Individual Lab Report

Bikramjot Singh Hanzra
Team B – Auto Pirates
Teammates – Tushar Chugh, Shiyu Dong, Tae-Hyung Kim, William Seto
ILR10

March 31, 2016

Contents

List of Figures

1 Introduction

During the past couple of weeks, we worked on preparing for our field test. Two of the key goals for this field test and also the progress review were to test the cost inflation of obstacles in the river and to test the updated path planner code. We also wanted to test our extended goal of following the rules of the road. I worked on inflating the cost of the obstacles. Tushar then integrated this code with the path planner code.

2 Implementation

In order to inflate the cost of the obstacles we tried a couple of approaches –

- Distance Transform
- Morphological Operations using Dilation and Blurring

2.1 Distance Transform

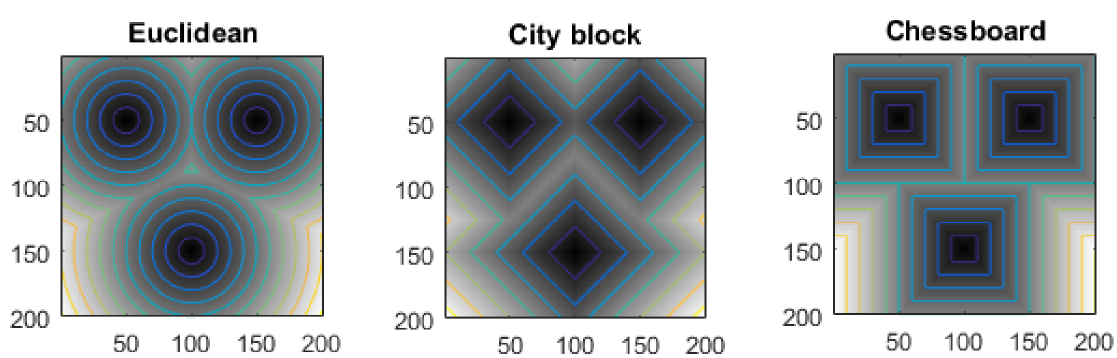
The distance transform of a binary image calculates the distance of each non-zero intensity pixel from the nearest zero intensity pixel. By definition, the distance transform of a zero intensity pixel is zero.

There are a number of different distance metrics that can be used and the most popular ones are –

- **Checkerboard Distance** – $\max(|x_1 - x_2|, |y_1 - y_2|)$
- **Manhattan Distance** – $|x_1 - x_2| + |y_1 - y_2|$
- **Euclidean Distance** – $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

where (x_1, y_1) is the non-zero intensity pixel and (x_2, y_2) is the zero intensity pixel.

We used the Euclidean Distance for computing the distance transform because it gave more smoother paths as compared to the other two distance metrics. Figure 1 shows the results of the various distance transform metrics.



(a) $t = 3$ (b) $t = 4$ (c) $t = 5$

Figure 1: Results of various distance transform metrics

2.2 Morphological Operations using Dilation and Blurring

The second method we used was to successively dilate and blur the image. We used a square kernel for dilating the image and a Gaussian Filter for blurring the image. So, in this we first dilate the obstacle image with the square kernel and then we convolve the resulting image with the Gaussian Filter to get the final resulting image. This results in a smooth cost drop from the maximum value to the minimum value near the boundaries of the obstacles. The maximum value in this case is 255 and the minimum value is 0.

2.3 Results

Below, I am showing the results of cost inflation on Figure 2 that I used while testing the code.

Figure 2: Input Image

Figure 3 shows the resulting image obtained by applying Distance Transform.

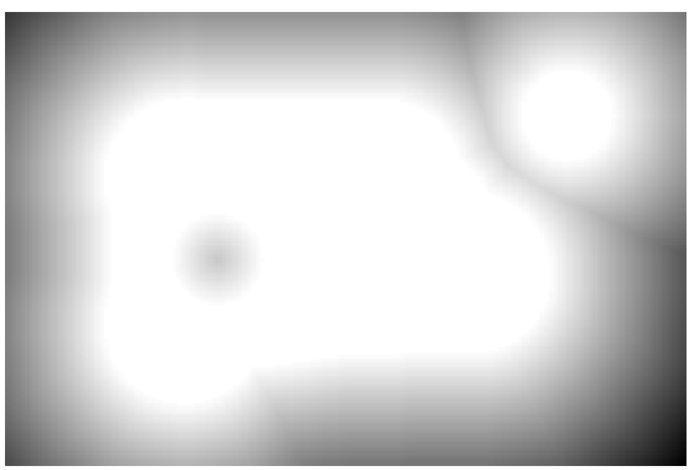


Figure 3: Resulting image obtained by applying Distance Transform

Figure 4 shows the resulting image obtained by successively dilating and blurring the image.

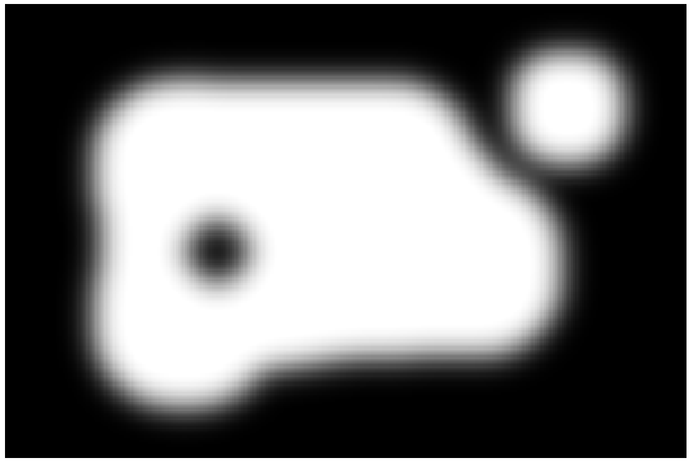


Figure 4: Resulting image obtained by successively dilating and blurring the image

2.3.1 Comparison of both the methods

From the two methods discussed above, we decided to use the more simpler approach of dilating and blurring an image because convolution can be done pretty fast as compared to calculating the distance transform for each pixel. This gives the path planner more time to search for the best path. On the other side, setting the kernel sizes in the latter case is a cumbersome task and requires considerable amount of time spent in tuning these parameters. OpenCV [1] functions were used to implement both the methods.

3 Challenges

There were some issues we faced during the field test. The key bottleneck in the whole system is the planning code. This takes more than 4 seconds to replan. After integrating the rules of the road it further alleviated this problem. Now, replanning takes even more time. This has the undesirable effect that the boat goes too near to the obstacles before the new path to avoid the collision is generated.

Another issue we faced was near one of the bridges where a lot of noise is generated behind the bridge. This noise get inflated and a path is generated that is very close to the shores. This issue was pretty weird and our proposition was that this bridge was way more wide than the rest of the bridges which caused the problem. We plan to engineer our way out by handling this bridge as a special case where we will ignore the noise behind the bridge.

As far as the rules of the road are concerned, we have not looked into how we will be tackling with situations when there is another boat coming head-on towards us. Implemented this is particularly difficult since within two successive radar frames the obstacle moves a lot of distance and it is difficult to predict the trajectory of the obstacles. Another concern is that other boats on the river do not always follow the rules of the road e.g. during our last field test we came across a situation where a cargo vessel was on the left hand side of the river and Mike had to take over manual control so that we did not the confuse the driver.

4 Teamwork

The work done by the rest of the team members is discuss below –

- **William Seto** – William worked with Tushar on rules of the road and integrated the various subsystems.
- **Tushar Chugh** – Tushar integrated the cost inflation code with the planning code. He also worked on rules of road with William.
- **Shiyu Dong** – Shiyu worked on preparing the maps that were used to test the rules of the road. He also added buttons in the GUI to add waypoints.
- **Tae-Hyung Kim** – Tae-Hyung worked on recording ros bag files using a package from Novatel.

5 Work Overview for Coming Week

For the next couple of weeks, we will be mostly working on tackling the issues discussed in the *Challenges* section. I will be working mostly on improving the cost inflation code. Right now, we inflate the cost equally of the obstacles. This inflates the smaller obstacles to pretty large sizes which is not desirable. An approach to tackle this will be to dilate and blur each individual obstacle based on their size i.e. the area they occupy in the costmap.

A vanilla version is ready of all the subsystem (Perception, Path Planning and Simulation) and we are currently spending time improving the overall system. Things seem to be in control right now and we are on schedule for SVE.

References

[1] `OpenCV` library
`http://opencv.org`