

# SIMULATION OF COLLISION AVOIDANCE BY NAVIGATION ASSISTANCE USING STEREOVISION

Syed Abdur Rauf Magrabi<sup>1</sup>

<sup>1</sup>Department of Aerospace Electrical and Electronics Engineering, Hyderabad, Andhra Pradesh

## Abstract

In the research study the fully automated vehicles is one of the most important factor to navigate the vehicle by avoiding obstacles and to maintain specific path. The methodologies to achieve this could be researched on robots which rely on their ability to sense and interact. Laser sensors and Ultra-sonic sound sensors have been a traditional way to detect objects in the recent past but they were not efficient at close range. By using stereo vision it is possible to detect objects and their distances just like how we humans detect object and their relative distances with the eyes. The objective of the project is to develop a system where it imports data a stereo vision camera, understand the nature of the environment and to choose the right direction to get an appropriate speed for a robot to navigate.

**Keywords:** X-box 360, MATLAB and SIMULINK.

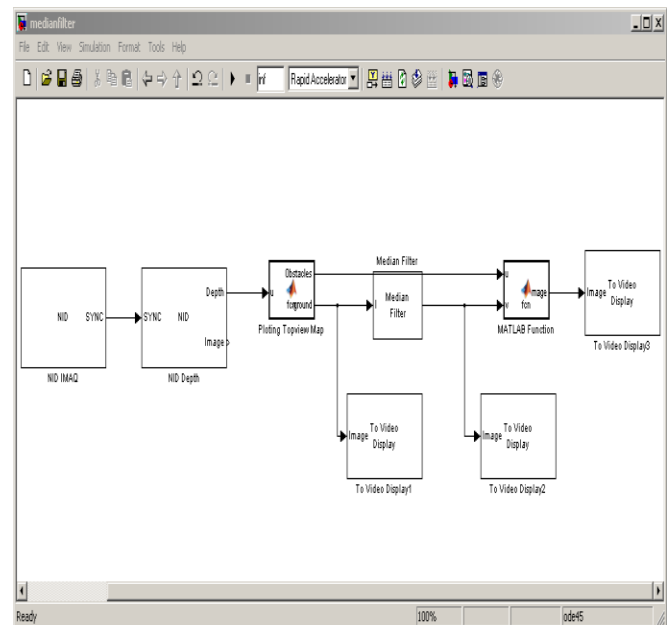
\*\*\*

## 1. INTRODUCTION

Stereo Vision refers to the ability to infer information on the 3D Structures and the distances of a scene from at least two images (Left and right), taken from different viewpoints. Stereo Vision utilizes the slightly different views of a scene, projected on to the right and left images, to recover depth information, which is known as Binocular Stereo Fusion. From a generic point of view, in both humans and machines, the problem reduces to a matching of the two views, in order to find the displacement (disparity) of corresponding patterns of the projected images. The use of automated machines in the study of other planets has emphasized the use of Stereo Vision where a precise depth perception is needed to detect and avoid obstacles to guide the route itself. Even the invention of automated robots, toys need the use of Stereo Vision to perform all the task to reach out and shake hands, pour into the container, grab a cup and serve and every human like activities.

## 2. BLOCK DIAGRAMS

We can see in the above figures that the ground plotted in blue consists of salt and pepper noise which is eliminated using a median filter with a neighbourhood size of  $18 \times 18$ .



**Fig 1:** Simulink model for applying median filter

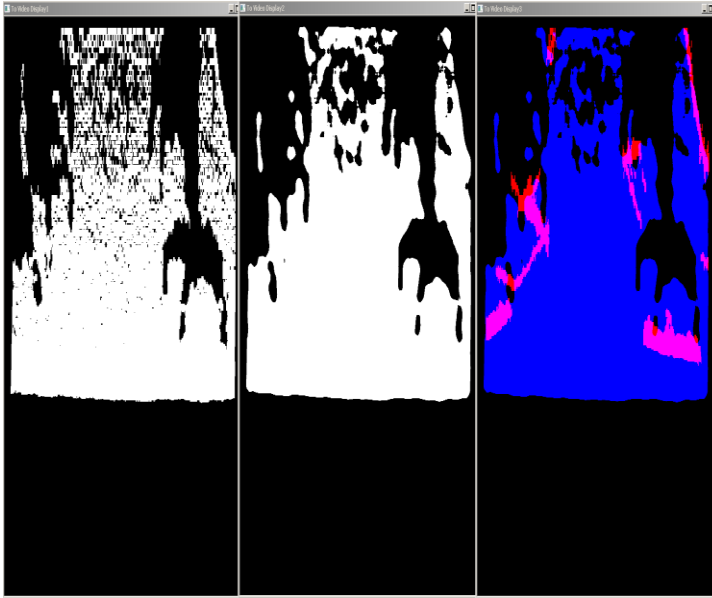


Fig 2: Outputs of above model.

It is seen that most of the salt and pepper noise is removed successfully removed by using a median filter. It is observed that Kinect fails to fully detect the depth at few random frames. This is rectified by using two control loops. In each of these control loops the output of the loop is delayed by one frame and given to the input. The loop is programmed in such a way that if the value of the pixel is zero then the pixel's value would be equal to that of the previous frame. The two loops are connected in so that the input of the second loop is the output of the first.

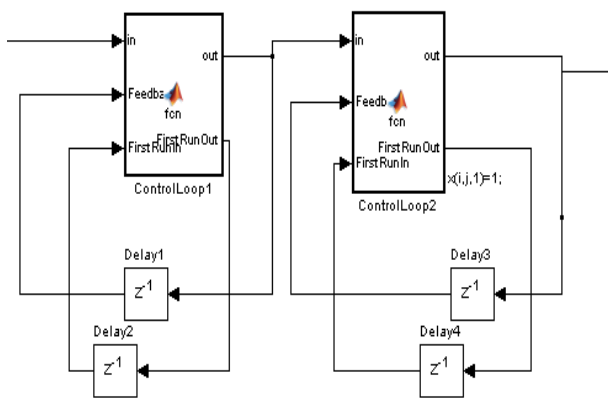


Fig 3: Control loops.

The edges detected from the depth map are more accurate than the edges detected from the image. A Roberts's edge detector is used to detect the edges.

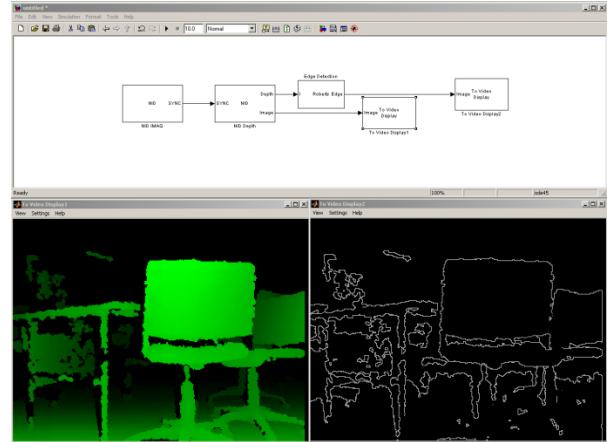


Fig 4: Edge Detection System.

### 3. HARDWARE AND SOFTWARE

MATLAB is a high-level language and Simulink is a tool for designing, simulating and analyzing dynamic systems in various domains.(Anon., 2010) Matlab allows matrix operations, implementing algorithms, plotting functions of data and interfacing devices. A grey scale image is represented in the form of a 2-D array with each pixel representing the value of intensity. A RGB image is represented in a 3-D array with three grey scale images of red, green and blue in an array. MATLAB R2012a with Computer Vision System Toolbox is used in this simulation. Microsoft® kinect is a sensor meant for X-box 360 console which enables users to interact with X-box 360 console without the need for a controller. This device can detect moving objects and track their motion using infrared projector and a stereo vision camera even in low light. (Anon., n.d.) .It has been observed that Kinect can map the depth of objects which are at a distance of 0.8m to 4m. The angle of view of the sensor is 57° horizontally and 43° vertically. (Microsoft, 2013) Interfacing Matlab to Kinect to acquire the depth images from the sensor requires a complex method. Please refer the Appendix for the procedure. The method which worked in the project is to include few additional Simulink blocks. Simulink® for Natural Interaction Device, to the Simulink browser.

### 4. IMPLEMENTATION

Natural Interaction Device for Simulink consists of six blocks, they are, NID IMAQ - This block acquires the data from the Kinect to be processed. It has an output port called 'SYNC'. This data is directly connected to the NID Depth/Motion/Image/Skeleton/IR blocks.NID Depth -This creates a depth image from the data acquired. It has two output ports. Depth -The output from this port a grey scale image with resolution 640X480. Each pixel of this image represents the distance of the object at that pixel in meters. Image -The output from this image is a RGB image which displays the depth in green color. NID Image - This block has an output

called Image which displays the actual image in RGB. NID IR – This block has an output port IR which returns the IR value at that pixel. NID Motion – This block detects the objects which are in motion and gives ID to each object give. Output the port motion is a grey scale image which each pixel denoted the ID(0 to 15) of object which is in motion and represents them in colors through the port Image. NID Skeleton – This block tracks the skeleton of the people in the image are represent the details of their position.

**5. MAPPING**

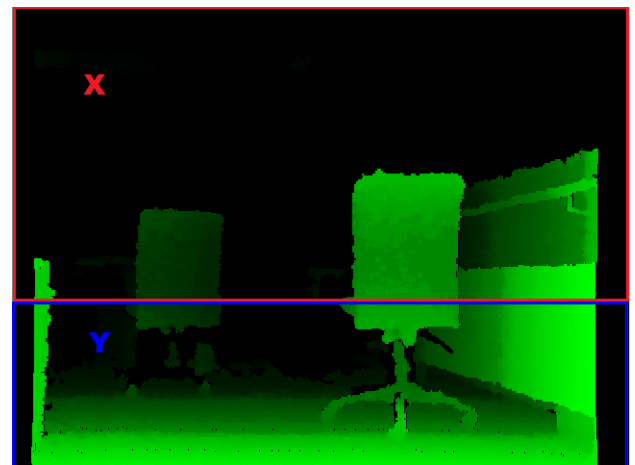
An image  $T$  representing the objects from the top view can be drawn from the data of the depth image, if  $D(i, j)$  is a pixel on the depth image, where  $i, j$  represent the row and column of the pixel and  $v$  is it's value., then a point is marked on the map at coordinates  $(n \times v, j)$ , where  $n$  is a constant. The pixels in the region  $X$  shown in **Error! Reference source not found.** are marked in red.

i.e If  $D(i, j) = v$ , then  $T(n \times v, j, m) = 1$

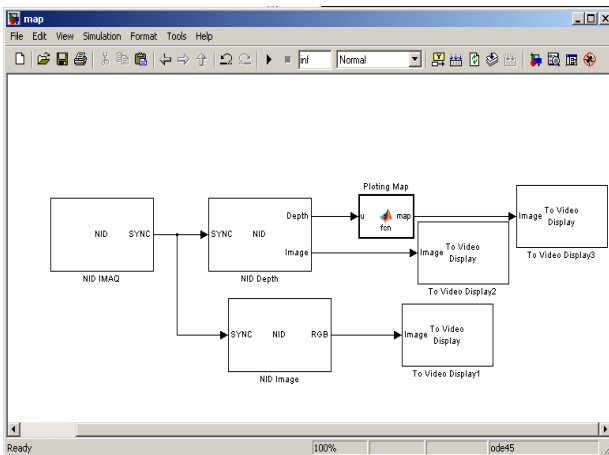
Where  $n = 200$ ,  $m = 3$  if  $D(i, j)$  lies in the region  $X$ ,  $m = 1$  if  $D(i, j)$  lies in the region  $Y$ . This is implemented in Simulink by using a MATLAB function. A written is developed in MATLAB language in the MATLAB function. Please refer the appendix for the code.



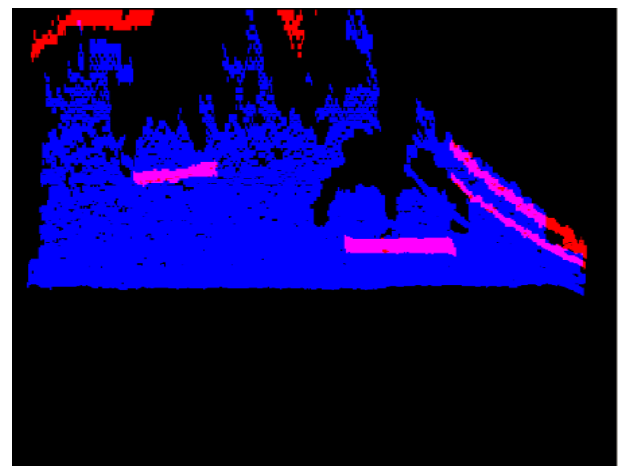
**Fig 6a:** shows actual image



**Fig 6b:** shows 2D image



**Fig 5:** Simulink model to generate map



**Fig 6c:** shows the RGB image, depth showing X and Y regions on MAP

## 6. CALCULATIONS

The safest direction to navigate to is calculated from the m. In order to mathematically calculate the safest direction, the following procedure is used. The pixels representing the obstacles are multiplied with the square their vertical coordinate. This is done to give a greater value for obstacles which are closer.

$$D(i, j, 1)^2 \times j$$

And the pixels representing the ground, blue value subtracted is multiplied by its vertical coordinate from a constant, here the constant is 1000. This is done to give a greater value for the surface detected at a greater distance

$$D(i, j) \times (1000 - j)$$

The difference is calculated in the vertical direction to form another array  $X$  of size  $j$ ,

$$X(j) = D(i, j, 3) \times (1000 - j) - D(i, j, 1) \times j$$

The values are then added along the vertical and put in an array of size  $j$ .

$$X(j) = \sum_{i=1}^{1000} [D(i, j, 3)^2 \times (1000 - j) - D(i, j, 1) \times j]$$

$X(j)$  represents the relative values of safeness to navigate in the direction of the column  $j$ .

So the column with the greatest value of  $X(j)$  could be considered as the best direction to navigate in.

Since the camera angle of the sensor is  $57^\circ$ . The angle in which the robot should rotate can be calculated by the formula.

$$\alpha = ((X(j) - 320) * 28.5) / 320$$

## 7. RESULTS AND CONCLUSIONS

This project demonstrates the basic idea to avoid obstacles and to find the safest direction for a robot. However there are many objectives which should be considered in order to design a better model for a robot. The map plotted in the image is not saved and only relates to the environment at that instance. A map which updates itself as the robot is moving could be designed. It can be seen that there is no particular destination for the robot to go to in this model. A model could be designed so that a destination could be given. Moving objects are detected by the Kinect but they have only been shaded in this model. More development on this could be done to avoid collision with the moving objects. Identifying the path to

navigate in by recognising the texture or markers on the path could be done.

The screenshots of the output images are shown in below figure 7. The original RGB image and the map plotted are shown in in the figures. In text in the figures shows the angle in degrees in which the robot should steer to 1 indicates left while 0 indicates right. The safest distance which could be detected by the system in meters is also shown in the image.

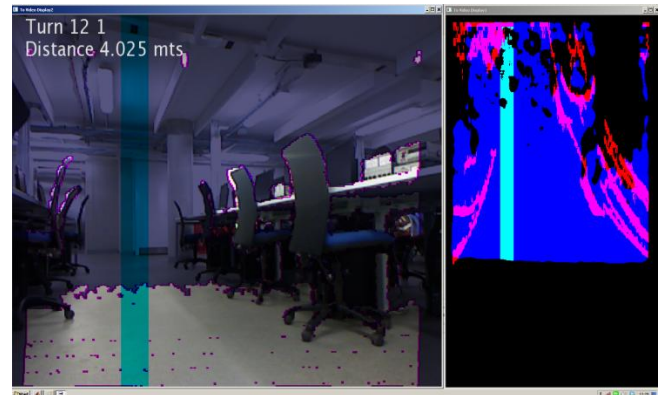


Fig 7: shows the output image1.

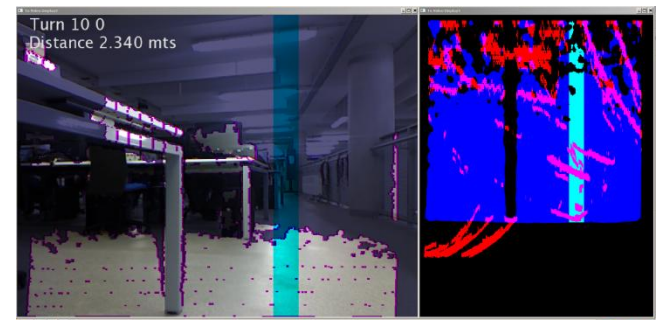


Fig 8: shows the output image2.

## REFERENCES

- [1]. Anon., 2010. Model based Design Accelerates the Development of Mechanical Locomotive Controls. SAE 2010 Commercial Vehicle Engineering Congress, p. 1 of 2.
- [2.] Anon., n.d. Primesense - MIT technology review. [Online] Available at: <http://www2.technologyreview.com/tr50/primesense/>
- [3]. Bertozzi, M., Broggi, A., Fascioli, A. & Nichele, S., 2000. Stereo vision-based vehicle detection, s.l.: IEEE.
- [4]. Hasan, A. H. R. & Johar, M. ,, 2009. Disparity mapping for navigation of stereo vision autonomous guided vehicle, s.l.: IEEE.
- [5]. Kelson R. T. Aires, A. M. S. A. A. D. M., 2008. Optical flow Using Color Information. New York, USA: ACM New York.

- [6]. Konolige, K., 1999. Stereo Geometry. s.l.:SRI International.
- [7]. Microsoft, 2013. Buy Kinect for X-box 360. [Online] Available at: [http://www.microsoftstore.com/store/msstore/en\\_US/buy/pageType.product/externalRefID.C737B081](http://www.microsoftstore.com/store/msstore/en_US/buy/pageType.product/externalRefID.C737B081)
- [8]. Rfisher, S. w. W., 2003. Spatrial Filters - Mean Filter. [Online] Available at: [homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm](http://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm)
- [9]. Rfisher, S. w. W., n.d. Spatrial Filters - Median Filter. [Online] Available at: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>
- [10]. S. S. Beauchemin, J. L. B., n.d. The computation of optical flow. New York, USA: ACM New York.
- [11]. S.S.a.b, D., 1999. Autonomous Cross-Country Navigation Using Stereo Vision, s.l.: Carnegi Mello University technical report.
- [12]. Tinku Acharya, A. K. R., 2005. Image Processing. s.l.:s.n.
- [13]. Wei, Z., 2008. Real-time accurate optical flow-based Motion Sensor. s.l.:s.n.
- [14]. Wilhem Burger, M. J. B., 2007. Digital Image Processing. s.l.:s.n.