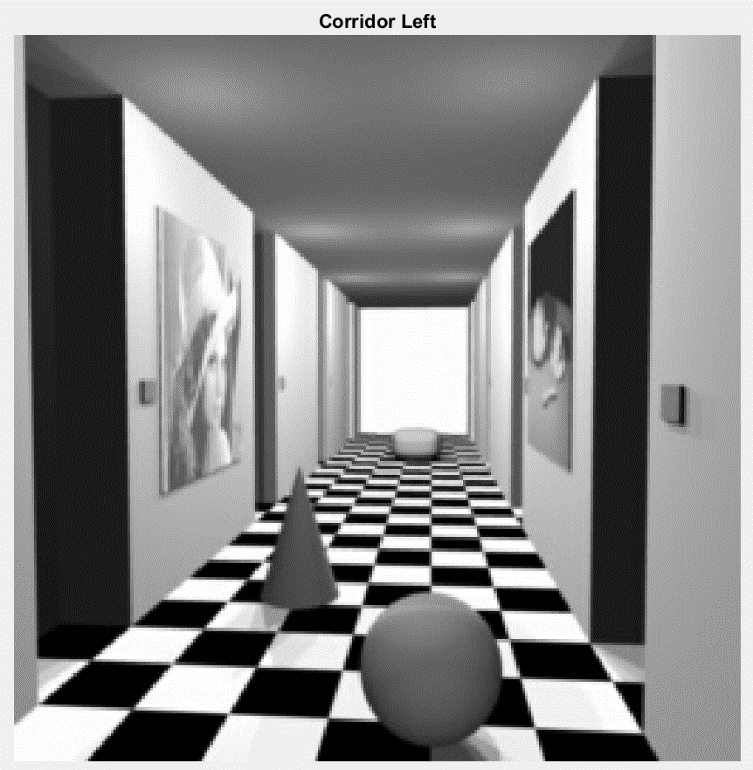
***Fig 2.1 Disparity Map Function Code***

1. Download the synthetic stereo pair images of ‘corridorl.jpg’ and ‘corridorr.jpg’, converting both to grayscale.

***Text

Description automatically generatedCode:***

***Fig 3.2.1 Code to run***

***Result:***

***Fig 3.2.2 Corridor Left***

**A room with a black and white checkered floor

Description automatically generated with low confidence *Fig 3.2.3 Corridor Right***

1. Run your algorithm on the two images to obtain a disparity map D, and see the results via

>> imshow(-D,[-15 15]);

The results should show the nearer points as bright and the further points as dark. The expected quality of the image should be similar to `corridor\_disp.jpg’ which you can view for reference.

Comment on how the quality of the disparities computed varies with the corresponding local image structure.

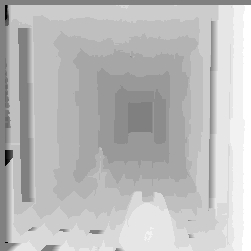
***Code:***

**Text

Description automatically generated**

***Fig 3.3.1 Code to run***

***Result:***

***A picture containing text, indoor, wall, bathroom

Description automatically generated Fig 3.3.2 Result of Corridor after Disparity Map D***

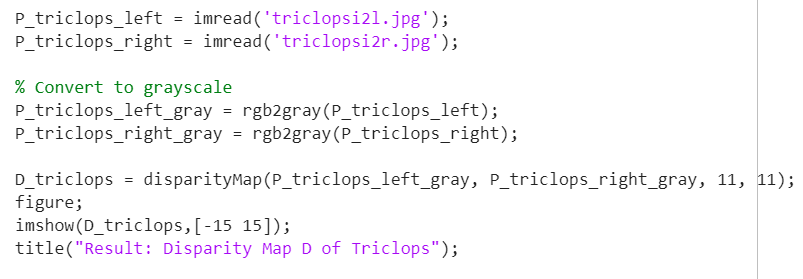
***Fig 3.3.3 Expected Output of Corridor***

**Answer:**

The result disparity map is similar to the expected disparity map. As expected, the pixel intensity decreases (gets darker) as we reach the centre of the image since the disparity decreases for far away objects. One area that can be improved upon is the centre of the image. Although the pixel intensities there are dark indicating low disparity which is correct, the results are not uniform across that section. This is due to the homogenous colour of that part of the image.

1. Rerun your algorithm on the real images of ‘triclops-i2l.jpg’ and triclops-i2r.jpg’. Again you may refer to ‘triclops-id.jpg’ for expected quality. How does the image structure of the stereo images affect the accuracy of the estimated disparities?

***Code:***

******

***Fig 3.4.1 Code to run***

***A picture containing text

Description automatically generatedResult:***

***A picture containing indoor, white, black

Description automatically generated Fig 3.4.2 Result of Triclops after Disparity Map D***

***Fig 3.4.3 Expected Output of the Triclops***

**Answer:**

The result disparity map we obtained is quite close to the expected disparity map. However, this result is not desirable as the original building structure edge information are lost in the disparity map. To conclude, appearance-based disparity maps like the one we have implemented have trouble calculating the correct disparity if there is a homogenous surface with the same intensity.