UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Autonomous indoor navigation for a wheeled robot

Diploma Thesis

Papadomanolakis Eleftherios

Supervisor: Lalis Spyros

Volos 2020

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Autonomous indoor navigation for a wheeled robot

Diploma Thesis

Papadomanolakis Eleftherios

Supervisor: Lalis Spyros

Volos 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αυτόνομη εσωτερική πλοήγηση για τροχοφόρο ρομπότ

Διπλωματική Εργασία

Παπαδομανωλάκης Ελευθέριος

Επιβλέπων: Λάλης Σπύρος

Βόλος 2020

Approved by the Examination Committee:

Supervisor **Lalis Spyros**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Bellas Nikolaos**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Antonopoulos Christos**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Date of approval: 20-9-2020

Acknowledgements

I would like to thank all the people that helped me in the whole journey of my studies, my professors for entrusting me with the lab's equipment, and my family for always supporting me to go forward. I also want to thank the whole open-source community (GNU/Linux, Numpy, ROS, OpenCV, Arduino, etc) for making it possible to perform this project without requiring any proprietary software.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Papadomanolakis Eleftherios
15-9-2020

Abstract

The abstract includes the scientific area and the purpose /subject of the thesis, the methodology, the main steps followed and the main results obtained. The total extent of the abstract will be up to one page.

A thesis written in english should include an english and a greek abstract.

Abstract

Η περίληψη περιλαμβάνει την επιστημονική περιοχή και το σκοπό - αντικείμενο της εργασίας, τη μεθοδολογία, τα κύρια βήματα που ακολουθήθηκαν και τέλος τα κύρια αποτελέσματα. Η συνολική έκταση της περίληψης θα είναι μέχρι μία σελίδα.

Διπλωματική εργασία γραμμένη στα ελληνικά, θα πρέπει να περιλαμβάνει ελληνική και αγγλική περίληψη.

Contents

List of Figures

List of Tables

Abbreviations

βλπ	βλέπε
κ.λπ.	και λοιπά
etc	et cetera is
WIFI	Wireless Fidelity
BPF	Band Pass Filter
IP	Internet protocol
RGB	Red, green, blue
ROS	Robot Operating System
SLAM	Simultaneous localization and mapping
SSH	Secure shell
URDF	Unified robot description format
XML	Extensible markup language
3D	3 Dimensions
GPIO	general-purpose input/output
USB	Universal Serial Bus
IMU	Inertial Measurement Unit

Chapter 1

Introduction

1.1 Subject of thesis

Εδώ αναφερόμαστε συγκεκριμένα στο τί θα κάνει η διπλωματική. Αναφέρουμε λεπτομερώς α) τα προβλήματα που θα λύσει (και που ήδη έχουν περιγραφεί γενικά στην προηγούμενη ενότητα), και β) πώς σκοπεύει να τα λύσει.

1.2 Organisation of this thesis

At 2nd chapter we present the hardware and software that was used to design and build the robotic system. Subsequently, at the 3rd we show how every component was integrated into the system and the steps needed to get it started. At the 4th chapter discusses the line recognition algorithm that was used and the optimizations made to achieve better performance/accuracy with the hardware present. At the 5th chapter should present photos/results of the autonomous robot working (pros/cons)

Για την τελική οργάνωση του κειμένου σας, συμβουλευθείτε τον επιβλέποντα της εργασίας.

Chapter 2

The Robot System

2.1 System Architecture

In this section, we will present the hardware we used to construct the robot/system with a briefing explanation on why each was chosen. The components listed are a Raspberry pi 3 (model B+), an Arduino mega 2560 board, 5 ultrasonic sensors, GY521 based on mpu6050, md25 Motor Drive with emg30 motors.

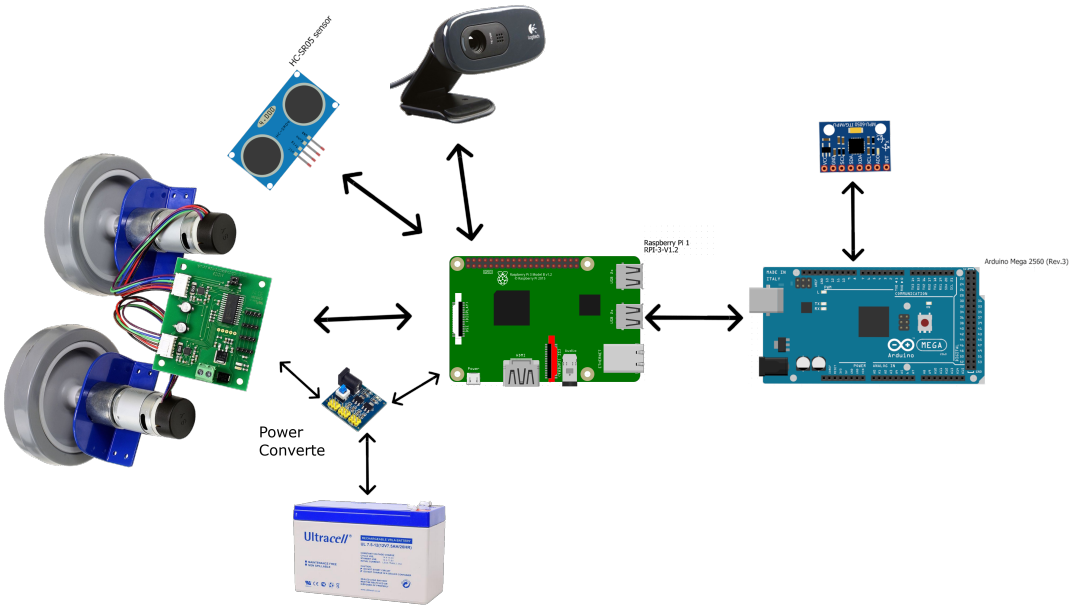

Figure 2.1: System Diagram

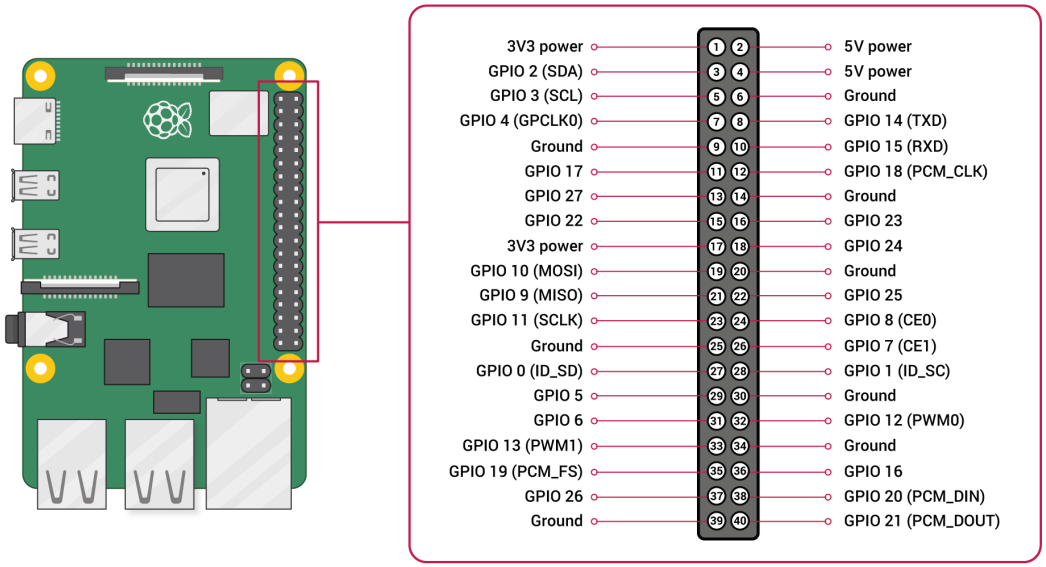

Figure 2.2: Raspberry pi 3 Model B+

2.2 Robot hardware

2.2.1 Raspberry pi

A raspberry pi 3 model B+ (fig.**??**) was employed for the MASTER because of its remarkable versatility in use. It is used in many IoT applications due to good hardware to price ratio, excellent documentation, and fairly big growing community, while various operating systems, each for their purpose, can be installed such as Ubuntu, Kali or RISC OS (original ARM OS), which can be simply exchanged with a switch of micro-SD cards. It comes with 1.2Ghz Broadcom BCM2837 quad-core ARMv8 processor, 1GB RAM, a 400MHz VideoCore IV multimedia GPU, build-in WiFi and bluetooth, as well as, several interfaces to connect a wide variety of sensors and gadgets. In order to power it up, a power converter was used to convert 12V AC from battery to 5v/ 2A DC while it needs [1] 5v supply and the current draw is from 600mA to 1.2A (max).

This board will serve as our master node or a gateway to an external server, as it's the one with the most processing capabilities. -include WiFi speeds/bandwidth?

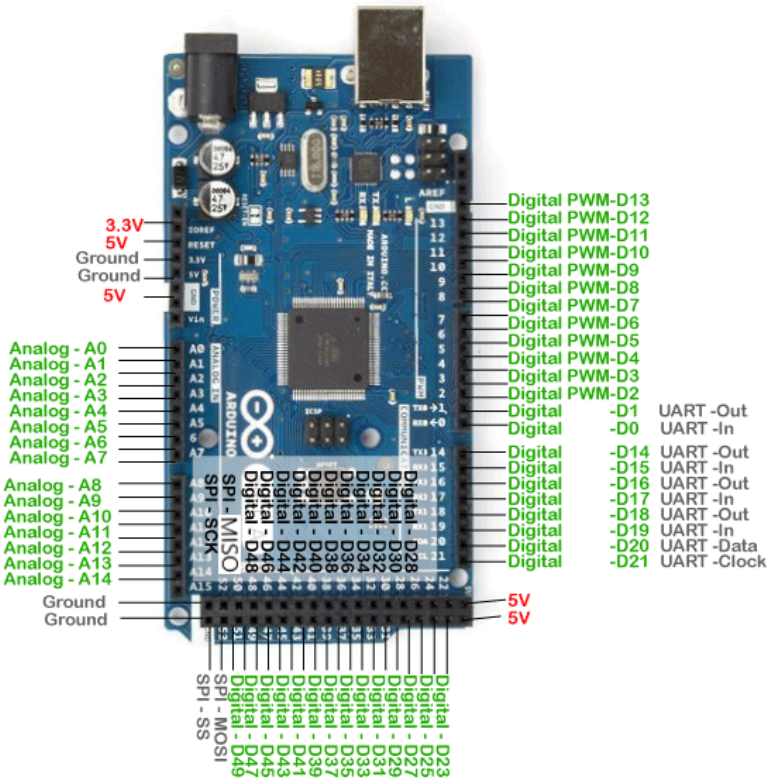

Figure 2.3: Arduino Mega 2560

2.2.2 Arduino board

The Arduino Mega 2560 (fig.2.3) is a microcontroller board and was mainly chosen to be used to relieve the Raspberry pi of all the tasks it has to do and to take advantage of the Digital Motion Proccesor (DMP) contained in the MPU6050 imu sensor. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. We can simply connect it to the raspberry with a USB cable in order to get it started, as well as, to connect with the master node using serial communication.

2.2.3 MD25 board and EMG30 motors

For the mobility of the robot, the MD25 board, a dual motor driver, was adopted, which is designed for use with the EMG30 motors (fig. 2.4), and is utilized with I2C communication protocol. The conjoint of these two allows us to drive two motors with independent or combined control as well as to read motor encoders and provide counts for determining the distance traveled and direction. Furthermore, input battery voltage and motor current are

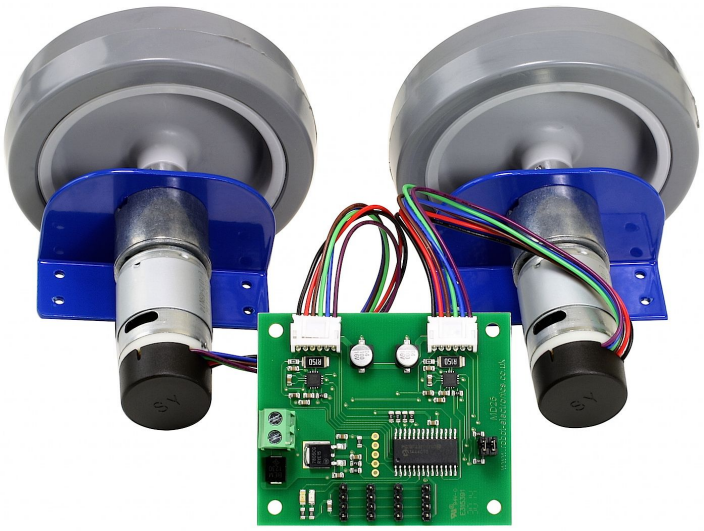

Figure 2.4: MD25 board with motors

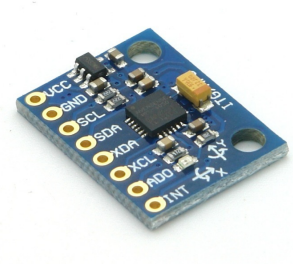

Figure 2.5: GY-521 imu

readable in case we want to take them into account(not in the scope of the thesis).

2.2.4 Sensors

2.2.4.1 GY-521

The GY-521 (fig.2.5) is a 6 axis motion tracking device based on the MPU6050 with 6 degrees of freedom meaning it has 3 axis accelerometer and 3 axis gyroscope integrated on a single chip. The gyroscope measures rotational velocity over the xyz-axis while the accelerometer linear acceleration over the three axis. We already have odometry to determine orientation and distance, however since the localization is incremental (based on the previously estimated location), measurement errors are accumulated over time and cause the estimated robot pose to drift from the actual location. We will use it in order to combine it with the odometry to have a more robust way of finding orientation

Figure 2.6: Ultrasonic sensor HC-SR04

2.2.4.2 Ultrasonic Distance Sensor - HC-SR04

In order to identify and avoid obstacles, we'll need a way to distinguish them and the most reachable low-cost option would be ultrasonic sensors. Five of them (fig.2.6), with 15 degrees field of view each, will be responsible for finding range at the front of the robot using ultrasonic sound waves. The distance of a target is measured by emitting ultrasonic sound waves, and converting the reflected sound into an electrical signal, then it's a simple equation of the speed of sound (it varies with the temperature). While its not affected by color or transparency of objects it's slightly affected by dust, dirt, or high-moisture environments. Sensing accuracy affected by soft materials and changes in temperature of 5-10 degrees or more

2.2.4.3 C270 Logitech HD WEBCAM

A USB camera is needed for our robot to find its way, so the c270 was the choice we moved with because of the low cost, lightness, and compactness. As long as, it fulfills our needs for high resolution, high rate video, any other camera could have been chosen. C270 video captures up to 1280 x 720 pixels (30fps) for lifelike detail and motion. The 16:9 aspect ratio provides a widescreen view.

2.2.4.4 Power supply

12V to 5V converter (why?)

2.2.4.5 Battery

12V 7Ah

- how long can it drive the system?

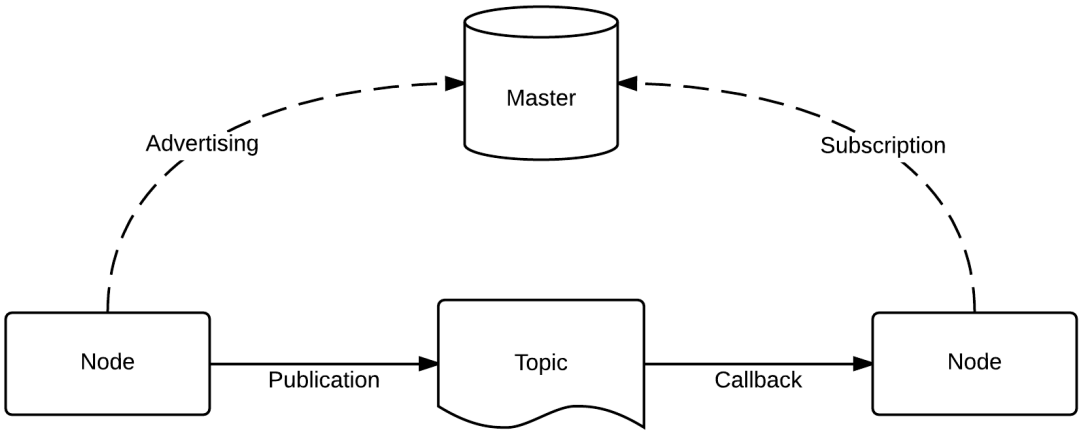

Figure 2.7: ROS nodes

2.3 Robotics Operating System (ROS)

explain

2.3.1 Urdf

The Unified Robotic Description Format (URDF) is an XML file format used in ROS to describe all elements of a robot. This specification cannot describe all robots as it assumes the robot consists of rigid links connected by joints; flexible elements are not supported. The description of a robot consists of a set of link elements, and a set of joint elements connecting the links together. The specification covers:

1. Kinematic and dynamic description of the robot
2. Visual representation of the robot
3. Collision model of the robot

2.3.2 ROS node

Explain how each process runs on each own node and the way they communicate
- topic subscribe/publish
- spawn/ restart etc
- Show created diagram (topics/nodes)

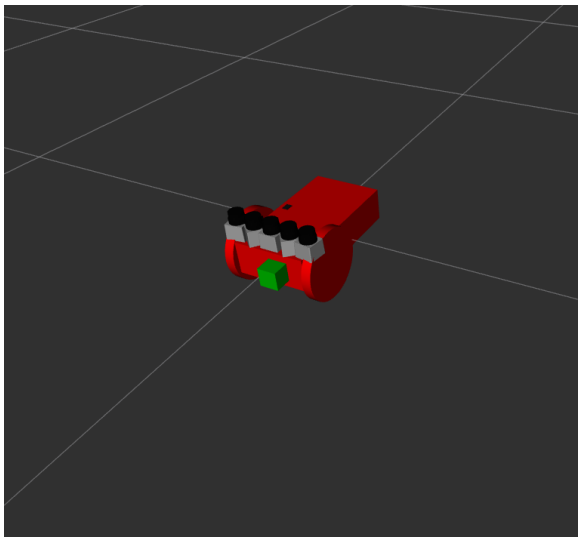

Figure 2.8: Robot designed in Rviz

2.4 The Gazebo simulator

Gazebo simulates multiple robots in a 3D environment, with extensive dynamic interaction between objects.

2.4.1 Robot model

urdf / Show how it was designed and why (collision etc)

2.4.2 World

map/ buildings/ world/ physics

2.4.3 Simulation

Explain how it was tested.

2.4.4 Test

Chapter 3

System developement and integration

3.1 Intro

In this chapter the process of developing the software that the robot will run and the tools used to develop our system. It translates in writing code for simulation in order to evaluate the logic, and after that deploy it on the actual robot system.

The robot provided from the university, meaning the chassis is already made with md25 and motors mounted so we have to integrate the rest of the modules on that.

3.2 Software installation

The first thing to do is choose the operating system of raspberry pi where our logic will be run. I chose ROS because it allows every component to be implemented/tested separately before deploying and it also provides a structured communications layer above the host operating systems of a heterogeneous compute cluster.

- install ubuntu/ros

- OpenCV

- Gazebo/rviz

install scripts and explain more

3.3 Components position

The Ultrasonic sensors are positioned in a uniform spread in 180° (left-right) to sense wider area and reduce overlapping. There exists a scientific paper that proposes a way to get the optimal position for our scenario, but my guess is that there is no need to go into that.

Having the fact that the ultrasonic beam has 15° radius and that we want to spread 5 of them in the front while also checking the left and right side of the robot for obstacles, I choose to assign 2 sensors facing ±90° and fill in the others afterwards. The final positioning of the 5 sensors is:

1. 0° - left
2. (180+39)° front_left
3. 90 front
4. (180-39)° front_right
5. 180° right

0° degrees - 1st

(15/2)-0+(15/2) radius

so we have to calculate [180+(15/2)+(15/2)]/N = 39° (N=5) instead of 180/N where N = number of sensors (or we could also follow this [2] paper for)

Accelerometer could be positioned anywhere and take the transform frame in accordance with an abstract point at the center of the robot.

Wheels could be placed in the center of the robot,but..

Battery in the center, above the wheels, because space-balance

USB Camera in the front (don’t obstruct sensors)

3.4 Urdf Model design

XML tags used and type of each joint

Rviz for confirmation

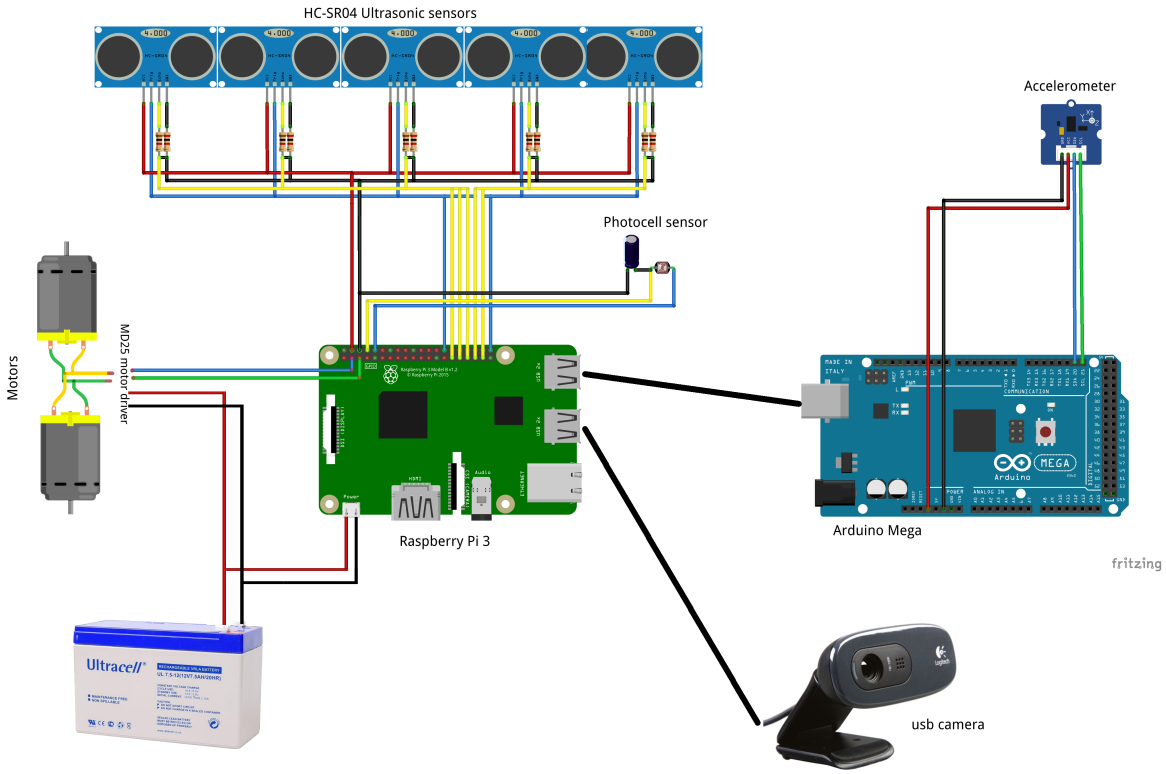

Figure 3.1: Robot Schematic

3.5 Simulation

TF_frames (from ros)

World made

Tests with obstacles etc..

3.6 Deploy, Operate and Monitor

Node for teleoperation

Rviz for confirmation on:

1. robotmodel
2. publish of topics
3. transformation frames

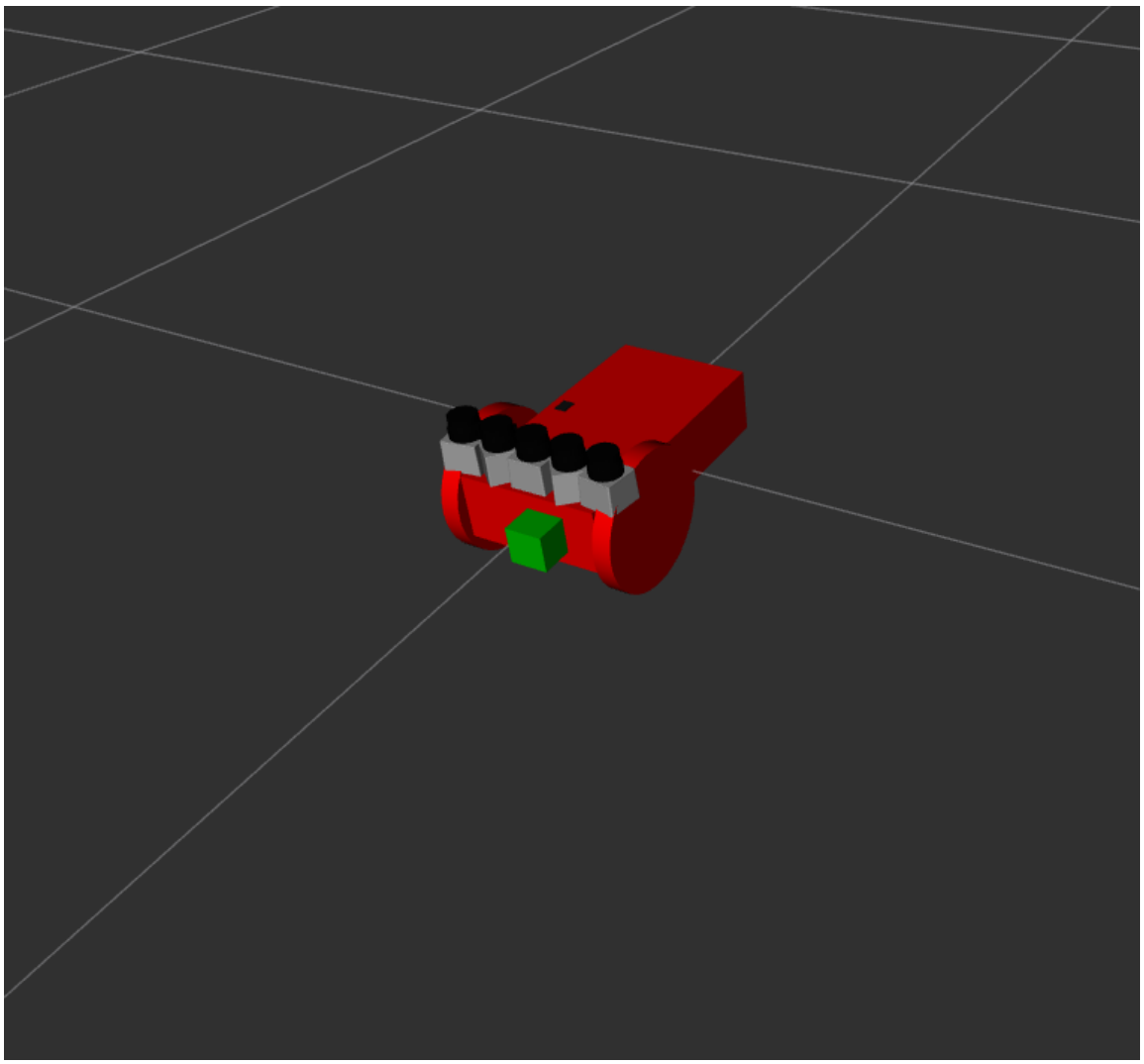

Figure 3.2: Robot Model

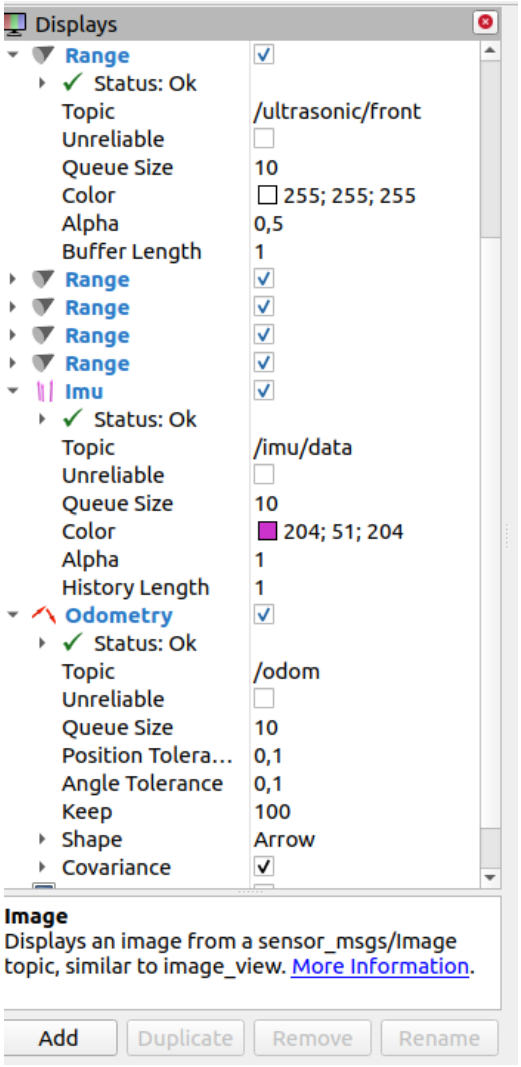

Figure 3.3: Rviz

Chapter 4

Navigation

4.1 Intro

The autonomous robot needs a way to find it's course which of course depends on the purpose of each robot. While SLAM could be used to determine the map and then navigate safely, in this thesis we will research ways to do line recognition using the easiest accessible component, a usb web camera.

4.2 Related work

Based on edges:

1. Sobel operator, Canny edge detector, Hough Line Transform etc.
2. the edges must be detected well for the Hough transform to be efficient => edges dont provide enough accuracy in various lighting conditions.

Based on color:

1. Line Detection using IR sensor, the most effective w/ zero computation (not present and doesn't have anything to do research on)
2. HSV - choose pixels in range of calibrated values. Pretty accurate but we need at least one RGB->HSV conversion as our camera outputs RGB.
3. Kmeans: good accuracy but slow

Figure 4.1: Gray without much red (no practical difference)

4. RGB - uses euclidean distance. $\sqrt{r^2+g^2+b^2} < threshold$

   Preproccessing:

   1. Histogram equalization: not robust enough

   2. GrayWorld colorbalance algorithm(link): it averages RGB channels separately and then multiplies every pixel by the gain coefficient calculated. Could be performed in various colorspaces but the conversion between colorspaces just adds more computations.

   3. GrayWorld(improvised): plant known color on field of view to ”force” better results. This happens because these kind of methods perform better with the existence of certain colors in certain quantities and in case they are missing the method performs poorly. You can see how image changes based on quantity of red color in figure

   4. Gaussian blur, not computational expensive and increases accuracy.

   Lane Detection Based on Connection of Various Feature Extraction Methods but every extra action counts so I tried to keep it as simple as possible

4.3 Proposed method

The finalization of the recognition task is trivial with some assumptions of course.

1. line could be found anywhere on input frame

2. route could have intersection

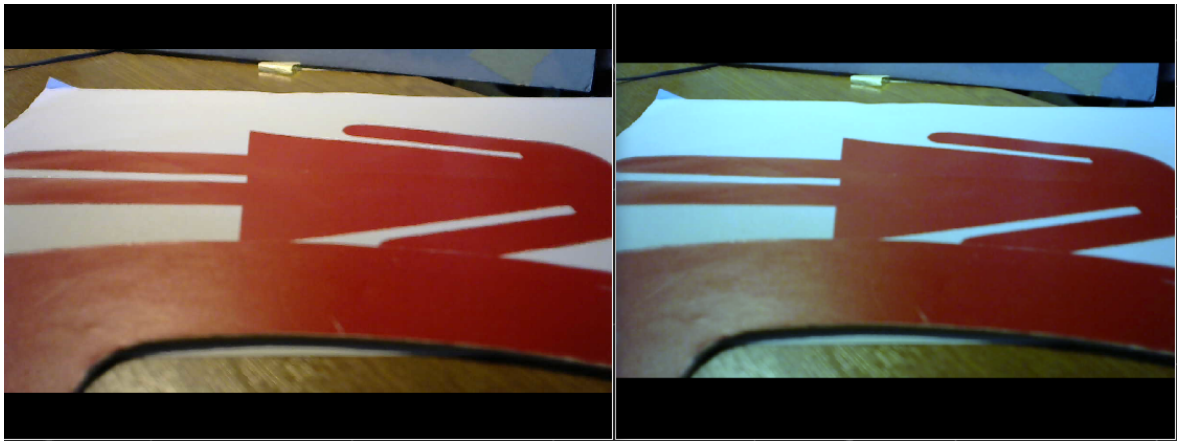

Figure 4.2: Gray with red

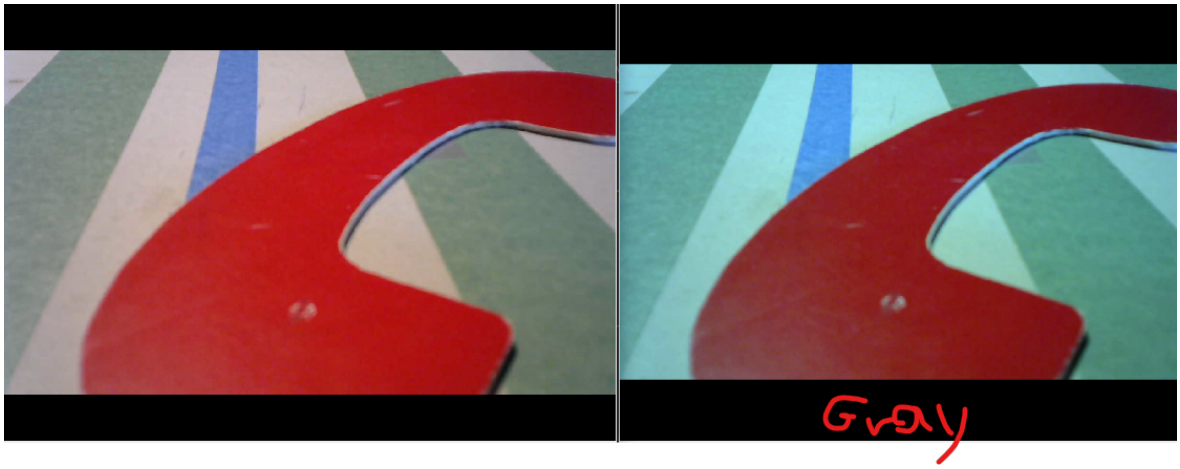

Figure 4.3: Gray wit red 2

Algorithm 1 Euclidean Distance in RGB colorspace

Get Frame
initialize Output_Frame(height,width) = zeros {one channel instead of three}
Gaussian_blur(Frame)
for $i = 0, \ldots, height$ **do**
  for $j = 0, \ldots, width$ **do**
    r, g, b = Frame[i][j]
    calculate S from HSV colorspace {we don't calculate H and V}
    calculate luminance of pixel
    $L = \sqrt{0.299 * r^2 + 0.587 * g^2 + 0.114 * b^2}$
    if low_threshold < S or S > up_threshold **then**
      $Output\_Frame[i][j] = 0$ {this pixel is invalid}
    else if $low\_luminance\_thres < LorL > up\_luminance\_thres$ **then**
      $Output\_Frame[i][j] = 0$ {this pixel is invalid}
    else if $\sqrt{(r - thresR)^2 + (g - thresG)^2 + (b - thresB)^2} > euclidean_distance$ **then**
      $Output\_Frame[i][j] = 255$ {this pixel is valid}
    end if
  end for
end for
Morphological Transformations (MORPH_OPEN) useful in removing noise.
Morphological Transformations (MORPH_CLOSE) useful in closing small holes inside the foreground objects, or small black points on the object.

3. intersection is defined by the cendroid of all the points found away from the borders of the input frame

4. some steep turns are being recognised as intersections, so if we follow them we will have a smoother cruise

Algorithm 2 Choose destination based on black/white image

Output_Frame(height,width) {black/white image}
findContours(Output_Frame, EXTERNAL_ONLY)
if $max(contour\_Area) < threshold$ **then**
    return 'no line'
end if
$epsilon = 0.01 * arcLength(max_contour)$
$aprox\_points = approxPolyDP(max_contour, aprox, epsilon, true)$ {now has a list of points on the perimeter of the detected object}
for $i = 1, 2, \ldots, size(aprox\_points)$ **do**
    if aprox_points[i] is close to one of the borders **then**
        add to candidates_list["North|South|East|West|Intersection"]
        and calculate centroid of each group
    end if
end for
if $candidates\_list > 2$ **and** (didn't found point_to_go in previous round) **then**
    choose the closest to the middle of the screen
else if found point_to_go in previous round) **then**
    choose the closest to the previous chosen point
else if found intersection **then**
    choose the intersection
else
    choose the highest we see
end if

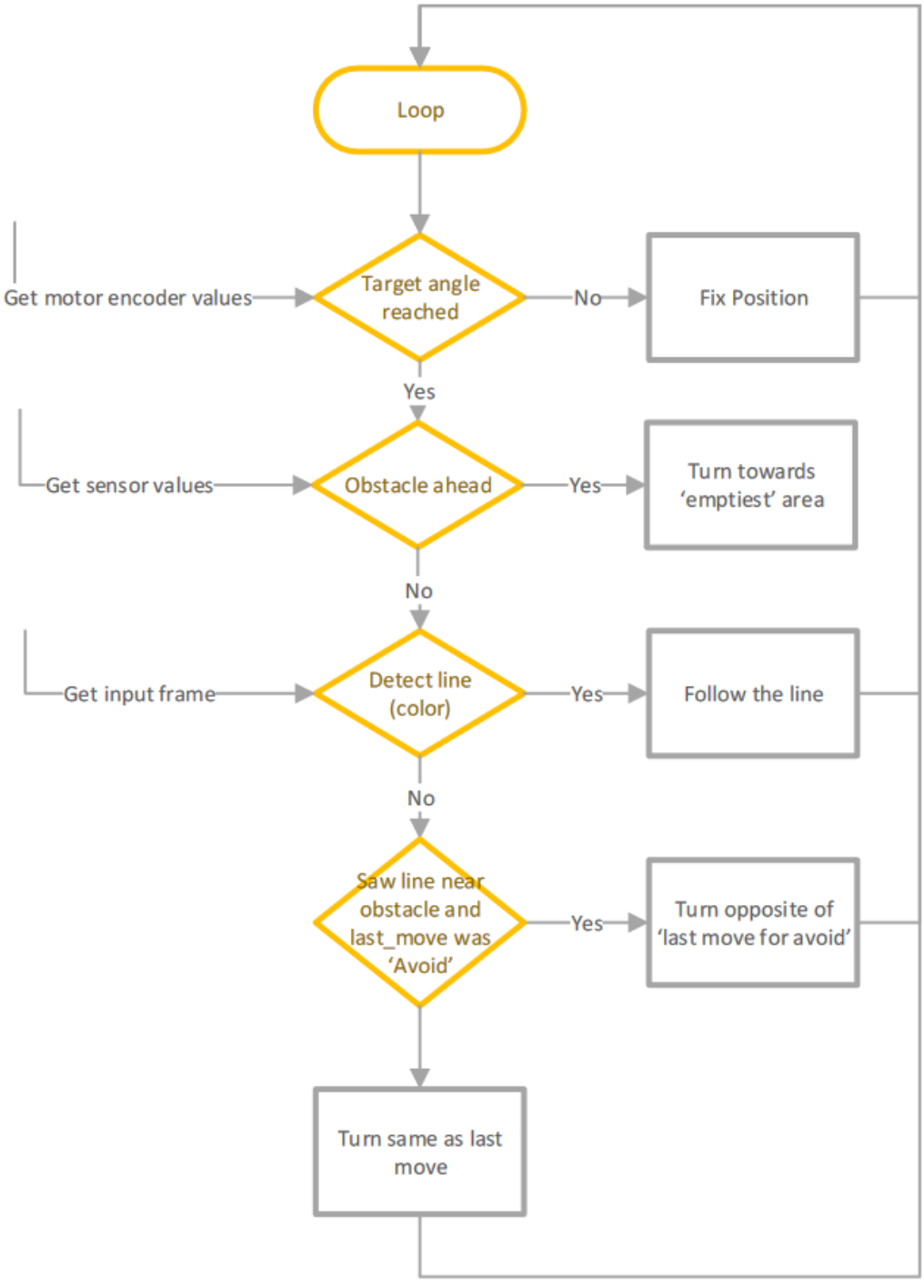

Figure 4.4: Line detect logic diagram

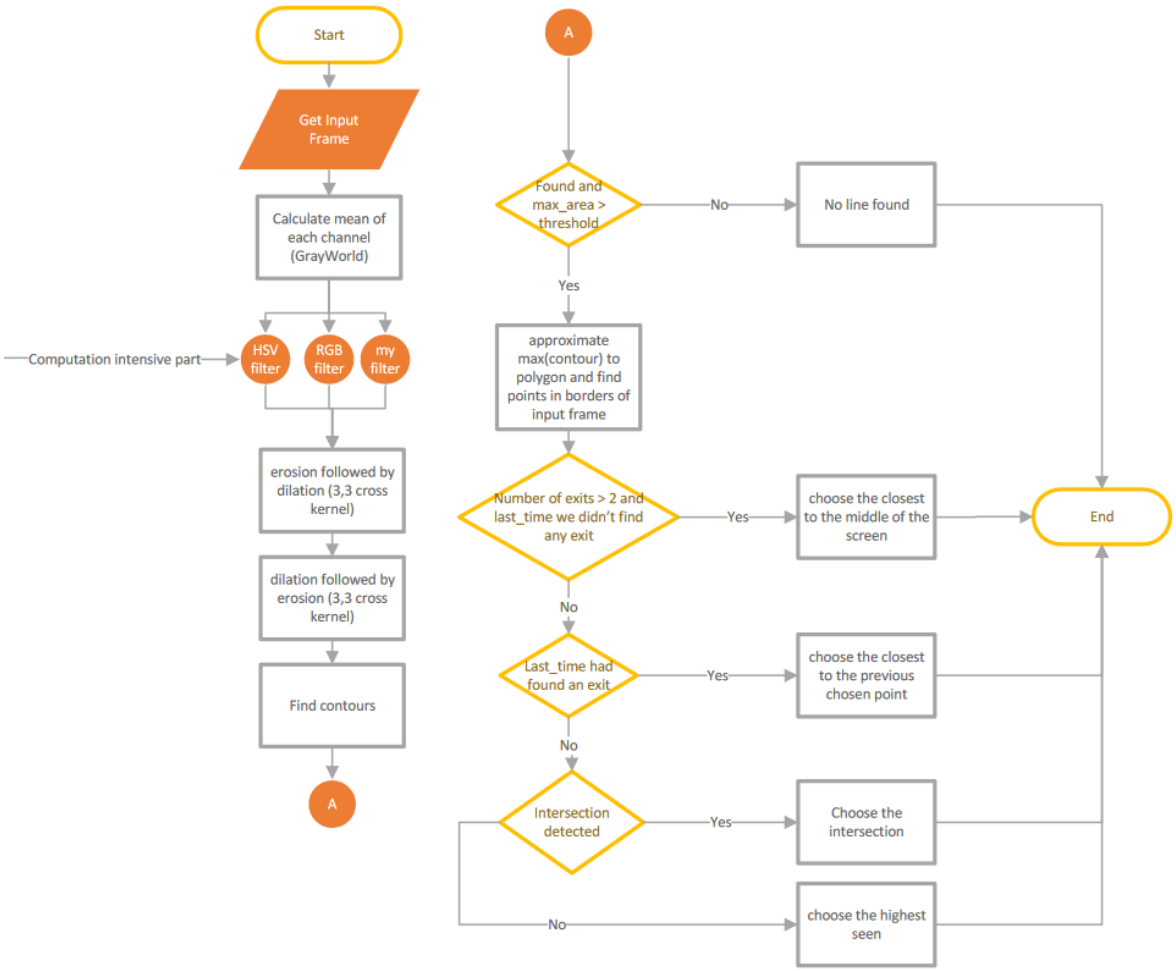

Figure 4.5: Line detect logic diagram

4.4 Optimizations

OpenMP

raspi qpu(gpu) test?

4.5 From points in 2D -> speed,turn

explain how

Chapter 5

Testing and Results

5.1 Color detection performance results

5.2 HSV filter + Euclidean RGB + Gray-world color balance + Custom criterion

Show + Compare pictures

5.2.1 1280*720 input frame

Raspberry 3 average times (fig.5.1)

1. (RGB) 524.496000 ms, 1.906592 fps
2. (RGB + CUSTOM) 523.855000 ms, 1.908925 fps
3. (HSV) 386.589000 ms, 2.586726 fps
4. (HSV + CUSTOM) 376.639000 ms, 2.655062 fps
5. (HSV + RGB) 717.869000 ms, 1.393012 fps
6. (HSV + RGB + CUSTOM) 701.764000 ms, 1.424980 fps
7. (HSV + RGB + CUSTOM + GRAYWORLD) 886.507000 ms, 1.128023 fps (BEST accuracy, but needs some tweaks if one of R,G,B is in much quantity on the input frame)

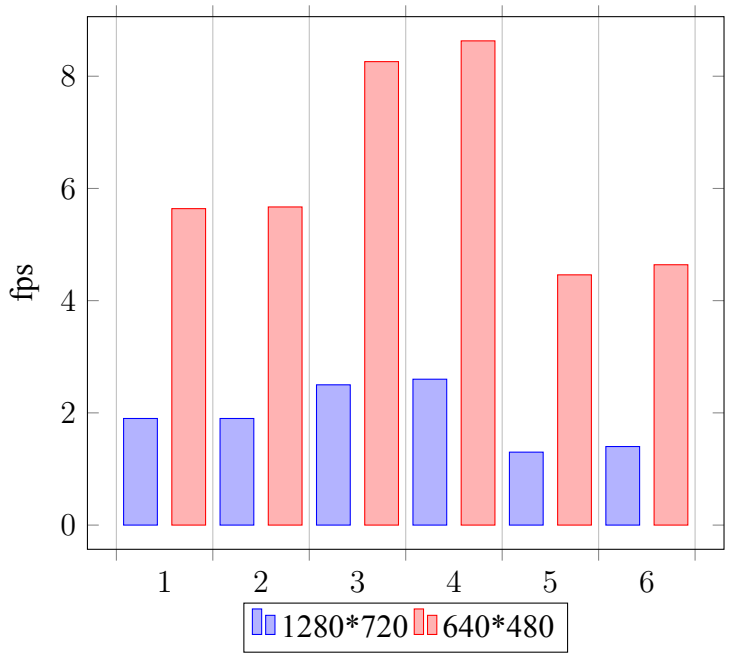

Figure 5.1: Raspberry pi 3 compute times(From worst accuracy to best)

PC (include specs ?):

1. TODO depending on what system is the best to measure it

5.2.2 640*480 input frame

Raspberry 3 average times (fig.5.1)

1. (RGB) 177.050000 ms, 5.648122 fps
2. (HSV + RGB) 223.800000 ms, 4.468275 fps
3. (HSV) 120.934000 ms/frame, 8.268973 fps
4. (RGB+CUSTOM) 176.092000 ms, 5.678850 fps
5. (HSV+CUSTOM)115.781000 ms, 8.636996 fps
6. (HSV + RGB + CUSTOM) 215.170000 ms, 4.647488 fps
7. (HSV + RGB + CUSTOM + GRAYWORLD) 286.658000 ms, 3.488478 fps (BEST accuracy)

Pc (include specs?):

1. TODO depending on what system is the best to measure it

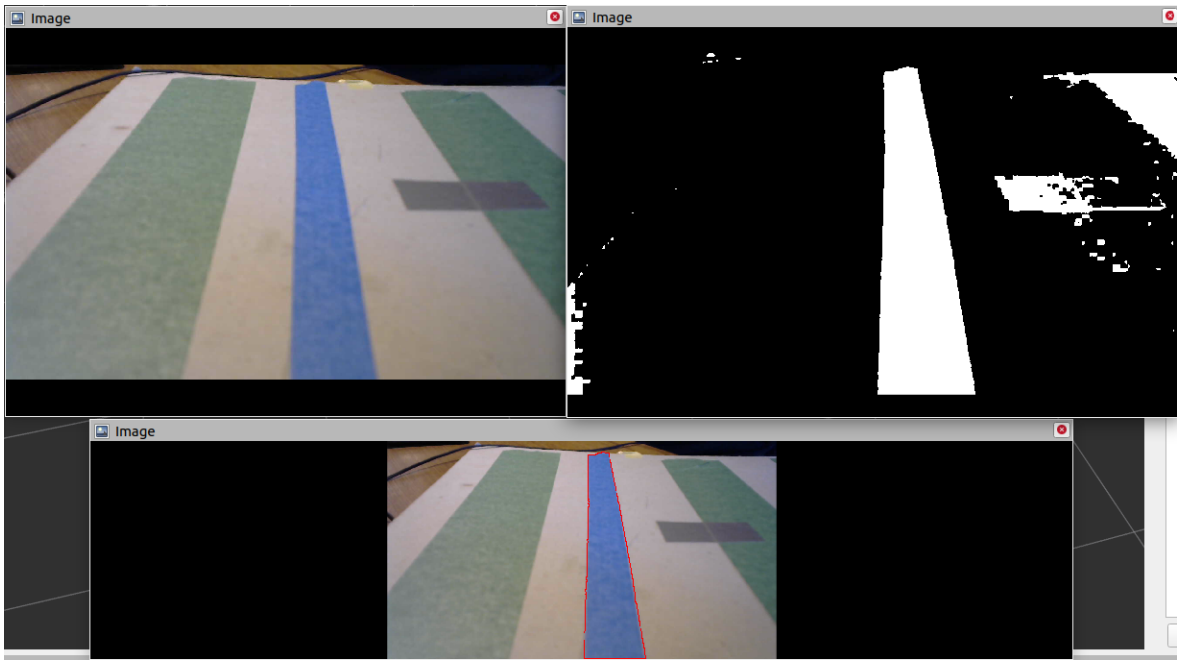

Figure 5.2: Find blue color using HSV/GrayWorld/custom criterion

5.3 Results

Some of the best

How should we incorporate videos?

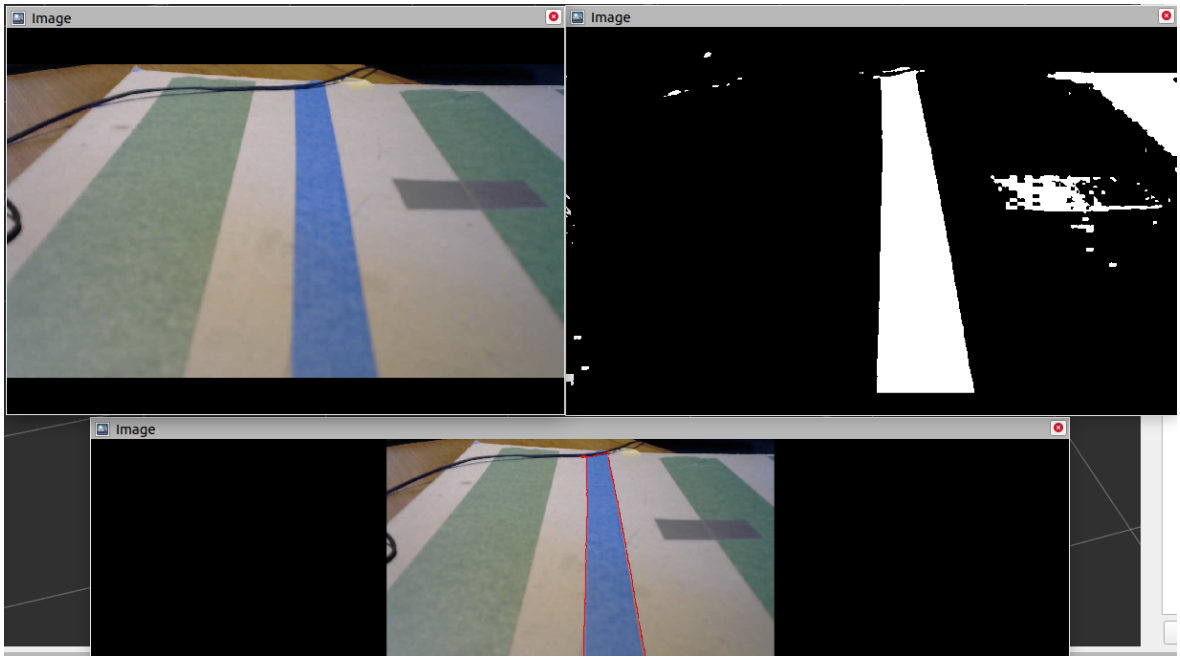

Figure 5.3: Find blue color using HSV/custom criterion

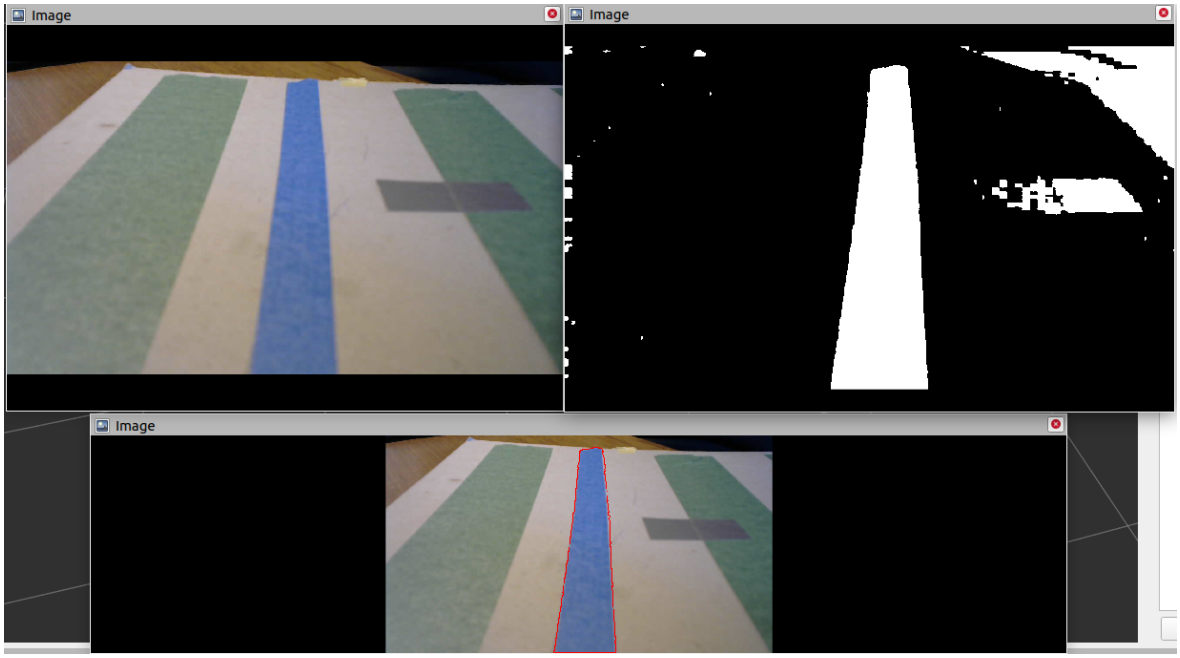

Figure 5.4: Find blue color using HSV criterion

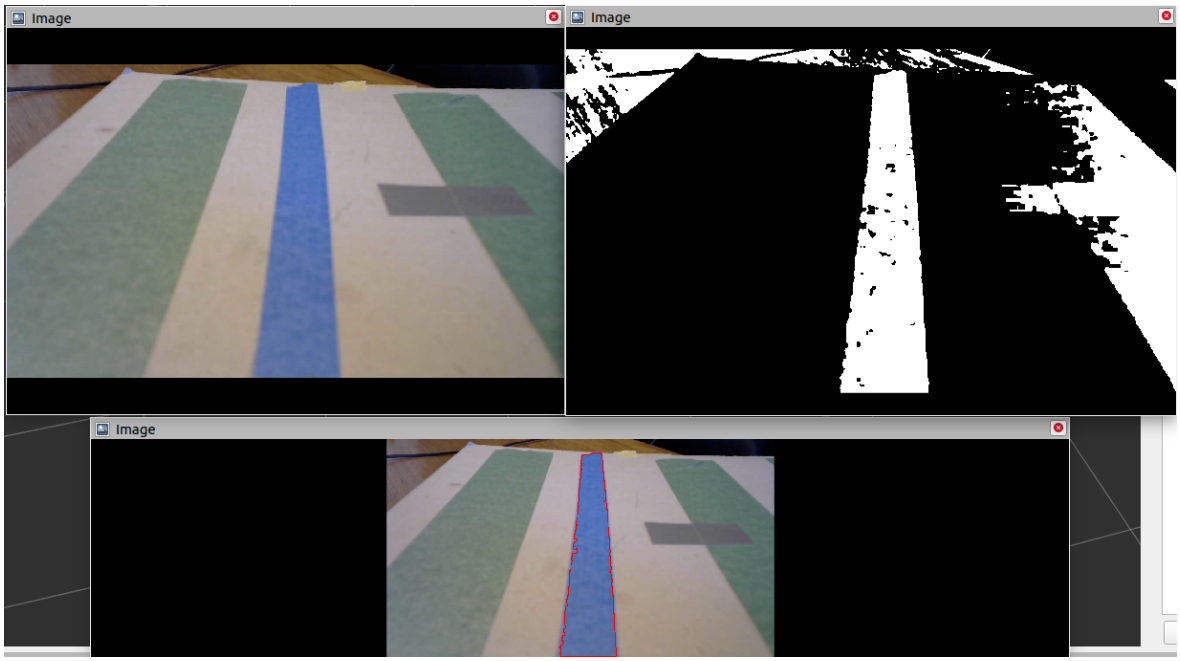

Figure 5.5: Find blue color using RGB/GrayWorld/custom criterion

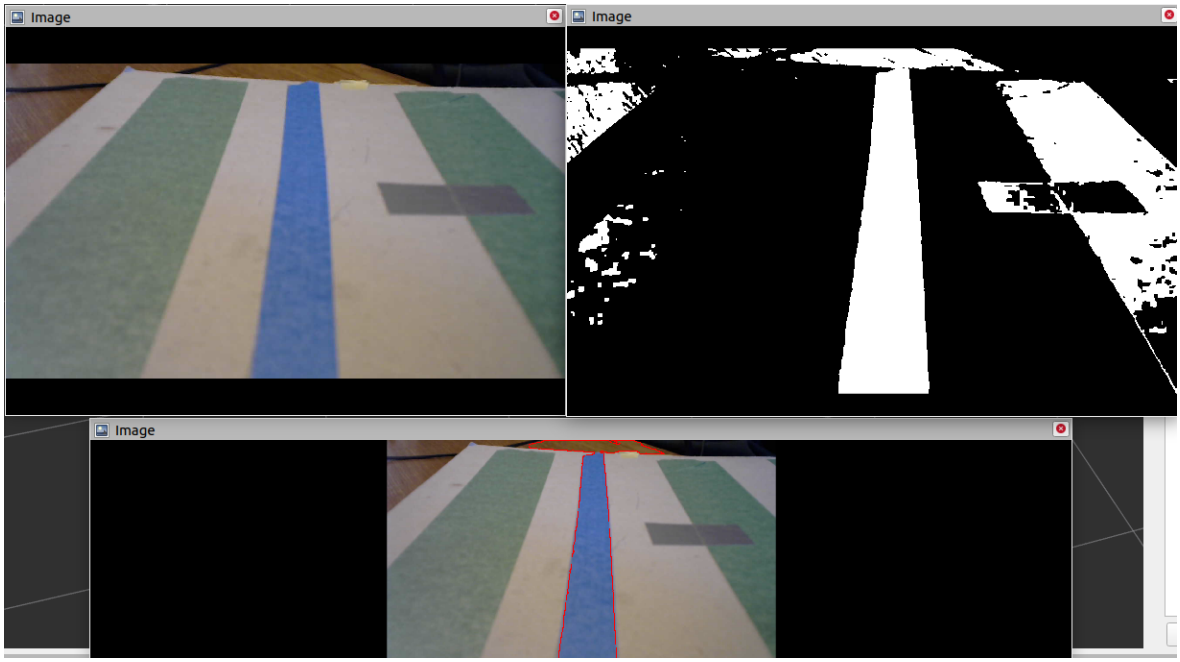

Figure 5.6: Find blue color using RGB/custom criterion

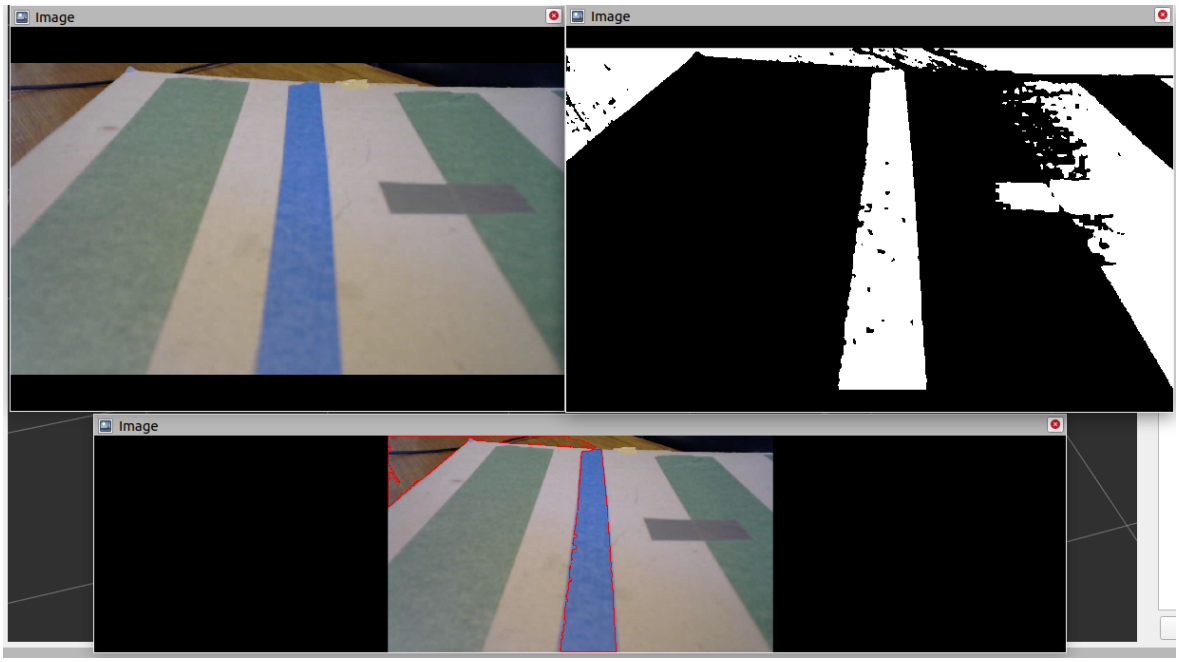

Figure 5.7: Find blue color using RGB/custom criterion

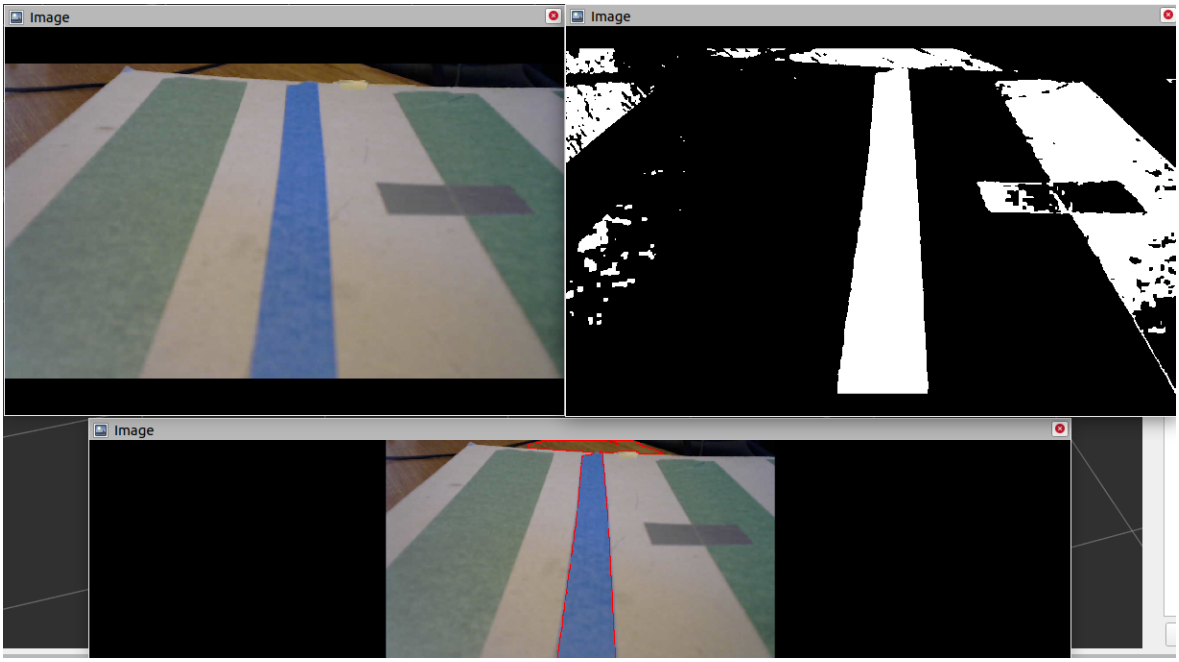

Figure 5.8: Find blue color using RGB/custom criterion

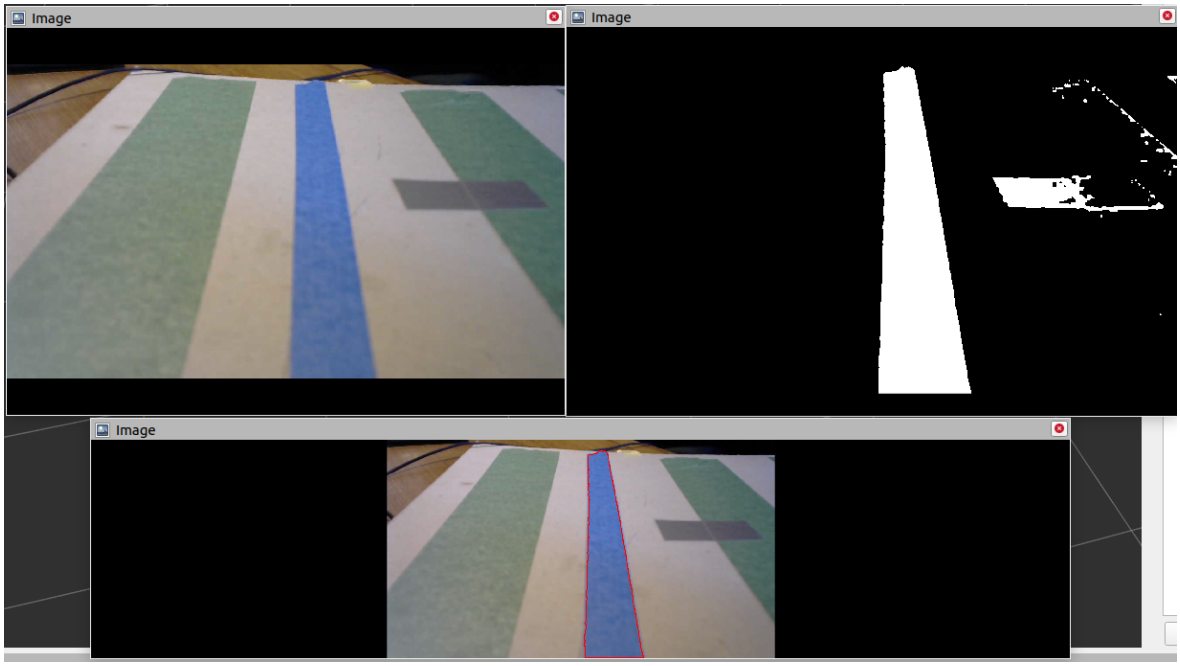

Figure 5.9: Find blue color using RGB/HSV/GrayWorld/custom criterion

Chapter 6

Conclusions

Εδώ εξηγούμε ότι θα συνοψίσουμε την μελέτη που εκπονήθηκε στα πλαίσια της διπλωματικής.

6.1 Summary

Εδώ συνοψίζουμε τα αποτελέσματα της διπλωματικής και περιγράφουμε τα συμπεράσματα που προέκυψαν, αρνητικά και θετικά. Επιβεβαιώνουμε την συνεισφορά της διπλωματικής στα προβλήματα που αναφέραμε στην εισαγωγή.

6.2 Future work

Εδώ δίνουμε ιδέες για επέκταση της διπλωματικής.

Bibliography

[1] Raspberry power supply. `https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md`. Ημερομηνία πρόσβασης: 2-2-2020.

[2] Sungbok Kim and Hyunbin Kim. Optimally overlapped ultrasonic sensor ring design for minimal positional uncertainty in obstacle detection. *International Journal of Control, Automation and Systems*, (8):1280–1287, 2010.

[3] M. Goossens, F. Mittelbach, and A. Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 9 edition, 1993.

[4] Ι. Κάβουρας. *Συστήματα Υπολογιστών*. Κλειδάριθμος, Αθήνα, 3 edition, 1991.

[5] J. Liaperdos, A. Arapoyanni, and Y. Tsiatouhas. Adjustable RF mixers' alternate test efficiency optimization by the reduction of test observables. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(9):1383–1394, Sept. 2013.

[6] I. Liaperdos, L. Dermentzoglou, A. Arapoyanni, and Y. Tsiatouhas. Fault detection in RF mixers combining defect-oriented and alternate test strategies. In *26th Conference on Design of Circuits and Integrated Systems (DCIS)*, San Sebastian, Spain, Nov. 2011.

[7] Latex project. `http://www.latex-project.org`. Ημερομηνία πρόσβασης: 13-11-2014.

[8] Ε. Ανδρουλάκη. Υλοποίηση Ενεργού Μηχανισμού σε Σύστημα Ομότιμων Βάσεων. Πτυχιακή εργασία, KDBS Lab, Εθνικό Μετσόβιο Πολυτεχνείο, Ιουλ. 2005.

[9] Ζ. Καούδη. Πρότυπο Σύστημα Αποθήκευσης και Διαχείρισης Σχημάτων rdfs. Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Ιουλ. 2004.

[10] Ζ. Λάσκαρη. Κοινωνική Ανάλυση των Ταινιών της finos films. Master's thesis, Εθνικό Μετσόβιο Πολυτεχνείο, Aug. 2012.

[11] Ζ. Κουρούκλη. *Κατανεμημένα Συστήματα.* PhD thesis, ΤΕΙ Πελοποννήσου, Dec. 2013.

[12] H. Cheng J. Gao and P.-N. Tan. A framework for incorporating labeled examples into anomaly detection. Technical Report MSU-CSE-05-29, Department of Computer Science, Michigan State University, East Lansing, Michigan, 2005.

[13] P. Viswanathan, G. Winner, and P. Vyas. Convenient provisioning of embedded devices with wifi capability. US Patent 8,665,744, 2014.

[14] K. Patroumpas and T. Sellis. Subsuming multiple sliding windows for shared stream computation. In Johann Eder, Maria Bielikova, and A Min Tjoa, editors, *Advances in Databases and Information Systems*, volume 6909 of *Lecture Notes in Computer Science*, pages 56–69. Springer, 2011. doi:10.1007/978-3-642-23737-9_5.

APPENDICES

Appendix A

Τίτλος Παραρτήματος

Τα παραρτήματα περιλαμβάνουν συνοδευτικό, υποστηρικτικό υλικό (πίνακες, φωτογραφίες, ερωτηματολόγια, στατιστικά στοιχεία, αποδείξεις, περιγραφές λογισμικών προγραμμάτων, παραδείγματα, περιγραφές πολύπλοκων διαδικασιών, λίστα με πρωτογενή στοιχεία, λεπτομερής περιγραφή και προδιαγραφές εξοπλισμού, οδηγίες εγκατάστασης λογισμικού, κ.λπ.), ή αλλιώς ό,τι θεωρείται χρήσιμο να περιγραφεί, αλλά δεν συνηθίζεται να εντάσσεται μέσα στο κυρίως κείμενο της Εργασίας. Στο κυρίως κείμενο της Εργασίας πρέπει να γίνονται οι κατάλληλες παραπομπές προς τα παραρτήματα, όπου το κείμενο σχετίζεται με υλικό που περιλαμβάνεται σε αυτά. Ένα παράρτημα, αναλόγως με το περιεχόμενό του, μπορεί να είναι ενιαίο, ή να χωρίζεται σε ενότητες.

A.1 Δυνατότητες του LaTeX

Καθώς το παρόν αποτελεί ένα πρότυπο συγγραφής διπλωματικών εργασιών, στην ενότητα αυτή επιδεικνύονται ορισμένες από τις δυνατότητες το LaTeXοι οποίες μπορούν να αξιοποιηθούν στο κείμενο μιας διπλωματικής εργασίας (μια καλή πηγή για τη διερεύνηση των δυνατοτήτων του LaTeXείναι η ιστοσελίδα `https://www.overleaf.com/learn/latex/Main_Page`).

A.1.1 Πίνακες

Ο Πίνακας A.1 είναι ένα παράδειγμα πίνακα σχεδιασμένου με εντολές του περιβάλλοντος `tabular`.

Table A.1: Παράμετροι πειραμάτων

Πλήθος κελιών καννάβου $c \times c$	50 × 50, 100 × 100, 200 × 200, **250** × **250**, 500 × 500, 1000 × 1000
Τυπική απόκλιση σ	25m, 50m, 75m, **100m**, 150m, 200m
Αριθμός εγγύτερων γειτόνων k	1, 2, **3**, 4, 5, 10, 20
Πιθανοτικό κατώφλι θ	50%, 60%, 70%, **75**%, 80%, 90%, 99%

A.1.2 Διαγράμματα - Γραφικές παραστάσεις

Στο Σχήμα **??**, παρουσιάζεται ένα παράδειγμα γραφικής παράστασης σχεδιασμένης με το Gnuplot.

A.1.3 Σχήματα

A.1.4 Αλγόριθμοι

Ακολουθεί ο Αλγόριθμος 3, ο οποίος είναι μορφοποιημένος με τα πακέτα algorithm και algorithmic.

Algorithm 3 Probabilistic $k\theta NN$ Monitoring

1: **Procedure** *VerifyCandidate* (focal query point q, threshold θ, object o, list of auxiliary objects P, distance $kMAXDIST$)
2: **if** $\Phi(o, kMAXDIST) \geq \theta$ **and** $L_2(q, o) \leq L_2(q, P.\text{top}())$ **then**
3: P.pop(); *//Replace the most extreme element in P, since candidate o ...*
4: P.push(o); *//... has enough probability and has its mean closer to focal q*
5: **end if**
6: **End Procedure**

A.1.5 Μαθηματικές εκφράσεις

Ακολουθούν παραδείγματα μαθηματικών εκφράσεων.

$$\hat{I}(x,u,t) \;\; = \;\; dist(y(t_f),\Gamma) + \int_t^{t_f} \mathcal{L}(y(s),u(s),s)\,ds \tag{A.1}$$

$$\frac{d}{dx}\left(\int_0^z f(u)\,du\right) = f(x).$$

A.1.6 Θεωρήματα, Πορίσματα, Ορισμοί, κλπ.

Ακολουθεί παράδειγμα θεωρήματος από την ιστοσελίδα `https://www.overleaf.com/learn/latex/Theorems_and_proofs`

Theorem 7.1. *Let f be a function whose derivative exists in every point, then f is a continuous function.*

A.1.7 Απαριθμήσεις

Μια απαρίθμηση (itemized list) βοηθά στην παρουσίαση μιας σειράς περιπτώσεων με σαφήνεια. Ακολουθεί παράδειγμα.

Η εκπαίδευση στην Ελλάδα διακρίνεται σε:

- Πρωτοβάθμια
- Δευτεροβάθμια
- Τριτοβάθμια

A.1.8 Είδη πηγών στις αναφορές

Στο references.bib μπορεί να δει κανείς πώς γράφονται διάφορα είδη πηγών (Βιβλία Ξενόγλωσσα [3], Βιβλία Ελληνικά [4], Άρθρα σε επιστημονικά περιοδικά [5], Άρθρα σε επιστημονικά συνέδρια [6], Ιστοσελίδες [7], Πτυχιακές Εργασίες [8], Διπλωματικές Εργασίες [9], Μεταπτυχιακές Διπλωματικές Εργασίες [10], Διδακτορικές Διατριβές [11], Τεχνικές Αναφορές [12], Διπλώματα Ευρεσιτεχνίας [13]), Κεφάλαια σε συλλογικούς τόμους [14].

Appendix B

Τίτλος 2ου Παραρτήματος

Τα παραρτήματα περιλαμβάνουν συνοδευτικό, υποστηρικτικό υλικό (πίνακες, φωτογραφίες, ερωτηματολόγια, στατιστικά στοιχεία, αποδείξεις, περιγραφές λογισμικών προγραμμάτων, παραδείγματα, περιγραφές πολύπλοκων διαδικασιών, λίστα με πρωτογενή στοιχεία, λεπτομερής περιγραφή και προδιαγραφές εξοπλισμού, οδηγίες εγκατάστασης λογισμικού, κ.λπ.), ή αλλιώς ό,τι θεωρείται χρήσιμο να περιγραφεί, αλλά δεν συνηθίζεται να εντάσσεται μέσα στο κυρίως κείμενο της Εργασίας. Στο κυρίως κείμενο της Εργασίας πρέπει να γίνονται οι κατάλληλες παραπομπές προς τα παραρτήματα, όπου το κείμενο σχετίζεται με υλικό που περιλαμβάνεται σε αυτά. Ένα παράρτημα, αναλόγως με το περιεχόμενό του, μπορεί να είναι ενιαίο, ή να χωρίζεται σε ενότητες.