NTNU – Trondheim
Norwegian University of
Science and Technology

Visual Sensing in Mobile Robots

Vegard S. Lindrup

Submission date:	December 2015
Responsible professor:	Firstname Lastname, Affiliation
Supervisor:	Firstname Lastname, Affiliation

Norwegian University of Science and Technology
Department of Telematics

Abstract

Preface

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

List of Figures

List of Tables

Contents

Chapter 1
Introduction

This section is intended to provide an overview of the contents and context of this report. The first part of this section gives a brief introduction to the field of mobile autonomous robotics and computer vision, as well as the benifits and potential applications for this technology. The robot system and tools used in the project is presented in subsection **??**. Lastly, each of the following sections will given short introductions.

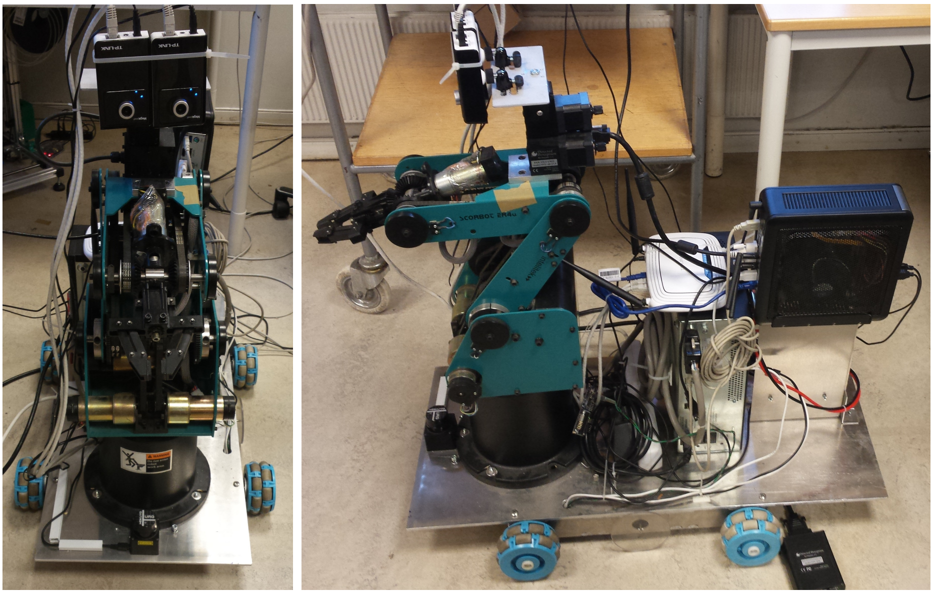
1.1 Mobile Autonomous Robotics and Computer Vision

Put the task into a larger context. Bring in some points on the societal impact of autonomous robotics and the increased potential of mobile robotics.

The field of computer vision has seen an enormous growth over the last few decades - not only in scale, but in accessibility and capability as well. As a consequence of this recent growth, tapping into the field of computer vision is bound to reveal applications that are useful for a mobile autonomous maintenance robot. Recent discoveries within computer vision includes robust feature recognition and object detection, face detection and video processing. The latest great additions to the field are Big Data and Artificial Intelligence.

1.2 System Overview

The mobile robot being worked on in this project is shown in figure 1.1a. The manipulator arm has been used in previous projects on robotic maintenance, and it was placed on the mobile platform during the master thesis of (Aspunvik og siter). This section provides a short description of the hardware. If a more detailed description of the robot and it's equipment is required, consult the thesis of Aspunvik[cite].



(a) Front view. (b) Side view.

Figure 1.1: The robot used in the project.

1.2.1 Propulsion

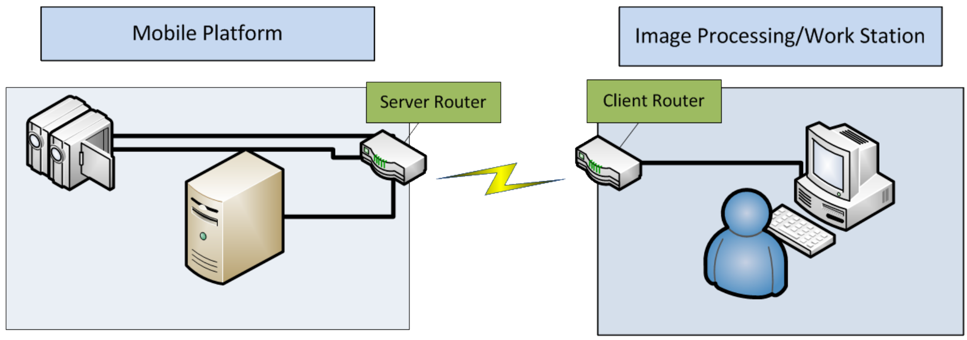
The steel chassis of the robot stands upon four omni-wheels. The wheel pairs are placed in parallel, making the vehicle uncontrollable along the lateral axis. Each wheel is powered by an electrical motor and motor driver. The motor drivers are controlled with pulse width modulation by an evaluation board from Atmel,(atmel kort).

1.2.2 Sensors

The robot was outfitted with several sensors over the course of previous projects. These are:

- Two odometer wheels with encoders. One on each side.
- Two infrared distance sensors.
- A LIDAR (Light Detection and Ranging).
- Two IP-cameras.

Only the cameras were used in this projects.



1.2.3 The Manipulator

1.3 Report Structure

How the report is structured, and a very brief description of the contents in each section.

Chapter 2

Background Theory

2.1 Introduction to Computer Vision

2.1.1 Introduction

This chapter contains the background theory which is necessary to understand the implementations in chapter 4 and how they are intended to work.

2.1.2 How it's Done

nanana

2.1.3 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source library with a vast number of advanced computer vision and machine learning algorithms. The library supports Windows, Linux, iOS and Android, and has interfaces to C, C++, Python, Java and MATLAB. All OpenCV applications in this project uses OpenCV 3.0.0 for Windows. OpenCV for Windows can be downloaded from sourceforge.net. This download contains source files, sample programs, sample data and a pre-built library for MSVC 2010 and 2013. The pre-build library can quickly be plugged into an IDE such as Qt Creator or Visual Studio 2013, thus giving the programmer access to all basic OpenCV features. A steb-by-step guide for using both the pre-built and a custom-built library can be found in Appendix **??**.

2.2 Stereo Vision and Depth Perception

Stereo vision and depth perception is one of the core topics within this report. Here, the theory behind a method using two cameras is presented, while some additional methods are mentioned to provide context.

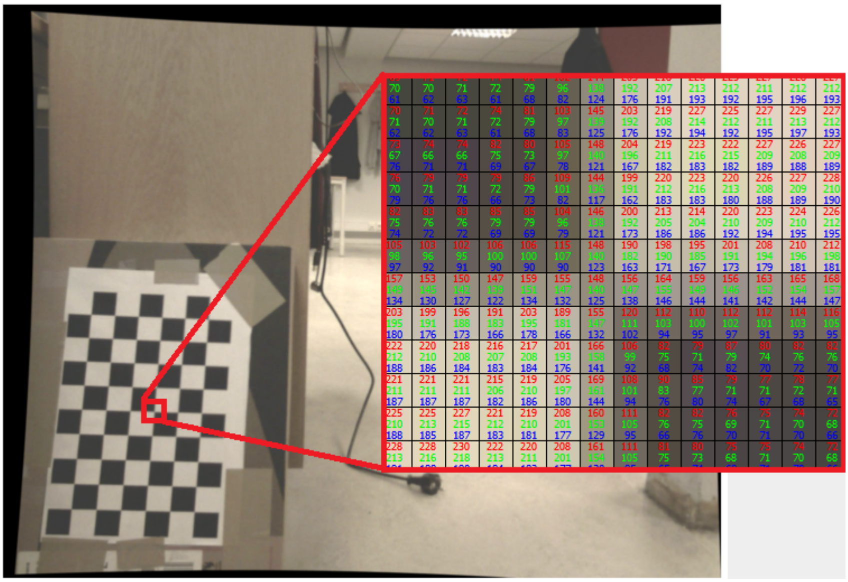


Figure 2.1: Elements in a cv::Mat representing an image. This is a 3-channel image with 8 bit RGB colors.

2.2.1 Various Methods

Methods for depth perception in computer vision can be separated into two main categories, i.e. active and passive. Active sensors will usually project a light pattern onto the scene to be perceived, before sensing how this pattern is displaced by the topology of the scene. The Kinect sensor and 3d-scanners using laser light are typical examples of active sensors. Passive depth perception makes use of many of the same cues we use to perceive depth. The most common passive sensors extract the depth information by observing observing a scene from at least two different positions.

Optical flow is another important method for depth perception. Optical flow may be either active or passive. The passive variant requires only one camera, but depends on motion and a stream of images to extract depth information. Observing how much some chosen features in a scene has moved in the image frame at $t = 1$ compared to the frame at $t = 0$ is the basis of depth sensing from optical flow. When the camera moves through a scene where all objects are stationary, objects that are far away will naturally have an optical flow field with a smaller magnitude than objects that are close.

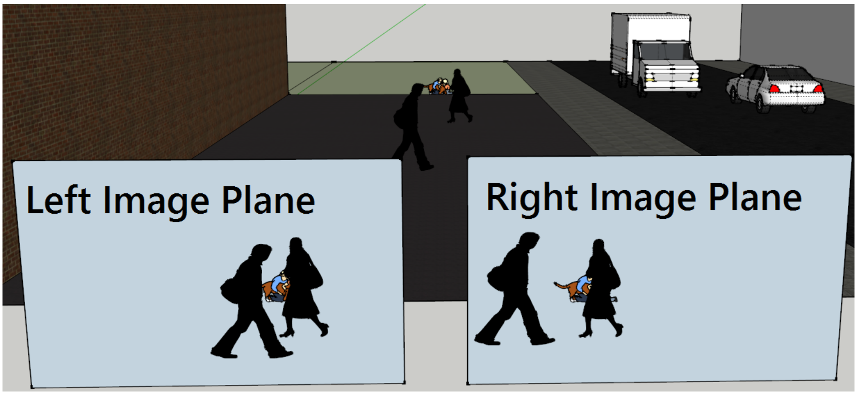


Figure 2.2: Left to right displacement on the image plane based on distance.

2.2.2 Stereoscopic Vision in General

In this project, passive stereoscopic vision is achieved by using two identical (in theory) cameras placed on the same plane. The gist of passive stereoscopic vision is based on the fact that objects close to the camera pair will have a large displacement from the left to the right camera compared to objects that are further away. This concept is illustrated in figure 2.2.

The Pinhole Camera Model Figure **??** illustrates the geometry of a pinhole camera. In a pinhole camera, a scene will be projected onto an image plane, often denoted π, through the camera projection point O which is located at the origin. The focal length f is the distance from the projection point O to the image plane π. In a true pinhole camera the image plane will be located behind the lens and the projection point O, and the scene projection will be rotated by 180°. A virtual image plane placed in front of O is intended to make the figure more straightforward.

Stereo Cameras Figure 2.4 shows an ideal stereo camera model. The model comprise two pinhole camera models where the virtual image planes are located on the same plane. The two image planes are separated by a horizontal translation B which is called the baseline. This implies that the projection point O_L in the left camera, relates to the projection point O_R

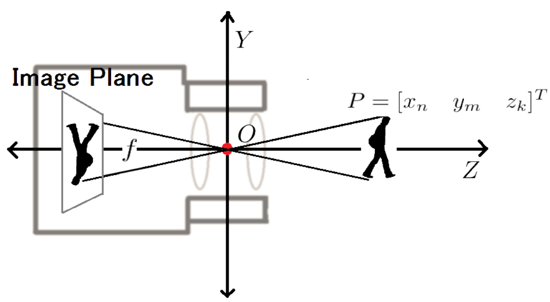


Figure 2.3: Geometry of a pinhole camera.

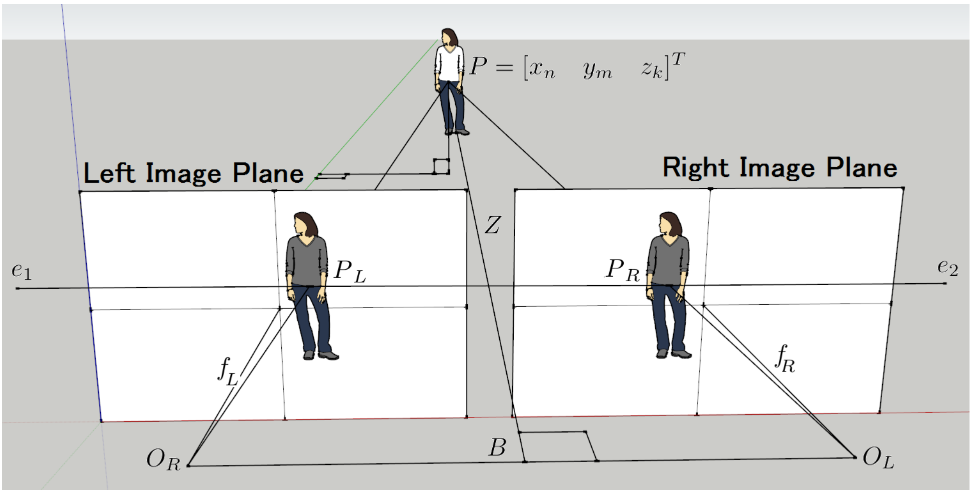


Figure 2.4: Geometry of stereo vision.

on the right camera through B: $O_R = O_L + B$. Each of the two image planes has a left handed pixel based coordinate system u, v, i.e. the origin is in the top left corner and the opposite pixel is in the bottom right corner.

Camera Distortion Cite [Ein05].

2.2.3 Stereoscopic Vision in OpenCV

The prebuilt version of OpenCV 3.0.0 comes with two stereo matching algorithms: Block Matching (StereoBM) and Semi Global Block Matching (StereoSGBM). Additional algorithms are available if OpenCV is built with, e.g. CUDA.

StereoSGBM [Hir08] blablabalb

Chapter 3
Example

Here is an example of how to use acronyms such as Norwegian University of Science and Technology (NTNU). The second time only NTNU is shown and if there were several you would write NTNUs. And here is an example[1] of citation [NN00].

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original

[1] A footnote

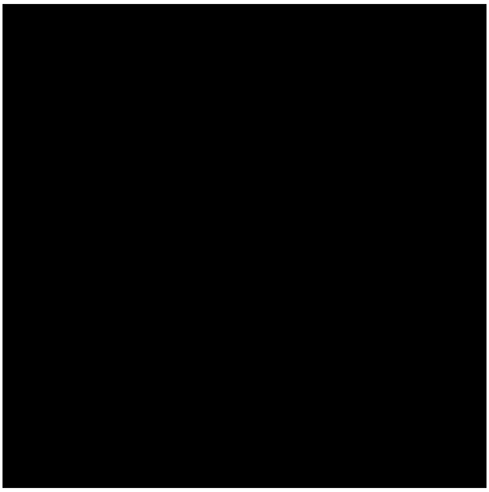


Figure 3.1: A figure

language. There is no need for special content, but the length of words should match the language.

3.1 First section

3.1.1 First subsection with some $\mathcal{M}ath$ symbol

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

- item1
- item2
- ...

3.1.2 Mathematics

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. $\sin^2(\alpha) + \cos^2(\beta) = 1$. If you read this text, you will get no information $E = mc^2$. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not

Table 3.1: A table

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y
z	æ	ø	å	

at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. $\sqrt[n]{a} \cdot \sqrt[n]{b} = \sqrt[n]{ab}$. This text should contain all letters of the alphabet and it should be written in of the original language. $\frac{\sqrt[n]{a}}{\sqrt[n]{b}} = \sqrt[n]{\frac{a}{b}}$. There is no need for special content, but the length of words should match the language. $a\sqrt[n]{b} = \sqrt[n]{a^n b}$.

B
X_L
X_R
P_R` P_L
$P = [x \quad y \quad z]^T$
$P = [x_n \quad y_m \quad z_k]^T$
e_1
e_2
O_R
O_L
$X = [x \quad y \quad z]^T$
f
Z
Y

Proposition 3.1. *A proposition... (similar environments include: theorem, corrolary, conjecture, lemma)*

Proof. And its proof. □

3.1.3 Source code example

You can refer to figures using the predefined command like Figure 3.1, to pages like page 10, to tables like Table 3.1, to chapters like Chapter 3 and to sections like Section 3.1 and you may define similar commands to refer to proposition, algorithms etc.

Algorithm 3.1 The Hello World! program in Java.

```
class HelloWorldApp {
  public static void main(String[] args) {
    //Display the string
    System.out.println("Hello World!");
  }
}
```

Chapter 4
Implementation

4.1 Vanishing Point Detection

4.1.1 Overview

4.1.2 Line Detection

4.1.3 Line Filtering

4.1.4 Vanishing Point Detection

4.1.5 Graphical User Interface

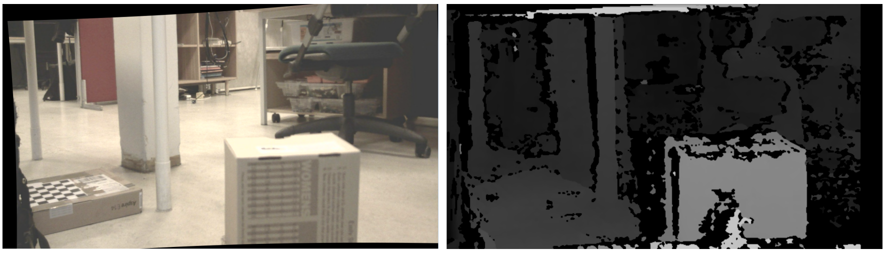
4.1.6 Where it Fails

4.2 Depth Perception and Obstruction Detection

4.2.1 Overview

4.2.2 The camera rig

The two IP cameras were moved together to form a stereo camera. This stereo camera was used in two positions. The first camera position is on the pan-tilt module on the robot arm, see figure **??**. The second position is just over the LIDAR in front of the robot arm base, see figure **??**. The workshop at ITK made a mounting bracket, so that the cameras could be placed over the LIDAR. In stereo vision, it is essential that the positions of the cameras relative to each other is constant. One problem encountered throughout the project was that the camera assemby, when placed either at the pan-tilt module and over the LIDAR, was not rigid enough. The severity of this problem was somewhat alleviated by wrapping a strap around both the cameras. This camera rig is ad hoc, i.e. suitable for the purpose of this project, but a better solution should be used for following projects.



(a) Left camera image. (b) Disparity map.

Figure 4.1: The result of StereoSGBM.

4.2.3 Calibration

As mentioned in chapter **??**, all cameras will have some distortion. If the distortion is too severe, as it often will be in the context of stereo vision, the camera must be calibrated. In addition, it was assumed that the image planes were located on the same plane, and that a projection pair, for example the projections X_L and X_R of an object X, form two equal epipolar lines, e_1 and e_2, on the two image planes. In practice, these conditions must be realized through stereo calibration. The second purpose of the calibration procedure is to relate the sensor data to real world quantities.

Single Camera Calibration

Stereo Calibration

Stereo Rectification

4.2.4 Stereo Matching

4.2.5 Finding Obstructions

4.2.6 Distance Measurment

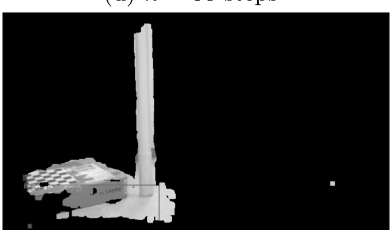
4.2.7 Problems Encountered During Implementation

(a) $n = 10$ steps

(b) $n = 25$ steps

(c) $n = 50$ steps

(d) $n = 50$ steps



(e) $n = 50$ steps

Figure 4.2: Classical Random

References

[Ein05] Albert Einstein. Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]. *Annalen der Physik*, 322(10):891–921, 1905.

[Hir08] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2), 2008.

[NN00] Firstname 1 Name1 and Firstname2 Name2. A dummy title. *A Fake Journal*, 1(1):000–000, June 2000.

Appendix A

Setting up a project with Qt and OpenCV