Optical Navigation by recognition of reference labels using 3D calibration of camera.

Qaiser Anwar
Abstract

In this thesis a machine vision based indoor navigation system is presented. This is achieved by using rotationally independent optimized color reference labels and a geometrical camera calibration model which determines a set of camera parameters. All reference labels carry one byte of information (0 to 255), which can be designed for different values. An algorithm in Matlab has been developed so that a machine vision system for N number of symbols can recognize the symbols at different orientations. A camera calibration model describes the mapping between the 3-D world coordinates and the 2-D image coordinates. The reconstruction system uses the direct linear transform (DLT) method with a set of control reference labels in relation to the camera calibration. The least-squares adjustment method has been developed to calculate the parameters of the machine vision system. In these experiments it has been demonstrated that the pose of the camera can be calculated, with a relatively high precision, by using the least-squares estimation.

Keywords: Least square estimation, Matlab, Pose, DLT, Optimized color symbols.
Acknowledgements

I would like to give thanks to my supervisor Benny Thörnberg for his encouragement and support throughout my thesis which has made it possible. I would also give thanks to Farooq Shah, Abdul Waheed Malik and Prasanna Kumar for helping me in my thesis. Finally I express my thanks to my family and friends for supporting me in my studies.
# Table of Contents

**Optical Navigation by recognition of reference labels using 3D calibration of camera.** ......................................................... i  
**Author:** Qaiser Anwar ........................................................................... i  

**Abstract** ................................................................................................. ii  

**Acknowledgements** ................................................................................ iii  

**Table of Contents** .................................................................................. iv  

**Terminology** .......................................................................................... vi  

**Acronyms** .............................................................................................. vii  

1 **Introduction** ......................................................................................... 1  
1.1 Background .......................................................................................... 1  
1.2 Overall aim .......................................................................................... 1  
1.3 Verifiable goals .................................................................................... 3  
1.4 Contribution ......................................................................................... 3  
1.5 Outline .................................................................................................. 3  

2 **Related Work** ..................................................................................... 5  

3 **Theory** .................................................................................................. 9  
3.1 HSI color model ................................................................................... 9  
3.2 HSI to RGB color conversion ............................................................. 10  
3.3 RGB to YCbCr color conversion .......................................................... 11  
3.4 Imaging Geometry .............................................................................. 11  
3.4.1 Perspective projection ................................................................. 12  
3.4.2 Translation ................................................................................... 14  
3.4.3 Rotation ....................................................................................... 15  
3.4.4 Camera calibration ................................................................. 16  

4 **Methodology** ...................................................................................... 17  
4.1 Method ................................................................................................. 17  
4.1.1 Machine vision system ............................................................... 18  
4.1.2 Environment Calibration ............................................................ 18  
4.1.3 Matlab ....................................................................................... 19  
4.2 Pose Measurement .......................................................................... 19
# Table of Contents

## 5 Design

5.1 VGA Camera ..................................................22
5.2 Field of view ..................................................22
5.3 Operating system .............................................23
5.4 Reference labels (Symbols) .................................23
5.5 Encoding .........................................................23
5.6 Segmentation ....................................................24
5.6.1 Mean value Filter ............................................25
5.6.2 Background subtraction and thresholding .............26
5.7 Decoding .........................................................27
5.7.1 Area of interest (AOI) .....................................27
5.7.2 Symbols Components .....................................28
5.7.3 Detection of reference circles ...........................29
5.7.4 Detection of bit circles ....................................29
5.8 Linear estimation of parameters ............................32
5.8.1 From pixel coordinates to image plane coordinates ..35
5.8.2 Least square solution ......................................36

## 6 Result ..............................................................39

## 7 Analysis ...........................................................47

7.1 Light intensity ...................................................47
7.2 Table of reference label ......................................47
7.3 Image Resizing ................................................48
7.4 Error calculation ...............................................48

## 8 Conclusion ........................................................49

References ................................................................50
Terminology

Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pose</td>
<td>Position and Orientation</td>
</tr>
<tr>
<td>AOI</td>
<td>Area of interest</td>
</tr>
<tr>
<td>DLT</td>
<td>Direct linear transformation</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green and blue</td>
</tr>
<tr>
<td>Cr</td>
<td>Red component of YCbCr color space</td>
</tr>
<tr>
<td>DOF</td>
<td>Dimensions of freedom</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background

Navigation is a very ancient art which has become a complex science with the passage of time. It deals with trajectory determination and guidance in relation to moving objects. The determination of trajectory is related to the derivation of the state vector of the moving object at any given time. Normally, a state vector consists of position, velocity, and altitude. A position of the moving object, at a given time, is a set of coordinates related to a well-defined coordinate reference frame [1]. A modern form of navigation is by following maps. The navigator in this process determines the position by observing geographical features such as hills, roads valleys etc which are drawn in the maps. These features are defined on the map with respect to a reference frame. Usually, the terrain features on the map are defined with respect to the equator of the earth and the Greenwich meridian in term of latitude and longitude. Thus the navigator’s place can be determined within the given reference frame which is fundamental to the process of navigation. Another method of navigation is to observe other objects or naturally occurring phenomena in order to determine the present position. In ancient time one of the well-established techniques was to take the sightings of certain fixed stars with regards to which the navigator is then able to relate a position. In this type of navigation, the fixed star defines a reference frame in the given space, which is commonly referred to as the inertial reference frame [2].

1.2 Overall aim

The indoor navigation system involves navigation inside buildings that is able to interactively guide the navigator to his/her destination. Unfortunately, the most modern and commonly used system, a satellite-based navigation is not capable of indoor navigation because of the accuracy, which lies from 50 to 500 feet. Thus, in relation to indoor navigation, more accurate measurements with regards to the present position are required. Visitors inside large buildings such as hotels, airports, trains station, hospitals etc. Generally find it difficult to follow directions, especially for buildings with many floors. For example, the user may be looking for his/her car in a multi-storey, attempting to retrieve lug-
gage at an airport or a patient’s room at a hospital. As GPS signals are usually unavailable inside a building, indoor navigation has now become a growth research area. Many techniques are used to implement efficient and low-cost indoor navigation. The most common approaches are, typically, special-purpose sensors and infrastructure which can be used in the implementation of indoor navigation inside a building. For example, NFC indoor navigation system proposed by Turkish scientists is described below.

Figure 1.1: Users touch NFC tags as they move around a building, with each tap orienting the app and allowing it to calculate a new best route to the destination [3].

The aim of this project is to implement an indoor navigation system using reference labels that can determine the position and orientation of the system with respect to a calibrated reference frame. The image data can compute the camera’s pose (position and orientation) in three dimensions of freedom (DOF). The 3D world space is calibrated based on the identity of an optimized color reference around the machine vision system. The design of the reference label or symbol is simplified for ease of recognition.

In this proposed technique, optimized color reference labels that calibrate the environment around the optical system are used for indoor optical navigation. As these reference labels contain 1 byte of information, it is thus possible to encode these symbols with difference values, which are thus the identity of each symbol. Symbols are then printed on paper and attached to the ceiling of the corridor inside a university building. A camera is required to observe 6 planar symbols in
order to compute its pose. An algorithm in Matlab is used to decode each symbol value. A one channel color technique is used in segmentation so as to highlight and extract the optimized color symbols in a complex background. The numbers of the segmented components in the binary image are reduced based on the applied color technique. A direct linear transformation (DLT) method is used to compute the camera parameters that reflect the relationship between the object-space reference frame and the image-plane reference frame.

1.3 Verifiable goals
Six or more symbols have been used in order to compute the pose of the machine vision system, which contains up to 10 bytes information in a reference frame. In the calibration model, the position of the machine vision system is calculated in a 3-D world space along the X-axis and Y-axis. The reconstruction system also includes the calculation of the rotation of the camera along the Z-axis.
In the corridor of the university building which was being used, 42 symbols were involved in calculating the pose of the system along the corridor, which is 10 meters in the horizontal and 1 meter in the vertical direction. The precision of this system for position measurement is ± 3 cm and ± 8 degrees of variation in rotational measurement.

1.4 Contribution
There are portions of this thesis which were conducted in 15 hp project by the author and Farooq Shah. The Matlab algorithm for mean value filter in segmentation process was provided by my supervisor Benny Thörnberg. Farooq Shah also contributed to the design of the symbol, color model’s calculation and the decoding of a symbol. The author of this thesis developed the encoding and decoding algorithm of the symbols using Matlab. The implementation of the reconstruction system using direct linear transformation in indoor optical navigation was performed by the author.

1.5 Outline
Chapter 2 describes related work of indoor optical navigation in comparison to this technique.

Chapter 3 describes the theory behind the color models and camera calibration.
Chapter 4 explains the approach of solving the problems related to indoor optical positioning.

Chapter 5 deals with the design of the reference labels and the implementation of the indoor navigation.

Chapter 6 shows the results of the indoor optical navigation.

Chapter 7 describes the limitations and errors which may occur during implementation.

Chapter 8 shows conclusion of the project.
2 Related Work

Indoor Optical navigation is currently an attractive or alternative field of navigation because of its reduced number of infrastructures, low cost and large coverage area. Optical positioning has a large range of application depending on the levels of accuracy. The advancement in field of detectors (CCD sensors) and the miniaturization of laser technology have contributed to the success of different optical methods. In addition to this, the development of algorithms and the increase in computational capabilities of image processing enables the implementation of an indoor optical positioning system.

There are many methods used in optical positioning system which are available in research literature [4]. A discussion in relation to some methods and a comparison with the proposed technique will now be provided.

Kohoutek et al [5], [4] used the digital spatiosemantic interior building model City Geography Markup Language (CityGML) and a Ranging Imaging sensor to implement an indoor positioning system. In order to determine the indoor positioning of the camera, it was necessary to obtain semantically rich geospatial data. In this method, the area of the machine vision system is firstly identified from the CityGML database. Thus, by using the range imaging sensor, the structural objects inside the building can be detected and their geometric properties are compared with the database. In the next step, the position of the camera is calculated based on spatial resection and trilateration. The accuracy of this system is in terms of decimeters (dm).
Some options use a floor plan of the building by taking images from the camera phone to compute their position. Hile and Borriello [6], [4] firstly work out a rough estimate of the position of the navigator by using WLAN connectivity. The features from the image are then extracted and compared with the database to calculate the location. The accuracy of the system is in decimeters.

Another approach used by many researchers is a comparison of multiple viewed images. Ido J and Shimizu [7], [4] used the template to match the images taken at different positions by the robot. The position of the machine vision system is calculated by determining the correlation between the templates and the current image. This technique provides accuracy up to 30 cm. The disadvantage in this method is that there is a significant computational load on the system, in relation to computing the position, due to a lack of references in the images.
Indoor optical navigation can also be implemented by deploying symbols or reference labels in the environment. This technique increases automation and provides a high accuracy for the navigation system. Mark [8], [4] used a rectangular reference label to detect the pose of the AR drones. The design of the rectangular coded symbol is given below.

![Figure 2.2: Color symbols used for AR drones navigation (Mark et al. [8])](image)

The problem associated with this method is that the accuracy level is unknown and, additionally, the reference label is a mixture of colors, pattern and QR code, which places an extra load on the computational process.

Sky-Trax.Inc [9], implements a real time tracking system for a forklift inside a warehouse. QR code bars are attached to the ceiling of the warehouse along the route of the forklift. The imaging sensor takes images of these reference labels in order to compute the location of the forklift in combination with RFID reader, which identify pallet. The accuracy of the system is from one inch to one foot.

Mulloni et al. [10], [4] use bar-coded fiduciary markers to implement a low-cost indoor positioning system for off-the -shelf camera phones. The accuracy of the system lies within the half meter range.
In this thesis, the indoor optical navigation is based on the detection of optimized colors reference labels and a camera calibration model in order to calculate the pose of the machine vision system. A one channel color technique was used to extract the symbol even in a complex environment. These symbols are attached to the ceiling of the corridor inside the university building and the identity of each symbol is used to specify the location of the navigator along the corridor. The reconstruction system uses a 2-D DLT method to calculate the pose of the system in the 3D world space. The computed orientation of the camera is along the Z-axis. In some of the above methods there are computational issues in determining the pose, accuracy and other required heavy infrastructure necessary in implementing them for an indoor environment. This new method can be implemented at very low cost and with fewer infrastructures. The accuracy level of this system is in the range of 0 to 3 cm and an accurate rotation can be calculated between 0 to 8 degrees, which is high when compared with the above methods.
3 Theory

This chapter is divided into two main sections and describes methods used in the implementation of an indoor optical navigation.

1. Color Models
2. Imaging Geometry

Color models describe the calculation of the foreground color and background color used in the design of the reference labels. In the following section the calculation of the camera geometry used in the indoor optical navigation will be discussed.

3.1 HSI color model

Humans describe viewed objects by means of their hue, saturation and brightness. Where Hue describes the purity of color, saturation is the measure of the degree to which color purity is diluted by white color. Brightness is a subjective descriptor that is, in practical terms, impossible to measure. Brightness embodies the achromatic notion of intensity and is an important factor in describing color sensation [11]. Intensity is a useful descriptor in relation to monochromatic images as it is easily measurable and interpretable. An HSI model decouples intensity from the hue and saturation, which is color carrying information in the color image. Thus the HSI model is the ideal tool for developing image process algorithms based on color description, which is natural in relation to the human eye [11].

Foreground color and background color were selected that were computed by Xin Cheng [12]. The color with the lowest mean value was selected as the background color. The foreground was selected based on the highest SNR with respect to the background color. The calculated value for the foreground color is given as I=0.5, H=210, S=0.996 and the background color is given as I=0.5, H=359, S=0.52.
3.2 HSI to RGB color conversion

The foreground and background colors, which are computed in the HSI color space, are converted to the RGB color space using Matlab in order to encode the information in RGB. As the computed hue of the HSI space is between 120 and 240 degrees, the conversion formula for the background color is given as:

\[
R = (1-S)I, \quad Eq\ (3.1)
\]
\[
G = I \left(1 + \frac{(\cos(H) \cos(120) + \sin(H) \sin(120))}{\cos(180) \cos(H) + \sin(H) \sin(180)} \right) \times S, \quad Eq\ (3.2)
\]
\[
B = (I+S) \left(1 - \frac{(\cos(H) \cos(120) + \sin(H) \sin(120))}{\cos(180) \cos(H) + \sin(H) \sin(180)} \right) \times S, \quad Eq\ (3.3)
\]

In the above equations R, G and B stand for red, blue and green color. Now, for the foreground color, the Hue is 359 which lies between 240 and 360 so the conversion formula becomes

\[
R = I + \frac{(1-(\cos(H) \cos(240) + \sin(H) \sin(240)))}{(\cos(300) \cos(H) + \sin(H) \sin(300)) \times S)}, \quad Eq\ (3.4)
\]
\[
G = I \times S, \quad Eq\ (3.5)
\]
\[
B = I + \frac{(\cos(H) \ cos(240) + \sin(H) \sin(240))}{(\cos(300) \cos(H) + \sin(H) \sin(300)) \times S)}, \quad Eq\ (3.6)
\]

The results of the computed foreground and background colors in RGB space are given below using the above formulas [13].
3.3 RGB to YCbCr color conversion

The YCbCr color space is used to encode an RGB color space, allowing for the bandwidth of reduced chrominance components. Y is the intensity and Cb, Cr are the blue and red Chroma components. RGB images have a great deal of redundancy thus it is desirable to convert the RGB color to an YCbCr color space, which is less redundant. The equation to convert RGB to YCbCr space is given below.

\[ Y = 0.299R + 0.587G + 0.114B \quad Eq \ (3.7) \]
\[ Cb = 128 - 0.1687R - 0.3313G + 0.5B \quad Eq \ (3.8) \]
\[ Cr = 128 + 0.5R - 0.4187G - 0.081B \quad Eq \ (3.9) \]

The result shows that Y (intensity) has more segmented components than Cr red and thus Cr components were used in these experiments [12].

3.4 Imaging Geometry

In this section, several transformations methods used in the image processing are discussed. This includes the development of the representation of problems related to image rotation, image translation and camera calibration. In order to describe the position of an object in any space, the Cartesian coordinate system is used for mapping between the 3-D world and the 2-D image plane. In the 3-D world, the coordinate position of an object is denoted by \((X, Y, Z)\) while in images, the convention is to use a lowercase representation of \((x, y)\) to denote the position of an object in a 2-D image plane.
3.4.1 Perspective projection

Perspective transformation is an approximate representation of an image as seen by the human eye. The effect of smaller objects at a distance is ignored in an orthographic projection for accurate measurements. While the perspective projection shows the distant objects as being smaller in order to provide additional realism [14]. It projects the 3-D points onto the 2-D imaging plane along the lines that emanate from the centre of the projection. This means that size of the object projection depends on the distance from the centre of projection and it thus plays a vital role in image processing. These transformations are, in the main, nonlinear and this involves division by the coordinate values [11]. It is possible to define the perspective transformation matrix as

\[
P = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1/\lambda & 1
\end{bmatrix}
\]  
\text{Eq 3.10}

In the above matrix (\(\lambda\)) is the focal length of Camera.

Now, suppose that (X, Y, Z) are the coordinates of any point in a 3-D World and (x, y) is the projection of the point in the image plane as shown in figure 3.3. The camera coordinate system (x, y, z) has an image plane coincident with the (xy) plane and the optical axis (established by the centre of the lens) along the z axis. Thus the centre of the image plane is at the origin and the centre of the lens is at coordinates (0, 0, \(\lambda\)) [11].

![Figure 3.3: Alignment of camera coordinates system (x, y, z) with the world coordinate system (X, Y, Z).](image-url)
To determine the relationship between the projections of a 3-D \((X, Y, Z)\) point on the image plane \((x, y)\), a similar triangle technique can be applied as shown in figure 3.3. However, the desire is to follow the convenient linear matrix form used to express rotation, translation and scaling. A homogeneous coordinates system is used for the implementation of a projective transformation because it can be easily represented by a matrix. The advantage of using a homogeneous coordinate system is that points at infinity can be represented using finite coordinates. The homogenous coordinate of a point in 3-D coordinates \((X, Y, Z)\) can be represented as \((kX, kY, kZ)\) for which \(k\) is a non-zero constant. A point in the 3-D Cartesian world coordinate can be shown as:

\[
W = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}
\]

The homogeneous coordinate of the given point is given as

\[
W_h = \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} \quad \text{............... Eq 3.11}
\]

Now the product of the \((P*W_h)\) can be

\[
ch = P*W_h
\]

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1/\lambda & 1
\end{pmatrix}
\begin{pmatrix}
kX \\
kY \\
kZ \\
k
\end{pmatrix}
\]

\[
ch = \begin{pmatrix} kX \\ kY \\ kZ \\ (-kZ/\lambda)+k \end{pmatrix} \quad \text{...............Eq 3.12}
\]
The \( c \) is the equation of the camera coordinates in the homogeneous form. The homogenous coordinates in Eq (3.12) are converted to the Cartesian form by dividing the all components by last fourth component. The final form of a point after in the camera coordinates system becomes [11].

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \begin{bmatrix}
  \lambda X/\lambda Z \\
  \lambda Y/\lambda Z \\
  \lambda Z/\lambda Z
\end{bmatrix}
\]

In the above equation, \( x \) and \( y \) are the coordinates in the image \((x, y)\) plane showing the projection of a 3-D point. At the same time, \( z \) acts as a free variable, which is of no interest in terms of the camera model.

### 3.4.2 Translation

This is a general problem in which the two coordinate systems are separated from each other, but the basic objective of the calculating the image-plane coordinates of any particular world point remains the same. In figure 2.4, shows the general model of the camera in the real world coordinate system \((X, Y, Z)\) which is used to locate an object and the camera from the origin [11].

![Figure 3.4: Imaging geometry with two coordinate systems. Where \( w \) is position point in the world space denoted by \((X, Y, Z)\) and \( w_0 \) is camera position world coordinate denoted as \((X_0, Y_0, Z_0)\).](image)

If the camera is moved away from the origin of the world coordinates and by applying exactly the same sequence of steps to all the world points, it becomes possible to, once again, achieve the normal position of the camera.
A transformation matrix is used to locate the origin of the world coordinate at the centre of the camera gimbal and this is given below [11].

\[
T = \begin{bmatrix}
1 & 0 & 0 & -X_0 \\
0 & 1 & 0 & -Y_0 \\
0 & 0 & 1 & -Z_0 \\
0 & 0 & 0 & 1
\end{bmatrix} \quad \text{......Eq 3.14}
\]

### 3.4.3 Rotation

The transformations used for 3-D rotation are inherently more complex than the transformation discussed so far. The orientation of the camera has been represented by using Euler angles. The simplest form of transformation is the rotation of the camera along the z-axis. The angle is measured between the x and X axis. It is the case that under normal conditions these axes are aligned with each other. The basic diagram of the rotated camera is given below [11].

![Diagram of rotated camera](image)

Figure 3.5: The camera is rotated along the z-axis and the pan angle is between the x and X axes.

The convention followed to measure the angle of the camera is that the points are rotated in the clockwise direction. The process of measuring the angle of the camera along the z-axis can be accomplished by using a rotation transformation matrix which is given as

\[
R_\theta = \begin{bmatrix}
\cos\theta & \sin\theta & 0 & 0 \\
-S\sin\theta & \cos\theta & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \quad \text{......... Eq 3.15}
\]
3.4.4 Camera calibration

Camera calibration is the process of determining the camera poses relative to a control point in the 3-D world coordinate system from a given image. This process is also called photometric camera calibration. A camera transformation matrix is used to represent the unknown extrinsic and intrinsic parameters of the camera. Although a direct calculation of these parameters is possible, using the camera itself as a measuring device is more convenient, especially when the machine vision system is in motion. It is necessary for there to be a calibration in relation to the required point’s positions of the camera in 3-D world coordinates and the pixel values of these points in the image plane [11].

If it is assumed that A is a 3X4 matrix, which contains all the unknown camera parameters and where $A=P*R_0*T$. Now, the homogeneous coordinates system is used to find the relationship between the image plane and the world coordinate i.e. $c_h = A * W_h$ and by considering $k=1$, the equation thus becomes

$$
\begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
c_4 
\end{bmatrix} =
\begin{bmatrix}
s_{11} & s_{12} & s_{13} & s_{14} \\
s_{21} & s_{22} & s_{23} & s_{24} \\
s_{31} & s_{32} & s_{33} & s_{34} \\
s_{41} & s_{42} & s_{43} & s_{44}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
$$

In these experiments, the position of each point and the camera was measured in 3-D world coordinates in order to compare this with the calibrated results. The experimental setup used is as shown in figure 3.5 and a number of pictures of the points at various distances were taken. The picture with the best binary image was chosen for further processing. A transformation matrix of unknown coefficients is solved by using least square minimization.
4 Methodology

In this chapter the approach, problems related to the indoor optical system implementation and the achievement of the verifiable goals for the project are described.

4.1 Method

In the proposed method, an image of the reference labels in 3-D space at any position in a calibrated environment is used to calculate the pose of the machine vision system. In this case, a university corridor was used as the calibrated area. The symbols are attached to the ceiling of the corridor. Six or more reference labels were used in an image in order to compute the pose of the machine vision system. By using the position of these reference labels it is possible to calculate the pose of the camera. As reference labels have encoded information, an algorithm has been developed to decode six or more symbol values. The identity of these symbols is used as the reference in the indoor navigation. After decoding the symbols, the information is used to discover the position of specific symbols in the corridor. The camera calibration model uses the symbol’s identity and position to compute the pose at any given place within the calibrated area. A maximum of ten symbols are used in this navigation process, which means that this particular indoor navigation system can be implemented over large indoor regions. The work flow of the method is given below.
4.1.1 **Machine vision system**

In this technique for an indoor optical navigation, a machine vision system is placed on a movable vehicle in order to calculate the pose at various points in the corridor. The components of machine vision system are cost effective and are readily available. The components of the system are an Usb camera and an operating system. The operating system has an Intel Core i3 processor and a 4 GB DDR3 memory.

4.1.2 **Environment Calibration**

The 3-D space around the machine vision system is calibrated by using optimized color symbols. The symbols are assigned with different encoded values which become the identity of each symbol. The indoor optical navigation prototype is implemented in the corridor of the university building. In the reconstruction model six or more symbols are used. The choice of placing the symbols on the ceiling is because the view of the camera is rarely disturbed in the upward direction towards the ceiling. The 3-D space in the corridor is carefully calibrated. Experiments were conducted with regards to the selection of the symbol size and the distance between the symbols. Small symbols may not be detected from the ceiling and they are also more sensitive to variations in the light. Symbols with a larger size provide better results in the segmentation process. The distance between the symbols was also measured in order to provide at least six symbols being viewed at a specific distance from the roof and the path is then divided into a number of small steps in relation to the pose measurements, as shown in the figure below.

![Symbols attached to ceiling in the corridor.](image-url)
The position of the symbols is measured in the XY plane of the ceiling from the starting point to the end of the corridor. The starting point of the corridor is defined as the origin of the 3-D World coordinates system.

4.1.3 Matlab

In this thesis all algorithms were developed in the Matlab environment. MATLAB® is a high-level language and is an interactive environment for numerical computation, visualization, and programming [15]. In Matlab, it is possible to analyse the data, create a GUI based application and create models for testing purposes. Matlab has a wide range of applications including image processing, signal processing, control system and video processing. In the present modern world, a very significant number of engineers and scientist are using Matlab as the technical language in their specific fields.

In Matlab, tables are developed relating to the symbols identities and their related positions in 3-D space. In the decoding process, the identities of the symbols are matched with the contents of the table and based on their matched values, the symbol's position in the 3-D world is defined.

4.2 Pose Measurement

In this experimental method a picture of at least six symbols is taken using a µeye camera. The image is then stored in the memory of the operating system. The decoding algorithm reads the input image from its memory location and applies a segmentation process to the symbols. Based on the use of a one channel color technique, the segmented components are reduced. The values of the symbols are then decoded and compared with the reference tables to determine the position of the symbols in the corridor. The orientation of the camera is measured along the Z-axis and its position in 3-D space by relating it to the origin. The reason for calculating the orientation only along the Z-axis is to avoid the many constraints in the camera calibration model. The pose calculation is simple and has a high level of accuracy. The measurement of the position of the machine vision system is based on its distance from the origin of the 3-D world space. The path in the corridor is divided into a number of steps which are perpendicular to each symbol in 3 D space. The distance between the steps is the same as the distance between the symbols.
At first, the pose of the camera was calculated at the measured points in the corridor and the machine vision system was translated along the corridor. The angle of the camera, for all the measurements, was near to 0 degrees. The calculated values of the position of the machine vision system were compared with the position of the symbol on which the camera is focused.

The machine vision system is then moved to random points in the corridor and the pose of the system is calculated. In this experiment, the system is moved a few centimeters back and forwards along the X-axis direction from these points. The position of the machine vision system at the new position is calculated by using the symbols.

The third type of experiment was conducted in order to calculate the different orientations of the camera along the Z-axis. The camera was rotated around its own axis for various angles and from the image of the symbols the pose of the system was calculated. The accuracy and the experimental results of the system are discussed in the following chapters.
5 Design

In the proposed technique, the indoor optical navigation system consists of printed optimized color reference labels that are attached to the ceiling inside the university building, an usb camera and an operating system. The 3-D space around the camera is calibrated by means of the symbols. All the reference labels contain 1byte of information and each symbol has been given a unique identity along the corridor. The calibration of the camera is computed by observing the optimized colors labels whose geometry is measured in the corridor. The camera is required to observe 6 planar symbols in order to compute its pose. The camera and the operating system are placed on a vehicle and moved horizontally along the corridor to determine the pose at various places. A one channel color technique is used in the segmentation to highlight and extract the optimized color symbols in a complex background and based on which, the number of segmented components in the binary image is also reduced. A direct linear transformation (DLT) method is used to compute the camera parameters that reflect the relationship between the object-space reference frame and the image-plane reference frame. The implementation process was divided into different steps, each of which is explained in detail.

Figure 5.1: Experimental setup used for indoor optical navigation. Where the reference labels are attached to ceiling in office corridor.

The indoor optical navigation system consists of a machine vision system and reference labels. The design of the reference labels is discussed
in detail, but, the discussion is firstly focussed on the basic information about the machine vision system components.

5.1 VGA Camera

In this project, the machine vision system consists of the WVGA camera, which is usb, 2 interfaces and a Windows operating system. This camera is an all-rounder with a 1/2”in Aptina CMOS sensor having 3.1 mega pixel resolutions. It can accept a signal up to 30 V and the input and output ports and the flash control are opto-coupled in order to minimize the damage to the camera [16]. The image quality was good throughout the experiments.

![Figure 5.2: Showing the architecture of the ueye color camera.](image)

The 64 bit ueye software was used in the interface of the camera. The quality of the image can be easily controlled using ueye software, which also enabled the camera to be interfaced with Matlab, which can assist in the automation of the image processing.

5.2 Field of view

The field of view of the camera is defined as the area that can be seen by the camera at a given moment. The field of view of the camera depends on the camera lens or the focal length. A camera with a smaller focal length has a greater angle of view and the camera can see a large area, which is useful in covering large objects in the image. In the case of a bigger lens, there is a smaller field of view [17]. Both type of lens are important, depending on their applications.

In this case a smaller lens was used in order to increase the field of view. A 3.5 mm focal length lens was used which enabled ten symbols to be viewed in the image.
5.3 Operating system

The operating system used in the indoor navigation system is a Laptop pc. It has an Intel core i3 2.4 GHz processor and a 4 GB DDR3 memory. Windows 7 is the operating software used in the processing and the ueye 64 bit software was downloaded from the ueye website as it is compatible with Windows 7. The speed of the operating system is sufficient to calculate the pose of the machine vision system from an image having 840X 480 resolutions.

5.4 Reference labels (Symbols)

The main aim in designing the optimized color reference labels is to provide a symbol which can carry some information data. The design of the symbol should be simple so that it can be easily recognized at a distance. The symbol should also be recognized at any orientation. Therefore, it is necessary to have references in the symbol. The symbol design consists of one big circle, inside which there are two reference circles and eight small bit circles. The distance of these small bit circles varies from the centre of the symbol according to its values. When its value is 0, the small circle is closer to the centre of the symbol lying in the threshold value, while, in the case of 1, its distance from the centre is greater than the threshold distance. The reference circles are slightly bigger than the small bit circles. The symbol is rotationally independent based on these reference circles, which are used in the decoding to select the starting of the bit circle in a clockwise direction. The symbol is divided into four Cartesian coordinate quadrants. Every quadrant has two small bit circles. The angles of the bit circles vary from 0 to 360 degrees. The threshold of the minimum distance between the circles must be fixed because, in the case of a smaller distance between the circles, the boundaries of the bit circles are attached to each other, thus making them unable to recognize the symbol correctly at various distance ranges. There are many steps involved in the identity and design of the symbol and these are given below.

5.5 Encoding

Encoding is defined as placing information data into code. An algorithm has been developed to design a symbol using optimized foreground and background colors to encode information. Circles of different radii were used in the design of the symbol and there is an outer circle followed by two reference circles and small bits circles. The outer circle defines the area of the symbol. The reference circles and small bit circles are filled
with foreground color so as to detect the whole circle in the segmentation process. The threshold of the minimum distance between each bit circle is specified. The value of the small bit circle is 0 or 1 depending on the position from the centre of the symbol. The symbol can be encoded up to 8 bits of information which means that the input value ranges from 0 to 255. In Matlab, it is the user who provide the values to the encoding function and the values are assigned to the symbol as its identity, as shown below.

![Symbol encoding](image)

Figure 5.3: The symbol is encoded for decimal value of 234.

### 5.6 Segmentation

```
RGB input image
  ↓
RGB to Cr component
  ↓
Mean value filter
  ↓
Background Subtraction
  ↓
Thresholding
```
In the segmentation process, the number of the segmented components in the binary image is analysed. The encoded symbols are printed using a colored laser printer. The printed symbols are attached to the ceiling of the corridor inside the office room. The images of the different symbols were taken using a smart camera in the raw format. Raw format images have minimally processed data from the smart camera as they are not yet processed [18]. These images of the symbol were taken at different distances and at different angles with a variety of the complexity in the image. A raw image occupies a larger memory than that of a normal format image and these are then converted to a bitmap format, which stores one bit per pixel. In indoor optical navigation more than one symbol is required in order to find the pose of the camera therefore experiments have also been conducted involving images with more than one symbol. Pictures of these symbols were also taken in the indoor environment using different complexity levels in the background image. The color segmentation was analysed using Matlab. The processing time varies depending on the size of the image and the complexity of the background. The image is resized in order to reduce the resolution of the image. Then the image RGB color space is converted to the YCbCr color space. RGB images have a great deal of redundancy which is the reason for converting the RGB color to an YCbCr color space, which is less redundant. Cr is the red component of the YCbCr color space and this is used in the segmentation process because of the reduced number of segmented components in the binary image than for the other components of the YCbCr color space [11] as shown in the flow graph.

5.6.1 Mean value Filter

In order to smooth the Cr images an 11 X 11 mean value filter was implemented. This is simple and easy to use and it reduces noise in the Cr image. A mean value filter is also called an averaging filter and is where each pixel in the image is replaced with the mean value of its neighbours. In this way it is possible to eliminate pixel values that are unrepresentative of their surroundings. The image is convolved using
the 11 X 11 kernel of the mean value filter and a background is thus computed from the image [19].

5.6.2 Background subtraction and thresholding

The computed background image is then subtracted from the original image. A threshold value is applied, which is computed as a percentage of the maximum pixel values. The threshold value depends on the distance between the symbol and the smart camera. At greater distances, the threshold is minimized in order to detect small components in a symbol. In the segmentation process, the threshold value is increased or decreased up to the required level so that the bigger object remains the symbol in the binary image and the remainder of the objects are smaller in size than the symbol. Thus, there is also a threshold on the size of the reference label because in a small symbol it is difficult to extract all the components of the symbols. The result of extracting the symbol from a complex background can be seen below.

Figure 5.5: Image of reference labels in the complex environment from high distance.

Figure 5.6: Binary image of reference label after segmentation. Numbers of segmented components other than Area of interest (AOI) are few, which is due Cr component.
5.7 Decoding
The information carried by the symbol is important in indoor optical navigation because it provides a reference to the machine vision system inside the corridor of a building. The decoding process is explained in a systematic manner by the following steps.

![Graph Flow Diagram]

5.7.1 Area of interest (AOI)
Finding the symbols in the image, deals with the area of interest (AOI) in the image. The proposed algorithm defines the area of interest (AOI)
in the image. After the segmentation process, the decoding algorithm is applied to the binary image. The detection of the outer circle of the symbol determines the area of interest in the image. To find the outer circle in the image, the threshold value of the mean value filter is adjusted so that the outer circle is the biggest object in the whole image.

In some experiments, the outer circle of the symbol is not detected fully and thus an `imclose` function may be used in order to complete the circle. The size and position of every object in the image is computed using a `regionprops` function. The position of every object is determined in term of the number of pixels. The horizontal axis shows, in this case, the number of columns and the vertical axis shows the rows in the image plane. The small circles and reference circles lie inside the outer circle of the symbol. The proposed algorithm can compute N number of AOIs if there is more than one symbol in the image. The small circles in the symbol carry the one bit information depending on the distance from the centre of the symbol. After the detection of the outer circle, the threshold distance for the small bit circles is calculated. This is the midpoint of the radius of the big circle. Based on this point, the value of the small bit circle is calculated from the centre of the reference labels.

### 5.7.2 Symbols Components

In the binary image, a check is applied on the position of every object with respect to the AOI. In the algorithm there is an “ignore” objects in the image which lie outside the AOI region because no information related to that symbol or not the component of the symbol is being carried. By applying limits to the position of the objects, the number of components lying in the AOI can be obtained. At this point it is possible to confirm the correct detection of the symbol by counting the numbers of symbol components. If the symbol components in the binary image proves to be less than or greater than the real symbol, then no further process is performed on the symbol and the output is shown without decoding the symbol. For subsequent applications, the threshold value in the segmentation process is decreased or increased to a particular level in order to detect the symbol correctly.

In some experiments, objects may be attached to each other in a complex environment which occurs because of changes to the or because of the distance between the symbol and the camera. The threshold value is adjusted in order to detect the correct symbol.
5.7.3 Detection of reference circles

For rotationally independent detection of the symbol, the determination of position of the reference circles is the main factor in the decoding. It is known that one reference circle is at the centre of the symbol and that the other is at some distance from the centre of the symbol, passing the threshold distance. These circles are detected based on their position from the centre of the symbol. The position of the first reference circle is computed by comparing the centre point of the symbol with the position of the symbol components in the binary image. In this manner it is possible to detect the position of the first reference circle. Then, the position of the second reference circle is computed by comparing the size of the first reference circle with every object inside the AOI. The circle, which matches the size, is the second reference circle and becomes the starting point in the decoding.

5.7.4 Detection of bit circles

The information data of the symbol is determined by the position of the small circle in the symbol. To decode the symbol it is necessary to measure the position of these circles with respect to the AOI of the symbol. In the 2-D image plane, the position of these circles is computed by applying limits to their position with respect to the AOI of the symbol. In relation to the determination of the MSB and LSB circles, the positions of small circles are converted to the polar coordinates system. Pythagorean principles are applied in order to calculate the distance r and the angle \( \theta \).

\[
\begin{align*}
\text{r}^2 &= x^2 + y^2 \quad \text{Eq (5.1)} \\
\theta &= \tan^{-1}(y/x) \quad \text{Eq (5.2)}
\end{align*}
\]

where \( x = x_0 - x_n \), \( y = y_0 - y_n \), and \( x_0, y_0 \) is the centre of the symbol and \( x_n, y_n \) is the position of the bit circle in the symbol [20].
Applying equation 5.1 the distance of each small bit circle was calculated from the centre of the symbol. Thus the value (0 or 1) is assigned to each small bit circle depending on its distance from the centre of the symbol. In order to determine the MSB and LSB of the symbol, the angles of the small bit circles were calculated using equation 5.2 in the clockwise direction. At first, the one symbol was decoded with the same orientation while neglecting the reference circles. The small bit circle with the greatest angle is the MSB and the smallest angled bit circle is the LSB as shown in figure 5.10.

If the image has more than one symbol, then a dynamic algorithm is required so that any number of symbols can be decoded if they have the same orientation. The results of two symbols in an image are shown below.
The desire was now to add another feature to the algorithm, which is the decoded value, and which should be independent of the symbol’s orientation. Reference circles are used to make the decoding process rotationally independent. Computing both the position and angle of the reference circles plays a key role because it defines the MSB and LSB circles. After determining the position and angle of the second reference circle, it is possible to compare its angle with the bit circles and use this as the starting point by assigning it the lowest angle. Then, the bit circle which has an angle greater than the second reference circle angle is the MSB circle. In this manner, every bit circle pose is arranged with respect to the second reference circle as shown in figure 5.11.
5.8 Linear estimation of parameters

The estimation process is begun by deriving the coefficients of the calibration matrix, which contains the unknown parameters of the camera. It was assumed that the camera centre is translated from the origin of the world coordinate. The translation of the origin of the world coordinate system to the location of the camera centre is accomplished by using the transformation matrix in Eq 3.14. There was also a desire to calculate the angle of the camera along the z-axis. The transformation matrix used to compute the camera’s rotation along the z-axis between the x and X axis is given in Eq 3.15. For this purpose it is necessary to derive the unknown coefficients of the matrix. The calibration matrix can be derived as
Indoor optical Navigation
Qaiser Anwar
2013-02-11

\[
R_\theta = \begin{pmatrix}
\cos \theta & \sin \theta & 0 & 0 \\
-\sin \theta & \cos \theta & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

\[
P = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1/\lambda & 1
\end{pmatrix}
\]

\[
W_h = \begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\]

\[
T = \begin{pmatrix}
1 & 0 & 0 & -X_0 \\
0 & 1 & 0 & -Y_0 \\
0 & 0 & 1 & -Z_0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

\[
A = (P^*(R_\theta^*T)) \quad \ldots \ldots \text{Eq 5.3}
\]

Inserting the values into equation 3.16, the final results becomes

\[
A = \begin{pmatrix}
\cos \theta & \sin \theta & 0 & -X_0 \cos \theta - Y_0 \sin \theta \\
-\sin \theta & \cos \theta & 0 & X_0 \sin \theta - Y_0 \cos \theta \\
0 & 0 & 1 & -Z_0 \\
0 & 0 & -1/\lambda & (Z_0/\lambda) + 1
\end{pmatrix}
\]

After multiplying the matrixes in the correct order the above results are obtained. Thus the unknown coefficients of the transformation matrix have been derived. After inserting the values in equation 3.16, the following is obtained

\[
\begin{pmatrix}
ch_1 \\
ch_2 \\
ch_3 \\
ch_4
\end{pmatrix} = \begin{pmatrix}
\cos \theta & \sin \theta & 0 & -X_0 \cos \theta - Y_0 \sin \theta \\
-\sin \theta & \cos \theta & 0 & X_0 \sin \theta - Y_0 \cos \theta \\
0 & 0 & 1 & -Z_0 \\
0 & 0 & -1/\lambda & (Z_0/\lambda) + 1
\end{pmatrix} \begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix} \quad \text{Eq 5.4}
\]

The coefficient of the image coordinates from the homogeneous coordinates are now computed which are given as

\[
x = ch_1/ch_4, \quad y = ch_2/ch_4 \quad \ldots \ldots \text{Eq 5.5}
\]

\[
ch_1 = xch_4, \quad ch_2 = ych_4 \quad \ldots \ldots \text{Eq 5.6}
\]

Inserting the values into equation 5.6 after expanding the matrix, the following is obtained

33
where \( C_{h3} \) is ignored as it is related to the Z-axis. Substitution of the \( C_{h4} \) in equations 5.7 and 5.8, yields two equations with 12 unknowns.

\[
\begin{align*}
X_{h4} &= s_{11}X + s_{12}Y + s_{13}Z + s_{14} \quad \cdots \text{Eq 5.7} \\
Y_{h4} &= s_{21}X + s_{22}Y + s_{23}Z + s_{24} \quad \cdots \text{Eq 5.8} \\
C_{h4} &= s_{41}X + s_{42}Y + s_{43}Z + s_{44}
\end{align*}
\]

The calibration procedure for coplanar labels consists of

- Imaging world labels on a planar surface requires 4 or greater than 4 points with known coordinates \((X_i, Y_i, Z_i)\)
- Calculate the values of these labels in the image plane from the pixel coordinates.
- Shift the label's position in both the image and in the 3-D world coordinates such that the center point is centered at the origin.
- Combine equation 5.10 and 5.9 in matrix form for imaged labels. Solve the matrix for unknown coefficients.

In the majority of the reconstruction process, the calibration model is based on the direct linear transformation (DLT), which was originally developed by Abdel-Aziz and Karara [21]. In this method, a set of points is used in the calibration of the camera, whose world coordinates and image plane coordinates are known. As in the case for the pinhole camera model, the non-linear radial and tangential errors are ignored, which may occur in the imaging process. The problem dealt with by the DLT is to calculate the mapping between the 3D world space and the 2D image plane. A linear estimation of the 3X4 transformation matrix Eq 5.4 is used to determine the relationship between these coordinate systems. In the experiments, the reference labels are coplanar and are attached to the ceiling inside the building. Thus, the problem is described as a 3D camera calibration using a 2D DLT algorithm. However, the only difference between this method and a 3D DLT is the problem of dimension. In a 2D DLT algorithm,
the transformation matrix has a 3X3 dimension which is given as; \( \mathbf{ch} = \mathbf{A} \mathbf{W} \).

\[
\begin{bmatrix}
\mathbf{ch}_1 \\
\mathbf{ch}_2 \\
\mathbf{ch}_4
\end{bmatrix} = 
\begin{bmatrix}
\mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{14} \\
\mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{24} \\
\mathbf{S}_{41} & \mathbf{S}_{42} & \mathbf{S}_{44}
\end{bmatrix}
\begin{bmatrix}
\mathbf{X} \\
\mathbf{Y} \\
1
\end{bmatrix}
\]

.................... Eq 5.11

The position of each point in pixel coordinates must firstly be measured, after which they are converted into image plane coordinates.

5.8.1 From pixel coordinates to image plane coordinates

Before the calibration of the camera, the position of each reference label in the pixel coordinate was calculated. A Centroid function is used to measure the position of each point in the binary image. The principal point, which is the centre of camera, can be determined by means of the camera focus. When the camera is focussed on any point in the 3-D space then, the principal point and that point have the same pixel coordinates. Pixel size is also important in the measurement of the camera pose. Pixel size is determined by the ratio between the sensor size and the resolution of the image. In these experiments, a 1/2" in Aptina CMOS sensor with a 3.1 (2048X1536) mega pixel resolution was used. For computational purposes, the resolution of the image was decreased to 840X480. Thus the pixel size is also changed accordingly. The pixel size for the given image can be measured as:

- Pixel width \((p_x) = \) width of sensor / image width
- Pixel height \((p_y) = \) height of sensor / image height

Inserting values into the above equations

\[ p_x = \frac{6.4 \text{ mm}}{840} = 7.6 \times 10^{-4} \text{ cm} \]
\[ p_y = \frac{4.8 \text{ mm}}{480} = 10 \times 10^{-4} \text{ cm} \]

Figure 5.15: Position of point from world coordinates to camera coordinates
In order to calculate the position of a point in the image plane the following formulas were used.

\[ x = -(x_{im} - o_x) p_x \quad y = -(y_{im} - o_y) p_y \quad \ldots \quad \text{Eq 5.12} \]

\[ x_{im} = -x/p_x + o_x \quad y_{im} = -y/p_y + o_y \quad \ldots \quad \text{Eq 5.13} \]

where \((o_x, o_y)\) are the coordinates of the principal point in pixels. In the case of the principal point being in the centre of the image, \(o_x=\text{width of image}/2, o_y=\text{height of the image}/2\) (if the principal point is in the centre of the image) and \(p_x, p_y\) are the size of a pixel in the horizontal and vertical directions [21].

### 5.8.2 Least square solution

Each point in the camera calibration model corresponds to two independent equations 5.9 and 5.10. As the reference labels are attached to a planar surface, the values of the reference labels along the Z-axis are zero. Inserting zero values for Z in equations 5.9 and 5.10 the following is obtained

\[ s_{11}X + s_{12}Y + s_{14} - s_{41}X - s_{42}Y - s_{44}X = 0 \quad \ldots \quad \text{Eq 5.14} \]

\[ s_{21}X + s_{22}Y + s_{24} - s_{41}Y - s_{42}Y - s_{44}Y = 0 \quad \ldots \quad \text{Eq 5.15} \]

The above two equations are used for 5 reference labels and then they are combined into a matrix. The DLT matrix and its unknown coefficients for 5 reference labels are given as

\[
H = \begin{pmatrix}
X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -Y_1X_1 & -x_1 \\
0 & 0 & 0 & X_1 & Y_1 & 1 & -y_1Y_1 & -y_1Y_1 & -y_1 \\
X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -Y_2X_2 & -x_2 \\
0 & 0 & 0 & X_2 & Y_2 & 1 & -y_2Y_2 & -y_2Y_2 & -y_2 \\
X_3 & Y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -Y_3X_3 & -x_3 \\
0 & 0 & 0 & X_3 & Y_3 & 1 & -y_3Y_3 & -y_3Y_3 & -y_3 \\
X_4 & Y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -Y_4X_4 & -x_4 \\
0 & 0 & 0 & X_4 & Y_4 & 1 & -y_4Y_4 & -y_4Y_4 & -y_4 \\
X_5 & Y_5 & 1 & 0 & 0 & 0 & -x_5X_5 & -Y_5X_5 & -x_5 \\
0 & 0 & 0 & X_5 & Y_5 & 1 & -y_5Y_5 & -y_5Y_5 & -y_5 \\
\end{pmatrix}
\]

\[
S = \begin{pmatrix}
s_{11} \\
s_{12} \\
s_{14} \\
s_{21} \\
s_{22} \\
s_{24} \\
s_{41} \\
s_{42} \\
s_{44} \\
\end{pmatrix}
\]

In this case, where \(S\) is the vector matrix of the unknown parameters of the camera and \(H\) is the matrix of the known coefficients. The above matrices can be written in the form of a linear equation for \(N\) numbers of reference labels.
The equation (Eq 5.16) is a homogeneous linear equation having one or an infinite number of solutions. The problem associated with linear equation systems is their trivial solution, which is not desirable in relation to normal solutions. A non-linear constraint is applied to the length of the coefficients of the vector matrix S in order to avoid the trivial solution. Abdel aziz and Karara applied a constraint on $s_{44} = 1$ [22]. This constraint adds one more row to matrix H and matrix S making the equation system over-determined. More labels can be included to further expand matrix H with two rows per label, see equations 5.14 and 5.15. There is no solution for this over-determined equation system but an approximate solution can be estimated by using the least squares method. In many cases, singularity is introduced even with the constraint on $s_{44}$, because $s_{44}$ can have values close to zero. Faugeras and Toscani applied a constraint on $s_{241} + s_{242} + s_{243} = 1$ for a singularity free solution [23].

In the calibration model the unknown parameters of the column vector matrix S were estimated by applying a constraint to the homogeneous equation. For a singularity free solution, the homogeneous linear equation Eq 5.16 is converted to a non-homogeneous form as shown below.

$$Y = H_1^* S_l \quad \text{Eq 5.17}$$
where the non-homogeneous equation is defined as

\[
\begin{bmatrix}
    x_1 \\
    y_1 \\
    x_2 \\
    y_2 \\
    x_3 \\
    y_3 \\
    x_4 \\
    y_4 \\
    x_5 \\
    y_5 \\
\end{bmatrix} =
\begin{bmatrix}
    X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1 X_1 & -Y_1 X_1 \\
    0 & 0 & 0 & X_1 & Y_1 & 1 & -y_1 Y_1 & -y_1 Y_1 \\
    X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2 X_2 & -Y_2 X_2 \\
    0 & 0 & 0 & X_2 & Y_2 & 1 & -y_2 Y_2 & -y_2 Y_2 \\
    X_3 & Y_3 & 1 & 0 & 0 & 0 & -x_3 X_3 & -Y_3 X_3 \\
    0 & 0 & 0 & X_3 & Y_3 & 1 & -y_3 Y_3 & -y_3 Y_3 \\
    X_4 & Y_4 & 1 & 0 & 0 & 0 & -x_4 X_4 & -Y_4 X_4 \\
    0 & 0 & 0 & X_4 & Y_4 & 1 & -y_4 Y_4 & -y_4 Y_4 \\
    X_5 & Y_5 & 1 & 0 & 0 & 0 & -x_5 X_5 & -Y_5 X_5 \\
    0 & 0 & 0 & X_5 & Y_5 & 1 & -y_5 Y_5 & -y_5 Y_5 \\
\end{bmatrix} \times
\begin{bmatrix}
    s_{11} \\
    s_{12} \\
    s_{14} \\
    s_{15} \\
    s_{121} \\
    s_{122} \\
    s_{141} \\
    s_{142} \\
    \ast \\
\end{bmatrix}
\]

\[Y = \begin{bmatrix}
    X_1 \\
    Y_1 \\
    X_2 \\
    Y_2 \\
    X_3 \\
    Y_3 \\
    X_4 \\
    Y_4 \\
    X_5 \\
    Y_5 \\
\end{bmatrix} * \begin{bmatrix}
    H_1 \\
\end{bmatrix} \ast \begin{bmatrix}
    \ast \\
\end{bmatrix}
\]

Since the coefficient \( s_{44} \) shows the translation of the machine vision system along the Z-axis, the real value of \( s_{44} \) is rarely null. In the above Eq 5.17, \( Y \) is a column matrix which is calculated from the multiplication of the coefficient \( s_{44} \) with the last column of the matrix \( H \). This transformation is only possible since \( s_{44} \) is being constrained to a constant equal to 1. Thus \( s_{44} \) is no longer a free variable for which a solution is being sought. Geometrically, the solution for Eq 5.17 is a translation of the solution set for Eq 5.16, which only occurs when the vector \( Y \) lies in the image of the linear transformation matrix \( H \). The equation Eq 5.17 is solved by means of a pseudo inverse technique. As the projection matrix \( H \) is homogeneous, the definitive projection matrix is obtained by multiplying the calculated elements of matrix \( S_1 \) with the translation vector of the camera along the Z-axis [24]. For those coefficients that have been estimated, some have a physical value and other must be solved. Many techniques have been applied for the extraction of the physical parameters for the DLT matrix in relation to more complex equations but, many have failed to extract the physical parameters from the equation [25]. In order to extract a physical value another constraint was applied to the equation of the coefficients and a physical value of \( Y_0 \) was inserted into the equation by referring to the identity of the reference labels. In this manner, the pose of the machine vision system in the corridor was determined for different points horizontally along the corridor and vertically between the side walls.
6 Result

In this chapter, the results for the calculated pose of the machine vision system at different positions in the corridor along horizontal and vertical directions are displayed. The start was at one end of the corridor where the origin of the 3D world space is defined. The pose of the machine vision system is calculated at different positions by merely taking the images of six reference labels or more. In the following images the pose of the system is calculated using reference labels.

Figure 6.1: Image of the ten reference labels which is near to origin of the 3D world coordinates.

In order to estimate the coefficients of the matrix $S_i$, a least square estimation method was used. The pseudo inverse function $\text{pinv}$ is applied to the DLT matrix $H_i$ and the results are multiplied by the vector matrix $Y$.

$$S_i = (\text{pinv}(H_i)) \cdot Y \quad \text{...............Eq 6.1}$$

where the coefficients of matrix $S_i$ are determined from Eq 5.11 which corresponds to Eq 5.4 which are given as

$$s_{11} = \cos \theta, \quad s_{12} = \sin \theta, \quad s_{14} = -X_0 \cos \theta - Y_0 \sin \theta$$
$$s_{21} = -\sin \theta, \quad s_{22} = \cos \theta, \quad s_{24} = X_0 \sin \theta - Y_0 \cos \theta$$
$$s_{41} = 0, \quad s_{42} = 0, \quad s_{44} = (Z_0 / \lambda) + 1.$$
By inserting values of the reference labels into Eq 6.1 from the image and 3D world space, it is possible to estimate the elements of matrix $S_1$. The definitive projection matrix is obtained by multiplying the calculated elements of matrix $S_1$ by the translation vector of the camera along the Z-axis. The physical properties of the camera are now calculated from the matrix $S_1$ coefficients. For example, in the above image, the machine vision system is translated from the origin in the 3D space. It is assumed that for a translation of an angle $\theta$ along the Z-axis that this is equal to 0 degree. Now inserting the value of the angle in the above coefficients of matrix $S_1$, the following is obtained:

\[
\begin{align*}
    s_{11} &= \cos(0) = 1, \\
    s_{12} &= \sin(0) = 0, \\
    s_{14} &= -X_0\cos(0) - Y_0\sin(0) = -X_0 \\
    s_{21} &= -\sin(0) = 0, \\
    s_{22} &= \cos(0) = 1, \\
    s_{24} &= X_0\sin(0) - Y_0\cos(0) = -Y_0 \\
    s_{41} &= 0, \\
    s_{42} &= 0, \\
    s_{44} &= \frac{Z_0}{\lambda} + 1.
\end{align*}
\]

The above values of the coefficients of matrix $S_1$ are normally the derived values for the translation of the machine vision system. The physical properties of the machine vision system from the above coefficients are calculated as:

\[
\begin{pmatrix}
    s_{11} \\
    s_{12} \\
    s_{14} \\
    s_{21} \\
    s_{22} \\
    s_{24} \\
    s_{41} \\
    s_{42}
\end{pmatrix}
= 
\begin{pmatrix}
    1 & 0 & 0 & 0 \\
    0 & 0 & 0 & 1 \\
    -X_0 & 0 & 1 & 0 \\
    0 & -Y_0 & 0 & 0 \\
    0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 \\
\end{pmatrix}
= 
\begin{pmatrix}
    1.0057 & 0.0860 & -0.8687 & 0.0022 & 0.0013 & 1.3398 & -0.1108 & 0.0128
\end{pmatrix}
\]

As the positions of the labels are shifted in the 3-D world coordinates, the center label is at the origin of the 3D space. The calculated values of $X_0$ and $Y_0$ show a shifted position of the machine vision system from the origin. The calculated values of the camera are then added to the real position of the centred reference label in order to shift back to the actual position in the 3D world space. The deviation of the camera from the actual position and angle is discussed in the next chapter.
Another example of calculating the pose of the machine vision system at a different angle is now provided. The camera was rotated by a $\theta$ angle of 90 degrees along the Z-axis and the image of the symbols was taken as shown below.

After placing values of the reference labels into Eq 6.1, the elements of matrix $S_i$ were estimated. The definite values of the matrix $S_i$ coefficients are achieved by multiplying the calculated elements of matrix $S_i$ by the translation vector of the camera along the Z-axis.
The physical properties of the machine vision are calculated from the estimated coefficients values of matrix $S_i$.

\[
\begin{pmatrix}
S_{11} \\
S_{12} \\
S_{14} \\
S_{21} \\
S_{22} \\
S_{24} \\
S_{41} \\
S_{42}
\end{pmatrix} = \begin{pmatrix}
\cos \theta \\
\sin \theta \\
-X_0 \cos \theta - Y_0 \sin \theta \\
-\sin \theta \\
\cos \theta \\
X_0 \sin \theta - Y_0 \cos \theta \\
0 \\
0
\end{pmatrix} = \begin{pmatrix}
0.0186 \\
1.0020 \\
-12.2980 \\
-1.0066 \\
0.0173 \\
2.5693 \\
-0.0337 \\
-0.0434
\end{pmatrix}
\]

$s_{11} = \theta = \cos^{-1}(0.0186) = 89^\circ$, $s_{12} = \theta = \sin^{-1}(1) = 90^\circ$

$s_{14} = -X_0 \cos \theta - Y_0 \sin \theta = 12.2$, Where $\sin \theta = 1$, $\cos \theta = 0.0186$

Putting the values

$s_{14} = -X_0(0.0186)-Y_0(1)=-12.2$ \quad \Rightarrow \quad Y_0 = 12.2 \text{ cm}

$s_{21} = \theta = \sin^{-1}(1) = 90^\circ$, $s_{22} = \theta = \cos^{-1}(0.0173) = 89^\circ$

$s_{24} = X_0 \sin \theta - Y_0 \cos \theta = 2.5693$

Putting the values of $\sin \theta$ and $\cos \theta$

$s_{24} = X_0(1) - Y_0 (0.0173) = 2.5693$ \quad \Rightarrow \quad X_0 = 2.56 \text{ cm}

$s_{41} = -0.0337$, $s_{42} = -0.0434$

Figure 6.4: Showing the decoding of rotationally independent symbols at 90 degrees.
In the above images, the pose of the machine vision system was calculated without any constraint on the coefficients. However it is generally the case that it is only in the rotation of the camera along the Z-axis a constraint is applied by inserting the measured value of the $Y_0$ into the coefficients of the matrix $S_i$.

Now the overall accuracy of the system is discussed. Experiments have been performed by calculating the pose of the system at different points in the corridor. In these experiments the supposition is that the angle $\theta$ is equal to 0, which shows only the translation of the machine vision along the corridor. The calculated position is represented in the following table and also shown in the graph.

Table 6.1:

<table>
<thead>
<tr>
<th></th>
<th>Measured value along X-axis (cm)</th>
<th>Measured values along Y-axis (cm)</th>
<th>Calculated value along X-axis (cm)</th>
<th>Calculated values along Y-axis (cm)</th>
<th>Deviation along X-axis (cm)</th>
<th>Deviation along Y-axis (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>295</td>
<td>60</td>
<td>293.3</td>
<td>62.4</td>
<td>1.7</td>
<td>2.3</td>
</tr>
<tr>
<td>2</td>
<td>364</td>
<td>62</td>
<td>363.09</td>
<td>61.5</td>
<td>0.9</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>424.2</td>
<td>62</td>
<td>424.7</td>
<td>61.4</td>
<td>0.50</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>483.5</td>
<td>60</td>
<td>482.48</td>
<td>61.7</td>
<td>1.01</td>
<td>1.7</td>
</tr>
<tr>
<td>5</td>
<td>539.5</td>
<td>60</td>
<td>535.79</td>
<td>59.2</td>
<td>3.71</td>
<td>0.8</td>
</tr>
<tr>
<td>6</td>
<td>896</td>
<td>61</td>
<td>893.9</td>
<td>60</td>
<td>2.01</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>955.2</td>
<td>60</td>
<td>955.48</td>
<td>58.3</td>
<td>2.7</td>
<td>0.28</td>
</tr>
<tr>
<td>8</td>
<td>1014.7</td>
<td>60</td>
<td>1011.9</td>
<td>59.7</td>
<td>2.8</td>
<td>0.3</td>
</tr>
</tbody>
</table>
For calculation of a differential error in the system other types of experiments were conducted by placing the system at different positions from the measured points in the corridor. At first the pose of the machine vision system was measured and calculated following which the position of the machine vision system was changed by 10 cm, 5 cm, 6 cm, 9 cm and 7.5 cm. Again the pose of the system was calculated and the results are shown below.

Table 6.2:

<table>
<thead>
<tr>
<th></th>
<th>Measured value along X-axis (cm)</th>
<th>Measured values along Y-axis (cm)</th>
<th>Calculated value along X-axis (cm)</th>
<th>Calculated values along Y-axis (cm)</th>
<th>Differential error along X-axis (cm)</th>
<th>Differential along Y-axis (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>58</td>
<td>9.7</td>
<td>55.9</td>
<td>0.3</td>
<td>2.1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>62.3</td>
<td>9.1</td>
<td>63.7</td>
<td>0.9</td>
<td>1.4</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>60</td>
<td>3.3</td>
<td>61.7</td>
<td>1.7</td>
<td>0.8</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>60</td>
<td>4.9</td>
<td>58.8</td>
<td>1.1</td>
<td>1.2</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>60</td>
<td>7.5</td>
<td>59.3</td>
<td>1.5</td>
<td>0.7</td>
</tr>
<tr>
<td>6</td>
<td>7.5</td>
<td>51</td>
<td>7.4</td>
<td>50.6</td>
<td>0.1</td>
<td>0.4</td>
</tr>
</tbody>
</table>
In order to determine a different orientation of the camera, it was rotated around its axis along the Z-axis. The orientation of the camera is measured and calculated by this system at different points in the corridor. The result of the pose calculation shows that the orientation of the camera is more sensitive to noise data.

Table 6.3:

<table>
<thead>
<tr>
<th></th>
<th>Measured value along X-axis (cm)</th>
<th>Measured values along Y-axis (cm)</th>
<th>Measured angle along Z-axis</th>
<th>Calculated value along X-axis (cm)</th>
<th>Calculated values along Y-axis (cm)</th>
<th>Calculated angle in degrees</th>
<th>Deviation Of Angle θ in degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>159.8</td>
<td>10</td>
<td>70</td>
<td>162.3</td>
<td>9.4</td>
<td>68</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>259.5</td>
<td>10</td>
<td>24.5</td>
<td>260</td>
<td>9.18</td>
<td>23.6</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>458.5</td>
<td>10</td>
<td>18.2</td>
<td>458.69</td>
<td>9.6</td>
<td>17.2</td>
<td>0.8</td>
</tr>
<tr>
<td>4</td>
<td>952.3</td>
<td>22</td>
<td>55</td>
<td>949.8</td>
<td>24.09</td>
<td>52.9</td>
<td>2.1</td>
</tr>
<tr>
<td>5</td>
<td>1059.5</td>
<td>10</td>
<td>47</td>
<td>1061.7</td>
<td>6.59</td>
<td>40.8</td>
<td>6.2</td>
</tr>
</tbody>
</table>
Figure 6.7: Showing the calculated angle of the camera at different positions in the corridor.
7 Analysis

7.1 Light intensity

A machine vision system measurement depends on the light source. In indoor optical navigation, the segmentation process may be affected by a change in the light intensity. For example, if the machine vision system is under a direct light source, the same threshold value of the mean value filter cannot extract the reference labels as a bigger object from this image as shown in figure 7.1. Thus a more dynamic algorithm for the decoding of the symbols has been written as shown in fig 7.2.

Figure 7.1: In the above image calculated AOI which are due to increase in light intensity.

Figure 7.2: The results of the dynamic algorithm which neglect the changes in the light intensity inside the building.

7.2 Table of reference label

42 symbols are used in the corridor to determine the pose of the machine vision system. When an image of the symbols is taken, these are attached to the ceiling of the corridor. The identity of each symbol is used to provide its position in 3D world coordinates in order to compute the pose of the machine vision system. The tables for the reference labels data have been created in the algorithm, which is used in the camera calibration along the corridor.
7.3 **Image Resizing**

An image of the system was taken from the USB camera, which has a default resolution of 2048x1536. Then the algorithm is applied to this image with the same resolution in order to calculate the pose of the system. As there may be six or more symbols to decode in the image with the same resolution, this may involve significant processing time depending on the complexity of the background. In this technique, the image was resized and the pixel size recalculated in both the horizontal and vertical directions of the pixel coordinate plane. The algorithm is then applied to the resized image which has a resolution of 840 X 480 and this resulted in a substantial reduction in the processing time. However, the drawback is that it may affect the accuracy of the system.

7.4 **Error Calculation**

The pose of the machine vision system was calculated at different places in the corridor. The absolute error in the system was calculated and this is the amount of physical error in a measurement period [26]. The absolute error in this system for the calculation of the position is ± 3 cm and in relation to the angle measurement, the absolute error is ± 8°. The cause of distortion in the machine vision system is the alignment issue of camera coordinates with the 3D world coordinates specifically in the rotation of the camera along the Z-axis and also inaccuracy in the physical measurement of the position of the symbols in the 3D space may contribute to the errors. Other types of distortion are linear distortion and radial lens distortion, which cause the position of the reference label to be displaced radially in the image plane. In the experiments it was also noticed that when light intensity changes on the label surface, cause slightly movement in the centre position of labels. This type of error occurs if the label surface is not uniformly illuminated. The error of the system can be reduced by using a high quality lens and by ensuring that the reference labels are distributed for calculation of the DLT matrix in the image, [27] and also by increasing the number of measured points in the image, which may be useful for improving the overall precision. The equations for the calculation of the error in the pose are given below.

Absolute error in position = (X₀ − Xₘ),
Absolute error in angle = (θ₀ − θₘ),
where X₀ and θ₀ are calculated values and, Xₘ and θₘ are the physically measured values.
8 Conclusion

In this project the main task has been achieved, by implementing an indoor navigation system with high accuracy. The results show that the identity of 10 color reference labels in an indoor office environment have been investigated, which provide 10 bytes of information in one image. The identity of the symbols is used as a reference in the implementation of geometrical camera calibration. The calculated absolute error for finding the position of the camera is ± 3 cm along both the horizontal and vertical directions. The absolute error in the calculated angle is ± 8° degrees. This method can be beneficial in a robot vision system as the accuracy falls within the required range.

In future this project could be implemented on a mobile operating system. In the marketplace, there are many mobile apps, which normally require an internet connection to discover the position of the navigator. It is the case nowadays that the majority of mobiles have a camera by taking images of these symbols using the mobile camera, a navigator can discover his/her position anywhere in the building.
References


