MOTION TARGET TRACKING OF MOBILE ROBOTS

by

NOOR AIMAN BIN NOOR RASHID

17117

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronic)

SEPTEMBER 2015

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
32610 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

MOTION TARGET TRACKING OF MOBILE ROBOTS

BY

NOOR AIMAN BIN NOOR RASHID

17117

A project dissertation submitted to the

Electrical and Electronic Engineering Program

Universiti Teknologi PETRONAS

In partial fulfilment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTRICAL AND ELECTRONIC)

Approved by,

____

(Abu Bakar Sayuti bin Hj Mohd Saman)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

SEPTEMBER 2015

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

NOOR AIMAN BIN NOOR RASHID

ABSTRACT

In the study of mobile robotics, it is essential to be able to perform precise measurements of the rate of changes in motion, directions and accuracy of the robots under study. Therefore, development of a hardware-software system that can track and record the movement of a mobile robot will improve pertaining research activities. This report describes the development of such system using an overhead camera linked to a computer that make track, record and graphically visualize the motion of a mobile robot on a flat surface. Two main methods will be employed, i.e. background subtraction and brightest point. The main hardware and has been tested with basic programs based on the algorithms of the said methods. Two-dimensional coordinate measurement and calibration has been achieved with varying degrees of accuracy depending upon the resolution selected for use on the camera. Both motion tracking methods were done by tracking a bright reference point that was placed on a robot under testing, however approached with different image processing operations. A graphical path, coordinates, speed and total distance travelled will be generated at the end of the program for both motion tracking methods.

ACKNOWLEDGEMENT

First and foremost, I would like to express my utmost gratitude to Allah the Most Beneficent, the Most Merciful. Without His blessing, none of these achievements could be realized. I also would like to thank my parents for all the support given. Also to my supervisor, Mr Abu Bakar Sayuti bin Hj Mohd Saman for all the guidance and advices given throughout this final year project. Not forgetting my course mates and laboratory technicians who have been assisting me during the development of this project.

NOOR AIMAN BIN NOOR RASHID

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
1.0 INTRODUCTION

1.1 BACKGROUND

One of the many challenges faced by researchers in robotics is precision in movement and navigation in relation with time. With this information, the effectiveness of certain algorithms implemented on robots in terms of localisation accuracy, response time, erroneous manoeuvre and mapping errors could be determined. Therefore, it is essential to have a motion tracking system for robots in relative to time in order to observe the performance of the mobile robots' movements and determine which part needs improvement.

1.2 PROBLEM STATEMENT

There are many studies and researches conducted on mobile robots in terms of precision of movements and navigation. In order to continuously improve the movements of these robots, a monitoring system is required. To assist in the study of ground-based mobile robots, a motion tracking system is needed whereby the motion of the robots can be recorded, tracked and visualized.

1.3 OBJECTIVES

To develop a motion tracking system using overhead camera that is able to:

- Record the motion of mobile robots in a designated area
- Track the movement in real time and
- Visualize the information graphically on a personal computer

1.4 SCOPE OF STUDY

- The study of motion tracking algorithms for one robot using a static camera
- 2D coordinate system of robot movements
- Graphical overlay for robot's path

CHAPTER 2
2.0 LITERATURE REVIEW

2.1 Motion Tracking

The process of tracking a mobile object with respect to time using a camera is called video tracking or motion tracking and it has a lot of uses such as security surveillance, medical imaging [4] and video editing. Besides that, this process can also be implemented in tracking a mobile target such as mobile robots. Motion tracking can be done in a lot of ways. However, two methods will be considered in this project; *Background Subtraction* and *Brightest Point*.

Background Subtraction

One of the commonly used algorithm for motion tracking is background subtraction. In a video signal, this process is able to distinguish between background (the static part) and foreground (the dynamic part) of the video. The part of interest is the foreground which is produced by this algorithm by identifying the pixels that contain the moving objects in a given scene [3]. Figure 1 below shows how background subtraction operates.

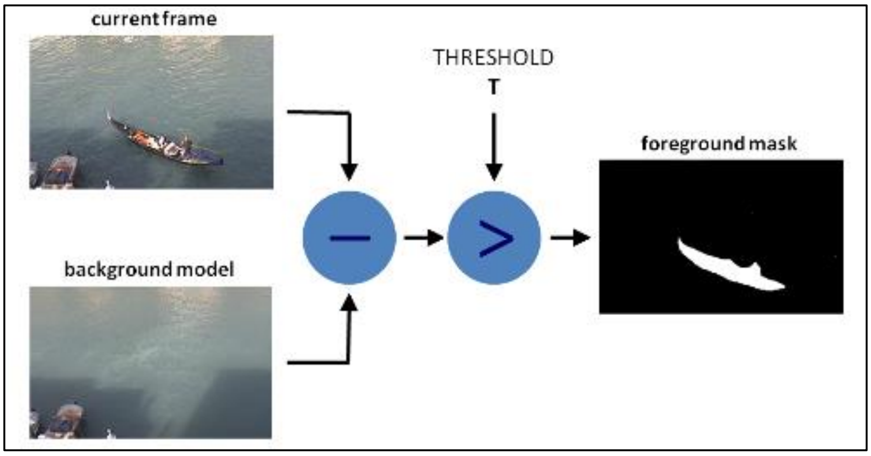


Figure 1: Background Subtraction

Background subtraction is an effective approach to motion tracking. However, it only detects the outline of the foreground or object of interest. This phenomenon might be useful for security surveillance but when location detection is concerned, it may require additional complex algorithms. The reason for this is a series of pixels make up the foreground (object), but the problem is which point of the object can be used as a continuous reference for precise tracking. To add to the problem, as the object moves from one point to another, the shape of the object changes from the camera's perspective [2]. Hence, reference point needed to be change to determine the object's location. Ultimately, if these problems could be addressed, background subtraction can be used in this project.

Brightest Point

Another approach to motion tracking is by tracking the brightest point of an object. With this algorithm, the object of interest (a robot) is required to be equipped with a reference point that is bright and in contrast with other elements of the scene.

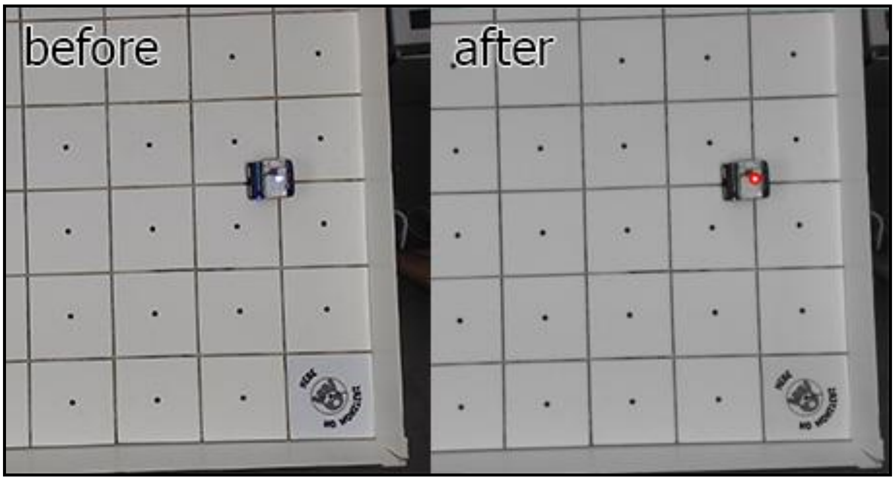


Figure 2: Image Processing Result of Brightest Point

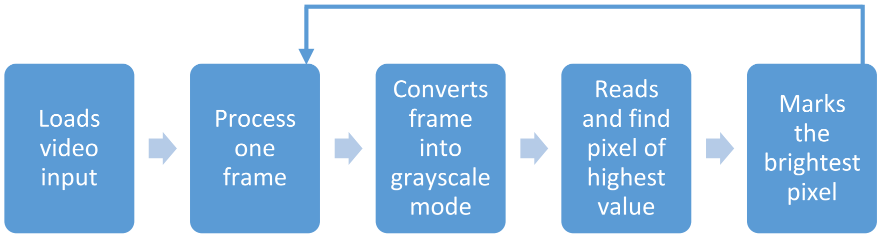

Loads video input
Process one frame
Converts frame into grayscale mode
Reads and find pixel of highest value
Marks the brightest pixel

Figure 3: Process Flow of Brightest Point

Since the bright reference point is small, the location of the robot can be identified with higher accuracy compared to background subtraction method. However, Odewahn and Borenstein stated that the result for this method can be disappointing in an environment with constantly-changing lighting conditions [8]. They further added a solution to this by controlling the lighting conditions to optimize the performance of this method.

2.2 2D Coordinate System

Motion tracking of a robot is useless without being able to define its location at a certain instance of time. This project considers the path of a robot on a 2D plane. Therefore, there is a need to study a means to declare the position of a robot on a 2D plane by using a processing software.

Pfister in 2002 had patented a measurement system which is able to determine tailor body measurements by using only one camera. The systems starts by a calibration process in which pixels per unit in both vertical and horizontal directions are determined [7]. The subsequent processes involved are irrelevant, however the calibration process implemented in this system can be very helpful in designing a 2D coordinate system for this motion tracking project.

2.3 Graphical Overlay

Overlay is required in this project to represent graphically the path that the robot follows. The main processing software called “Processing” is sufficient to provide the visuals required. For every detected point of reference, an ellipse shape can be drawn on the screen.

CHAPTER 3

3.0 METHODOLOGY

3.1 Tools

1) Laptop Computer
2) "Processing" Software by Processing Foundation
3) Matlab Software
4) Camera (DCS-932L), MJEPG video codec, max resolution of 640 x 480 at 20 fps, with IR LED for Night Viewing
5) Wireless Router (DIR-300), 54 Mbps, supports 802.11 b/g

3.2 Overall Setup

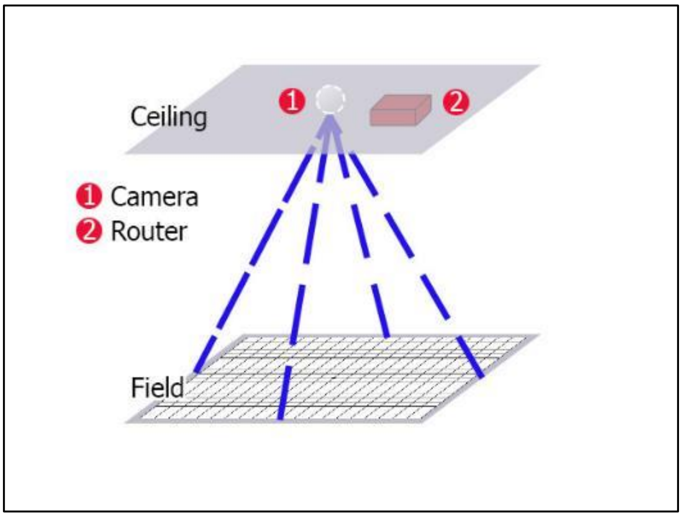


Figure 4: Isometric View

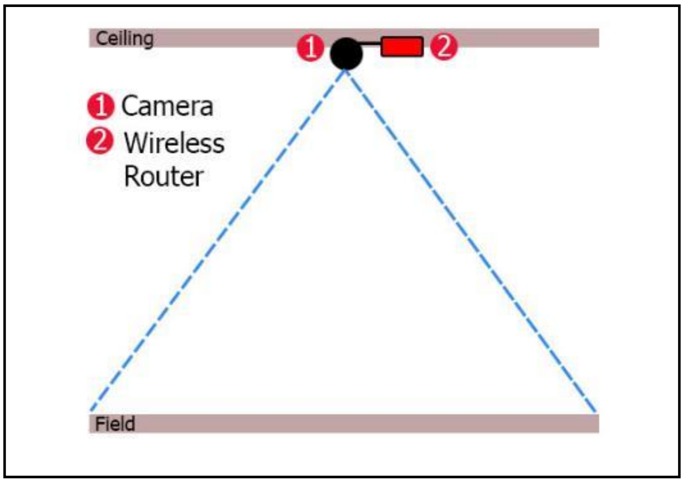


Figure 5: Side View

This project basically consists of three hardware; a camera, a wireless router and a laptop (not shown in Figure 4 above). Firstly, the mounted camera will scan the field size. Once ready, this system will start recording robot's movement on the field and relay this information to a laptop via a wireless router for motion tracking processes.

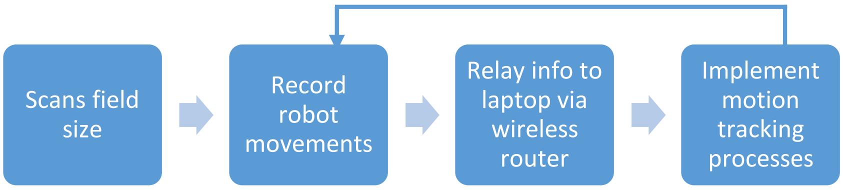

Scans field size
Record robot movements
Relay info to laptop via wireless router
Implement motion tracking processes

Figure 6: Operations Flow of Motion Target Tracking System

3.3 Robot Tracking

Robot tracking can be done in many ways such as segmentation, background subtraction, brightest point and more. Of the methods for motion tracking discussed earlier, brightest point has a higher potential to be implemented. It works by detecting a distinctive part of a given frame of a video such as a bright spot or a unique marker as motion references.

For this project, there are many possible elements that can be used as a motion reference. The robot to be tested will be equipped with an LED on top to allow the software to detect it as a distinctive bright spot as shown in Figure 7. For every frame of the video, this spot will be detected and marked to visualize the path of the robots as graphics overlay.

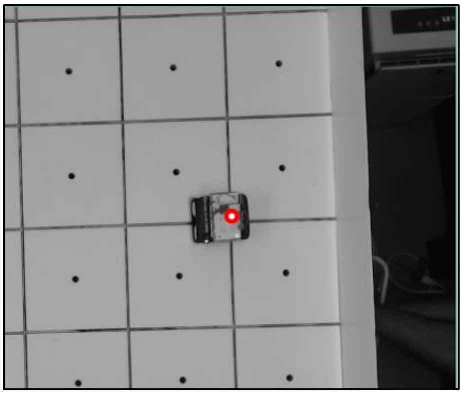


Figure 7: Sample of Motion Tracking and Graphic Overlay

3.4 2D Coordinate System

Once the movement of the robot is determined by a reference point, it is essential to translate the reference point in terms of location such as creating a virtual 2D Map that has X and Y axes. Subsequently, once the reference point of the robot is detected at a certain location in a given video frame, the location can be marked and defined a set coordinates.

By adopting the calibration method in [7], this can be achieved by first defining the relationship between pixels and length. To attain this relationship, a reference line of known length have to be included and calibrated by drawing a line on the screen which is indicated as a red line in Figure 8. Next, any two vertices of the field that are diagonally-opposite to each other were declared to form a rectangular outlining the test field for the robot.

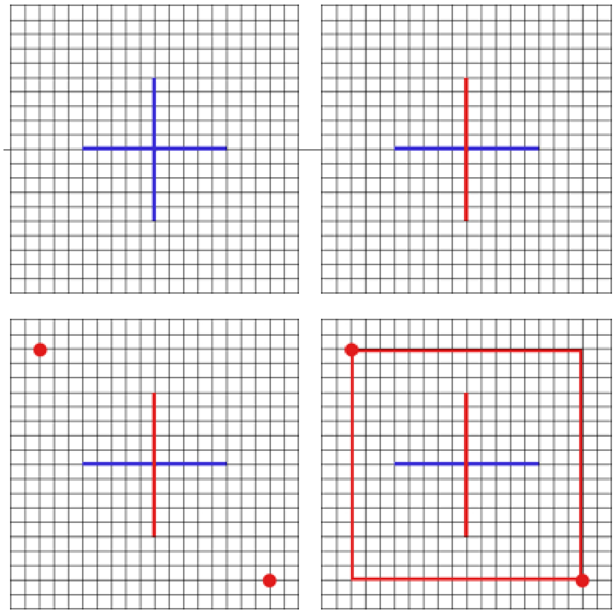


Figure 8: Calibration of Coordinate System

For example,

- The detected pixels for width and length are 500 pixels by 400 pixels respectively.
- The reference line is 1 meter and the detected pixel length of the line is 100 pixels

This yields,

$$Pixel\ to\ Length\ Factor\ (p) = 100\ pixels/m$$

Therefore,

$$Actual\ Width = \frac{500}{p} = 5\ meters$$

$$Actual\ Height = \frac{400}{p} = 4\ meters$$

For this method, there is also a need to study the optimal height of the camera from the ground although the decision may be quite intuitive. Depending on the quality of the camera, mounting it too high may result in reduced precision of measurement during calibration. Since the calibration points needs to be manually chosen by a user, the resulting points for a highly-mounted camera may have higher coordinate errors as compared to a camera that is optimally-mounted which may yield lower errors.

To define the location of the robot, virtual grids can be used either visibly or invisibly. However, the size of each grid plays an important role in the precision of locating the robot. Big-sized grids will lead to less sensitive measurements, meaning the reading of the coordinates may not change even though the robot moved a little. Inversely, making the grids smaller will lead to more sensitive measurements, consequently providing more accurate coordinates. The effect of grid sizes are demonstrated in Figure 9 and Figure 10.

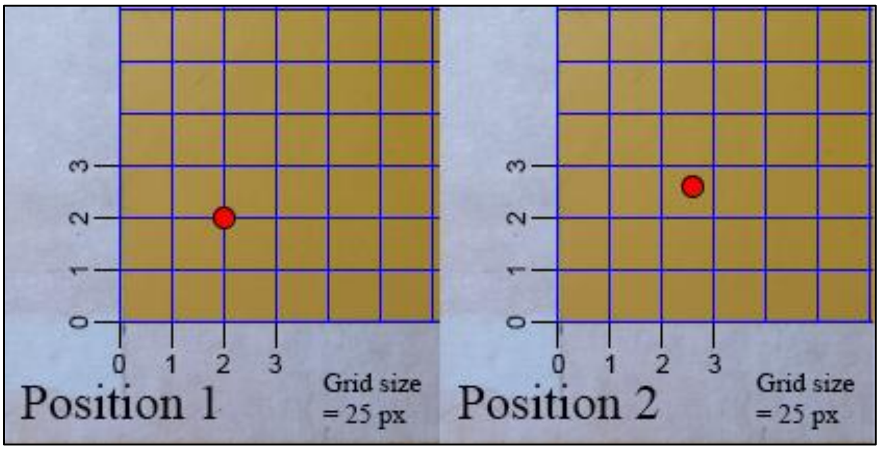


Figure 9: Position 1, Grid Size = 25 px; Position 2, Grid Size = 25 px

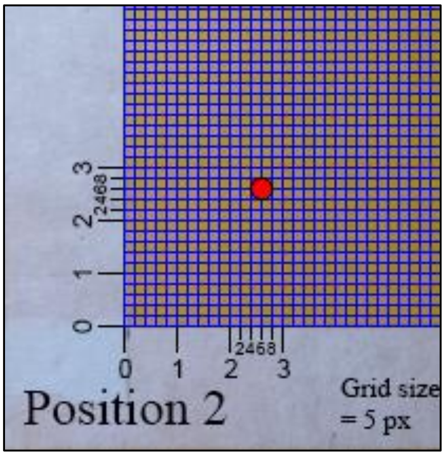


Figure 10: Position 2; Grid Size = 5 px

3.5 2D Coordinate and Calibration System Testing Setup

A setup was done to observe the accuracy of the coordinate system on a 20 cm by 20 cm field size with a reference line of 10 cm shown in Figure 11 and Figure 12. As mentioned before, this process will be done manually before starting motion tracking from the program. From this process, the width and height of the test field can be determined.

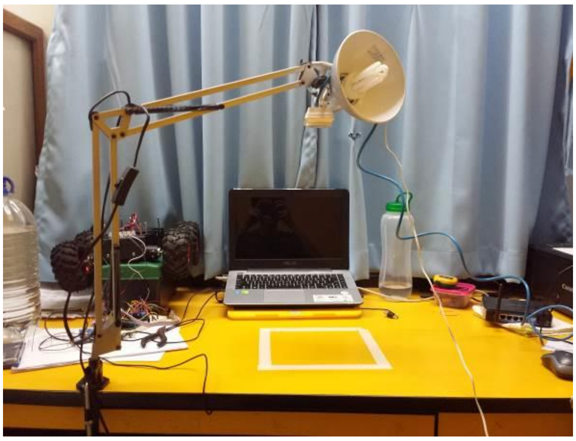


Figure 11: 2D Coordinate System Testing Setup

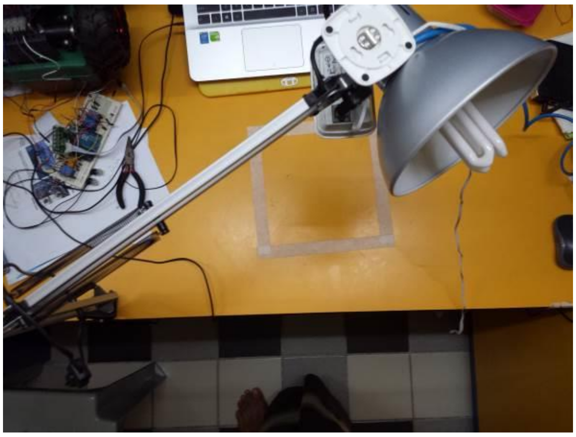


Figure 12: Top View of the Setup

3.6 Implementation of Brightest Point

As discussed earlier, Brightest Point tracks motion by detecting the brightest point of a video feed frame by frame. The point will then be marked and accumulated for every frame, which finally resembles the path that a robot takes. The testing area for Brightest Point method was set up as shown in Figure 13 and a bright yellow ping pong ball was used as the target to be tracked.

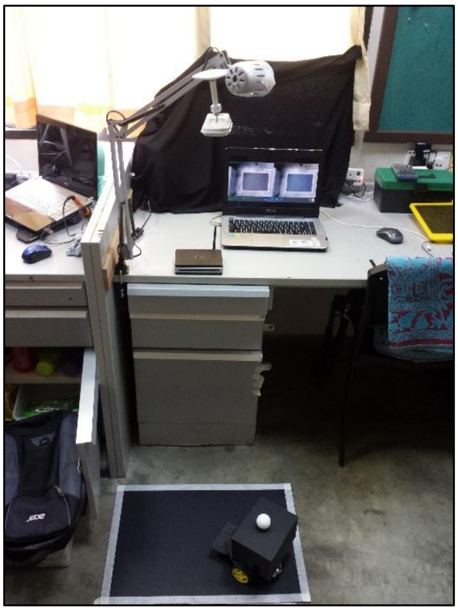


Figure 13: Testing Setup for Motion Tracking

The Brightest Point is quite a straightforward method to track motion. It detects only the brightest pixel of a video frame. This is achieved by comparing all the pixels that make up the frame which has the highest value of brightness.

However, simply applying the discussed algorithm without prior processes to the frames may not yield consistent results due to various bright points on testing floor. Hence, pre-processing is important to further set the brightness values apart.

3.7 Implementation of Background Subtraction

The basis of Background Subtraction is simply separating out foreground elements from the background in a sequence of video frames [9]. One way to achieve this is by comparing a sequence of frames to a model background [9] and this process is called image differencing. However, sometimes in low-lit environment the quality of the video frames received may degrade due to the appearance of scan lines as shown in Figure 14. Scan lines appear randomly and if this frame is used to compare with other frames, these scan lines will be detected, subsequently affecting motion tracking of the ping pong ball.

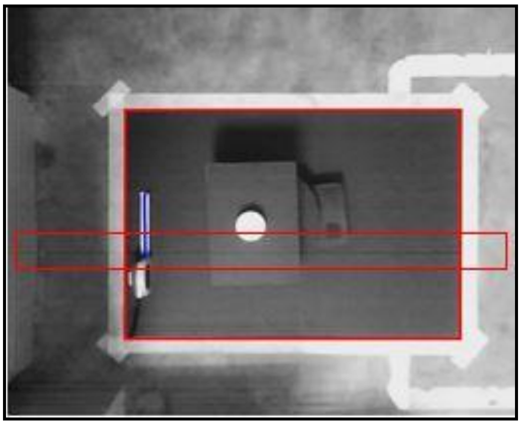


Figure 14: Scan Lines

To prevent this, the background subtraction was approached in a different way which is by using threshold. Every pixel on the frame takes a certain brightness value. The brighter the pixel, the higher the value. This property of pixel can be used to filter the pixels that are desired. Threshold function of a certain value was used filter these pixels that make up the ping pong ball since it is one of the the brightest objects of the frame.

However, it can be observed in Figure 14 that the boundaries of the test field are also bright. If threshold were done globally (entire frame), it would pick up the boundary as a reference as well, hence causes undesired detections. To overcome this problem, threshold process will be done based on region of interest. Region of interest allows further processes to be done only within the specified area of interest, consequently eliminating any undesired detection of the boundaries or any other element beyond the boundaries. Another advantage of this technique is that the processing power and time will be reduced and thus, resulting in less lag in real-time processing.

Before contour detection is implemented, the frame requires edge detection process to assist the contour detection process. Upon completing contour detection process, points that make up the contours will be generated. Based on the coordinates of the points that make up the contour, the centroid can be located, tracked and marked for every frame.

3.8 Robot Specifications

To test the effectiveness of the motion tracking system, a robot as shown in Figure 15 with the following specifications will be used. For tracking purposes, the robot was equipped with a line sensor to allow it to move around within the test field without exceeding the boundary. The programming of the robot was done on the ROMEO BLE board using Arduino IDE.

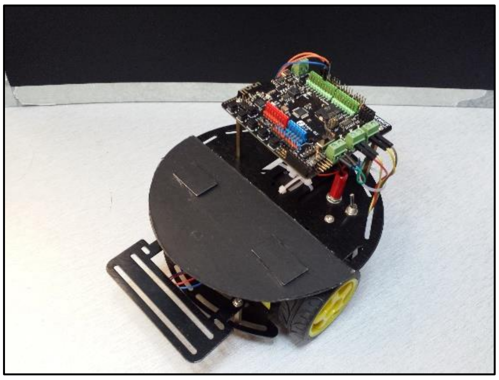


Figure 15: 2WD Mobile Platform

- 2-Wheel Drive Mobile Platform
- 2 DC Motors
- ROMEO BLE board
- UltraFire Li-Ion Battery 3.7V, 3000 mAh
- Cytron Line Sensor LSS05

There are areas on the robot which are reflective like the metal platform for instance. If uncovered, these areas will be picked up by the system as a reference point due to its high intensity. Therefore to prevent this, the platform was covered entirely with cardboard and the ping pong ball which acts as a reference point was placed on top of the robot as shown in Figure 16.

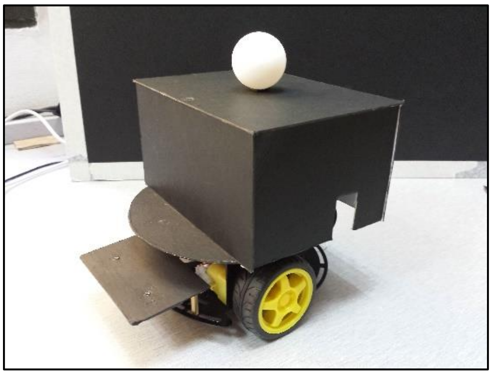


Figure 16: Covered Robot Platform

3.9 Project Activities

The following Gantt charts shows the main activities of the project and their progress to date.

- Camera-Laptop Communication Setup
- Identify Tracking Algorithms
- Identify method for Coordinate Systems
- Development of Coordinate System
- Develop a Setup for Coordinate System
- Development of Motion Tracking System
- Develop a Setup for Motion Tracking Testing
- Develop GUI for Results in Processing
- Integrate motion tracking, coordinate system, and GUI
- Troubleshoot and Modification
- Finalize system and setup for project demonstration

Week Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Camera-Laptop Communication Setup		▨	▨											
Identify Tracking Algorithms		▨	▨	▨	▨									
Identify Method For Coordinate Systems		▨	▨	▨	▨									
Testing of Tracking Algorithms			▨	▨	▨	▨	▨							
Testing of Coordinate Systems			▨	▨	▨	▨	▨	▨	▨	▨	▨			
Development of Coordinate of System									▨	▨	▨			
Develop a Setup for Coordinate System											▨	▨	▨	

Figure 17: Gantt Chart for FYP 1

Week Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Development of Motion Tracking System		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨			
Develop a Setup for Motion Tracking Testing		▨	▨	▨	▨									
Develop GUI for Results in Processing						▨	▨	▨	▨	▨	▨	▨	▨	▨
Integrate motion tracking, coordinate system, and GUI									▨	▨	▨	▨	▨	▨
Troubleshoot and Modification									▨	▨	▨	▨	▨	▨
Finalize system and setup for project demonstration									▨	▨	▨	▨	▨	▨

Figure 18: Gantt Chart for FYP 2

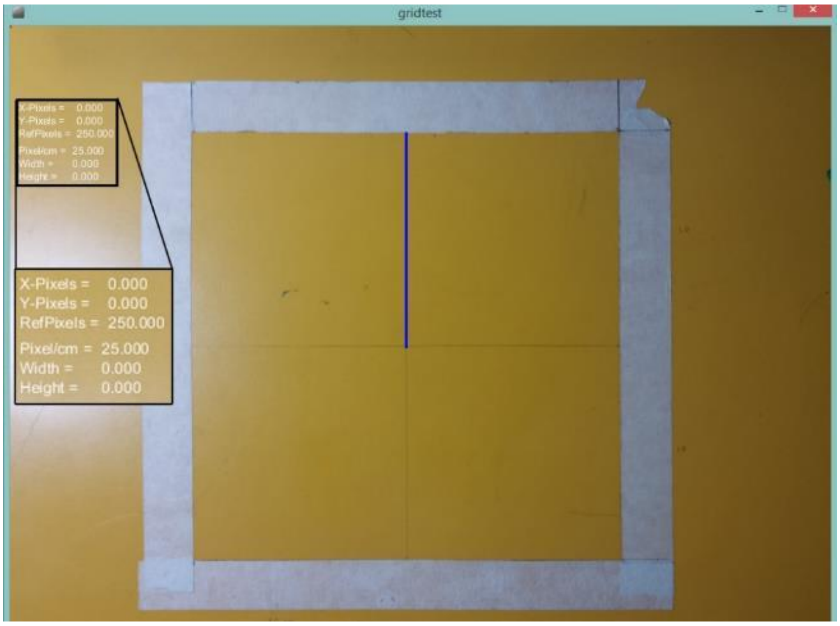
3.10 Key Milestone

- Camera-Laptop Communication – OK
- Basic Coordinate System Programming – Done
- Basic Motion Tracking Programming – Done
- Basic GUI development – Done
- Develop a Setup for Coordinate System - Done
- Testing of Coordinate System – OK
- Development of Motion Tracking System – Isolation OK; Issues regarding centroid location
- Develop a setup for Motion Tracking Testing – Done
- Develop GUI for results in Processing – Done
- Integrate motion tracking, coordinates system, and GUI – Done
- Troubleshoot and Modification – Done
- Finalize system and setup for the project demonstration – Done

CHAPTER 4
4.0 RESULTS

4.1 2D Coordinate system

As explained earlier, the system is initiated by a calibration process first. The most important part of calibration process is to generate the Pixel-Length factor, p. This was achieved by clicking on both ends of the reference line. The generated reference line is blue in colour and the Pixel-Length factor is shown on Figure 19 as “Pixel/cm” which for this example shown **25**.



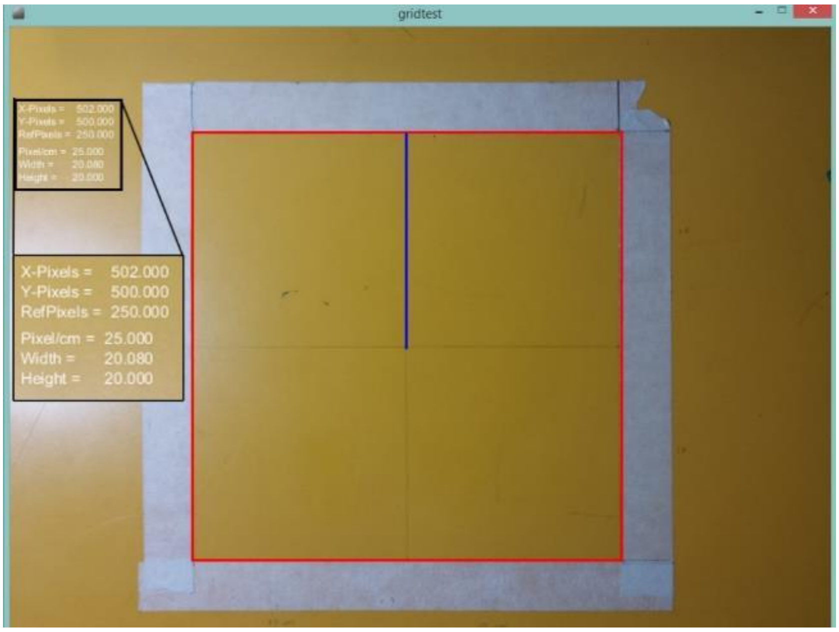
```
X-Pixels =   0.000
Y-Pixels =   0.000
RefPixels =  250.000
Pixel/cm =  25.000
Width =      0.000
Height =     0.000
```

```
X-Pixels =   0.000
Y-Pixels =   0.000
RefPixels =  250.000

Pixel/cm =  25.000
Width =      0.000
Height =     0.000
```

Figure 19: Step (1) – Reference Line

Next, the size of the field was define by first clicking any two vertices that are diagonally-opposite to each other. As a result a square or a rectangle depending on the shape of the field was generated indicated by the red square as shown in Figure 20.



```
X-Pixels =    502.000
Y-Pixels =    500.000
RefPixels =  250.000

Pixel/cm =  25.000
Width =      20.080
Height =     20.000
```

```
X-Pixels =    502.000
Y-Pixels =    500.000
RefPixels =  250.000

Pixel/cm =  25.000
Width =      20.080
Height =     20.000
```

Figure 20: Step (2) – Field Size

Based on the results obtained in Figure 20, it is shown that the width and height of the field is **20.08 cm** by **20.00 cm** respectively. It was mentioned before that the size of the field is 20 cm by 20 cm, which means the width has an error of 0.08 or 0.4%. Hence, this test proves that the algorithm used to develop 2D coordinate system is reliable with very minimal error. However, it is important to note that the accuracy of this system is dependent on how well the test field is calibrated since it is done manually.

4.2 Brightest Point

Based on what has been described in the methodology section, the Brightest Point algorithm consists of the following processes:

1) Captures a frame of a video
2) Thresholds the frame
3) Detects the brightest pixel

A brightness function was used to retrieve the brightness value. As stated earlier, running the Brightest Point process directly after capturing a frame may result in inconsistency of tracking as shown in Figure 21. This problem occurs due to uneven lighting on the target object which is inevitable, considering the basic setup.

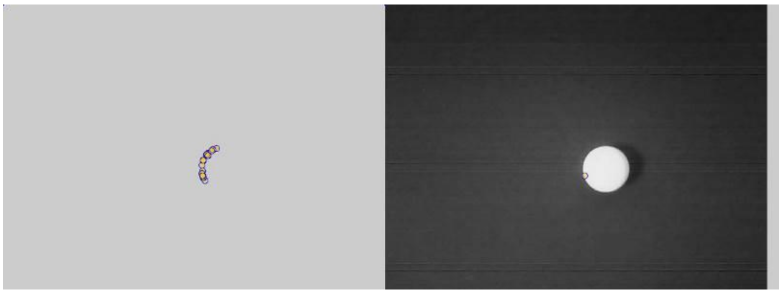


Figure 21: Brightest Point without Threshold

However, when the frame is given a threshold process prior to Brightest Point process, the resultant frame is a binary image. Which means, the whole target (ping pong ball) is now a group of bright pixel as shown in Figure 22. This way, other elements such as lighting on the testing floor can be filtered out and ruled out as false detection. The end result of the Brightest Point method is shown in Figure 23.

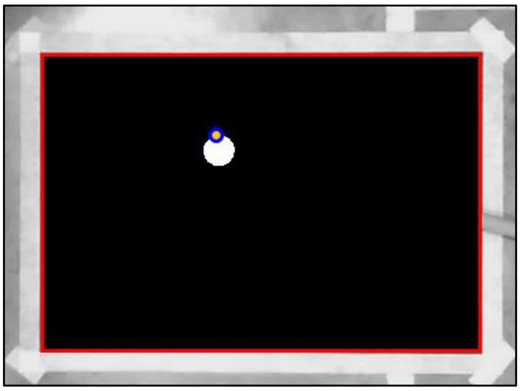


Figure 22: Brightest Point with Threshold

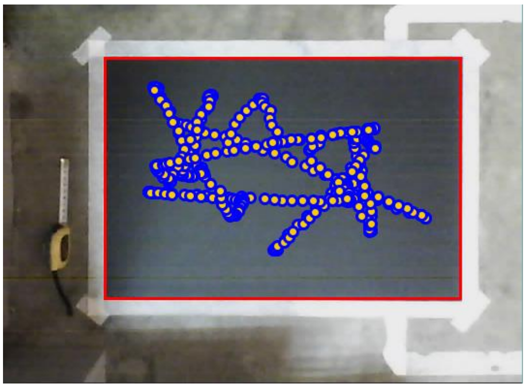


Figure 23: Brightest Point Result

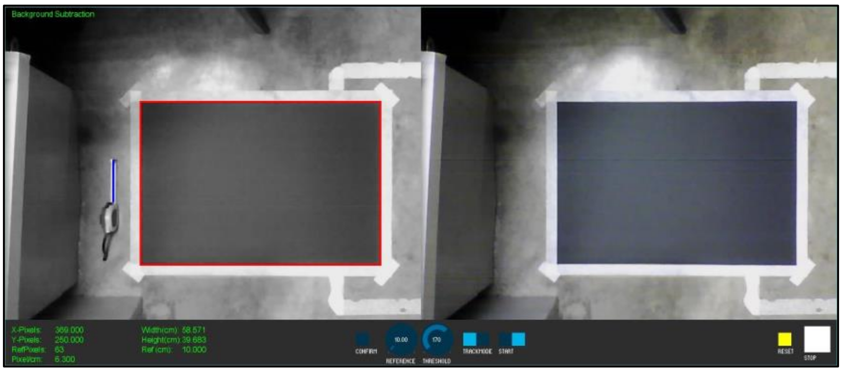
Despite having stable motion tracking, this method is still inaccurate due to the fact that it detects the top edge of the ping pong ball. This means that the center of the reference cannot be identified, resulting in a less accurate result.

4.3 Background Subtraction

Based on what has been described in the methodology section, the Background Subtraction algorithm consists of the following processes:

1) Region of Interest
2) Threshold
3) Edge Detection
4) Find Contours
5) Find centroid from points of contours

Upon finishing the calibration of the test field, the region of interest will be automatically declared with the size of the test field as shown in Figure 24.



X-Pixels: 369.000
Y-Pixels: 250.000
RefPixels: 63
Pixel/cm: 6.300

Width(cm): 58.571
Height(cm): 39.683
Ref (cm): 10.000

Figure 24: Calibrated Test Field

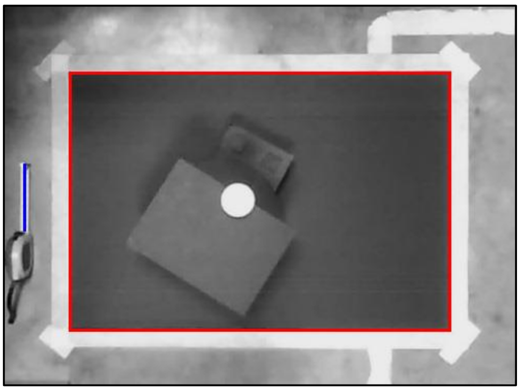


Figure 25: Robot Ready to be Tracked

To isolate the ping pong ball reference shown in Figure 25, background subtraction was done by the means of threshold which is shown in Figure 26. Then, the edge detection was done by using Canny Edges function from OpenCV as shown in Figure 27. At this point, “find contours” function was done to bind the edges formed from the edge detection process. The result can be seen in Figure 28 where an array of points make up the contour.

At this point, the motion of the target still cannot be tracked with respect to its centroid, considering the centroid of the contour is yet to be identified. By manipulating the points that make up the contour, the centroid can be located. To achieve this, minimum and maximum values of x and y coordinates from all the array of points that make up the contour were used. By using these values midpoint of these two axes can now be determined by adding the minimum and maximum values of each axes and divided by two.

Hence, for each frame the centroid can be marked and later resembles the path that the robot or target is taking. From the coordinates of these contours, the center of the reference can be determined as illustrated in Figure 29.

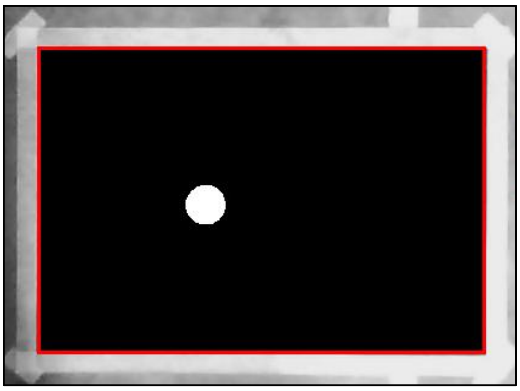


Figure 26: Threshold Process

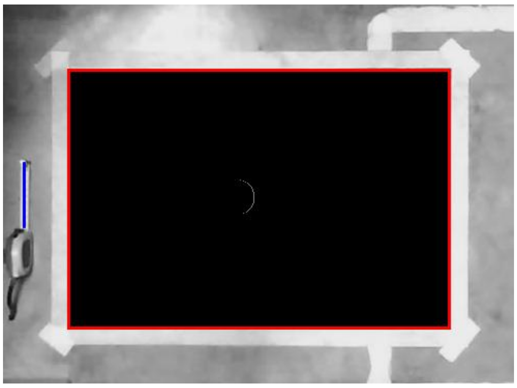


Figure 27: Canny Edges

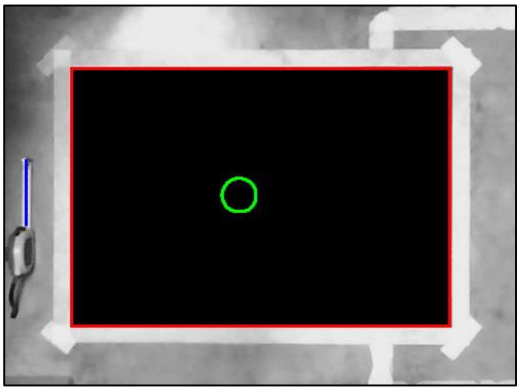


Figure 28: Find Contours

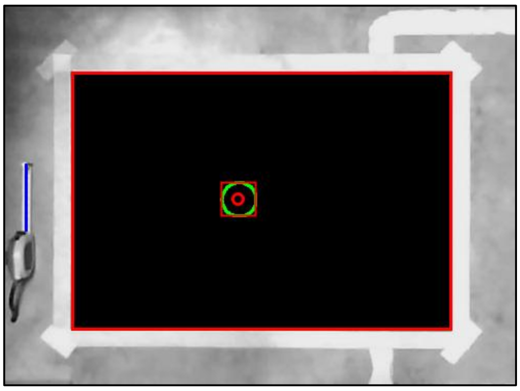
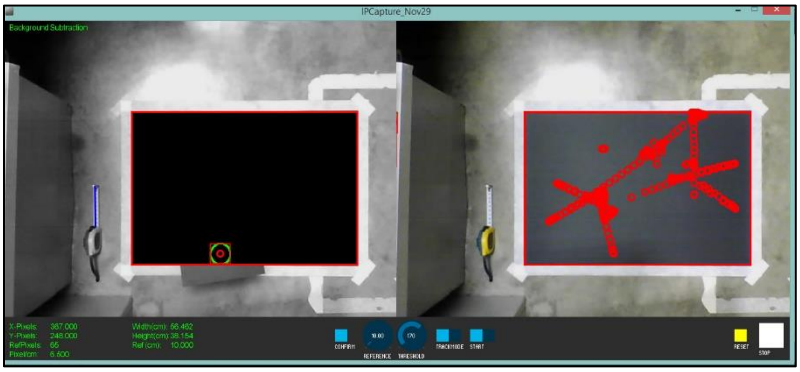


Figure 29: Localization of Centroid

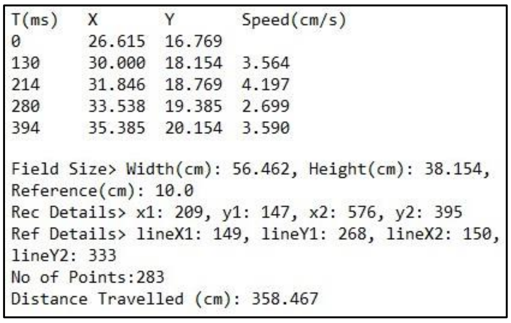
These processes will be repeated for every frame received from the camera for the Background Subtraction Method. The end result of this method is shown in Figure 30.

For both Brightest Point and Background Subtraction methods, a screenshot of the last frame from the program shown in Figure 30 will be stored as well as an output data file containing the coordinates of centroid against time, speed, distance travelled and test field measurement details as shown in Figure 31.



```
X-Pixels:   367.000        Width(cm): 56.462
Y-Pixels:   248.000        Height(cm):38.154
RefPixels:  65             Ref (cm):    10.000
Pixel/cm:   6.500
```

Figure 30: Background Subtraction Result



```
T(ms)   X       Y       Speed(cm/s)
0       26.615  16.769
130     30.000  18.154  3.564
214     31.846  18.769  4.197
280     33.538  19.385  2.699
394     35.385  20.154  3.590

Field Size> Width(cm): 56.462, Height(cm): 38.154,
Reference(cm): 10.0
Rec Details> x1: 209, y1: 147, x2: 576, y2: 395
Ref Details> lineX1: 149, lineY1: 268, lineX2: 150,
lineY2: 333
No of Points:283
Distance Travelled (cm): 358.467
```

Figure 31: Output Data File

CHAPTER 5

5.0 CONCLUSION

In conclusion, the motion target tracking system had successfully achieved the objectives which are to record the motion of robots, track the motion in real time as well as visualize the path graphically.

It can be seen both Brightest Point and Background Subtraction methods can be used to track motion. However, Brightest Point suffered from a little flaw in terms of localization in which it only detects the top edge of the ping pong ball. The reason behind this is because Brightest Point uses a much more simplistic approach compared to Background Subtraction in which it detects the first bright pixel it encounters. Since the whole ping pong ball is of the same pixel intensity due to threshold process, the top most and left most pixel of the ping pong ball would be detected every time.

Background Subtraction on the other hand provides a much more accurate results in terms of localization. This was made possible due to the addition of centroid localization algorithm. Since it uses contour points to represent the shape of the ping pong ball, the coordinates of these points were manipulated to obtain the center of the reference point.

For both of these methods, the accuracy of the motion tracking system is fully dependent on how accurate the manual calibration of the 2D coordinate system was done.

CHAPTER 6
6.0 RECOMMENDATION

It was found that the calibration process may affect the accuracy of the tracking system because it is a process done manually. Therefore, in future this process is recommended to be done automatically by using corner detection algorithm which can automatically detect corners. Apart from that, the angle of the camera was set up also may affect accuracy due to tilting errors. A warp perspective algorithm which can automatically correct tilted images into a perfect top view can be incorporated into this system to reduce this type of error.

From the motion tracking standpoint, both brightest point and background subtraction can currently track only one reference point per frame. To expand the system’s uses and robustness, multiple point tracking can be featured in this system. With this feature, the coordination and accuracy of locomotion of a swarm of robots can be tracked.

REFERENCES

[1] KaewTraKulPong, P. and R. Bowden (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. Video-based surveillance systems, Springer**:** 135-144.

[2] Motwani, M., et al. (2010). Towards Benchmarking of Video Motion Tracking Algorithms. Signal Acquisition and Processing, 2010. ICSAP '10.

[3] Vacavant, A., et al. (2013). A benchmark dataset for outdoor foreground/background extraction. Computer Vision-ACCV 2012 Workshops, Springer.

[4] Mountney, P., et al. (2010). "Three-Dimensional Tissue Deformation Recovery and Tracking." Signal Processing Magazine, IEEE 27(4): 14-24.

[5] Nihei, Y., et al. (2014). A method of localisation and multi-layered 2D mapping using selective update for particle filter. Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference.

[6] Lee Siau, Y., et al. (2008). Developing a blind robot: Study on 2D mapping. Innovative Technologies in Intelligent Systems and Industrial Applications, 2008. CITISIA 2008. IEEE Conference.

[7] Pfister, H. (2002). Camera measurement system, Google Patents.

[8] Odewahn, A., & Borenstein, G. (2014, May 22). Atduskgreg/opencv-processing-book. Retrieved June 21, 2015, from https://github.com/atduskgreg/opencv-processing-book/blob/master/book/tracking/brightest_point.md

[9] V. H. Carvalho, “Automatic Visual Surveillance of Vehicles in Complex Scenarios,” in Image Processing: Methods, Applications and Challenges, US: Nova Science Publishers, 2012, pp. 234–235.