Computer vision for autonomous mobile robotic applications

Author: Seyed Sadegh Mohammadi

Department of Signals and Systems
Division of Biomedical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2011

Report No. EX047/2011

Master of Science Thesis

Seyed Sadegh Mohammadi
Department of Signal & Systems
Chalmers University of Technology, 2011

Thesis Supervisor: Juan Alvaro Fernandez
Departement of Electrical, Electronic and Automation Engineering
University of Extramadura
Badajoz, Spain

Examiner:Yngve Hamnerius
Departement of Signals and Systems
Chalmers University of Technology
SE-412 96
Goteborg, Sweden

The work presented in this thesis report has been realized at University of Extremadura under the supervision of Dr. Juan Alvero Fernandez during the year 2010-2011.

Master's Thesis 2011: June

Department of Signals and Systems
Division of Biomedical Engineering
Chalmers University of Technology
SE-41296 Göteborg
Sweden

Tel. +46-(0)31 772 1000

Abstract

The aim of the thesis was to develop and implement several computer vision methods that would be needed in the implementation of an autonomous mobile robot for delivering inter-office mail/messages. In particular we present algorithms for camera calibration, robot home position identification, and the detection and recognition of objects present in the robots operating environment.

The robot camera calibration is performed using the Bouguet toolbox. For the robot positioning the Hough transform is applied to detect the dominant lines in the images. The orientation of the lines is used for moving the robot to the home position, which is the starting point for subsequent robot navigation. The Harris corner detector is used in an algorithm for detecting office doorplates. An external OCR (optical character recognition) system was evaluated for the task of recognizing the name on a doorplate. Finally a classifier (kNN) was developed for discriminating between two objects, the ceiling and floor, present in the robots operating environment based on grey-level co-occurrence matrix textural features. A feature selection strategy based on the Bhattacharyya distance was used.

The accuracy of the camera calibration was verified by reconstructing a 3D object of known dimensions and estimating its size from the reconstruction. The positioning and detection algorithms were developed and assessed in a controlled office operating environment. The orientation of the lines detected using the Hough transform was sufficient for controlling the robots position. The corners of the doorplates were correctly detected except when images were affected by saturation or strong reflection. In the texture based classification experiment a classification accuracy (leave-one-out) of 97 % was obtained. The results collectively demonstrate the efficacy of the developed algorithms within the experimental operating environment. Planned future work includes extending and developing these algorithms to work robustly in a more realistic office environment and to implement a fully functioning autonomous mobile robot for delivering office mail and messages.

Acknowledgements

This thesis owes its existence to the help, support, and inspiration of many people. In the first place, I would like to express my sincere appreciation and gratitude to my supervisor Juan Alvaro Fernández for his support and encouragement during this thesis's work. He has provided for an optimum working environment at the university of Extremadura. His uncompromising quest for excellence significantly shapes everyone at the university. Thanks to my best friend, PhD student Hassan Hoseinnia at the department of the industrial engineering for his kind help and advise in review and correction of my thesis. Great thanks and a lot of respect to Dr. Artur Chodorowski for his efficient comments and suggestions. I am very grateful to Prof. Yngve Hamnerius for being the examiner of this thesis work.

Finally, I owe special gratitude to my family for continuous and unconditional support of all my undertakings, scholastic and otherwise.

Contents

1 Introduction

.

During the last decade, robot vision has found widespread applications in industry, medical, public service and etc. Several studies have been done in this area; a new global vision system for tracking of multiple mobile robots is suggested in [1], a new 3D sensing system, in which the laser-structured-lighting method is basically utilized because of the robustness on the nature of the navigation environment and the easy extraction of feature information of interest, is proposed in [2]. Computer vision has been also widely used in biomedical engineering applications, for instance, in [3] a robot-tweezer manipulation system used to solve the problem of automatic transportation of biological cells, where optical tweezers function as special robot end effectors is presented. The accuracy and the precision of the daVinci (Intuitive Surgical, Sunnyvale, CA, USA) surgical system using robot are quantified in [4], and a prototype robotic system designed to assist vitreoretinal surgery and to evaluate its accuracy and maneuverability is developed in [5] Moreover, researchers try to design robots for public service and industrial applications, for instance a robot is used to guide people through the museum [6], a generic methodology for the synthesis of industrial robot applications with sensory feedback at the end-effector level is presented in [7], and a neural network-based vision inspection system interfaced with a robot to detect and report IC (Integrated Circuit)lead defects on-line is presented in [8].

The main concept of this thesis is divided in two parts. The first one called geometric robot positioning, which is moving the robot to a desired location in the robot operation environment. The second application is to use the robot for message delivery purpose. And two effective algorithms are presented to solve these tasks.

In addition, it is essential for the robot to distinguish objects within its operation environment. There are different methods for reaching this goal. In this thesis some methods such as the texture analysis approach mainly based on Haralick ´s features [9] are used. In order to classify objects placed in the robot environment kNN classifier [10] is used.

The rest of thesis is organized as follows.

Chapters 2 gives overview the general coordinate systems involved in robotic applications, and make a relation between them based on algebraic tools will be presented. Then, the basic structure of the pinhole camera model and its calibration process will be studied. Finally, different lens distortions which can affect image quality will be explained briefly.

Chapter 3 the general structure of the robot used in this thesis will be introduced. In addition, the robot operation environment will be explained in detail. Then, two applications for the autonomous mobile robot will be defined. Chapter 4, object discrimination using texture analysis and classification approach will be discussed. Chapter 5 will present the experimental results. Finally, conclusion and future work will be discussed in Chapter 6.

2 Background

This chapter introduces the general coordinate systems involved in robotic applications, and make a relation between them based on algebraic tools such as, rotation and translation transformation. The basic structure of the pinhole camera model and its calibration process will be introduced. Then, different lens distortions which can affect image quality will be explained briefly.

2.1 Coordinate system

Movement in robotics is mainly considered as the local change of a rigid object in relation to another rigid object [11]. In this context, translation is defined as the movement of all mass of a rigid object with the same direction and speed on parallel tracks and rotation is running the mass points along concentric tracks by revolving a rigid axis. Thus, every movement in robot point of view can be modeled as combination of the rotation and the translation [11]. Moreover, object position measurement is usually performed in the Cartesian coordinate system. In the robotic applications it is necessary to have several coordinate systems. Apart from the robot operation environment coordinate system, the number of coordinate systems mainly depend on the number of sensors used on a robot and these coordinate systems can be connected with the simple algebraic transformation such as the rotation (**R**) and the translation (**T**), the principle idea of the rotation and translation is also shown in Fig.2.1. As an example, the equipped robot with a camera, the origin R and C could be based on the robot and the camera respectively, and coordinate of the robot operation environment is specified by the origin W (Fig.2.2). In order to calculate the orientation of the camera coordinate system with respect to the robot coordinate system, the transformation matrix between both coordinate systems are calculated using (2.6).

$$\alpha_C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{2.1}$$

$$\beta_C = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{2.2}$$

$$\gamma_C = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$\mathbf{R} = \alpha_C . \beta_C . \gamma_C \tag{2.4}$$

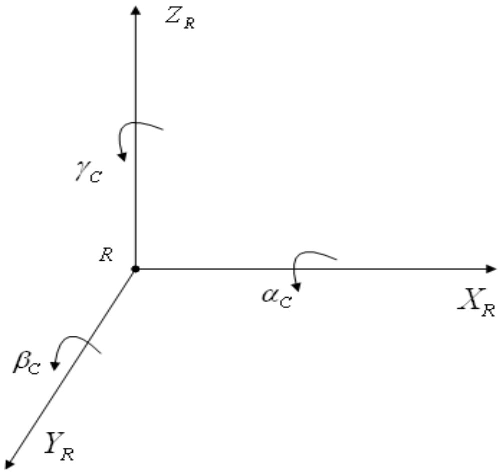


Figure 2.1: Six degree of freedom (i.e., X_R, Y_R and Z_R are translation, and $\alpha_C, \beta_C, \gamma_C$ are rotation in XYZ coordinates).

where angles $\alpha_C, \beta_C, \gamma_C$ are defined as the rotation of the camera around the X_R-axis, Y_R-axis, Z_R-axis of the robot origin (R), respectively.

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & X_R \\ 0 & 1 & Y_R \\ 0 & 0 & Z_R \end{bmatrix} \tag{2.5}$$

where X_R, Y_R and Z_R are the position of the robot in the robot reference coordinate system.

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \mathbf{R} \cdot \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} + \mathbf{T} \tag{2.6}$$

where X_C, Y_C and Z_C are the position of the camera mounted on the robot.

2.2 The Pinhole Camera Model

The pinhole camera is the simplest 3D camera model. Its simplicity enables better understanding of the camera parameters, but also it is the basis for the definition of more complex camera models [13]. In the pinhole camera model, each point of an object in space is projected on a point in a plane called the image plane, which forms the image of the 3D scene. Since the hole diameter is infinitesimal (optical center), every ray of light coming from each 3D point in the scene will pass only through it. Thus, each image sensor element will be affected by a single ray of light. As a result, all points in space are focused. In fact pinhole camera is perfectly suited for modeling the focal length parameter, which controls the zoom factor [13].

However, the major drawback of this model is that it does not cover most of the characteristics and parameters of real cameras, such as the blur associated to out of focus objects and

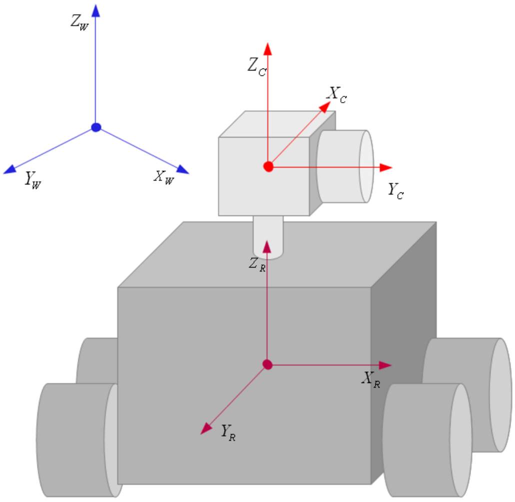


Figure 2.2: Coordinate system of robot-vision system where the origin R and C are based on the robot and the camera respectively, and W is coordinate of the robot operation environment.

the control of the amount of light incident on the sensor. Typically these visual effects are associated to the focal lens and the iris or diaphragm mounted alongside the camera.

2.2.1 The Pinhole Camera Calibration

The pinhole camera calibration procedure requires two steps [11], and it should be noted that the calibration doesn't include the nonlinear effect of the lens distortion (i.e., radial and tangential). In the first step the relationship between a point $(X_i, Y_i, Z_i)^T$ in the world coordinate system and its corresponding point in the image coordinate $(x_i, y_i)^T$ is extracted using a homogeneous projection 3x4 matrix, which is denoted by **A**. The goal is to calculate the projection matrix coefficients which are defined by a_{ij}. In projection matrix; equations are used to show the relationship between a point in the robot operation environment coordinate system and its corresponding point in the image coordinate.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \tag{2.7}$$

$$\begin{bmatrix} Kx \\ Ky \\ K \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.8}$$

Using the previous matrix transformation (2.8), the output coordinate values are obtained in the homogeneous coordinates. Thus it is necessary to perform final conversion between the homogeneous coordinates $(X, Y, Z, 1)^T$ to the Cartesian coordinates $(x, y)^T$. This is accomplished by:

$$x = \frac{a_{11}X + a_{12}Y + a_{13}Z + a_{14}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}}$$

$$y = \frac{a_{21}X + a_{22}Y + a_{23}Z + a_{24}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \tag{2.9}$$

$$a_{11}X + a_{12}Y + a_{13}Z + a_{14} - x(a_{31}X + a_{32}Y + a_{33}Z + a_{34}) = 0$$

$$a_{21}X + a_{22}Y + a_{23}Z + a_{24} - y(a_{31}X + a_{32}Y + a_{33}Z + a_{34}) = 0 \tag{2.10}$$

where it can be assumed that $K = 1$ [14].

According to (2.10) for a known point in the world coordinate and its corresponding point in the image coordinate system two linear equations are available. Consequently, for N projection points, the equation can be written [15]:

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -(x_iX_i) & -(x_iY_i) & -(x_iZ_i) & -x_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -(y_iX_i) & -(y_iY_i) & -(y_iZ_i) & -y_i \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{31} \\ a_{32} \\ a_{33} \\ a_{34} \end{bmatrix} = 0 \tag{2.11}$$

where $i = 1...N$, which is the number of points in the world coordinate and its corresponding points in the image coordinate. In addition, for estimating the parameters (i.e., $a_{11.....}a_{34}$) it is assumed that $a_{34} = 1$ [16]. Based on this assumption (2.11) can be solved with pseudoinverse technique. But the drawback of this estimation is that if the correct value of a_{34} is close to zero, a singularity is introduced. Because of this fact, the constraint $a_{31}^2 + a_{32}^2 + a_{33}^2 = 1$ were suggested [17], which is singularity free. The parameters $a_{11.....}a_{34}$ don't have any physical meaning. Therefore the first step where their values are estimated can be considered as the implicit camera calibration stage [14].

In summary (2.11) can be written [15]:

$$\mathbf{WX} = 0 \tag{2.12}$$

where **W** is the matrix obtained from the calibration points, **X** is the coefficients of the projection matrix .

2.2.2 Internal and External Camera Parameters

In the second step of the pinhole camera calibration, the internal and external parameters of camera are calculated [11]. Moreover, in order to extract physical camera parameters from (2.11) a method based on RQ decomposition were proposed [15]. The decomposition is defined as follows:

$$\mathbf{A} = \lambda \mathbf{V}^{-1}\mathbf{B}^{-1}\mathbf{FRT} \tag{2.13}$$

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

$$\mathbf{B} = \begin{bmatrix} 1+k_1 & k_2 & 0 \\ k_2 & 1-k_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

$$\mathbf{F} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

where $\mathbf{A}$ is projection matrix, λ is an overall scaling factor, and the matrices $\mathbf{V}$, $\mathbf{B}$, and $\mathbf{F}$ contain internal parameters of the camera which are focal length f, principal point (x_0, y_0), and coefficients for linear distortion (k_1, k_2). The External parameters of the camera are defined in the matrices $\mathbf{R}$ and $\mathbf{T}$ which are the rotation and translation, respectively,from the object coordinate system to the camera coordinate system.

2.2.3 Lens Distortion

As mentioned before, the pinhole camera is a efficient model for understanding the general structure of the camera but the problem is, since it is not equipped with a lens it does not have ability to model the possible effect of the lens on the image quality. In [18] three types of distortion are introduced. The first one, named tangential distortion, is caused by imperfect lens shape that is produced manufacturing. But the effect of this type of distortion is insignificant. Because of this fact in most practical situations the effect of this type of distortion is ignored. But radial distortion is one of the most ordinary visual distortions of images seen through a lens. This happens because the magnification at the edge of a lens is differing from its center. There are two types of radial lens distortion [18].

Equation of Optical Distortion In radial distortion the pixels value (gray level) doesn't change at all, the only thing that is changed the pixel coordinates. In the other words, they are distorted radially from the center of image. The optical distortion is governed by [18]:

$$r^{'} = r + k_1 r^3 + k_2 r^5 + k_3 r^5 + k_5 r^7 + ... \tag{2.17}$$

where r is the original radius and $r^{'}$the distorted radius, and the degree of distortion is given by the distortion coefficient (k_1, k_2, k_3, k_5). This equation is modeled in terms of infinite series.

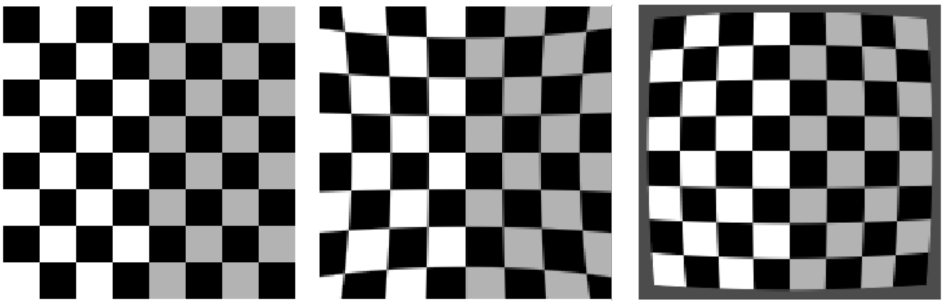


Figure 2.3: Original image (left), effect of the barrel (middle), effect of the pincushion (right)

According to [12], for vast majority of computer vision applications the radial distortion can be safely approximated to finite series using the first two terms of (2.17), which is given by:

$$r^{'} = r + kr^3. \tag{2.18}$$

Type of the radial distortion strongly depends on the sign of the distortion coefficient (i.e. pincushion radial distortion where k is positive and negative k is referred to as barrel distortion). The effect of pinchusion and barrel distortion are shown in Fig. 2.3 in the right and middle, respectively.

2.2.4 Camera Calibration toolbox

The Calibration process is an important step in any computer vision application. It is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3-D position and orientation of the camera frame relative to the world coordinate system (extrinsic parameters) [12]. In many cases, accuracy of the machine vision performance strongly depends on the accuracy of the camera calibration process. There have been many different proposed methods of calibration (e.g. [16], [15]). But one of the most efficient camera calibration toolbox is presented by Bouguet [19]. This toolbox is developed based on calibration methods that are defined by Heikkilä [14] and Zhang [21]. This method requires some images of planar patterns with known geometry. In our calibration process the patterns consist of 9 by 7 squares like a chest board. This software is found to be efficient in this application. In the rest of this section the process of the camera calibration based on Bouguet toolbox [19] will be presented.

Unit conversion

The focal length which is calculated by Bouguet's toolbox is in pixel units [19]. In practical applications it is necessary to convert it to the millimeter units. For reaching this goal, size of each pixel should be calculated. In this project a CCD camera [20] with 640x480 pixel resolution has been used. In order to calculate each pixel dimension, relation between the image size and dimension of the camera monitor should take into account. Assume that the image size is nxm

where n is number of the pixels in the row of the image, and m is number of columns of the image, and dimension of monitor of the camera is NxM centimeter, where N and M are the length and height of the monitor respectively. Therefore, dimension of each pixel is calculated by:

$$\alpha = \frac{N}{n}cm \tag{2.19}$$

$$\beta = \frac{M}{m}cm \tag{2.20}$$

where α and β are length and height of each pixel respectively. Thus, pixel value could be easily converted to millimeter value by:

$$mm = (pixels * 25.4)/dpi \tag{2.21}$$

where dpi is number of pixel in each inch. The result of implementing this formula has been shown in Table 5.2.

Object extraction

In this step, it is possible to extract an object dimension from the image plane using the focal length and the dimension of each pixel. As it is shown in Fig.2.4 two triangle (CPP',Cii') are similar, therefore, objects can be extracted using:

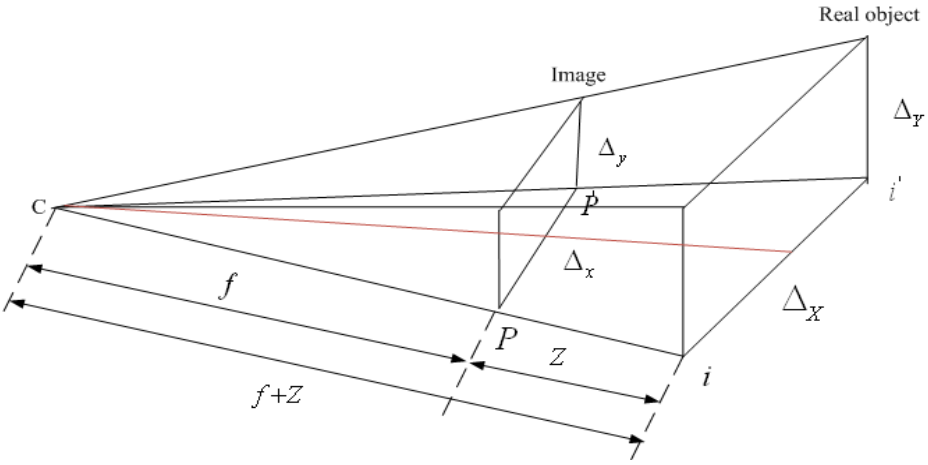


Figure 2.4: Object extraction

$$\frac{f}{f+Z} = \frac{\Delta_x}{\Delta_X} = \frac{\Delta_y}{\Delta_Y} \tag{2.22}$$

$$\Delta_X = \frac{\Delta_x.Z + f.\Delta_x}{f} \tag{2.23}$$

$$\Delta_Y = \frac{\Delta_y.Z + f.\Delta_y}{f} \tag{2.24}$$

where Δ_X and Δ_Y are the dimension of the object in the robot operation environment, Δ_x and Δ_y are the dimension of the object in the image plane, f is the focal length and Z is the known distance from the object to the image plane.

3 Autonomous Mobile Robot Applications.

In this chapter the general structure of the robot and environment of the robot operation will be introduced. Besides, two missions for the autonomous mobile robot will be defined. Finally, two algorithms will be presented in order to handle the robot positioning and message delivery applications.

3.1 Robot Structure

In this section, the basic structure of the robot named mBase, which is used in this project will be introduced.

3.1.1 mBase Robot

Mbase-MR5 is a robot designed and developed by the company MoviRobotics [22], to be used by research groups and universities.To work with this platform a personal computer (PC) with RS232 port, Linux operation system, and at least four megabytes free hard disk is required. This robot is not equipped with a camera and it works based on control engineering techniques. A physical view of the robot is shown in Fig. 3.1. The model and characteristic of the robot is also detailed in Table (3.1).

3.2 Environment Specification

In this section the robot operation environment will be explained. The robot is tested in a known corridor with a following specification (i.e., 6 meters long, 2.5 meters width, 2.5 meters height, there are four rooms whose access doors symmetrically placed by pairs on both sides, and one exit door is placed at the end of corridor). A sketch of this model is shown in Fig. 3.2.

3.3 Robot Missions

There are two missions suggested as applications in this thesis. The first one called geometric robotic positioning, this process strongly depends on assignment task for the robot [11]. Using this method can help the autonomous robot for measuring the robot position at any time. In our robotic application, the robot is drive in the environment, which was explained in the previous section, in this application it is necessary for it to be placed in the center of corridor.

There are two justifications for it: If the robot equipped with a fixed camera it will have be able to visualize both sides of the corridor. Besides, it can be good idea if the user locates the safe distance between robot and the walls, because in some situations robot sensors have some

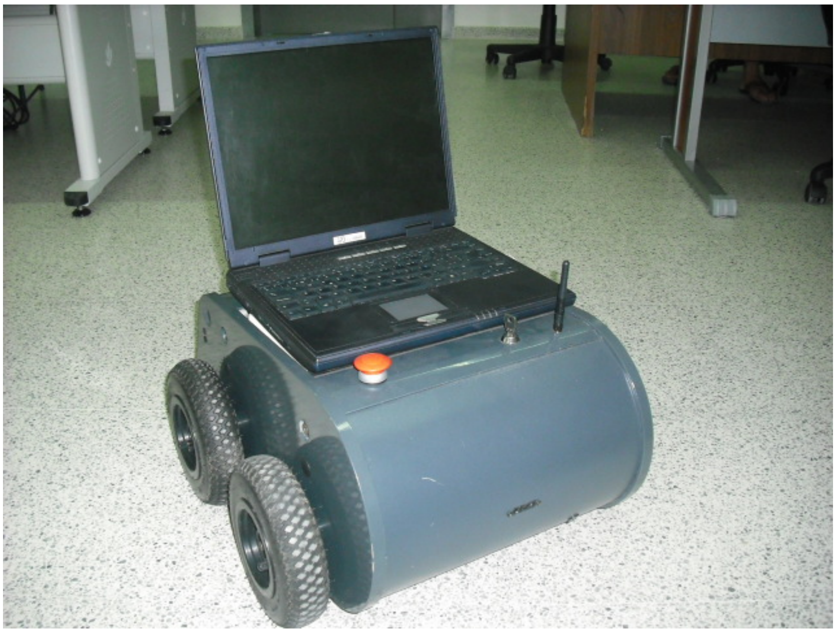


Figure 3.1: A view of mBase robot in the lab.

Table 3.1: Robot basic features

Model	mBase -MR5
Robot measures	53 $cm.$ (L) x 46 $cm.$ (W) x 31 $cm.$ (H)
Lateral wheels	Diameter:20 $cm.$ Pneumatic and air tube.
Castor wheel	Diameter: 5 $cm.$
Maximum Trans. Vel.	1.5 m/s
Minimum Rot. Vel.	$90^0/s$
Motors	12V / 6.1 Nm max / Redu \cing 19.7 : 1
Encoder resolution	500 CPR
Weight	10 Kg (platform), 26 Kg (robot with battery)
Load capacity	16 Kg
Sensors	Four infrared , Encoders, Gyro, Accelerometer
Case construction	Aluminum and plastic machining

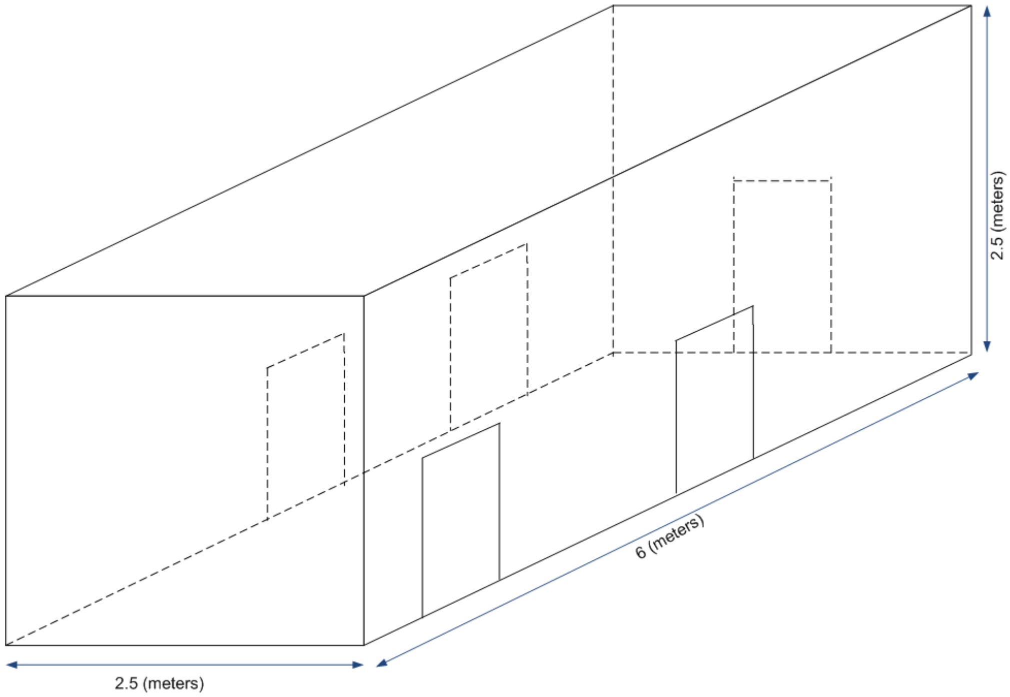


Figure 3.2: Robot operation environment.

delays to send a command about objects. Therefore the probability of robot crashing with the objects placed in its work place will reduce.

The second mission is assigned for message delivery purpose. The robot should navigate in the corridor and send a message after recognizing the address . As it was mentioned there are four room placed in the corridor so the robot has to search for suitable features to detect the doors. The doors generally can introduce by two vertical lines. Line detection algorithms need edge structure of image as an input. Consequently, thus, the edge detection techniques should be applied on image before line detection. In the next step, the robot should look for doorplate, which is placed on the door, the same as before the doorplate should be defined with some features. The doorplates are designed as rectangle-shapes, therefore extract four corner points of the doorplates could be a good solution for doorplate recognition.

Besides, the robot has to be able to read the names, which assures a correct delivery. As a result, it is essential to include an OCR (Optical Character Recognition) [23] algorithm as a part of the process, which will be presented in the following. Therefore, the robot can read the doorplates during the navigation process. So, it is able to determine which address the message must be delivered. Furthermore, the objects in the environment also can be classified using texture analysis and supervised classifier.

The proposed algorithms for the robot missions will be explained in more details in the following section.

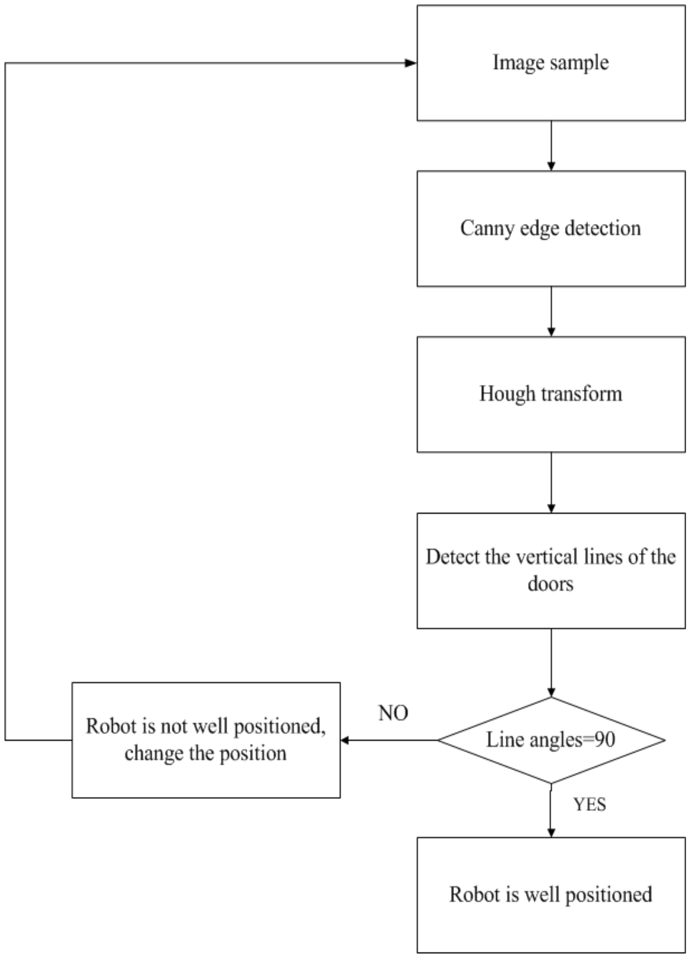


Figure 3.3: Algorithm for geometric robot positioning.

3.4 Algorithms for Handling the Missions

In this section, two efficient algorithms will be proposed to handle the robot missions. The first algorithm involve in the geometric robot positioning, then the calibrated robot will be assigned for it application (i.e., message delivery).

3.5 Geometric Robot Positioning

In this thesis the robot is designed for the message delivery application in the known corridor [11]. For this application it is essential for the robot to be placed in the center of the corridor in order to have ability to visualize both sides of the corridor. Consequently, in order to measure the robot position in the environment reference point(s) has/have to be defined. In this application vertical lines of the door side are defined as the reference points. Main steps of the algorithm are illustrated in Fig. **3.3.**

3.5.1 Canny Edge Detection

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Several edge detection algorithms exist, and this part focuses on a particular one developed by John F. Canny [24]. Due to its well performance, it has become one of the standard edge detection methods and it is still used in research [25].

The Canny operator works in a multi-stage process. At the initial step all the images are smoothed by Gaussian convolution. Then a simple 2-D first derivative operator such as Sobel operator, is applied to the smoothed image in order to highlight the regions with high first spatial derivatives (3.1). After this process the edge strengths can be determined using (3.2) but it doesn't give information about gradient orientation in the image. Thus, the gradient orientation can be estimated as (3.3). The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non-maximal suppression. The tracking process exhibits hysteresis controlled by two thresholds: T_1 and T_2, with $T_1 > T_2$. Tracking can only begin at a point on a ridge higher than T_1. Tracking then continues in the directions out from that point until the height of the ridge falls below T_2. This processing helps to ensure that noisy edges are not broken up into multiple edge fragments.

In general, the width of the Gaussian kernel used in the smoothing phase, and the upper and lower thresholds used by the tracker can control the effect of the Canny operator [24]. Detector's sensitivity to noise is reduced by increasing the width of the Gaussian kernel, at the same time some of the finer detail in the image will be loosed. Moreover, the localization error in the detected edges also increases slightly as the Gaussian width is increased [24]. Setting the lower threshold too high will cause noisy edges to break up (see Fig. 3.5), on the other hand, setting the upper threshold too hight remove efficient edges (see Fig. 3.6). Normally, the upper tracking threshold can be set high and the lower threshold low for good results this statement is illustrated in Fig. 3.7.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3.1}$$

where G_x, and G_y are Sobel operators in x,y direction, respectively.

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{3.2}$$

$$\theta = \arctan(\frac{|G_y|}{|G_x|}) \tag{3.3}$$

where $|G|$, and θ are defined as amplitude of gradient, and direction of gradient, respectively.

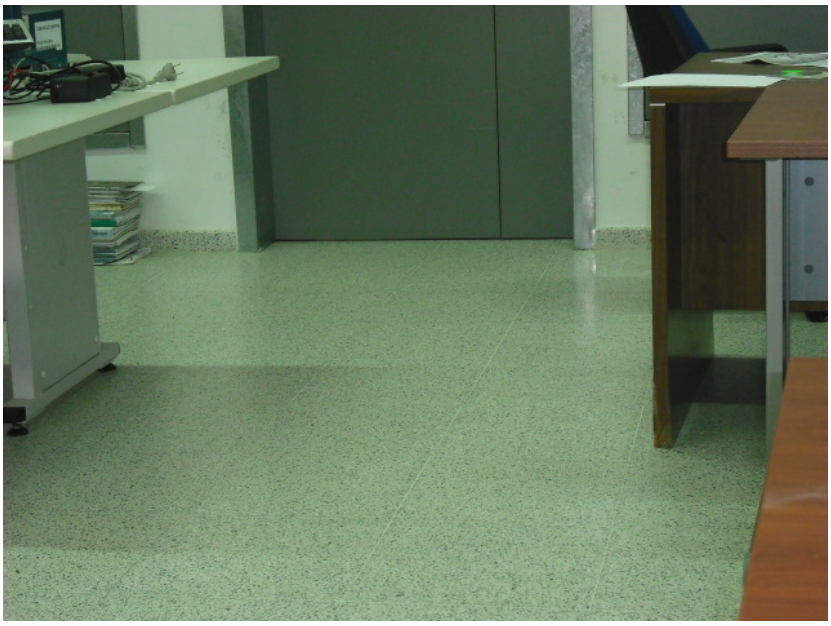


Figure 3.4: Original test image.

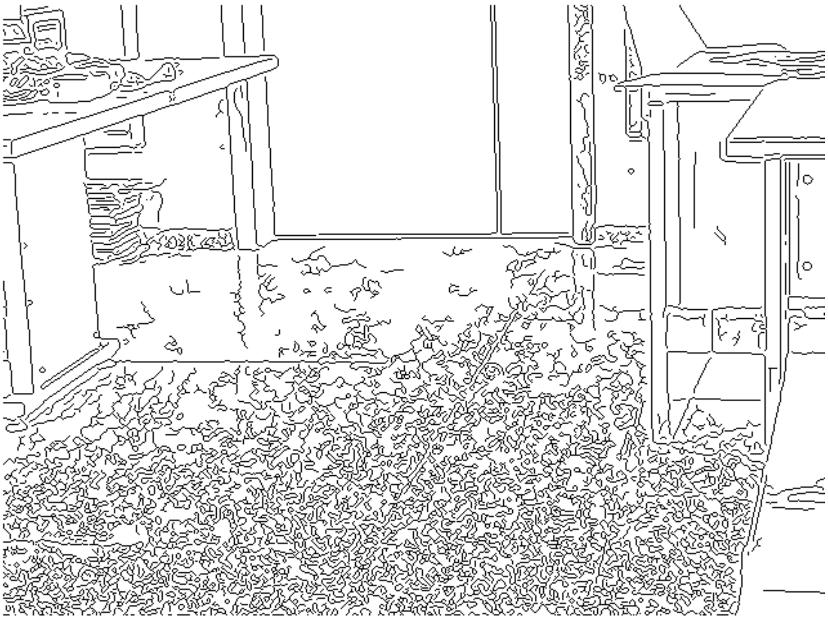


Figure 3.5: Canny result $T1 = 0.0169$, $T2 = 0.0388$.

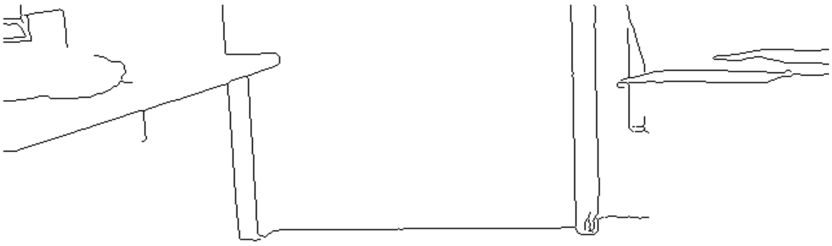


Figure 3.6: Canny result $T1 = 0.58$, $T2 = 0.0188$.

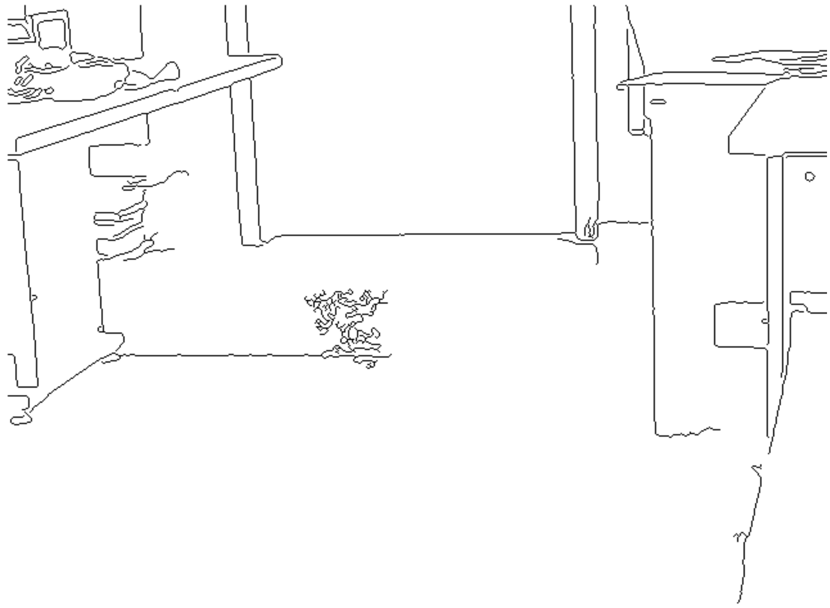


Figure 3.7: Canny result $T1 = 0.3469$, $T2 = 0.0008$.

Hough Transform

The Hough transform (HT), named after Paul Hough who patented the method [26], is a powerful global method for detecting lines. It transforms between the Cartesian space and a parameter space in which a straight line (or other boundary formulation) can be defined.

Theory of Hough Transform

Hough transform (HT) mainly consider to detect the possible straight lines in an image. For every point or pixel coordinates in the image, all the straight lines passing through that point with varying values of line slope and its intersect parameters should be calculated by:

$$y_i = mx_i + c \tag{3.4}$$

where (x_i, y_i) is a pixel coordinate in an image, and m and c , respectively, are slope and intersect of the straight lines passing through the pixel in the image. In addition, This statement has been shown in Fig. 3.8.

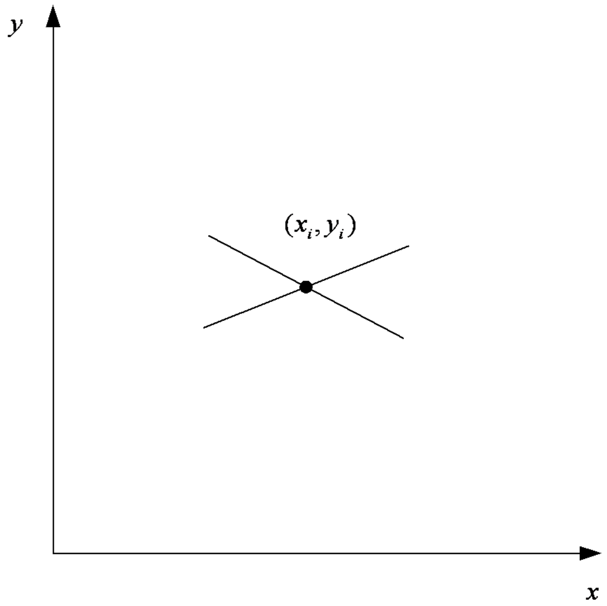


Figure 3.8: Lines through a point in the Cartesian Coordinate.

If the previous equation is expressed in terms of (m, c) instead of (x_i, y_i), then (3.4) can be defined by:

$$c = y_i - mx_i \tag{3.5}$$

Moreover, the idea behind (3.5) is shown in Fig. 3.9.

According to these statements, it is clear that each different line through the point or pixel (x_i, y_i) corresponds to one of the point on the line in (m, c) domain.

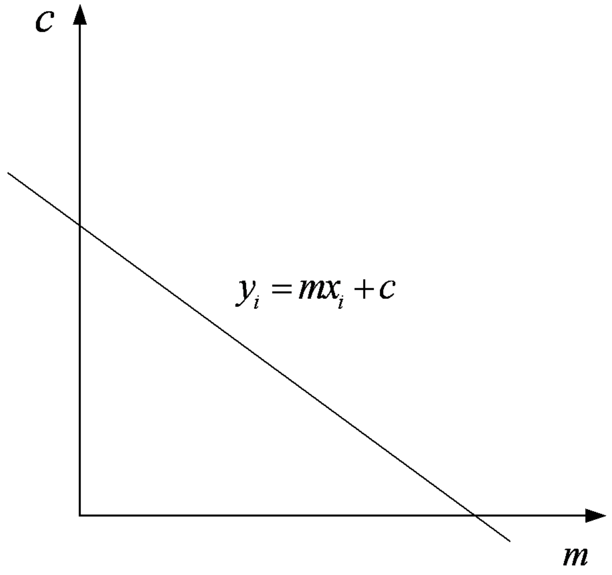


Figure 3.9: Parameter space.

Hough Transform Algorithm

1. Detect all the edge points utilizing suitable edge detection (often Canny edge detection).
2. Quantize the (m, c) space into a two-dimensional matrix H with appropriate quantization level.
3. Initialize the matrix H to zero.
4. Each element of $H(m_i, c_i)$ which corresponds to an edge point is incriminated by 1. The result is a histogram or a vote matrix showing the frequency of edge points corresponding to certain (m, c) (i.e., points lying on a common line).
5. The histogram H is thresholded where only the large valued elements are taken. These elements correspond to lines in the original image [27].

Practical Issue

But previous definition of Hough transform has some significant problems. It can give misleading results when objects happen to be aligned by chance. This causes another disadvantage which is that the detected lines are infinite lines described by their (m, c) values, rather than finite lines with defined end points. To avoid the problem of infinite m values which occurs when vertical lines exist in the image. In the following the alternative formula can be given using:

$$x \sin(\theta) + y \cos(\theta) = \rho \tag{3.6}$$

According to Fig. 3.10 a point (x, y) space is now represented by a curve in (ρ, θ) space rather than a straight line.

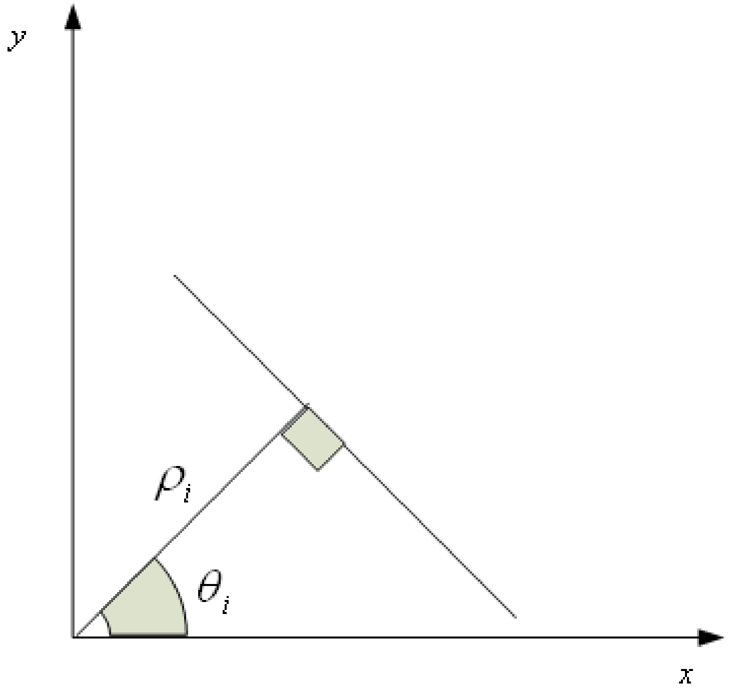


Figure 3.10: The representation of a line (x, y)in (ρ, θ)coordinates.

3.6 Message Delivery Application

In the following a efficient algorithm for the message delivery mission will be explained in detail. Initially,the positioned robot gather information about its operation environment using camera mounted over it. Since the robot task is message delivery, the room should be detected, the vertical lines of the door sides (which are defined the door position) are detected using Hough transform, which was explained in the previous section. The robot should also have ability to detect the doorplates, which is given the office owner name. In order to reach this goal corner detection should be applied. Afterward, the robot must be able to read the names, which assures a correct delivery. An OCR module must therefore be part of the process. The explained algorithm has been depicted in Fig. 3.11.

3.6.1 Corner Detection

Corners in images represent critical information in describing object features that are essential for pattern recognition and identification. As a result a number of corner detection methods have been proposed in the past like [28], and [29]. In this part of the thesis the Harris corner detector is used to detect the rectangle shape of the doorplates. The Harris corner detector is a popular interest point detector due to its strong invariance to rotation, scale, illumination variation and noise [30]. This detector is based on the local auto-correlation function of a signal, this function measures the local changes of the signal with patches shifted by a small amount in different directions. A discrete predecessor of the Harris algorithm was presented in [29],

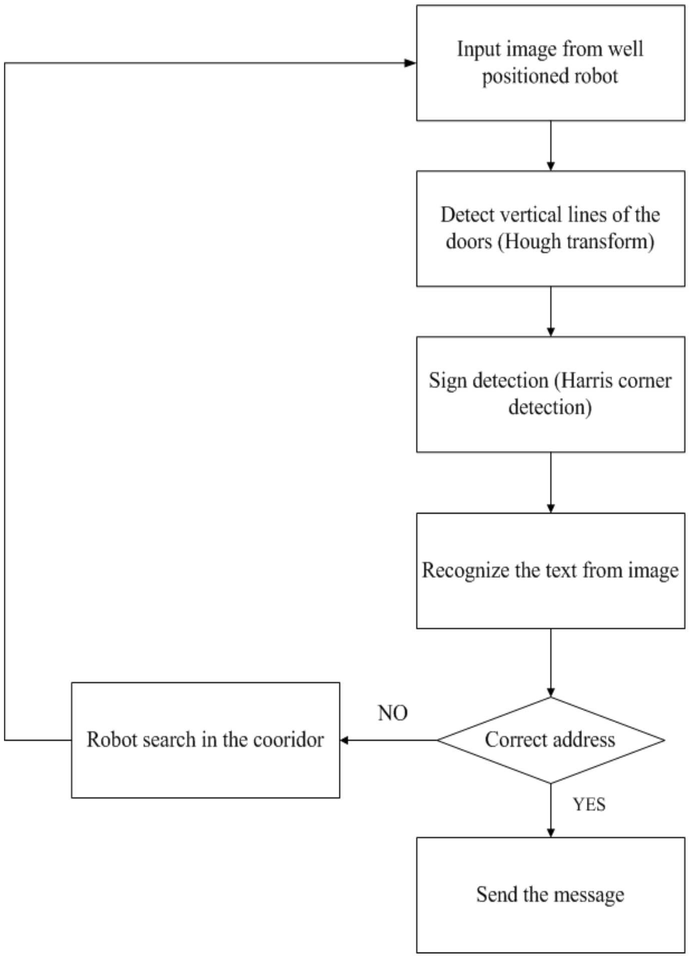

Input image from well positioned robot
Detect vertical lines of the doors (Hough transform)
Sign detection (Harris corner detection)
Recognize the text from image
Correct address
NO
Robot search in the cooridor
YES
Send the message

Figure 3.11: Algoritm of message-delivery robot.

the discreteness refers to the shifting of the patches. The Harris corner detection is governed by the following equations:

$$S(x, y) = \sum_{W} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \tag{3.7}$$

where $S(x, y)$ is defined as auto-correlation function, $I(.,.)$ denotes the image function, and point (x_i, y_i) in the Gaussian window W centered on (x, y).

The shifted image is approximated by a Taylor expansion truncated to the first order terms.

$$I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i) I_y(x_i, y_i)] \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{3.8}$$

where $I_x(.,.)$ and $I_y(.,.)$ denote the partial derivatives in x and y direction, respectively.

Substituting approximation (3.8) into (3.7) yields.

$$S(x, y) = [\Delta x \ \Delta y] \cdot \ \mathbf{C}(\mathbf{x}, \mathbf{y}) \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{3.9}$$

$$\mathbf{C}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} < I_x^2 > & < I_x I_y > \\ < I_x I_y > & < I_y^2 > \end{bmatrix} \tag{3.10}$$

where $\mathbf{C}(\mathbf{x}, \mathbf{y})$ captures the intensity structure of the local neighborhood. Let λ_1, λ_2 be the eigenvalue of the matrix $\mathbf{C}(\mathbf{x}, \mathbf{y})$

1. If $\lambda_1 = 0$, $\lambda_2 = 0$ then this pixel (x, y) has no features of interest.
2. $\lambda_1 = 0$, and $\lambda_2 > 0$, then an edge is found.
3. $\lambda_1 >> 0$, and $\lambda_2 >> 0$, then a corner is found.

Harris [31] suggested that exact eigenvalue computation can be avoided by calculating the response function which is defined by:

$$R(\mathbf{C}) = det(\mathbf{C}) - K \cdot trace^2(\mathbf{C}) \tag{3.11}$$

where $det(\mathbf{C})$ is the determinant of the local structure matrix $\mathbf{C}$, and K is tunable parameter where its value is from 0.04 to 0.15, and $trace^2(\mathbf{C})$, is the sum of the elements in the main diagonal of $\mathbf{C}$.

Harris corner detector algorithm

1. Filter the image with a Gaussian kernel.
2. Estimate intensity gradient in two perpendicular direction for each pixel, $\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}$. This is performed using a 1D convolution with the kernel approximating the derivative.
3. For each pixel and a given neighborhood window:

- Calculate local structure matrix $\mathbf{C}$.
- Evaluate the response function $R(\mathbf{C})$.

4. Choose the best candidates for corners by selecting a threshold on the response function $R(\mathbf{C})$ and perform non-maximal suppression [27].

3.6.2 Optical Character Recognition

Optical character recognition (OCR) is defined as the process of converting images of machine printed or handwritten numerals, letters, and symbols into a computer-processable format. This is a powerful technique, that has great ability to handle less restricted forms of text, according to this issue OCR has been introduced as the most important application area in machine perception [23].

OCR Systems As it is shown in Fig. 3.12 the structure of OCR system contains three logical components [23]:

1. Input text : Input text can be text images.
2. OCR software block.

- Document analysis (extracting individual character images).
- Recognizing these images (based on shape).
- Contextual processing (either to correct misclassification made by recognition algorithm or to limit recognition choice).

3. Recognition result which is sent to application using an output interface.

OCR Software Block

Document analysis: In general, extracting the text from image document is performed by a process called *document analysis* [23]. In this process poor quality of text image is compensated. Some image processing techniques such as image enhancement, line removal and noise removal are applied. Image enhancement methods emphasize character versus non-character discrimination. Line removal, lines which may interfere with character recognition, and noise removal suppress portions of the image that are not part of characters. In addition, many OCR systems use connected component analysis in order to isolate individual characters from the text image [23].

Character recognition: Two essential components in a character recognition algorithm are the feature extractor and the classifier [23]. All the characters are described by a set of descriptors which are determined by feature analysis. The derived features are then used as input to the character classifier. One of the most common classification methods is template matching in this method matching, individual image pixels are used as features.

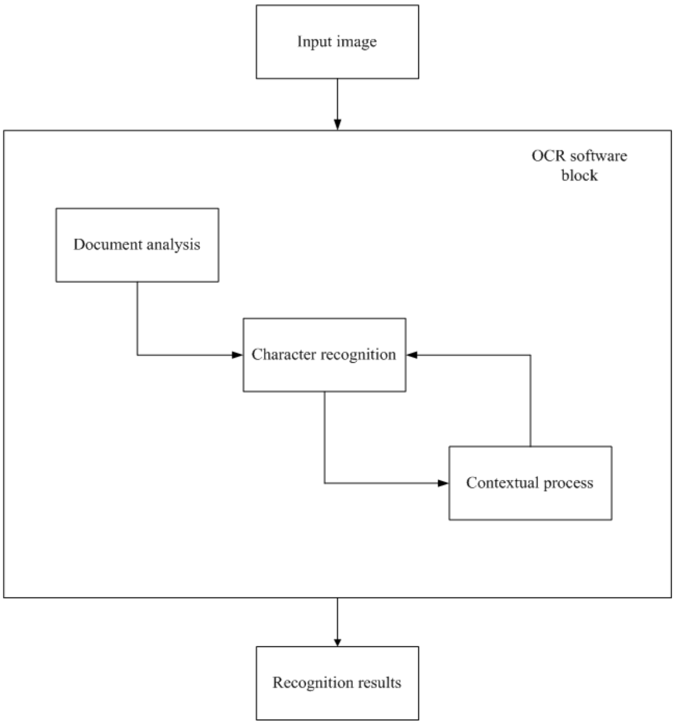


Figure 3.12: General structure of an OCR system [23].

In this thesis template matching classifier is performed by comparing an input character image with a set of templates (or prototypes) from each character class. Each comparison results in a similarity measure between the input character and the template. One measure increases the amount of similarity when a pixel in the observed character is identical with the same pixel in the template image. If the pixels differ, the measure of similarity may be decreased. At the end, the character identity is assigned as the identity of the most similar template [23].

Contextual Processing: Result from character recognition is not perfect and it is normally merged with some misspelling error. Thus, the result of recognition can be post-processed to correct the recognition errors [23].

4 Object detection

In this chapter object detection methods based on color models and texture analysis will be discussed. In addition, supervised classifier will be used for evaluating discrimination power of the texture features.

4.1 Texture Analysis

4.1.1 Introduction

Texture is one of the most important characteristics used in determining objects or regions of interest from different types of images [9]. Like photomicrograph, a aerial, or satellite images. Different studies have shown that human eyes are mostly sensitive to three fundamental features (i.e., spectral, textural, and contextual) [9]. These pattern elements used in human interpretation of color photographs [9]. Spectral features are described in the context of electromagnetic spectrum, which varies between the visible or infrared band of electromagnetic field, whereas texture features contain information within spectrum bands. On the other hand, contextual features contain information around interested regions in the images being analyzed.

Among these three features (spectral, textural, and contextual) our eyes are mostly sensitive to tonal and textural features. This thesis is mainly focused on texture features. Texture is repeating pattern of local variations in image color/intensity or something consisting of mutually related elements. Some types of the texture which can be analyzed with texture analysis are shown as an example in Fig. 4.1.

In fact, most of the significant information about structural arrangement of surface, and their relationship to the surrounding environment are included in texture. Texture features are quite computational resistant, therefore different studies had been done to extract efficient texture information [9]. Relative frequencies of various gray levels on the normalized image [32], power spectra restricted first-and second-order Markov meshes employed autocorrelation functions [33]. Their methods were efficient in some extent, but all of them have the same drawback, no specific definition or model for texture were proposed in their methods.

But the most effective texture features were suggested by Haralick [9], he defined fourteen efficient features based on the statistical nature of texture.

4.1.2 Textural Features Equation

Texture analysis is one of those areas in image processing which still lacks fundamental knowledge. Many different empirical and semi-empirical methods to texture analysis have been proposed. In general, texture analysis problems are divided into four main groups:

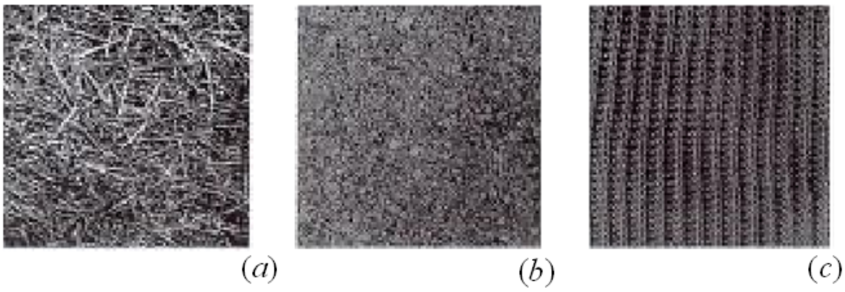


Figure 4.1: Textures example: (a) grass, (b) cork, (c) knitted fabric [34].

- *Classification*: The goal is to assign an unknown sample image to one of a set of known texture classes.

- *Segmentation*: Partitioning of an image into homogeneous properties with respect to texture.

- *Synthesis*: Building a model of image texture, which can then be used for generating the texture.

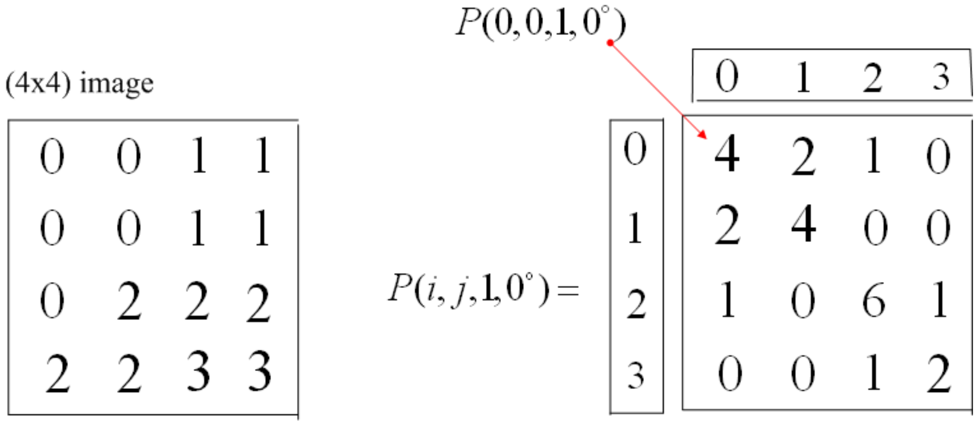
- *Shape from texture*: Texture in the 2D image is used to estimate surface orientations in the 3D scene.

As it was mentioned previously, the procedure that was suggested by Haralick [9] for obtaining the textural features of an image is based on the assumption that the texture information on an image I is contained in the overall or average spatial relationship, which the gray tones in the image I have to one another. More specifically, Haralick assumed that this texture information is adequately specified by a set of gray tone spatial-dependence matrices which are computed for various angular relationships and distances between neighboring resolution cell pairs on the image. All of Haralick's textural features are derived from these angular nearest-neighbor gray tone spatial-dependence matrices.

In this method, image to be analyzed is rectangular and has Nx, and Ny resolution cells in the horizontal and vertical directions, respectively. The method is applied to a quantized version of the input image (gray scale with $Ng = 8$ level). He also defined "Gray Level co-occurrence Matrix" (GLCM) which are based on the repeated occurrence of gray level in the texture. The GLCM $P_{\varphi,d}(a,b)$, for an image of size MxN, is a matrix of non-normalized frequencies describing how frequently two pixels with gray level a, b are separated by distance in d direction φ, and it is implemented on the four different directions $(0, 45, 90, 135)$ degrees and distance d. An example of co-occurrence Matrix in horizontal direction with the unit distance is shown Fig. 4.2.

GLCM is performed in the four different directions :

$$P(i,j,d,0^\circ) = \#\{[(k,l),(m,n)] \in L, k-m=0, |l-n|=d, I(k,l)=i, I(m,n)=j\} \tag{4.1}$$



$$\begin{array}{|cccc|}\hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \\ \hline\end{array} \qquad P(i,j,1,0^{\circ}) = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 4 & 2 & 1 & 0 \\ 1 & 2 & 4 & 0 & 0 \\ 2 & 1 & 0 & 6 & 1 \\ 3 & 0 & 0 & 1 & 2 \end{array}$$

Figure 4.2: 4x4 image (right), spatial relationships of pixels in horizontal direction with the unit distance (right) ($d = 1$).

$$P(i,j,d,45^{\circ}) = \#\{[(k,l),(m,n)] \in L, |k-m| = d, |l-n| = -d, I(k,l) = i, I(m,n) = j\} \tag{4.2}$$

$$P(i,j,d,90^{\circ}) = \#\{[(k,l),(m,n)] \in L, |k-m| = d, l-n = 0, I(k,l) = i, I(m,n) = j\} \tag{4.3}$$

$$P(i,j,d,135^{\circ}) = \#\{[(k,l),(m,n)] \in L, |k-m| = d, |l-n| = d, I(k,l) = i, I(m,n) = j\} \tag{4.4}$$

$$L = L_x x L_y \tag{4.5}$$

where #{...} refer to number of set, and L_x and L_y are horizontal and vertical spatial domain, and L is the set of resolution cells of the image ordered by their row-column designations, and # denotes the number of elements in the set. Moreover, these matrices are symmetric:

$$P(i,j,d,a) = P(j,i,d,a) \tag{4.6}$$

Haralick [9] defined fourteen features based on GLCM, and all of them are addressed in **Appendix I**.

4.1.3 Classification

Introduction

Pattern recognition, one of the most important aspects of artificial intelligence, is an appropriate field for the development, validation and comparison of different learning techniques: statistical or structural, supervised, inductive or deductive [10].

In machine learning and pattern recognition an object is described using pattern, which can be provided utilizing a transducer or sensor connected to the recognition system in the form of a data set. For the classification process it is essential that characteristic information about observed pattern is confined in this data set, so-called a *pattern-signature* or *fingerprint* [10].

The data which is provided using a sensor or transducer have a random nature, as a result the statistical approach mainly is used for analyzing data for pattern recognition. Furthermore, a supervised classifier is utilized for analysis of the data. In this database there are labeled patterns belonging to the M predefined classes $\omega_1, \omega_2, \omega_3$,...., ω_M. The labeled patterns are repeatedly introduced to the classifier in order to derive a decision rule optimizing a given criterion [10].

The general structure of a supervised statistical classification chain is presented in Fig. **??**. In general, the aim of classification is to classify any unknown pattern using its signature.

The supervised classification consists of the three following steps [10]:

- Feature vector extraction from the data recorded by a sensor, named signatures, for instance Haralick [9] feature which will allow the different classes to be separated.
- Data analysis: it is used to reduce the initial dimension of the feature space, at the same time keeping the discriminant properties of extracted features.
- Classification: is the decision step and associates a label with an unknown pattern [10].

For extracting a feature vector, there is no specific rule for finding the best salient features and their nature mainly depends on the application. However, many different techniques such as statistical methods are involved in this step [10].

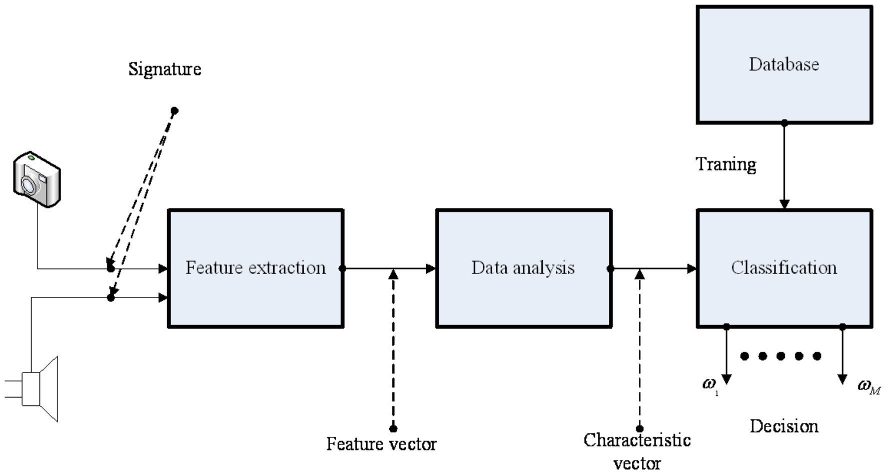


Figure 4.3: Supervised statistical classification flowchart [10].

Data Analysis method

The data analysis methods are mainly concerned about selecting a subset of basic elements from a set of feature vectors, while keeping discriminant properties of extracted features. As it is shown in Fig. 4.3, it is the step between extracted features from signatures and the classification process.

Generally, there are two types of data analysis approaches for reducing the feature space using feature selection. The methods using sequential methods, for instance SFS (sequential forward selection), SBS (sequential backward selection). But, they don't have the ability to remove the whole feature redundancy [10].

Principal Component Analysis (PCA) One of the well known linear method for data analysis is the Kahunen-Loeve transform (KLT) or the principal component analysis method (PCA), which maximizes the variance of projected vectors [10]. It is defined by a matrix having as rows the eigenvectors of the feature space covariance matrix Σ_x.

This matrix, the PCA algorithm reduces the feature vector dimension based on redundancy removal between the projection vector components. This process performed continuously, since the PCA removes any redundancy between the components of the projected vectors. The final covariance matrix results in the transformed space in diagonal form [10]:

$$\Sigma_y = \mathbf{K}\Sigma_x\mathbf{K}^T = diag[\lambda_2\lambda_3....\lambda_n] \tag{4.7}$$

where $\Sigma_x$and feature space covariance matrix and $\{\lambda_i\}_{i=1..n}$ referred as the eigenvalues of matrix. But the PCA output results don't give any information about their labels [10]. For instance, if this algorithm use on a feature matrix which contains information like temperature, pressure, etc, it has ability to reduce the number of dimensions of feature vector, without much loss of information [35], but it doesn't have ability to clarify which label (temperature, or pressure) is more valuable, in the other word the PCA algorithm is an unsupervised data analysis method.

Supervised Classifiers

The classifier is the decision element of a pattern recognition system. The result of the two previous stages, which extracted the most essential information about input pattern (images or signals), will be evaluated using a classifier in order to make the decision about the quality of the features. The general structure of a supervised classifier is shown in Fig. 4.4. The implementation such a classifier is equivalent to finding out, during the training phase, and to using, during the test phase, which is defined by:

$$x \in \omega_i \Rightarrow g_i(\mathbf{x}) = max[g_j(\mathbf{x})] \tag{4.8}$$

where x is feature vector, $\omega_1, \omega_2, \omega_3,....,\ \omega_M$ are labeled patterns belonging to the M predefined classes, and $g_1, g_{2,}g_3, ..., g_M$ are called the discriminant functions.

kNN (k-nearest neighbor) classifier kNN classifiers are the most commonly supervised classification techniques in pattern recognition applications. In this classifier the voting kNN (k-Nearest Neighbour) classifier assigns an unknown sample to a majority class of its k nearest neighbours. The decision rule is very simple and can be generalized to the multi class case (see Fig. 4.5).

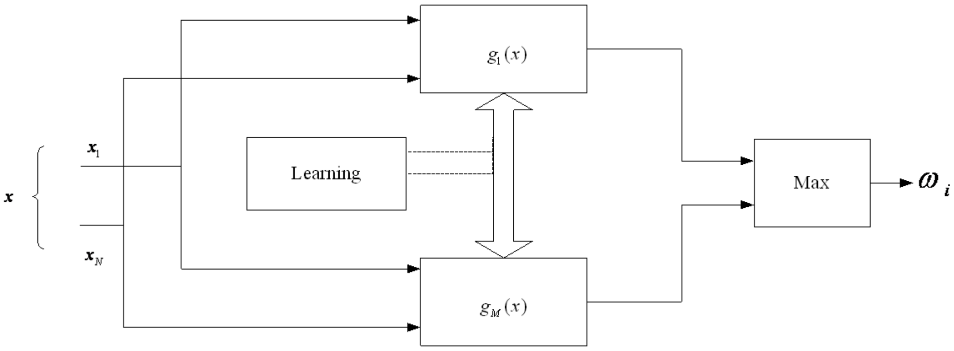


Figure 4.4: General structure of supervised classifier [10].

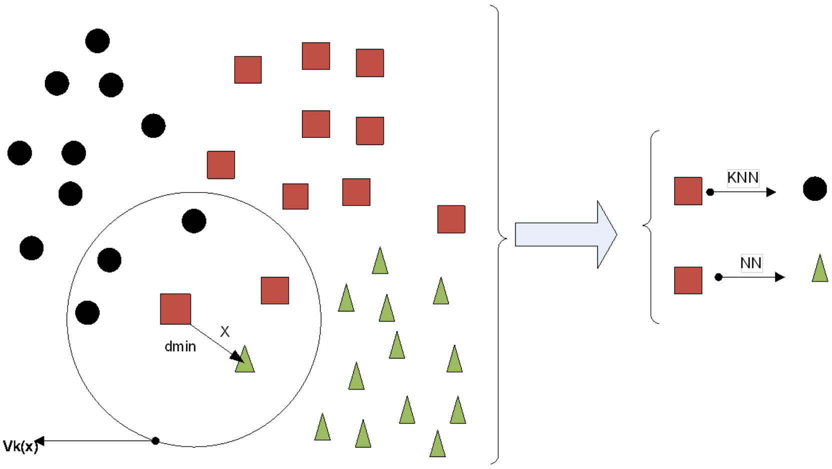


Figure 4.5: kNN classifier principle. $V_k(x)$ is the k^{th} order neighborhood of the unknown vector x and $dmin$ is a distance to the nearest neighbor [10].

For computing a given x the k-nearest neighbors from training set the Euclidean pattern distance is measured (4.9). Then, $\mathbf{x}$ will be assigned to a majority class of its k-nearest neighbors.

$$d(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{D} (\mathbf{x}_j - \mathbf{x}_{ij})^2} \tag{4.9}$$

where $d(\mathbf{x}, \mathbf{x}_i)$ is an Euclidean distance between vectors x and x_i and D is dimensionality of the feature vector x = $[\mathbf{x}_1 \mathbf{x}_2 \ \mathbf{x}_3 ... \mathbf{x}_D]$ and $x_i = [\mathbf{x}_{i1} \mathbf{x}_{i2} \ \mathbf{x}_{i3} ... \mathbf{x}_{iD}]$.

Classifier testing The final step is the classifier's performance evaluation. We need to estimate the classifier's probability of error (error rate), i.e. how many errors (wrong classifications) we expect when using the classifier. The straightforward way to evaluate a classifier is simply counting the number of errors on an independent test data set. The estimate of the classifier's error rate is then the ratio:

$$E = \frac{N_e}{N} \tag{4.10}$$

where E is error rate, N is the number of test samples, and N_e is the number of misclassified samples.

5 Experimental Results

This chapter is divided in two parts; first, the camera which is used in this thesis will be calibrated using Bouguet toolbox [19], and the size of a test object will be reconstructed from the image plane. The second part is organized in three examples: example 2 will demonstrate the geometric robot positioning. The message delivery task will be examined in example 3. The objects, which are placed in the robot operation environment, will be classified using texture analysis and kNN classifier and the results of the process will be presented in example 4.

Example 1 *The camera used in this project (Finepix J10 [20]) will be calibrated using Bouguet's toolbox [19]. In addition, an object will be extracted from image plane using mathematical approach which were explained in chapter 2.*

1. Necessary Patterns For Calibration. Different images of the pattern grids from different view angles and orientations are captured (see Fig. 5.1).

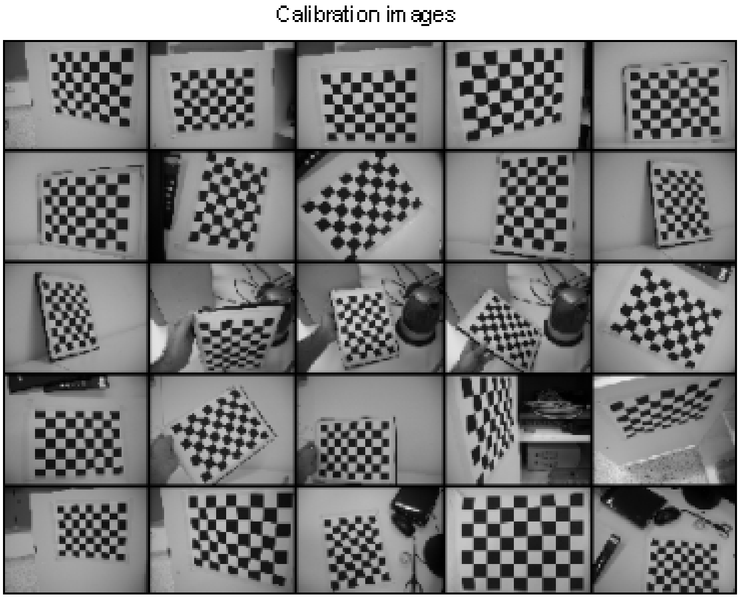


Figure 5.1: 25 patterns captured by camera

2. Corner Extraction

   In order to compute the amount of lens distortion, it is essential to extract the corners of each square in the pattern grids (see Fig. 5.2).

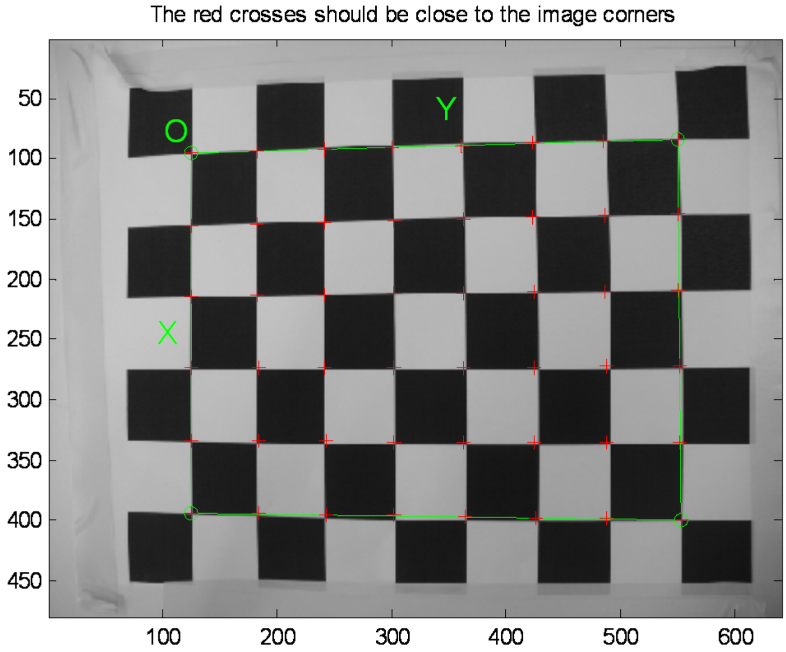


Figure 5.2: Detect Corners on Pattern grid.

3. Calculating Camera Internal Parameters

   In this toolbox four camera internal parameters are considered:

   - Focal length (In pixel value): f
   - Displacement of the center of the image (Principal point) .
   - Distortion coefficients (Radial and tangential): k
   - Angle between x and y pixel axes (Skew coefficient).

   Internal parameters of the camera which are calculated using the toolbox, are shown in Table 5.1. The toolbox also has the ability to calculate rotation and translation of the camera. Besides, it is able to plot the relative positions of the grids with respect to the camera (see Fig. 5.3) and the position of the camera regarding to cthe coordinate of robot operation environment in a 3D plot like the one shown in Fig. 5.4.

   In order to make a decision on the appropriate distortion model to use, it is sometimes very efficient to visualize the distortion effect on the image, and the importance of the radial component versus the tangential component of distortion [19].

Table 5.1: Camera internal parameters

Paramenter name	Parameters value	Parameters error
Focal Length (f)	[677.852 680.841]	± [2.896 2.825]
Principal point	[309.828 245.802]	± [3.478 3.019]
Skew coefficient	[0.00000]	±[0.00000]
Distortions (K)	[$-$0.16807 0.11262 0.00008 $-$0.00223]	± [0.01482 0.08262 0.00091 0.00105]

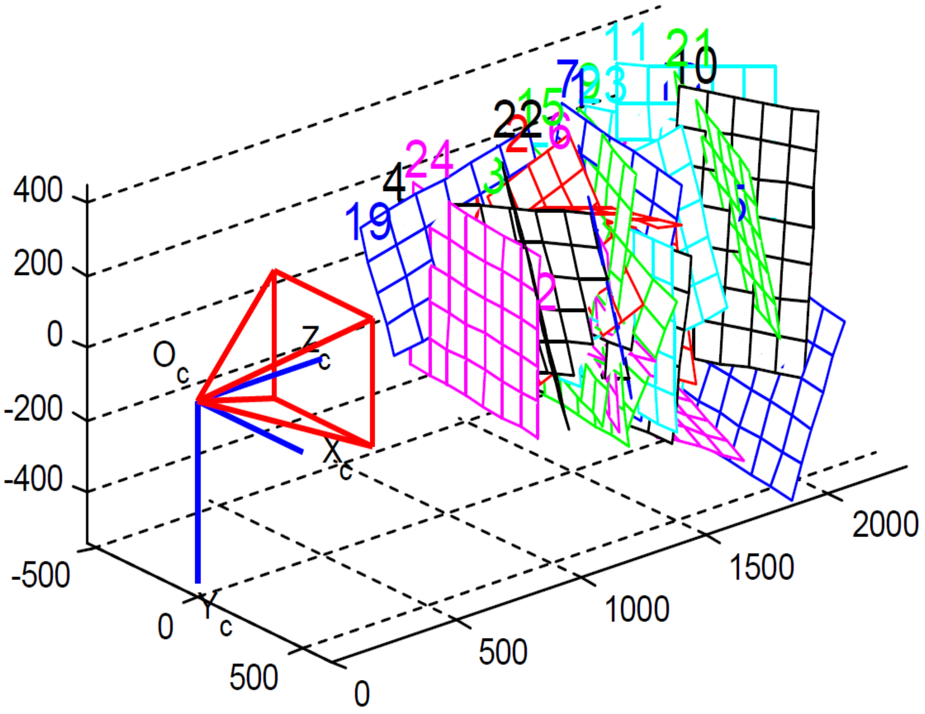


Figure 5.3: Extrinsic parameters (camera-center).

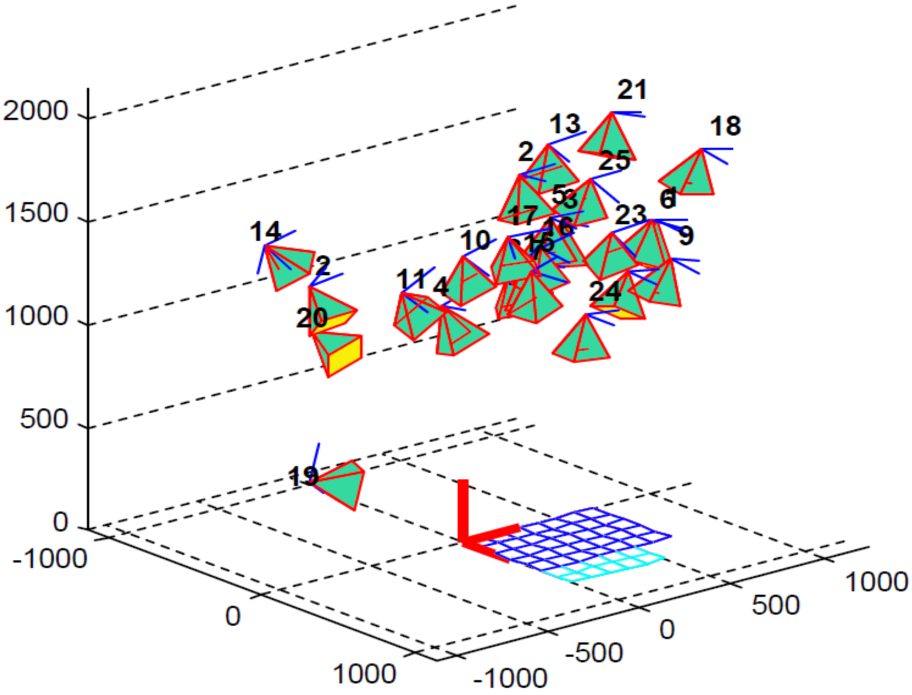
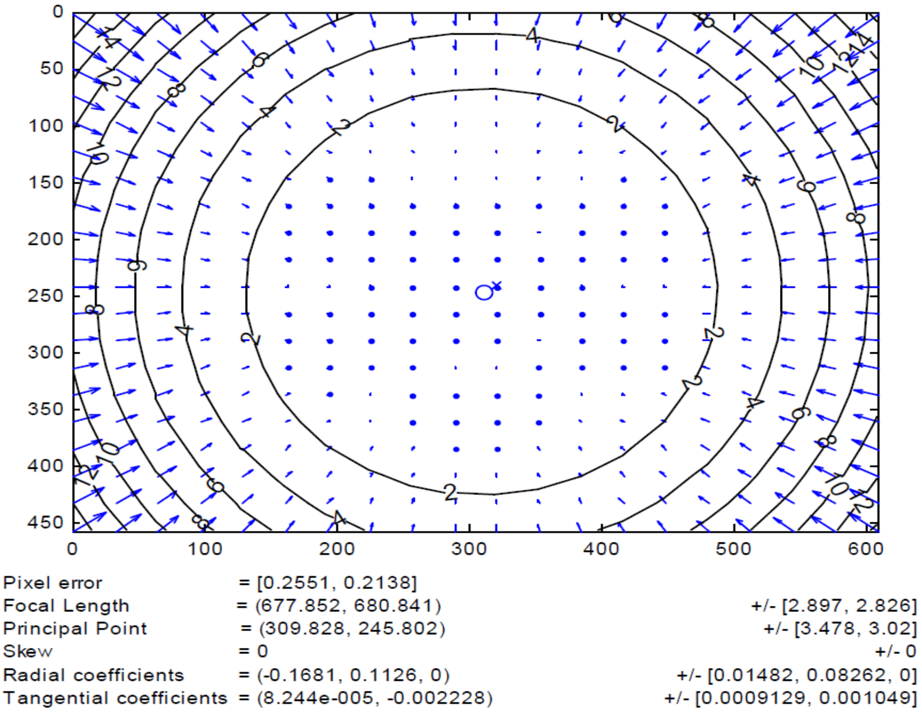
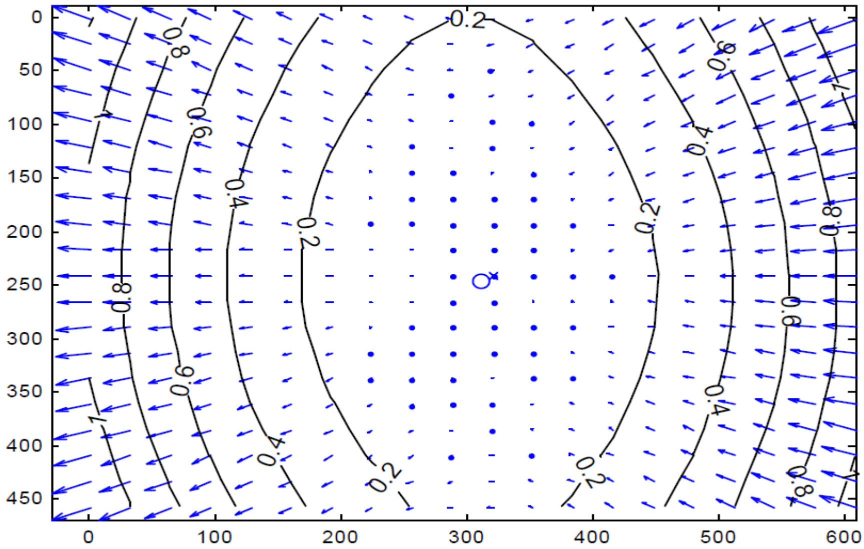


Figure 5.4: Extrinsic camera parameters (world center).



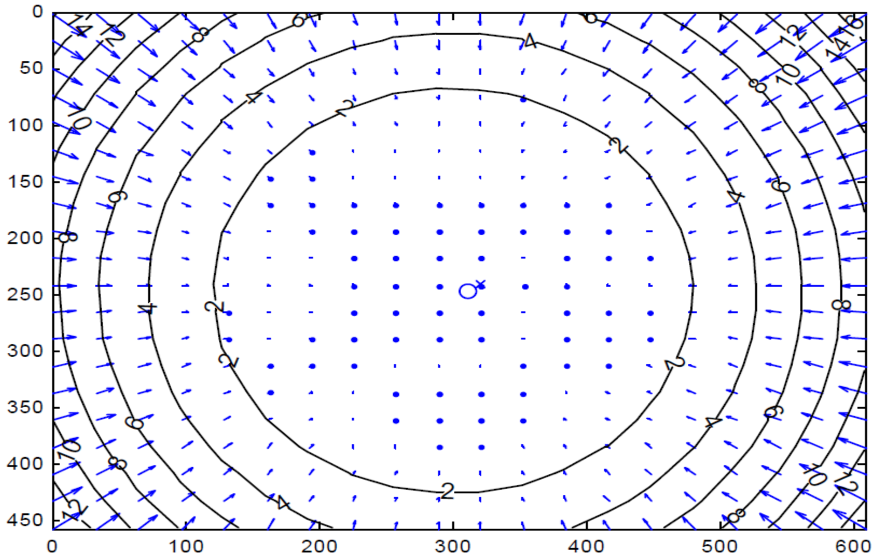
```
Pixel error            = [0.2551, 0.2138]
Focal Length           = (677.852, 680.841)               +/- [2.897, 2.826]
Principal Point         = (309.828, 245.802)               +/- [3.478, 3.02]
Skew                   = 0                                          +/- 0
Radial coefficients    = (-0.1681, 0.1126, 0)        +/- [0.01482, 0.08262, 0]
Tangential coefficients = (8.244e-005, -0.002228)   +/- [0.0009129, 0.001049]
```

Figure 5.5: Radial component of the distortion model.



```
Pixel error              = [0.2551, 0.2138]
Focal Length             = (677.852, 680.841)          +/- [2.897, 2.826]
Principal Point          = (309.828, 245.802)          +/- [3.478, 3.02]
Skew                     = 0                                       +/- 0
Radial coefficients      = (-0.1681, 0.1126, 0)   +/- [0.01482, 0.08262, 0]
Tangential coefficients  = (8.244e-005, -0.002228) +/- [0.0009129, 0.001049]
```

Figure 5.6: Tangential component of the distortion model.



```
Pixel error              = [0.2551, 0.2138]
Focal Length             = (677.852, 680.841)          +/- [2.897, 2.826]
Principal Point          = (309.828, 245.802)          +/- [3.478, 3.02]
Skew                     = 0                                       +/- 0
Radial coefficients      = (-0.1681, 0.1126, 0)   +/- [0.01482, 0.08262, 0]
Tangential coefficients  = (8.244e-005, -0.002228) +/- [0.0009129, 0.001049]
```

Figure 5.7: Complete distortion model.

Table 5.2: Camera internal parameters in millimeter

Size of each Pixel (cm)	dpi	Focal length (mm)	Error (mm)	Object dimension (cm)
[0.008 0.0079]	[312 312]	[55.095 55.338]	[0.2354 0.2296]	[8.45 26.73]

The radial component of distortion is shown in Fig. 5.5. Each arrow represents the effective displacement of a pixel induced by the lens distortion. According to Fig. 5.6, which shows the impact of the tangential distortion, maximum induced displacement is 0.6 pixels (at the upper left corner of the image). Furthermore, the effect of the complete distortion model (radial + tangential) on each pixel of the images observes that points at the corners of the image are displaced by as much as 14 pixels (see Fig. 5.7). This plot is very similar to the radial distortion plot, showing the tangential component could very well be discarded in the complete distortion model. In these figures '+' indicates distorted corner points, and 'o' represents the actual corner point.

4. Object reconstruction

   In the following the object which is shown in Fig. 5.8 will be extracted from the image plane. For reaching this goal size of each pixel in x, y direction is calculated using (2.19- 2.20), and the pixel values are converted to the millimeter (2.21). Assuming that the distance to the object is known, the size of the object is calculated (2.22- 2.23). The process is summarized is in Table Table 5.2.

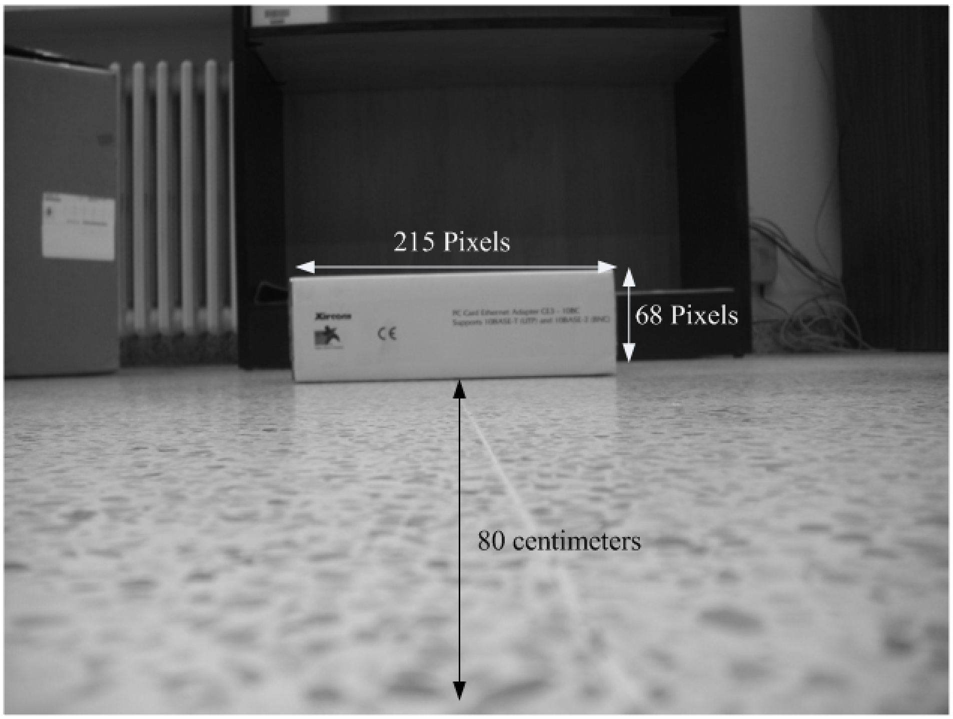


Figure 5.8: Reconstruct the object dimension from the image plane. Z =80cm, x=68cm, y=215cm

Example 2 *The presented algorithm for the geometric robot positioning has been tested in the known environment (corridor with length 6 meters long 2.5 meters height, and 2.5 meters width). In addition, there are four rooms whose access doors symmetrically placed by pairs on both sides (in the right and the left side), the whole corridor is shown in Fig. 5.9. In the initial step, the robot should be centered within its operation environment. According to the algorithm 3.3 choosing the reference points are quite necessary. The reference points must be a rigid objects [11]. In this case, two vertical lines which belong to the edges around the doors are chosen as the reference points (see Fig. 5.10). The practical issue of this algorithm has been presented in this example.*

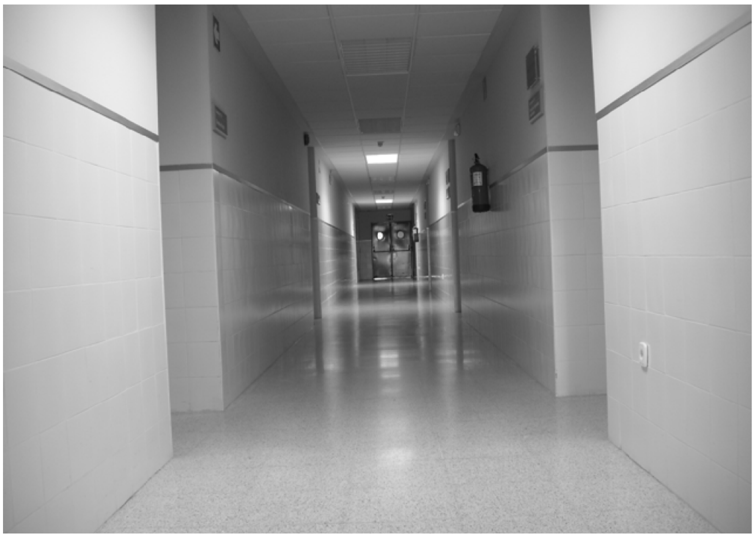


Figure 5.9: The whole corridor image.

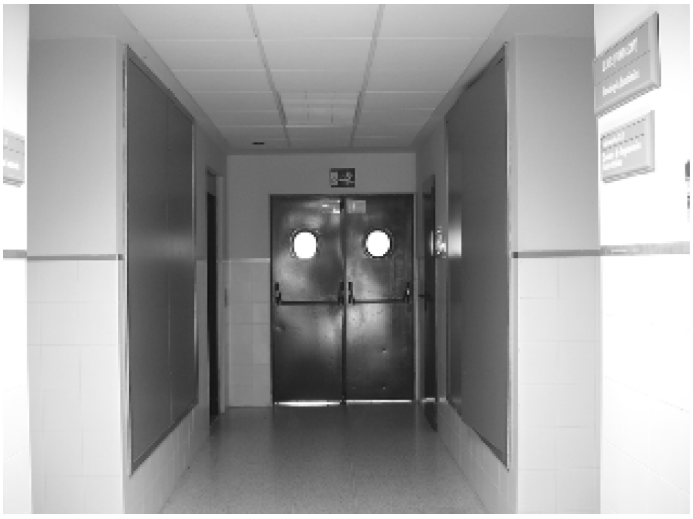


Figure 5.10: Reference image for the robot positioning.

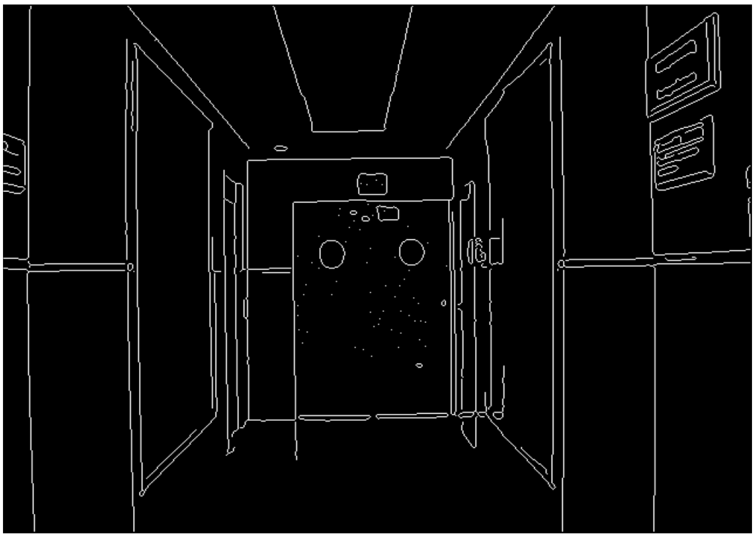


Figure 5.11: using Canny edge detection. Strongest lines are extracted from image (white edges).

According to the geometric robot positioning, in the first place it is essential to extract the edges of images Canny edge detection (see Fig. 5.11).

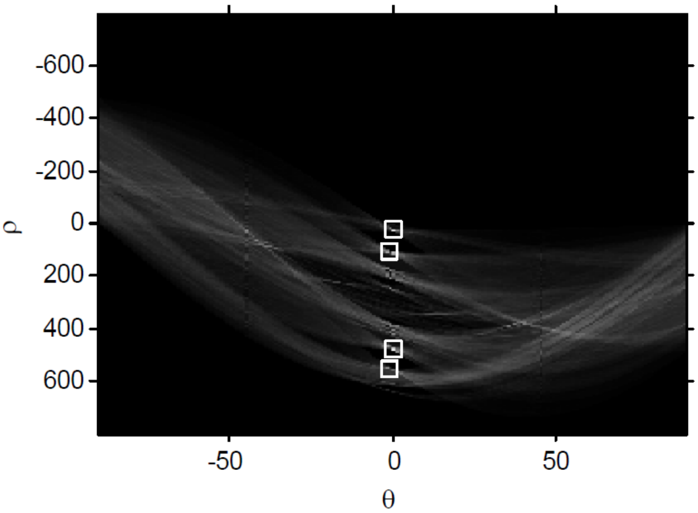


Figure 5.12: Detected lines marked with'⊡' on the vote histogram.

In the next step, Hough transform will be applied to the result of the previous step in order to segment the vertical lines which are used to determine the robot orientation in the scene. If the angle of each line is closer to 90 degrees then robot is centered within its operation environment. The vertical detected lines on the Hough 's vote histogram, which are presented with distance (ρ)-orientation (θ), is shown in Fig. 5.12. In addition detected lines on the image have been shown in Fig. 5.13. Other possible orientations of the robot have been shown in Figs. 5.14-5.19.

According to Table 5.3 the orientation of the robot, which is presented in Fig. 5.12 is closer to 90 degrees which means that it is centered. Therefore, it can be used for rest of the process.

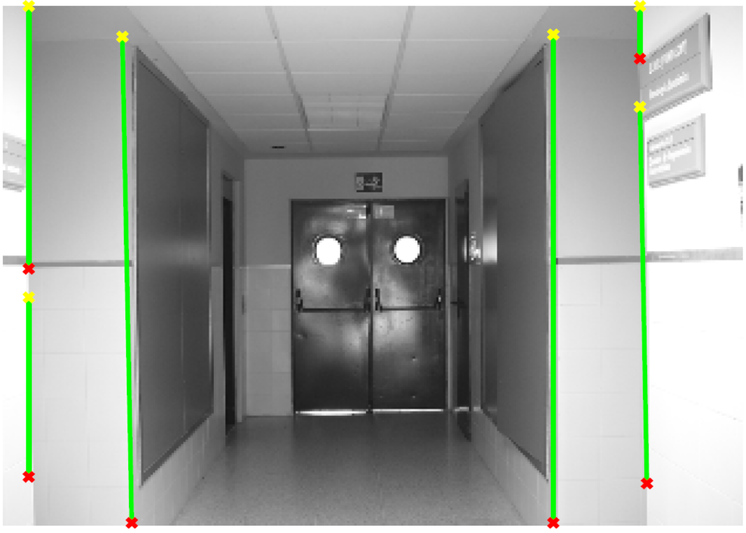


Figure 5.13: Line segments in the centered (well positioned) robot.

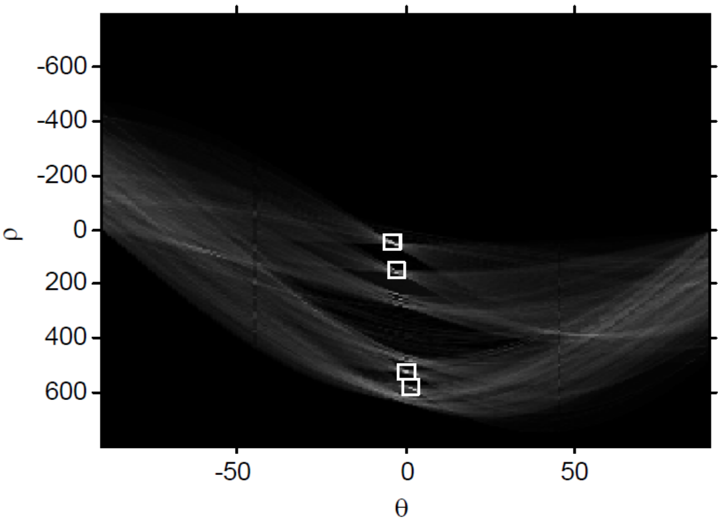


Figure 5.14: Detected lines marked with'□' on the vote histogram (right-tilted).

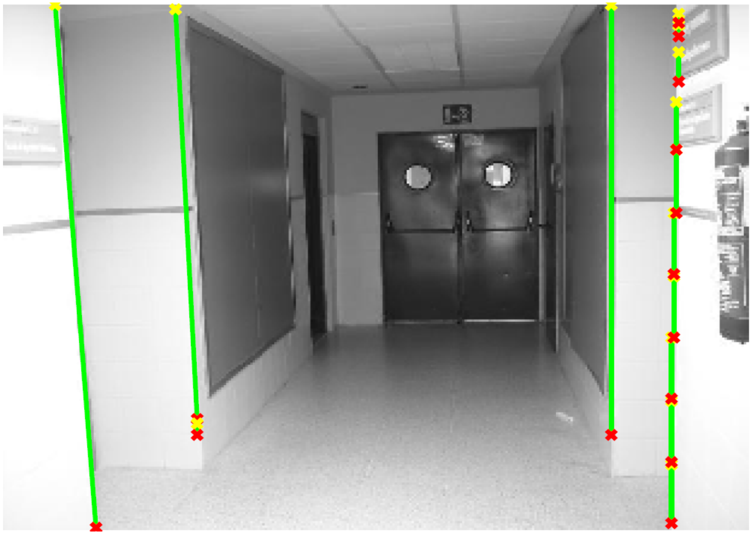


Figure 5.15: Line segments in the right-tilted robot.

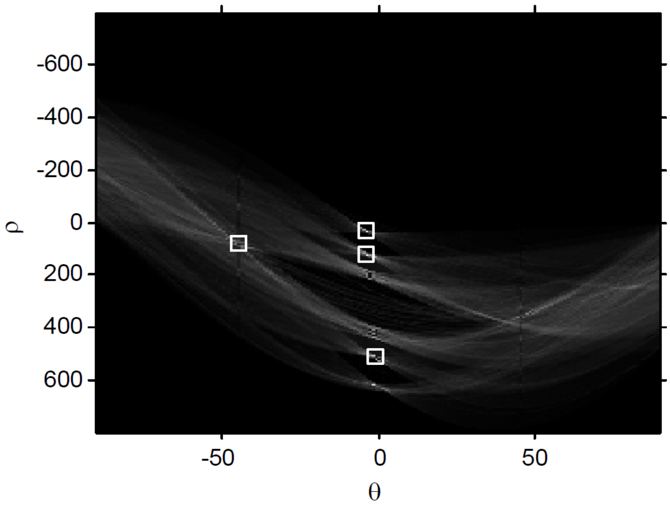


Figure 5.16: Detected lines marked with'□' on the vote histogram (left-tilted).

Table 5.3: Result of the robot positioning of the robot in different orientation

Figure	The angles of the detected lines and the robot orientation	orientation
5.14	[86, 91, 90, 87]	right-tilted
5.16)	[86, 45, 86, 89]	left-tilted
5.18	[84, 85, 92, 90]	down-tilted
5.12	[90, 89, 90, 89]	well positioned

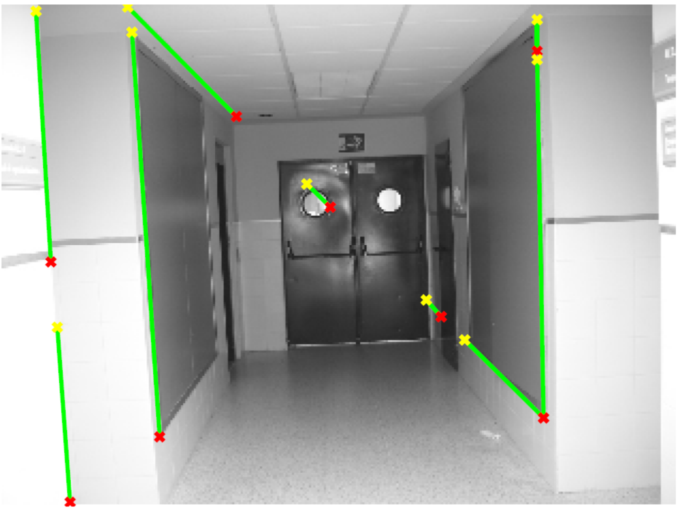


Figure 5.17: Line segments in the left-tilted robot.

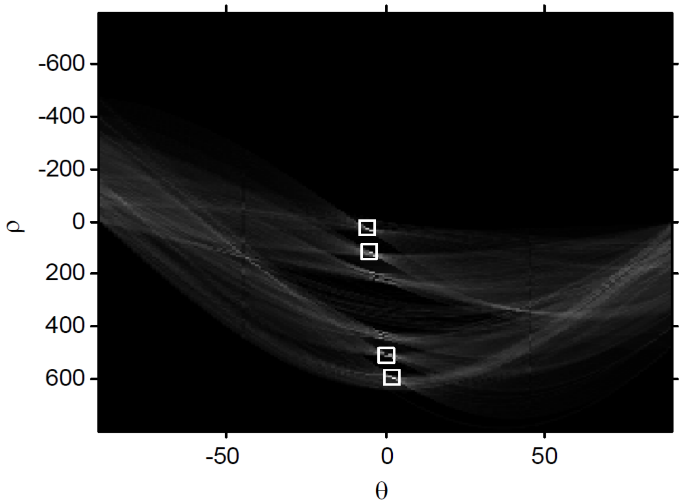


Figure 5.18: Detected lines marked with '⊡' on the vote histogram (down-tilted).

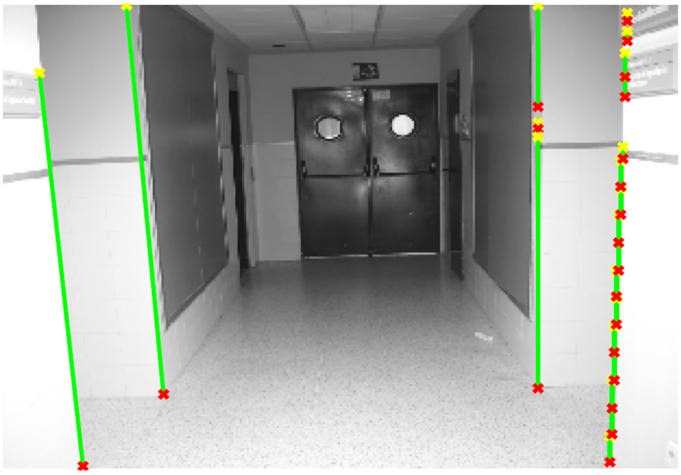


Figure 5.19: Line segments in the down-tilted robot.

Example 3 *The well-positioned robot (not tilted robot) will be utilized for a message delivery mission (see Fig. 3.11). In this example the robot should search a door and then detect a doorplate. The doorplate image will be applied to the OCR program for character recognition. The robot must also be able to recognize, which assures a correct delivery (OCR algorithm must apply).The results of these process are shown in Fig. 5.21. This process is repeated continuously until the robot find the correct address.*

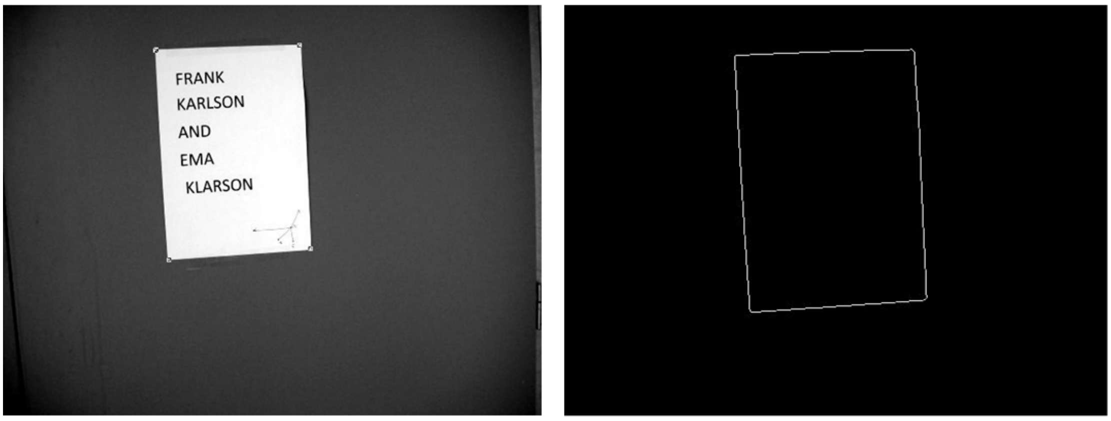
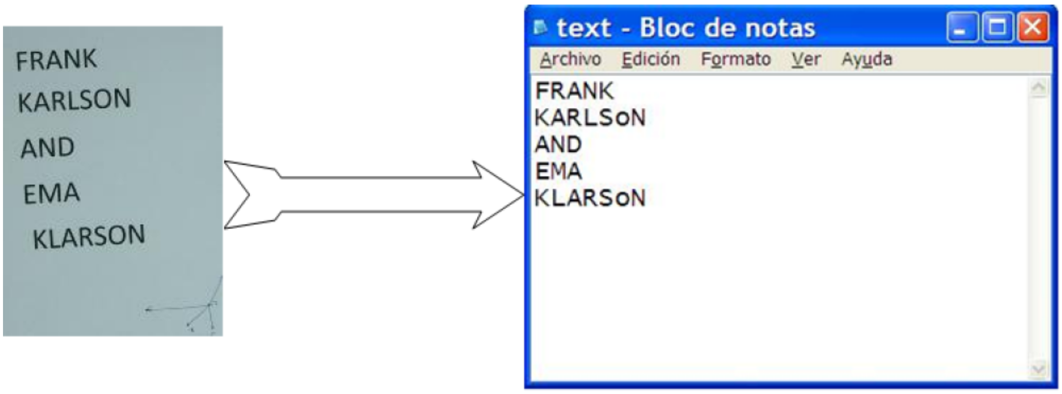


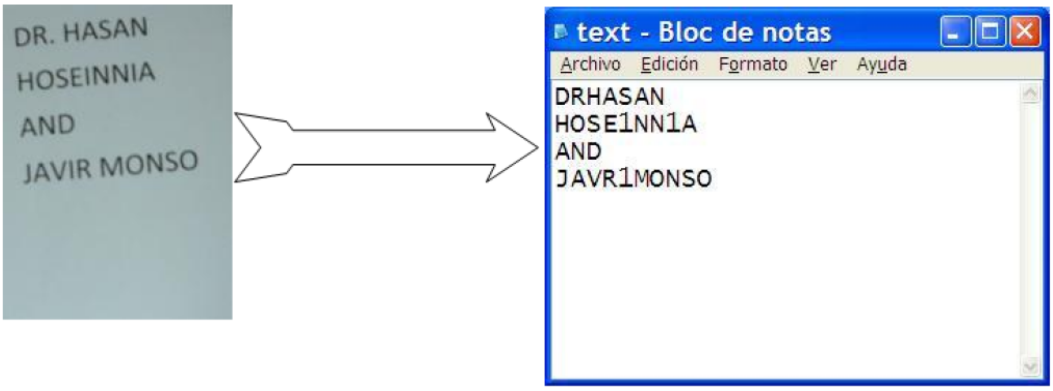
Figure 5.20: Corner detection (left), edge map of the detected corner (right).

The OCR program used in this thesis is based on the template matching [23]. It only has ability to recognize the upper case. Moreover, the space and dot (.) between two words do not recognize, also it has a problem to distinguish between (I and 1), and non-English characters such as Spanish ñ (see Figs. 5.22, and 5.23). But if these issues are considered this algorithm can work satisfactory (see Figure 5.21).



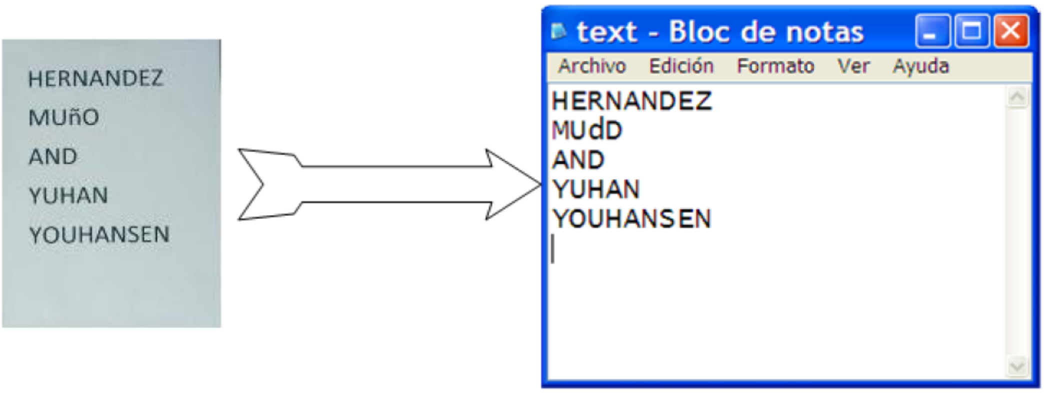
```
FRANK
KARLSoN
AND
EMA
KLARSoN
```

Figure 5.21: Result of using OCR program on text part of the image.



```
DRHASAN
HOSE1NN1A
AND
JAVR1MONSO
```

Figure 5.22: Result of using OCR program on text part of the image.



```
HERNANDEZ
MUdD
AND
YUHAN
YOUHANS EN
```

Figure 5.23: Result of using OCR program on text part of the image.

Example 4 *The purpose of this experiment is to classify the objects which are placed on the robot operation environment based on the texture analysis and using classification method (kNN classifier). In the robotic application the designed classifier should have the ability to distinguish between, the floor and the ceiling images of the corridor (see Fig. 5.24).*

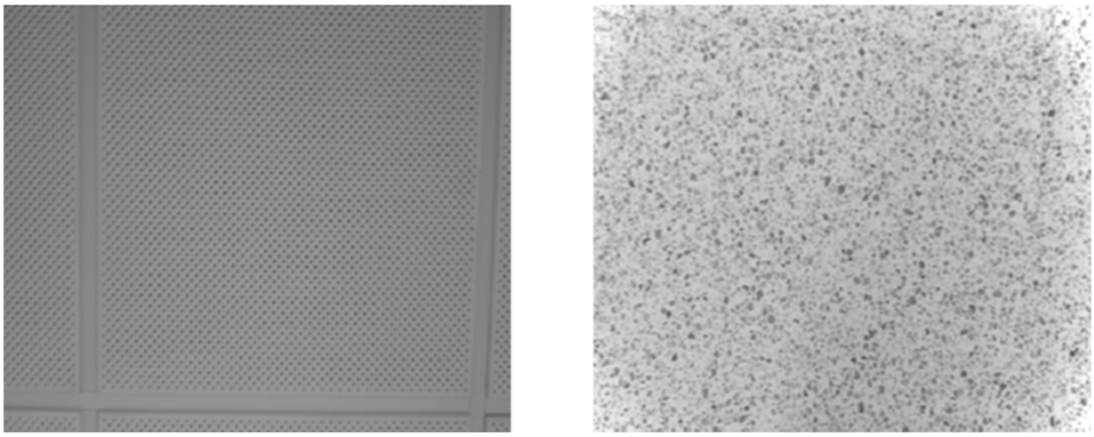


Figure 5.24: Texture image ceiling (left), floor (right).

The texture analysis and classification consist of following steps:

Step1:Image histogram equalization and image quantization: Normally, qualities of images which are captured by camera are affected by variation in lightening (illumination), and the lens distortion. Consequently, the images lose some of their contrast and efficient information. In order to overcome this problem it is effective to perform histogram equalization which can improve areas of lower contrast to gain a higher contrast. Therefore, usable data can be extracted from the images. The input images are originally represented with 8 bits (256 gray levels). This will result in huge GLCM (i.e., 256x256) in order to avoid this problem Haralick [9] suggested that images can be quantized into 8 gray level ($N_g = 8$). The result of these process is shown in Fig. 5.25.

Step2: Feature Extraction (textural features): In the second step, feature vectors are extracted from the quantized image. Originally, Haralick [9] suggested a set of 14 textural features (see **Appendix I**). Here we select 4 of them (i.e., angular second moment, contrast, correlation, and entropy) and average of them in four different directions (0,45,90,135), with a unit distance. In total, feature vector will consist of twenty features (see Table 5.4). One image gives us only one instance of the feature vector. To obtain a better representation of texture, the image could be divided into a set of smaller parts and calculate the features from them. In this thesis, images are divided into sub-images (blocks) of 32x32 pixels. It is also desirable to normalize the feature values to the same mean and the same variance. In this way we make the features equally important (see Fig. 5.26).

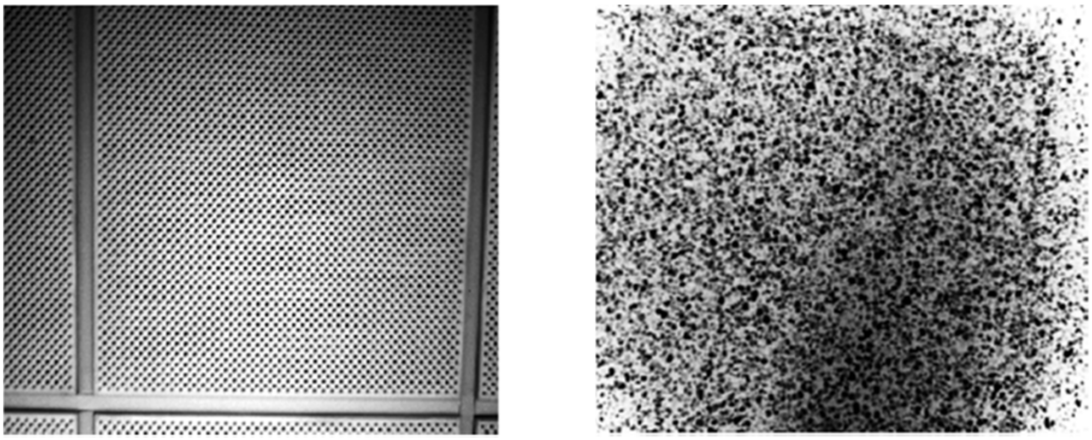


Figure 5.25: Histogram equalized and quantized to eight gray levels ($N_g = 8$) images of ceiling (left), floor (right).

Table 5.4: Texture features used in this experiment (asm=angular second momentum, con=contrast, cor=correlation, ent=entropy, see Appendix I).

Feature number	**Feature name (distance d=1)**
1	asm 0 deg
2	asm 45 deg
3	asm 90 deg
4	asm 135 deg
5	con 0 deg
6	con 45 deg
7	con 90 deg
8	con 135 deg
9	cor 0 deg
10	cor 45 deg
11	cor 90 deg
12	cor 135 deg
13	ent 0 deg
14	ent 45 deg
15	ent 90 deg
16	ent 135 deg
17	asm average
18	con average
19	cor average
20	ent average

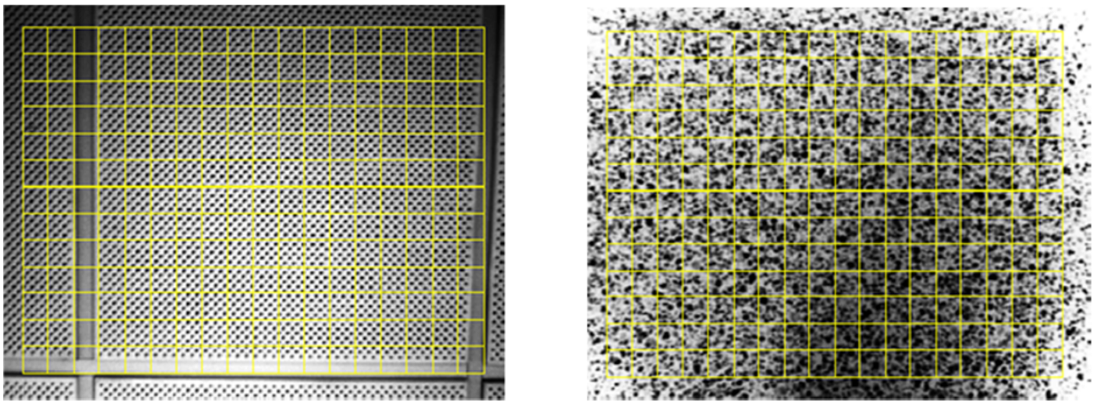


Figure 5.26: Training images ceiling (left), floor (right). The grid square represents 32x32 sub-image taken as training samples.

Step3: Feature Selection:

Feature selection is a process of finding the best feature subset from the fixed set of the original features. Usually more features leads to better performance. However, irrelevant features may result in performance degradation thus we need to select an optimal set of features [36]. To determine the best feature subset one needs to examine all possible subsets of size p. To choose the best subset of size p from set of d thus requires examination:

$$\begin{pmatrix} d \\ p \end{pmatrix} = \frac{d!}{p!(d-p)!} \tag{5.1}$$

In this thesis some efficient visual tools are used (i.e., Non-parametric density function using 1-D plot, which is used to show overlapping between samples for the univariate case (just one feature), and 2-D scatter plot, which is a chart for displaying values for two variables (feature pairs)) in order to investigate the distribution of the overlapping region between classes. Apart from visual investigation, it is possible to use numerical values using Bhattacharyya distance (BD) [37] in order to select the features. This method measures the separability of the classes. For multivariate Gaussian distributions the Bhattacharyya distance (BD) is expressed :

$$BD = \frac{1}{8}(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{S}^{-1}(\mathbf{m}_1 - \mathbf{m}_2) + \frac{1}{2}\ln\{\frac{\det(\mathbf{S})}{\sqrt{(\det(\mathbf{S}_1)\det(\mathbf{S}_2)}}\} \tag{5.2}$$

$$\mathbf{S} = \frac{\mathbf{S}_1 + \mathbf{S}_2}{2} \tag{5.3}$$

where $\mathbf{m}_i$ is the class mean matrix and $\mathbf{S}_i$ is the class covariance matrix.

- **Numerical values using Bhattacharyya distance for evaluating the 1-D and 2-D features**: In this step, BD of the single features (1-D) and feature pairs (2-D) are calculated using (5.2). And, their results are summarized in Tables (5.5-5.6). The results of Bhattacharrya distance are also presented as diagrams in Figs. 5.27-5.28.

Table 5.5: Numerical values for Bhattacharyya distance for the univariate distributions (1-D features).

number of the feature	**name of the feature**	**BD**
1	asm0	0.21
2	asm45	0.25
3	asm90	0.23
4	asm135	0.02
5	con0	0.34
6	con45	0.14
7	con90	0.25
8	con135	0.42
9	cor0	2.1
10	cor45	0.29
11	cor90	0.79
12	cor135	2.8
13	ent0	0.12
14	ent45	0.02
15	ent90	0.14
16	ent135	0.11
17	asm average	0.22
18	con average	0.29
19	cor average	2.2
20	ent average	0.14

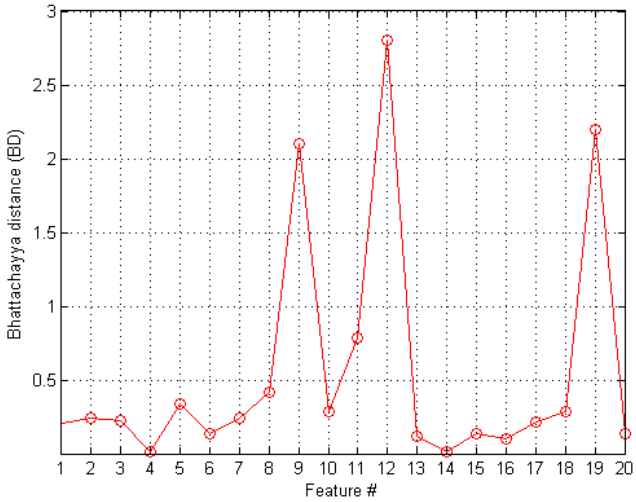


Figure 5.27: The Bhattacharyya distance (BD) for univariate distributions (1-D features). The highest BD is for feature # 9 (cor0), feature# 12 (cor135), and feature# 19 (cor average) .

Table 5.6: Numerical values for Bhattacharyya distance for the 2-D features.

Feature pairs number	number of the features	name of the feature	BD
1	(9,4)	(cor0,asm135)	27
2	(17,4)	(asm average,asm135)	0.16
3	(11,16)	(cor90,ent135)	0.78
4	(9,10)	(cor0,cor45)	2
5	(9,12)	(cor0,cor135)	8.2
6	(12,19)	(cor135,cor average)	0.37
7	(9,19)	(cor0,cor average)	5.7
8	(12,16)	(cor135,ent135)	8.2
9	(12,1)	(cor135,asm0)	13
10	(13,18)	(ent0,con average)	0.5
11	(9,20)	(cor0,ent average)	7.4
12	(19,5)	(cor average,con0)	1.8
13	(14,19)	(ent45,cor average)	8
14	(7,10)	(con90, cor45)	1.2
15	(9,12)	(cor0,cor135)	8.1
16	(3,7)	(asm90,con90)	2.3
17	(15,9)	(ent90, cor0)	21
18	(10,12)	(cor45,cor135)	3.1
19	(14,8)	(ent45,con135)	0.6
20	(4,2)	(asm0,asm135)	0.25

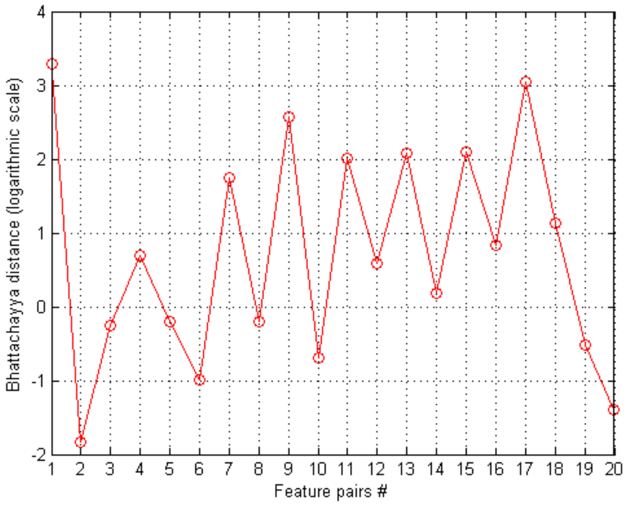


Figure 5.28: The Bhattacharyya distance (BD) for bivariate distributions (2-D features). The largest BD values are for feature pairs# 1 (cor0, asm135), feature pairs # 17 (ent90, cor0), feature pairs # 8 (cor135, ent135), and feature pairs# 3 (cor90, ent135).

- **1-D density plots, and 2-D scattersplot**: In Fig.5.29, and Fig.5.30 we present 1-D density plot and 2-D scatterplot, with the highest BD values. The visual judgment of overlapping regions of the classes shows that, there is a small overlapping between the classes. Therefore, the features with large BD values usually cause well separation between classes.

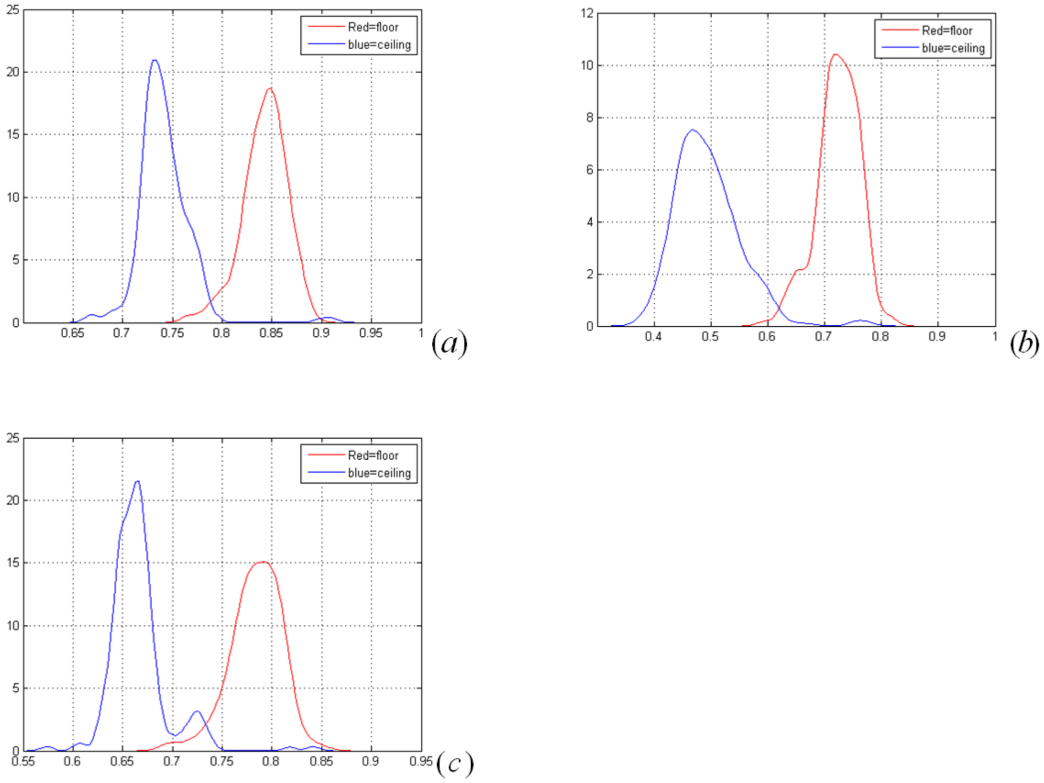


Figure 5.29: Non-parametric density function estimation for the best univariate features. Cor0 (a), Cor135 (b), Cor average (c) .

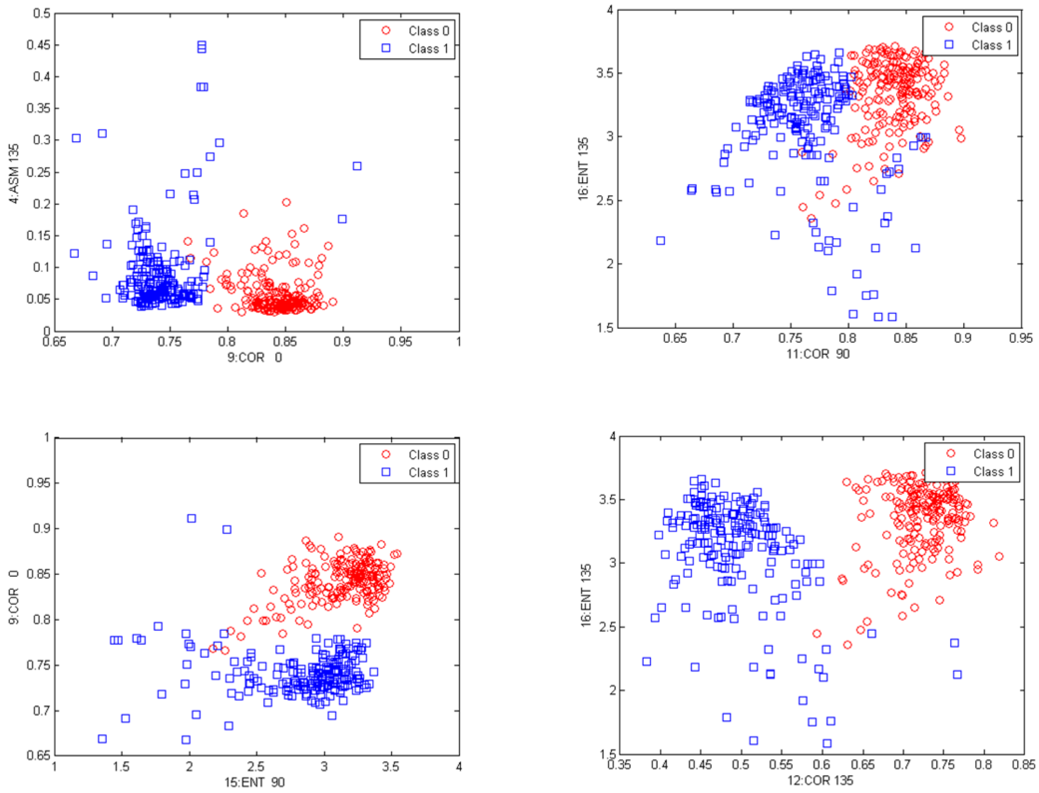


Figure 5.30: 2-D scatter plots of features having high discriminatory power (high BD) between ceiling and floor (class1==floor sample,class0==ceiling sample).

Table 5.7: LOO error rates of selected 1-D features, using kNN classifier (k=1, number of samples N=234).

feature number	name of feature	error rates
9	cor0	0.06
12	cor135	0.03
19	cor average	0.06

Table 5.8: LOO error rates of selected 2-D features, using kNN classifier (k=1, number of samples N=234)

feature pairs number	name of feature	error rate
(9,4)	(cor0, asm135)	0.04
(15,9)	(ent90, cor0)	0.04
(12,16)	(cor135, ent135)	0.03
(11,16)	(cor90, ent135)	0.04

Step5: Classifier 's performance evaluation.

The best subset of the 1-D and 2-D features, which are selected in the feature selection using BD are used in KNN classifier in order to classify an unknown samples, and evaluate the performance of the classifier by calculating the error rates (4.10). Here we use leave-one-out (LOO) error rate estimation method.

According to Tables (5.7-5.8) error rate for 1-D and 2-D, using most discriminative features from the feature selection part are efficient for classification problem, in order to make decision between ceiling and floor. In this experiment we obtained 97% classification accuracy, which maybe not sufficient in the robot applications. Moreover, this performance may be optimistically biased because we did not use an independent test data set for evaluation.

6 Discussion

In this thesis we used the Bouguet toolbox for camera calibration purpose. The extracted camera parameters were used for reconstruction of the test object from the image plane. The estimated size of the object was in agreement with the real size, which means that the calibration part works satisfactory.

In the robot positioning task, we have investigated orientation of the extracted lines and the real orientation of the robot (i.e., how it is tilted). We could use orientation of the lines to infer the tilt of the robot. However, in some cases the Hough transform could detect some false lines and these lines should be excluded when making decision about the robot movement. In Canny edge detection step we used some predefined thresholds. To avoid many unwanted edges we manually selected different thresholds. However, this would be preferable to make this step automatically.

In the task of message delivery we use the corner detection for finding the door plates. The result of corner detection using Harris operator can be affected by illumination and it can lead to the false positive results. Therefore, in order to use this algorithm we should avoid saturated images and detect possible reflections. In this thesis we have used the external OCR program which made some errors in distinguishing non-English characters which means OCR part should be improved.

The classification problem we have studied consisted of discrimination of relatively simple objects like ceiling and floor. We obtained high classification accuracy (97 %). However, this maybe not sufficient in the real robotic applications. To improve the accuracy we may introduce invariant features with respect to scale and rotation. Many of the images (captured by the robot) are distorted because of the perspective projection. In such a case it will be beneficial to perform the perspective distortion correction before the step of image feature extraction. The classifier should be tested test on the independent test set in order to get less biased estimator of the accuracy. Moreover, we need system that can recognize different objects (e.g., doors, lightening, tables, etc). We have presented different algorithms for the robotic applications. These algorithms solve only a part of the problems encountered in robot vision and robot navigation. The developed algorithms are planned to be used in larger systems using mobile robot applications

Bibliography

[1] Brezak M, Petrovic I, Ivanjko E, (2008), *Robust and accurate global vision system for real time tracking of multiple mobile robots,* Robotics and Autonomous Systems, Vol. 56, No. 3, pp. 213-230.

[2] Kim M.Y, Cho H, (2006), *An active trinocular vision system of sensing indoor navigation environment for mobile robots*, Sensors and Actuators A, Physical, Vol. 125, No. 2, pp. 192-209

[3] Hu S, Sun D, (2011), *Automated Transportation of Single Cells Using Robot -Tweezer Manipulation System,* Journal of Laboratory Automation, Vol. 16, NO. 4, pp. 263-270.

[4] Kwartowitz DM, Herrell SD, Galloway RL, (2007),*Toward image-guided robotic surgery: Determining intrinsic accuracy of the da Vinci robot*, International Journal of computer assisted radiology and surgery, Vol. 1, No. 3, pp. 157-165.

[5] Ueta T, Yamaguchi Y, Shirakawa Y, Nakano T, Ideta R, Noda Y, Morita A, Mochizuki R, Sugita N, Mitsuishi M, Tamaki Y, (2009), *Robot-Assisted Vitreoretinal surgery: Development of a Prototype and Feasibility Studies in an Animal Model*, Ophthamology, Vol. 116, No. 8, pp. 1538-1543, 1543.e1-e2.

[6] Burgard W, Cremers A. B, Hhnel D, Lakemeyer G, Schulz D, Steiner W, Thrun S.,(1999), *Experiences with an interactive museum tour-guide robot*, Journal of Artificial Intelligence, Vol. 114, No. 1-2, pp. 3–55.

[7] Kosmopoulos. D.I, Varvarigou, T.A, Emiris, D. M., Kostas A. A., (2002), *MD-SIR: a methodology for developing sensor-guided industry robot,* Robotics and Computer-Integrated Manufacturing, Vol. 18, No. 5-6, pp. 403-419.

[8] Edinbarough I, Balderas R, Bose S, (2005), *A vision and robot based on-line inspection monitoring system for electronic manufacturing*, Computers in Industry, Vol. 56, No. 8-9, pp. 986-996.

[9] Haralick R.M, Shanmugam K, and Dinstein I, (1973), *Textural Features for Image Classification*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 3, No. 6. pp.610-621.

[10] Quinquis A.,(2008), *Digital Signal Processing Using Matlab,* 1 edition, Wiley-ISTE Publication, ISBN:9781848210110.

[11] Florczyk S, (2005), *Robot Vision: Video-based Indoor Exploration with Autonomous and Mobile Robots,* Wiley-VCH publication, ISBN: 3-527-40544-5.

[12] Tsai R, (1987), *A versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*, IEEE Journal of Robotics and Automation, Vol. 3, No. 4, pp. 323-345.

[13] Hartley R, Zisserman A,(2004), *Multiple View Geometry in Computer Vision*, 2^{nd} edition, Cambridge University Press, ISBN: 0521540518.

[14] Heikkilä J, Silvén O, (1997), *A four-step camera calibration procedure with implicit image correction*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Washington DC, USA, pp.1106-1112.

[15] Melen T, (1994), *Geometrical Modelling and Calibration of Video Cameras for Underwater Navigation*, doctoral dissertation, Norwegian Univ. of Science and Technology, Trondheim, Norway.

[16] Abdel-Aziz Y. I, Karara H. M, (1971), *Direct linear transformation into object space coordinates in close-range photogrammetry*, Proc. Symposium on Close-Range Photogrammetry, Urbana, Illinois, p. 1-18.

[17] Faugeras O. D, Toscani G, (1987), *Camera calibration for 3D computer vision*, International Workshop on Machine Vision and Machine Intelligence, Silken, Japan, pp. 240-247.

[18] Weng J, Cohen P, Herniou M,(1992), *Camera Calibration with distortion Models and Accuracy Evaluation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.14, No. 10, pp.965-980.

[19] Bouguet J, *Camera calibration toolbox for Matlab,* (http://www.vision.caltech.edu/bouguetj/calib_doc).

[20] Fujifilm company, *finepix j10*, (www.fujifilm.com).

[21] Zhang Z.Y., (1999), *Flexible Camera Calibration by Viewing a Plane from Unknown Orientations*, The Proceeding of the Seventh IEEE International Conference on Computer Vision, pp. 666-673.

[22] MoviRobotics company, *mBase-MR5robot*, (www.movirobotics .com).

[23] Srihari SN, Shekhawat A, Lam SW, (2003), *Optical character recognition (OCR)*, John Wiley & Sons Ltd. Chichester, UK, ISBN:0-470-86412-5.

[24] Canny J.(1986), *A computational approach to edge detection Pattern Analysis and Machine Intelligence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, No. 6, pp. 679-698.

[25] Mai F, Hung Y, Zhong H, Sze W.,(2008), *A hierarchical approach for fast and robust ellipse extraction Pattern Recognition*, Journal of Pattern Recognition, Vol. 41, No. 8.

[26] Hough P.V.C, (1962), *A Method and means for recognition complex patterns*, US pattents, No. 3,069,654.

[27] Sonka M, Hlavac V, Boyle R, (2007), *Image Processing, Analysis, and Machine Vision*, 3rd edition,Publisher Cengage-Engineering, ISBN: 049508252X.

[28] Kitchen L, Rosenfeld M, (1982), *Gray level corner detection* Pattern recognition, Pattern Recognition Letters, vol.1, No.2, pp. 95-102.

[29] Moravec H.P, (1977), *Towards automatic visual obstacle avoidance*, Proceedings of the 5th international joint conference on Artificial intelligence, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, vol.2.

[30] Schmid C, Mohr R, Bauckhage C, (2000), *Evaluation of interest point detectors*, International Journal of Computer Vision, Vol. 37, No.2, pp.151-172

[31] Harris C, Stephens M, (1988), *A combined corner and edge detector*, In Proceedings of the 4th Alvey Vision Conference. pp. 147-151.

[32] Darling E. M, Joseph R. D, (1968), *Pattern recognition from satellite altitudes*, IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 1, pp. 38-47.

[33] Kaizer H, (1955), *A quantification of textures on aerial photographs*, Boston University. Res. Lab., Boston, Mass, Tech. Note 121, AD 69484.

[34] Brodatz P., (1966), *Textures A Photographic Album for Artists and Designers*, New York, Dover Publications, ISBN: 0486216691.

[35] Smith L.I, (2002), *A Tutorial on Principal Component Analysis*, http://www.cs.otago.ac.nz.

[36] Chodorowski A, (2010), *Laboratory of texture analysis and classification*, Image analysis course,Chalmers University of Technology.

[37] Fukunaga K, (1990), *Introduction to Statistical Pattern Recognition*,CA, Academic press, ISBN: 0122698517.

AppendixI: Haralick ’s texture features

Feature equation	Description
$f_1 = \sum_i \sum_j (i,j)p(i,j)$	Autocorrelation
$f_2 = \sum_{k=0}^{N_g-1} k^2 p_{x-y}$	Contrast:(con) A difference moment and measures the contrast or the local variation present in an image.
$f_3 = \sum_i \sum_j p(i,j)^2$	Angular second moment (asm)
$f_4 = -\sum_i \sum_j p(i,j) \log(p(i,j))$	Entropy (ent)
$f_5 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p^2(i,j)$	homogeneity: A measure homogeneity of the image
$f_6 = \underset{i,j}{MAX} p(i,j)$	Maximum Probability
$f_7 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-\mu)^2 p(i,j)$	Sum of Square : Variance
$f_8 = \sum_{i=2}^{2N_g} i p_{x+y}(i)$	Sum of Average
$f_9 = \sum_{i=2}^{2N_g} (i-f_8)^2 p_{x+y}(i)$	Sum of Variance
$f_{10} = -\sum_{i=2}^{2N_g} p_{x+y}(i) \log(p_{x+y}(i))$	Sum of Entropy
$f_{11} = \mathit{variance}$ of p_{x-y}	Difference variance
$f_{12} = -\sum_{i=2}^{2N_g} p_{x-y}(i) \log(p_{x-y}(i))$	Difference Entropy
$f_{13} = \frac{HXY-HXY1}{\max(HX,HY)}$	Information Measure of Correlation1
$f_{14} = \sqrt{1-\exp(-2(HXY2-HXY))}$	Information Measure of Correlation2

Feature notation	Description
N_g	Number of distinct gray levels in the quantized image.
$R = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)$	Number of neighboring resolution cell pairs.
$p(i,j) = \frac{P(i,j)}{R}$	$(i,j)th$ entry in a normalized gray-tone cooccurrence matrix .
$p_x(i) = \sum_{j=1}^{N_g} p(i,j)$	ith entry in the marginal-probability matrix obtained by summing the rows of $p(i,j)$.
$p_y(j) = \sum_{i=1}^{N_g} p(i,j)$	jth entry in the marginal-probability matrix obtained by summing the rows of $p(i,j)$.
$p_{x+y} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)$	$i+j=k$ and $k = 2,3,...,2N_g$
$p_{x-y} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)$	$\|i+j\| = k$ and $k = 0,1,...,N_g - 1$
$HXY1 = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log(p_x(i)p_y(j))$	Parameter of IMCORR1
$HXY2 = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_x(i)p_y(j)\log(p_x(i)p_y(i))$	Parameter of IMCORR2
$\mu_x = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} ip(i,j)$	Mean value of the $p_x(i)$
$\mu_y = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} jp(i,j)$	Mean value of the $p_y(j)$
$\sigma_x = \left\{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} i^2 p(i,j) - (\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} ip(i,j))^2\right\}^{0.5}$	Standard deviation of the $p_x(i)$.
$\sigma_y = \left\{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} j^2 p(i,j) - (\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} jp(i,j))^2\right\}^{0.5}$	Standard deviation of the $p_y(j)$.