YILDIZ TECHNICAL UNIVERSITY
FACULTY OF ELECTRIC AND ELECTRONICS
DEPARTMENT OF COMPUTER ENGINEERING

SENIOR PROJECT

DETERMINING SHORTEST PATH AND OBSTACLE RECOGNITION OF A MAP BUILDING ROBOT

Project Supervisor: Assist. Prof. Dr. Sırma YAVUZ

Project Group
03011503, Necmi, Öztürk
03011502, Can, Yüksel

İstanbul, 2007

TABLE OF CONTENTS

ABBREVIATION LIST

FIGURE LIST

TABLE LIST

PREFACE

This study, which is made in Yıldız Technical University Computer Engineering Department is for senior project. We thank to Zeynep KURT, Tarkan DINC, İbrahim ÖK, Hilmi Kemal YILDIZ, Fatih AYDINOGLU and our project supervisor Asist. Prof. Dr. Sırma YAVUZ for their valuable help.

ABSTRACT

We are implementing on an object recognition system that localises and identifies objects for an autonomous robot system which is developed in Yıldız Technical University, Computer Science Department. The porpose of the robot is to recognize some specified objects and develop a good algorithm for robot way.

We build a labyrinth for the movement area of robot. RF modules used to communicate with computer. We use camera with robot to recognize some obstacles and give decision for the right way of robot. Object localisation and identification is a crucial problem for advanced mobile service robots. Because sometimes we need to know what we meet in an unknown area and determine the best way for robot by avoiding obstacles.

ÖZET

Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümünde geliştirilen eş zamanlı konum belirleme ve haritalama yapan otonom robot için şekil tanıma, cismin yerini belirleme ve cismi tanınma işlemini gerçekleştiriyoruz. Projenin amacı bazı belirlenmiş cisimleri tanıması ve iyi bir yol bulma algoritması geliştirilmesidir.

Robotun hareketini sağlayacağı bir labirent oluşturuldu. Rf modülüde robotun bilgisayarla haberleşmesinde kullanılmaktadır. Kamerayı ise robot ile birlikte cisim tanıma ve robotun doğru yolu bulmasında kullanıyoruz. Objenin belirlenmesi robot sistemleri için önemli ve kritik bir meseledir. Çünkü bazen bilinmeyen alanlarda neyle karşılaştığımızı bilmeye ve engellerden sakınarak en iyi yolu belirlenmesine ihtiyaç duyarız.

1. INTRODUCTION

1.1. Robot Systems

Robots are comprised of several systems working together as a whole. The type of the job robot does dictates what system elements it needs. The general categories of robot systems are controller, body, mobility, power, sensors and tools.

The controller is the robot's brain and controls the robot's movements. It's usually a computer of some type which is used to store information about the robot and the work environment and to store and execute programs which operate the robot. The control system contains programs, data algorithms, logic analysis and various other processing activities which enable the robot to perform.

The body of a robot is related to the job it must perform. Industrial robots often take the shape of a bodyless arm since its job requires it to remain stationary relative to its task. Space robots have many different body shapes such as a sphere, a platform with wheels or legs, or a ballon, depending on its job.

Robots movement depends on the job they have to do and the environment they operate in. In the water, conventional unmanned, submersible robots are used in science and industry throughout the oceans of the world. Land based rovers can move around on legs, tracks or wheels. Robots that operate in the air use engines and thrusters to get around.

Power for industrial robots can be electric, pneumatic or hydraulic. Electric motors are efficient, require little maintenance, and aren't very noisy. Pneumatic robots use compressed air and come in a wide variety of sizes. A pneumatic robot requires another source of energy such as electricity, propane or gasoline to provide the compressed air. Hydraulic robots use oil under pressure and generally perform heavy duty jobs. This power type is noisy, large and heavier than the other power sources. A hydraulic robot also needs another source of energy to move the fluids through its components. Pneumatic and hydraulic robots require maintenance of the tubes, fittings and hoses that connect the components and distribute the energy.

Sensors are the perceptual system of a robot and measure physical quantities like contact, distance, light, sound, strain, rotation, magnetism, smell, temperature, inclination, pressure, or altitude. Sensors provide the raw information or signals that must be processed through the robot's computer brain to provide meaningful information. Robots are equipped with sensors so they can have an understanding of their surrounding environment and make changes in their behavior based on the information they have gathered.

As working machines, robots have defined job duties and carry all the tools they need to accomplish their tasks onboard their bodies. Many robots carry their tools at the end of a manipulator. The manipulator contains a series of segments, jointed or sliding relative to one another for the purpose of moving objects. The manipulator includes the arm, wrist and end-effector. An end-effector is a tool or gripping mechanism attached to the end of a robot arm to accomplish some task. It often encompasses a motor or a driven mechanical device. An end-effector can be a sensor, a gripping device, a paint gun, a drill, an arc welding device, etc.

1.2. The Advantages of Obstacle Recognition in Robotics

While wheeled robots are moving from one point to another point in an area, they need to learn about area’s characteristic. They have to see the obstacles and move regularly inside the area. In addition to this sometimes, robots have to know what they meet in some places. Because people sometimes can’t go and search dangerous places for their health or security. For example, in a factory some chemicals are harmful for humans and using robots is helpful. In a war military forces use robots. Because they can easily spy and they are not important as a human.

Using a camera with robotics is useful for many reasons. Another advantage is making good decision for robot’s dirrection and avoiding from obstacles in a complex area. With some technics we can learn about the object’s shape and object characteristic and we can use them when we need.

2. SYSTEM ANALYSIS and FEASIBILITY

2.1. Requirements Analysis

Requirement analysis shows the system's requirements for this project.

First of all, we need a robot which has programmed on it's microcontroller and has all electronical staff on it. In this project we need a wireless module camera and a Tv card for getting images from the atmospheric. The basic thing is a computer that we will be able to write and run java codes in a good performance. To move robot in an area we need to build a labyrinth.

2.2. Feasibility Study

2.2.1. Technical Feasibility

Technical feasibility shows the system's hardware and software requirements for this project.

2.2.1.1. Hardware Requirements

We need a robot which is programmed with some functions. For programming object recognition and path finding methods with java programmers and computers are needed. Also a wireless module camera and Tv card are needed for the image taking from the athmosphere.

2.2.1.2. Software Requirements

Microsoft XP Home Edition is needed for operating system and Netbeans IDE in order to write programmes with java.

2.2.2. Legal Feasibility

In this project all the softwares and hardwares are licensed. Yıldız Technical University students are prepared all the things in this project. Nobady can use any staff of this project without a permission.

2.2.3. Economical Feasibility

Economical feasibility shows the financial reports for this project. Software and hardware requirements' all expenses are shown with the cost table.

Robot	4.348.60 YTL
CMOS Wireless Module Camera	65.00 YTL
LifeView Tv Card	65 YTL
Windows XP Professional	260 YTL
NetBeans IDE 5.5	0 YTL (Free)
J2SE 1.5 Java Development Kit	0 YTL (Free)
Work Force	2000 YTL
TOTAL	6.738.60 YTL

Table 2.2.3.1. Cost Analysis Table

2.2.4. Usable Feasibility

Our project is developed by latest technology equipment and meets all personal needs. Evereybody will be able to use easily our project when they meet with it first time.

2.2.5. Alternative Feasibility

There are many different ways to solve a problem like this. We tried to find best way to solve the problem. Other alternatives may also be tried

2.2.6. Gantt Chart

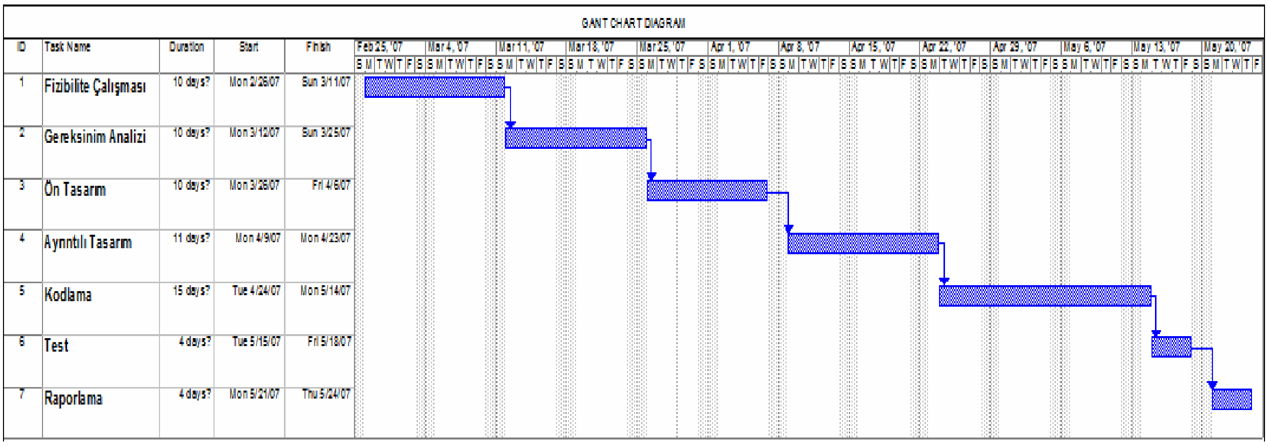


Figure 2.2.6.1. Gant Chart

3. SYSTEM ARCHITECTURE

In this project, a shortest path algorithm for autonomuos robot is implemented which is devoleped by Computer Engineering Department. SLAM robot is used while implementing image processing method and shortest path algorithm. Robot has 6 infrared sensors, a wireless camera and base technical equipments. Robot has RF communication system. Image processing methods and a good performance algorithm of shortest path finding was implemented.

4. IMAGE PROCESSING FOR ROBOT VISION

Image processing is a very important field within industrial automation, and more concretely, in the automated visual inspection. For example, in our ptoject we use a recognition system which provides a good performance for avoiding obstacles, finding best way and recognizing some objects in an area.

When we recognize an object, we have to apply some image processing methods in order to implement a good recognition system. At first we will explain the basic methods which we use in this project.

4.1. Median Filter

Median filter is the nonlinear filter more used to remove the impulsive noise from an image. Furthermore, it is a more robust method than the traditional linear filtering, because it preserves the sharp edges.

Median filter is a spatial filtering operation, so it uses a 2-D mask that is applied to each pixel in the input image. To apply the mask means to centre it in a pixel, evaluating the covered pixel brightnesses and determining which brightness value is the median value.

For example the figure shows that how to apply a median filter with an image.

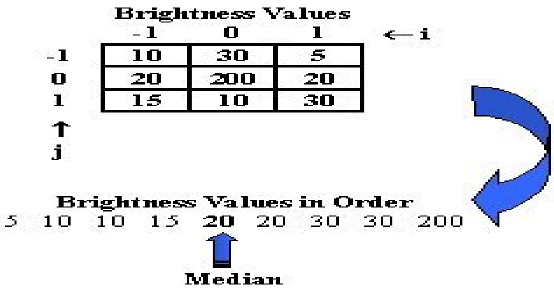


Figure 4.1.1 Implementing Median Filter

The picture which is shown below shows an example of median filter's effect.

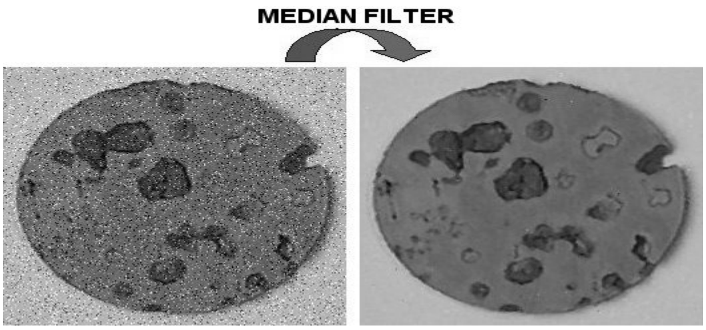


Figure 4.1.2 Median Filtered Picture

4.2. Averaging Filter

The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used for it. Averaging filter is used for bluring picture for a good segmentation.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figure 4.2.1 Averaging **Filter Mask**

Figure 4.2.2 Averaging Filtered Picture

4.3 Gaussian Smoothing

The Gaussian smoothing operator is a 2-D convolution operator that is used to blur images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. This kernel has some special properties which are detailed below.

In 2-D, an isotropic (*i.e.* circularly symmetric) Gaussian has the form,

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The idea of Gaussian smoothing is to use this 2-D distribution as a `point-spread' function, and this is achieved by convolution. The mask operation of gaussian smoothing is shown below.

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 4.3.1 Gaussian Smoothing Mask Operator

4.4. Edge Detection

Edge detection is one of the most commonly used operations in image analysis, and there are probably more algorithms. An edge is the boundry between an object and the background and indicates the boundry between overlapping objects. Edge detection is part of a process called segmentation, the identification regions with in an image [3].

In our projects we use edge values to determine best threshold value for segmenting an object from background. We use kmeans algorithm and group color pixel values with this algorithm. With kmeans we decrease our color values and we just have 3 color pixel values.

4.5. Connected Component Labeling

Detection of connected components between pixels in binary images is a fundamental step in segmentation of an image objects and regions, or blobs. Each blob is assigned a unique label to separate it from other blobs. All the pixels within a blob of spatially connected 1's are assigned the same label. It can be used to establish boundaries of objects, components of regions, and to count the number of blobs in an image . Its applications can be found in automatic inspection, optical character recognition, robotic vision.

We use this algorithm to find obstacles and eleminate blobs in the image. This method help us to find object's starting point (x,y) , object's width and object's height. Each seperated objects are labeled different numbers and so we can determine the number of objects. Our algorithm for conected component labeling is shown below with a pseudo code.

4.5.1. Connected Component Labeling Pseudo Code

- Get the Image Data into an array.
- Create a Label array and assign zero all values.
- Assign “1” at first cell for Label Array .
- for (i=0; i < (height); i++)
- for (j=0; j < (width); j++)
- Look up 4 neigboorhood for a pixel. (Left, up left, up, up right)
- if its color is same with its neigboor’s color and if its label is zero, get its neighboors label
- if its color is same with its neigboor’s color and if its label is not zero and if its neighboors label different from pixel join the all pizel to new label.
- If there is no same color neighboor assign a new label value.

All different blobs are labeled with different grup numbers. The blow image shows an example of labeled image.

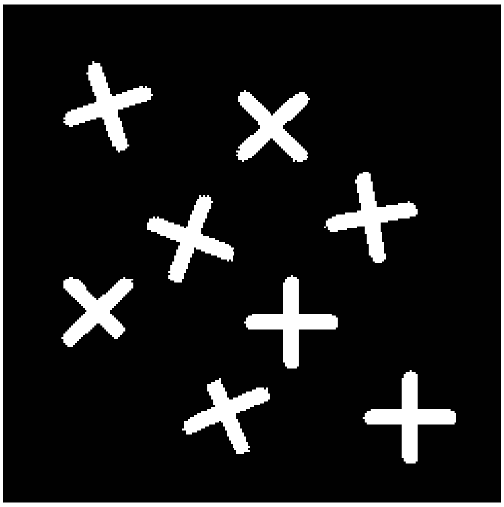


Figure 4.5.1.1. Gray level (binary) image

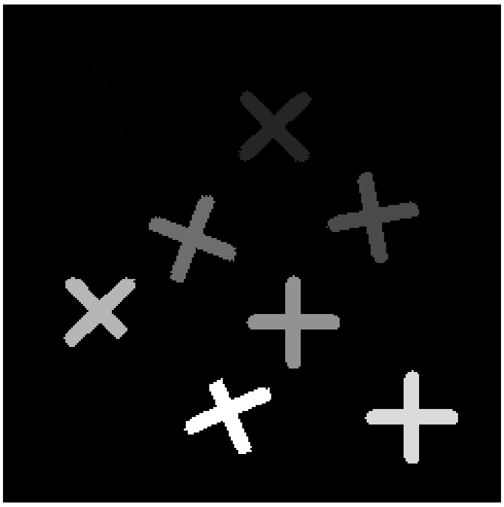


Figure 4.5.1.2. Gray level Labeled image

We use this method in our project in order to find some feature of obstacles. Area, objects start and finish point and weight. After finding feautures we check the area for eleminating the small blobs from the image.

4.6.Feature Extraction

After connected component labeling algorithm we got necessary values from image for obstacle analaysis. We need to know distance from robot to obstacle in order to go through to final destination without any collision. We collected a few image which has obstacles from different distances. We know the real distance which we take pictures and we know the obstacle position. So we can calculate d distance from the image with lineer regression analysis.

	Real Distance	Pixel Number
1-	120	186 pixels
2-	110	180 pixels
3-	100	174 pixels
4-	90	168 pixels
5-	80	162 pixels
6-	70	156 pixels
7-	60	144 pixels
8-	50	126 pixels
9-	40	132 pixels

Lineer Regression Analysis: y= ax + b

$$\begin{bmatrix} N & \Sigma x \\ \Sigma x & \Sigma x2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \Sigma y \\ \Sigma xy \end{bmatrix}$$

Y = distance

X = pixel numbers

a= 0.8752

b=-64.62

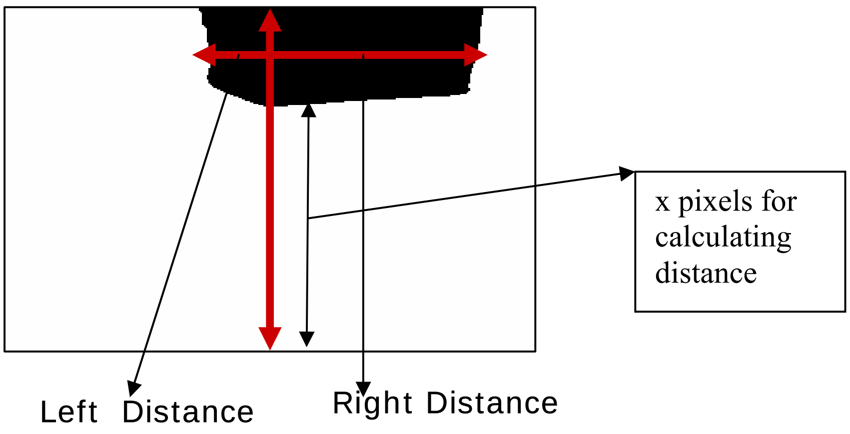


Figure 4.6.1.1. First Scenario (Obstacle)

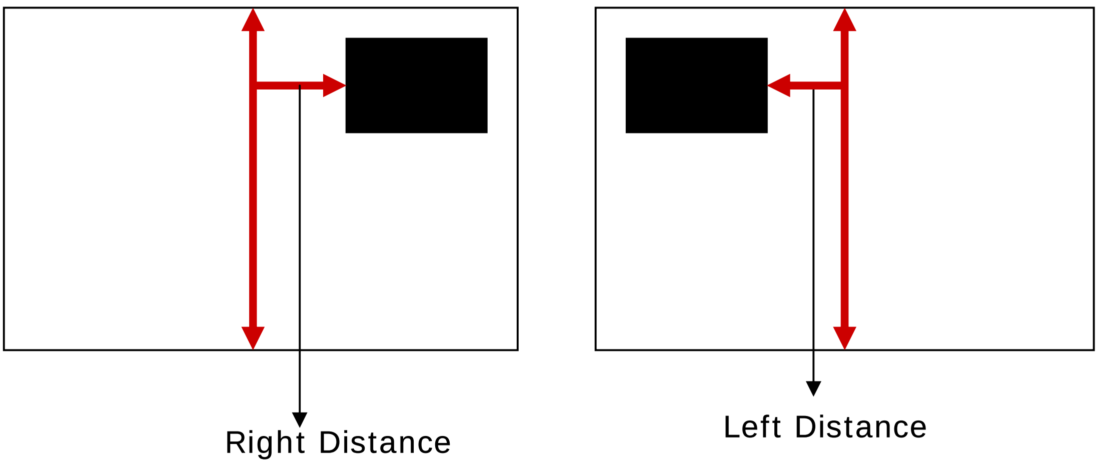


Figure 4.6.1.2. Second Scenario (Obstacle) Figure4.6.1.3. Third Scenario (Obstacle)

Now we know the distance to object and we can calculate the left side distance or right side distance for object with the same technic.

5. SHORTEST PATH FINDING

5.1. Robot Motion

In this project, while the robot is moving, it has to know its coordinations to find the shortest path and control itselfs correctly in an area. We need to know coordinates of the robot in order to go through to final destination without any collision to obstacles.

We have to think robot motion like vektorel not like a point to calculate correct coordiantes. In our project we use two motion type calculation: lineer motion and circular motion.

5.1.1. Lineer Motion

In this motion robot follows a direct way to the choosen point. Also robot's slope doesn't change. We get the distance (d) value from robot's encoder and the calculations which are shown below we find new coordinates.

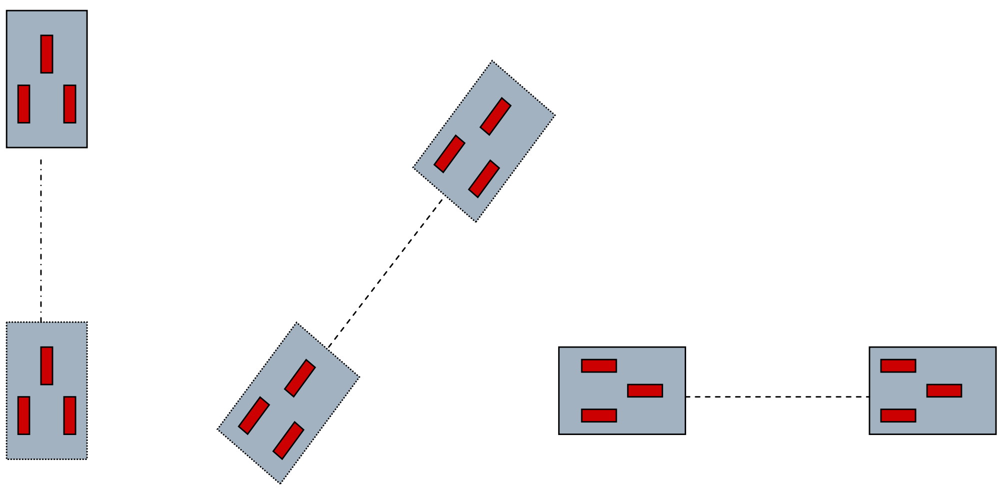


Figure 5.1.1.1. Linear Motion Types

<u>Linear Motion Coordinate Calculation:</u>

Robot's Slope = m= arctan((ony-arkay)/(onx-arkax))

on'x = onx + d* cos(m)

on'y = ony + d* sin(m)

arka'x = arkax + d* cos(m)

arka'y = arkay + d* sin(m)

5.1.2. Circular Motion

Usually we need to change direction of robot, so we have to find new slope with this motion. The calculation for circular motion is shown below.

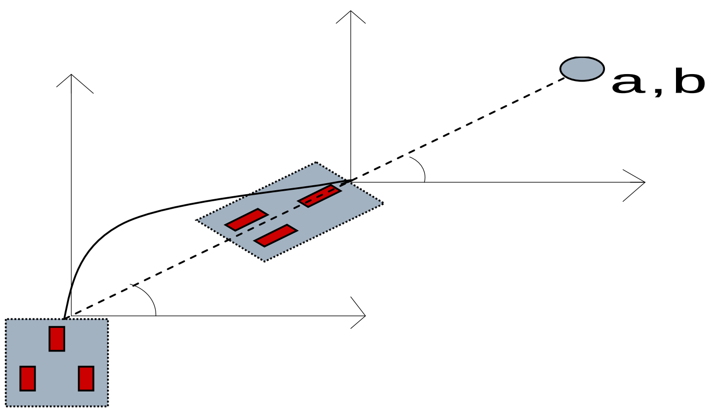
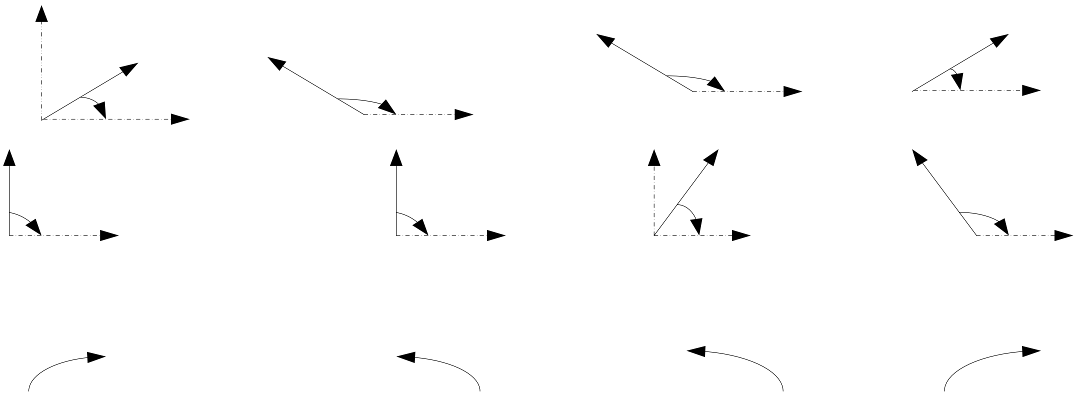


Figure 5.1.2.1. Circular Motion Types



m1 = old slope, m2= new slope

Figure 5.1.2.2. Robot Rotation Direction

Circular Motion Coordinate Calculation:

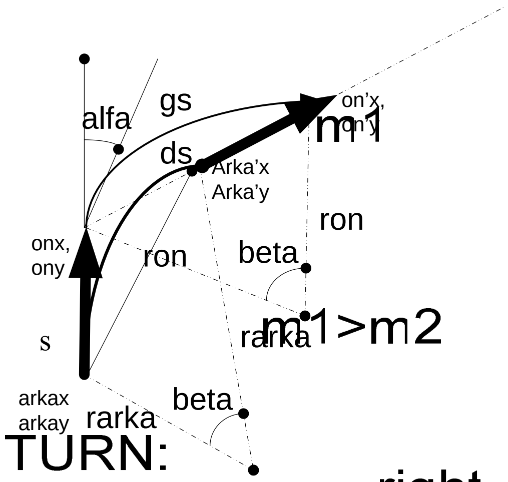


Figure 5.1.2.3. Robot Rotation Calculation

gs= encoders value

alfa= rotation angle

s= length of vektor

ron= s/ sin(alfa)

rarka= s* cos(alfa)/sin(alfa)

gs= 2*pi*ron*beta/360

we know gs from encoder and r from s then

beta= 360* gs / (2*pi*ron)

After we can calculate ds;

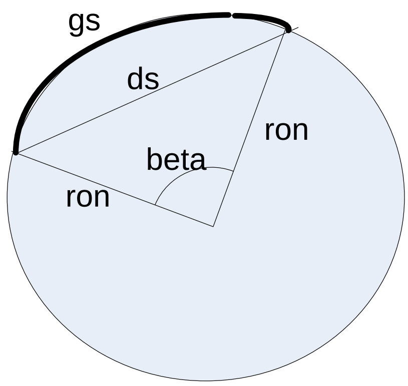


Figure 5.1.2.4. ds Calculation

Cosinus Teorem ;

$$ds^2 = ron^2 + ron^2 - 2 * ron * ron * \cos(beta)$$

$$m_1 = \arctan [(ony - arkay) / (onx - arkax)]$$

$$on'x = onx - ds * \cos (m_1 - alfa - beta/2)$$
$$on'y = ony - ds * \sin (m_1 - alfa - beta/2)$$

$$arka'x = arkax - ds * \cos (m_1 - beta/2)$$
$$arka'y = arkay - ds * \sin (m_1 - beta/2)$$

Then we can calculate new slope of robot.

5.1.3. Shortest Path Algorithm

In this method, we try to go to final destination with the shortest path. Firstly robot finds its direction to final points. “YorungeyeGir()” method is a circular motion and “DumduzGit()” is the linear motion. The robot takes pictures and decides the best points to go untill final position. Algorithm is shown Figure 5.1.2.4.

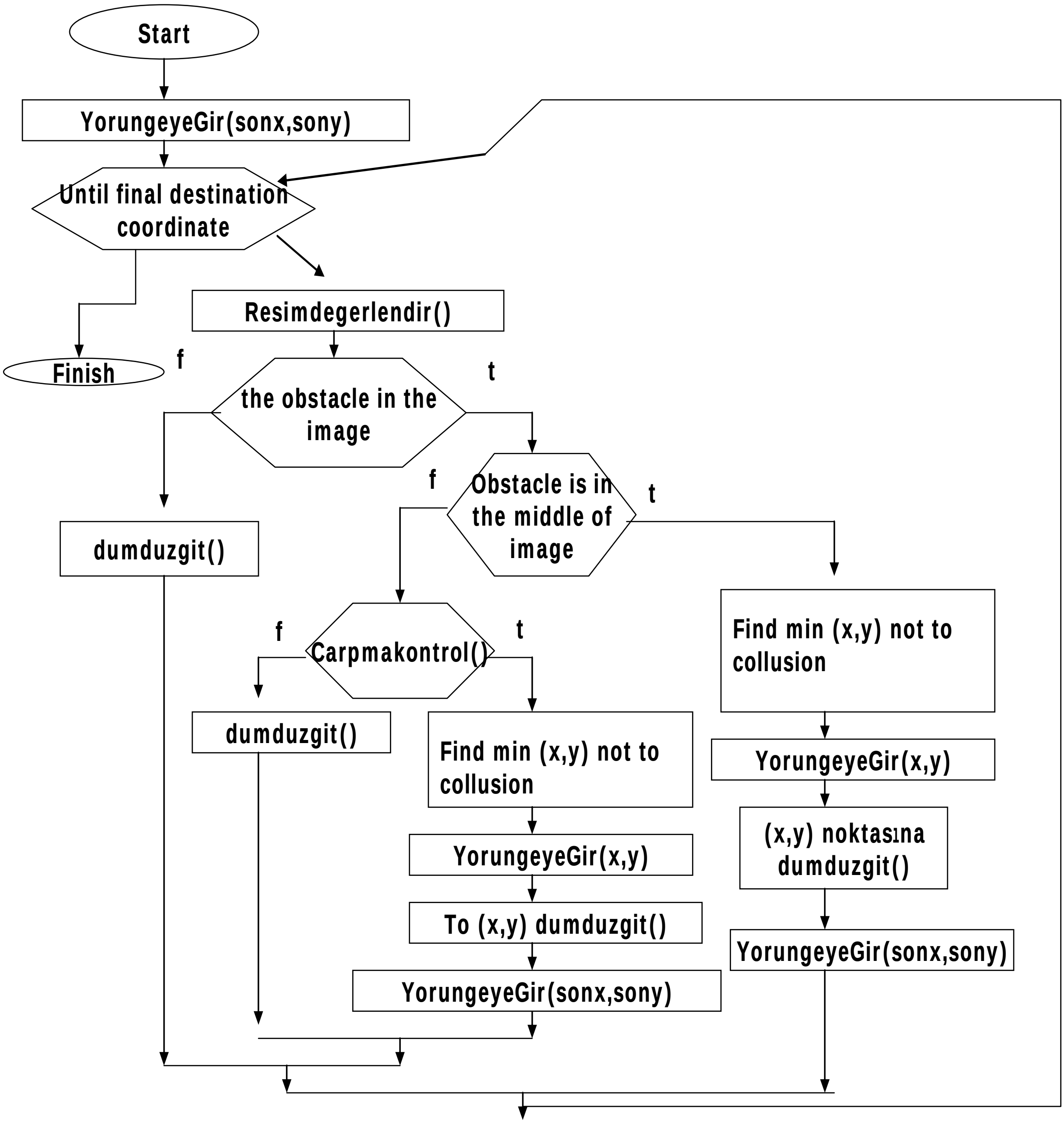


Figure 5.1.3.1 Shortest Path Algorithm

6. RESULT

We tested system and had %90 success. With some enviromental and equipmental reasons we had failed.

With this project we gained a large amount of information on robotics and image processing. We used different technichs from image processing for robot vision in order to solve the problem. Altough we had some problems with robot's mechanical staff, we wanted to get over all problems.

Java is an important object oriented programming language in the information technology. We used advanced java APIs and learned a lot about java. Robot's communication system, vision system, image processing system's codes are written with this technology. So we used different classes which have different duties, on the same framework.

RF communication is a better way for robotic communication system. We had a large amount of information about this subject too. Between robot and computer, communication with RF played a basic role. We had to learn all the details of RF communication.

The important thing is, we learned how to work with a project group and we tried to orginize project schedule. Helping each other in this project made a good effect on our performance. We had an excellent experience with this project.

7.REFERENCES

1. "**Algorithms For Image Processing And Computer Vision**", J. R. Parker, Wiley Computer Publishing, 1997 Canada
2. "**The Median Filter Problem**", http://tracer.lcc.uma.es/problems/mfp/mfp.html#ref4
3. "**Learning Object recognition in a Neurobotic System**", Rebecca Fay, Ulrich Kaufmann, Friedhelm Schwenker and Günther Palm, University of Ulm
4. "**Kmeans Algorithm**", http://en.wikipedia.org/wiki/K-means_algorithm
5. "**Gaussian Smoothing**", http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm
6. **"Reuter J., (1998). Mobile robots trajectories with continiously differentiable curvature: An optimal control approach"**, IEEE/RSJ Proc. Of Int. Conf on Intelligent Robots and System, Victoria B.C., Canada.

RESUME (CV)

1-

Name Surname : Necmi Öztürk

Birthdate : 01-07-1983

BirthPlace : Düzce

High School : Düzce Anatolian High School

Internship : Nortel Netaş Networks, Yıldız Technical University, I-3S Software Company

2-

Name Surname : Can Yüksel

Birthdate : 22-11-1984

BirthPlace : Alanya

High School : Alanya Ayşe Melahat Erkin Anatolian High School

Internship : Ipek Software Company