Czech Technical University in Prague
Faculty of Electrical Engineering

DIPLOMA THESIS

Bc. Lenka Mudrová

Entropy-Like Estimation Technique in Mobile Robot Localization

Department of Cybernetics
Supervisor: **Ing. Jan Faigl, Ph.D.**

Prague, 2013

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 3. 1. 2013 ....

podpis

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Lenka Mudrová

Studijní program: Kybernetika a robotika (magisterský)

Obor: Robotika

Název tématu: Technika odhadu parametrů založená na entropii v úloze navigace mobilního robotu

Pokyny pro vypracování:

1. Seznamte se s technikou ,,Entropy-Like" odhadu parametrů (LEL – Least Entropy Like) [1, 2].
2. Aplikujte techniku LEL v úloze lokalizace z dat dálkoměrných senzorů.
3. Seznamte se s úlohou SLAM [3] a přístupy jejího řešení.
4. Porovnejte techniku LEL s přístupem ICP.
5. Seznamte se s přístupem vizuální navigace [4] a jejím rozšířením [5].
6. Aplikujte techniku LEL v úloze vizuální navigace mobilního robotu ve venkovním prostředí.

Seznam odborné literatury:

[1] Giovanni Indiveri: An Entropy-Like Estimator for Robust Parameter Identification. ENTROPY, vol. 11; p. 560-585, 2009.

[2] Francesco Di Corato, Lorenzo Pollini, Mario Innocenti, Giovanni Indiveri: An Entropy-like approach to vision based autonomous navigation. ICRA 2011: 1640-1645.

[3] Sebastian Thrun, Wolfram Burgard, Dieter Fox: Probabilistic Robotics, The MIT Press, 2005.

[4] Tomáš Krajník, Jan Faigl, Vojtěch Vonásek, Karel Košnar, Miroslav Kulich, Libor Přeučil: Simple, yet stable bearing-only navigation, J. Field Robot., 2010.

[5] Hana Szücsová: Computer vision-based mobile robot navigation. Master's thesis, Czech Technical University in Prague, 2011.

Vedoucí diplomové práce: Ing. Jan Faigl, Ph.D.

Platnost zadání: do konce letního semestru 2012/2013

prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry

L.S.

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 9. 12. 2011

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Lenka Mudrová

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Diploma Thesis: Entropy-Like Estimation Technique in Mobile Robot Navigation

Guidelines:

1. Study the entropy-like estimation technique (LEL – Least Entropy Like) [1, 2].
2. Apply LEL in the localization task using range sensors.
3. Study the SLAM problem [3] and approaches addressing the problem.
4. Compare LEL with the ICP approach.
5. Study the visual navigation method [4] and its extension [5].
6. Apply LEL in the visual navigation task for outdoor environment.

Bibliography/Sources:

[1] Giovanni Indiveri: An Entropy-Like Estimator for Robust Parameter Identification. ENTROPY, vol. 11; p. 560-585, 2009.

[2] Francesco Di Corato, Lorenzo Pollini, Mario Innocenti, Giovanni Indiveri: An Entropy-like approach to vision based autonomous navigation. ICRA 2011: 1640-1645.

[3] Sebastian Thrun, Wolfram Burgard, Dieter Fox: Probabilistic Robotics, The MIT Press, 2005.

[4] Tomáš Krajník, Jan Faigl, Vojtěch Vonásek, Karel Košnar, Miroslav Kulich, Libor Přeučil: Simple, yet stable bearing-only navigation, J. Field Robot., 2010.

[5] Hana Szücsová: Computer vision-based mobile robot navigation. Master's thesis, Czech Technical University in Prague, 2011.

Diploma Thesis Supervisor: Ing. Jan Faigl, Ph.D.

Valid until: the end of the summer semester of academic year 2012/2013

L.S.

prof. Ing. Vladimír Mařík, DrSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, December 9, 2011

Abstrakt

Tato diplomová práce se zabývá problémem lokalizace mobilního robota v neznámém venkovním prostředí. Pozice robota je zde odhadována na základě pozorovaní okolí robota senzory (dálkoměrem či kamerou). Tato pozorování jsou tvořena množinou dílčích měření. Dvě takové množiny pozorování (skeny prostředí) pořízené ze dvou posobě následujících pozic robota jsou použity v přístupu „scan-to-scan", který odhaduje parametry transformace popisující relativní pohyb robota. Přesnost odhadu je ovlivněna šumem a pozorováními, jejichž číselná hodnota je značně vzdálená od většiny pozorování, tzv. „outlier" měřeními. V této práci je studována nová technika pro odhad parametrů transformace pozice robota, která je založena na měření disperze reziduí s využitím vlastností entropie. Tato technika (LEL) byla vytvořena s důrazem na robustní chování v případě, kdy jsou vstupních data se značným množstvým „outliers" . S ohledem na toto robustní chování představuje metoda LEL slibný přístup pro řešení problému lokalizace mobilního robota.

Hlavním cílem této diplomové práce je posouzení a ověření chování lokalizačního systému s využitím metody LEL v úloze lokalizace mobilního robota v neznámém reálném prostředí. Uvažovaný systém pro lokalizaci robota je založen na systému dvou kamer, které poskytují pozorování prostředí robota ve stereo obrázcích. Významné objekty prostředí, vhodné pro lokalizaci, jsou z těchto obrázků získána metodou „Speeded-Up Robust Feature" (SURF), pro které jsou určeny hloubky na základě určení rozdílu pozice pozorování v pravé a levé kameře. Lokalizační metoda je také ovlivněna způsobem určení korespondujících pozorování mezi dvěma po sobě následujícími pozicemi robota, a proto je tento vliv v práci také diskutován. Mimoto, je v práci navržena nová metoda hledání silně korespondujících dvojic, která pozitivně ovlivňuje chování metody LEL. Kromě toho, je v práci také uveden přehled lokalizačních technik a jejich porovnání. Na závěr této práce jsou diskutovány zjištěné vlastnosti metody LEL v úloze lokalizace mobilního robota.

Abstract

This diploma thesis deals with a problem of the mobile robot localization in an unknown outdoor environment. The studied localization problem is based on processing observations of the robot surroundings sensed by its exteroceptors providing a set of measurements called scan. Such a scan contains a percepted features of the environment that are used in a scan-to-scan localization method. The method provides an estimation of the robot pose transformation describing the robot motion (from which the global robot pose is determined) and the precision of the estimation is influenced by a noise and outliers in the input datasets (scans). In this thesis, a new estimation technique called Least Entropy-Like (LEL) to find parameters of the transformation is studied in the context of the mobile robot localization problem. This technique has been designed to be robust to a dataset corrupted by a significant amount of outliers, and therefore, it is a promising technique to solve the localization problem.

The main goal of the thesis is to evaluate and verify the performance of LEL in a serie of experiments and realistic scenarios of mobile robot localization to provide a realistic expectation of the performance in a real deployment of the method. The considered robot localization system is based on a stereoscopic camera system and extraction of features from the image using the Speeded-Up Robust Feature (SURF) detection and estimation of the feature's depth from the disparity between the left and right images and known parameters of the cameras. In addition to evaluation of the estimation technique for outliers, the technique is evaluated also according to the quality of found correspondences between features in two consecutive scans. Moreover, a new data association method is proposed to extract only strong feature correspondences, which positively impact the performance of the LEL technique. Besides, an overview of localization techniques and their comparison is presented. Finally, the properties and discovered findings are presented in the conclusion.

Velice děkuji Ing. Janu Faiglovi, Ph.D. za trpělivost, ochotu a cenné rady, jak k obsahu práce, tak správnému psaní technického textu v angličtině. Dále mu děkuji za všechen čas, který mi při konzultacích věnoval i v rámci svého osobního volna. Děkuji za jeho zapálení do problematiky, které mě pomáhalo překonat těžké chvíle, a za úžasnou pečlivost, ke které mě také vedl. Na závěr bych mu ráda poděkovala za pochopení a podporu v ostatní situacích mého života.

Dále bych chtěla poděkovat bratrovi a jeho rodině za psychickou podporu. Děkuji mému příteloví za to, že mě vždy vyslechl, což mi mnohdy pomohlo si ujasnit směrování diplomové práce. V neposlední řadě děkuji rodičům za podporu při studium a svým kamarádům ze školy, se kterými jsem sdíleli radosti i strasti při studiu.

Contents

List of Figures

Introduction

Mobile robots have become a part of humans life and they are designed to help with many tasks, for example exploring unknown environment, transfer material from one location to another, clean or entertain. An example of nowadays robotic platforms for industrial as well as home use is depicted in Fig. 1. The most of such tasks can be defined by a request to move a robot from its initial location to the desired location. Therefore the robot needs to know its location according to the selected coordinate system; thus, be able to localize itself. The localization problem is difficult due to the following reasons. Although global localization systems like GPS [39] or GLONASS [42] are available in outdoor areas, they are not sufficiently precise or available in all areas, e.g., indoor. On the other hand, systems like VICON [40] can be used in indoor with very precise estimation of the robot position. However, they require expensive infrastructure. Also, such systems do not determinate all parameters of the robot pose, for example GPS estimates 3D position but it cannot directly determine orientation. Because of such limitations other sensors and localization methods are combined together to overcome such issues and provide an independent localization system and a more precise estimation of the robot position and orientation [5].

(a) (b)

Figure 1: Examples of mobile robots; a) industrial robot for transferring [52]; b) a robotic vacuum cleaner [51].

Sensors sense states of the robot or its surrounding environment and the robot pose is then determined using such observations. Localization methods can be divided into two main groups according to the type of measured values. Sensors from the first group measure inner states of the robot. An example of such sensoric system is an odometry for a wheeled robot, where rotations of the wheels are used to estimate robot position changes. The second group are exteroceptors that observe the robot surroundings and estimation of the robot position is based on observation of the environment. Such a sensor can be laser or sonar based rangefinders or a camera.

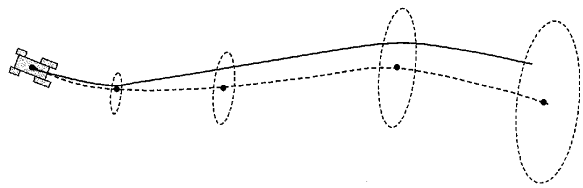


Figure 2: Increasing uncertainty during a robot localization [32].

The localization methods can be also divided to local and global [33]. The *local* localization, also known as tracking, is a dead-reckoning problem. An initial position is required and the robot pose is estimated using estimation of the traveled distance, robot heading and the robot's previous pose. The main limitation of the local localization is an integration of the measurement noise that increases uncertainty in the estimation of the robot pose. An example of increasing uncertainty in localization in terms of covariance matrix visualized as ellipses [38] is shown in Fig. 2.

(a) (b)

Figure 3: Global Monte Carlo localization method, the posterior mobile robot pose is represented by set of weighted particles that are updated; a) the initial states, the current robot pose is unknown; b) during localization, the uncertainty in robot pose is reduced; [9].

In contrast, the *global* localization works with many hypotheses about the robot position. Based on new observations of the world and the current available map of the environment, the hypotheses are adjusted and the most promising hypothesis stands for the estimation of the robot position in the global coordinate frame, see Fig. 3. These methods can also solved the so-called "kidnapped robot problem" [10], it means that robot is transferred by another force to unknown position. For example, a modern *iRobot Mint* home robot cleans the room more systematically, it builds a map of the environment and plans its movements to cover the whole floor. However, sometimes the cleaning pad has to be replaced; so, a human lifts up the robot, changes the pad and puts the robot on the floor again. The robot will be unlikely placed to the identical location; hence, the robot has to solve the "kidnapped robot problem" and has to localize itself according to the previously build map.

The precision of localization methods also depends on the robot surroundings properties [29]. A surface, where a robot operates, can be rough, for example path made from pavers; thus, information from sensors can be influenced by robot shaking. The robot surrounding is very miscellaneous and it is difficult to specify all possibilities. Even more, moving objects often exist and they are influencing the robot environment. Localization systems have to be capable to handle with these and more challenges.

The **goal of this thesis** is to evaluate a localization system for a mobile robot that operates in an outdoor environment and uses a perception sensor to sense robot surroundings. These sensed measurements form a dataset, which includes important features and outliers that are observations numerically distant from the rest of the features. These datasets are used in different *local* localization approaches as *scan-to-scan matching, scan-to-map matching* or *simultaneous localization and mapping*. In this thesis, we assume that the new proposed **Least Entropy-like technique** [7] is a promising optimization technique for localization and therefore, its performance is evaluated and compared with well-know localization methods of the *scan-to-scan* matching approach.

The thesis is organized as follows. The problem definition is given in Chapter 1. A summary of considered sensors and types of the robots is given in Chapter 2. The perception process is described in Chapter 3. The Least Entropy-like localization technique is introduced in Chapter 4 and other localization techniques are described in Chapter 5. The scenarios for verification of the localization techniques are proposed in Chapter 6. The Chapters 7 and 8 are dedicated to experimental evaluation in the indoor and outdoor scenarios, respectively. The concluding remarks are in Chapter 9.

Chapter 1

Problem Definition

Consider a mobile wheeled robot that operates in an unknown environment and is equipped with odometry and perception sensors. An odometry sensor measures relative movements of the robot according to the relative coordinate system, which is placed at the robot center of rotation. The global coordinate system is placed at the specific place of the environment, see Fig. 1.1. The odometry mea-

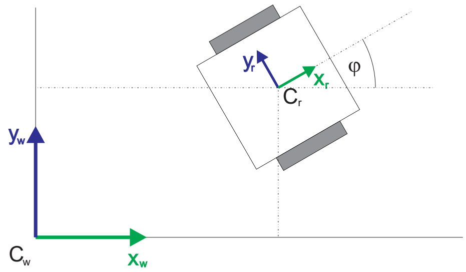


Figure 1.1: An example of the global and relative coordinate systems.

surements are recorded at particular time instants, for time T the odometry values are

$$U_T = u_1, u_2, u_3, ..., u_T.$$

A perception sensor senses the robot surrounding environment features f at different time instants t. Let the set of these measurements be

$$Z_T = z_1, z_2, z_3, ..., z_t.$$

The map $\mathcal{M}$ of the environment is determined using the sensed features as

$$\mathcal{M} = f_1,\ f_2, \ldots, f_n.$$

The variables, their dependencies and relations can be seen in the graphical model shown in Fig. 1.2.

The goal of localization techniques is to estimate the robot poses at particular time instants. The robot poses can be defined in the global coordinate system and the robot path can be defined as a sequence of poses

$$X_T = x_0, x_1, x_2, ..., x_T.$$

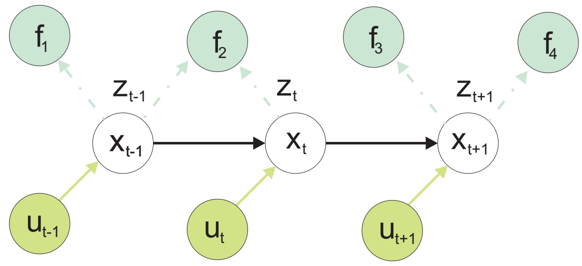


Figure 1.2: Graphical model of the localization variables.

A naïve localization method uses the odometry measurements transformed to the global coordinate system and the robot initial position x_0. If the odometry measurements are without a noise, these process will estimate the exact robot path. However, the odometry measurements suffer from a sensor noise and the error is cumulated along the path; therefore, this approach is not sufficient for long robot paths. Another localization technique is based on the estimation of a transformation between two consequent scans that includes the observed features of the environment. This thesis is focused on this approach, its main idea is described in the following section. Also, the phenomena that influence the localization precision are introduced here.

1.1 Localization Using Observations of the Environment

Having two scans P and Q acquired at the robot positions P_p and Q_p, respectively, the localization techniques solve the problem of finding a transformation $\mathbb{T}$ projecting the scan Q to P. This situation is visualized in Fig. 1.3 for one pair of sensed features. The feature sets P and Q have coordinates in different basis ac-

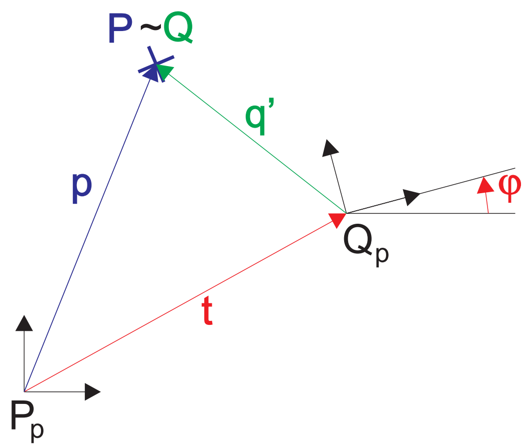


Figure 1.3: Relation of two different robot positions and corresponding observations.

cording to the robot relative position; so, the coordinates Q have to be transformed to the base of P

$$Q' = RQ,$$

where R is the rotation matrix. Then, equation

$$P = Q' + T = RQ + T, \tag{1.1}$$

holds, where $T = [dx,\ dy,\ dz]$ is the translation matrix. The robot relative transformation can be defined as

$$\mathbb{T} = [dx,\ dy,\ dz,\ \varphi,\ \beta,\ \gamma], \tag{1.2}$$

where φ is yaw (the rotation about z-axis), β is pitch (the rotation about y-axis) and γ is roll (the rotation about x-axis).

Definition of Studied Problem

Having the aforementioned preliminaries, the problem studied in this thesis can be defined as follows. *Let the robot operational environment be static, sensors have fixed position on the robot's body, the problem is to determine parameters of the transformation $\mathbb{T}$ Eq. 1.2 describing the robot motion from the positions P_p to Q_p using the associated observations to these positions.*

1.2 Precision of the Localization

A precision of a localization technique can be influenced by the following issues, which are considered in this thesis.

Quality of feature sets

The projection of the feature sets Q to P is exact only if the features represent the same environment landmarks. However, the measurements are always influenced by a noise. Moreover sets can include measurements, so-called outliers. Outliers significantly affect the estimation of the transformation, see Section 7.1; so, a detection of the outliers and discarding them from the estimation process is another challenging problem that affects the precision of the localization.

Features extraction

The precision is also influenced by an extraction of the features from the input scans, some data adjustments can be made before estimation process to improve results. The process of feature extraction and data adjustment is a challenging problem itself and it is describe in Section 3.1 and Section 3.2.

Data association

Data association methods find for each features in the set P the best corresponding feature from the set Q. Such pairs are then used in a particular localization technique to estimate the robot pose. A brief introduction to the data association problem is given in Section 3.4.

Robot velocity

The perception sensors scans environment during the robot movements. The robot velocity can also influence the quality and precision of the measured observations. This is happened also for a laser rangefinder, because it does not make n measurements at one time instant but it makes measurements in a sequence. It means that n measurements are not captured from the same robot position as localization techniques suppose. Therefore, the robot localization using these measurements cannot be ever precise. The error caused by the robot velocity can be reduced if the speed of sensor measurements (e.g., laser rangefinder) is significantly higher than the robot velocity.

In this thesis, it is assumed the scan contains measurements taken at the same time instant for simplicity as it is assumed in literature using slow robots and fast laser rangefinders or cameras.

Localization vs navigation

The localization can be considered as a passive process, which estimates the mobile robot poses based on new measurements provided by the robot sensor system and the robot movements are controlled by an independent process. On the other hand, in the navigation, the estimation of the actual mobile robot pose is directly used to control the robot motion. Although an employment of the estimation in the robot control problem can influence the performance of the robot pose estimation itself, we assume an independent systems of robot control and localization in this thesis. This allows us to be focused on issues of the localization method and estimation technique itself.

Long corridors problem

When a robot operates in a specific environment, such as long corridors, the robot's laser rangefinder does not measure any obstacle in front of it. Moreover, the side walls of the corridors do not provide any significant features that can improve matching of two consecutive scans. Hence, the robot can be placed at different position along the corridor, while the laser scanner provides almost identical scans. Such a dataset is badly conditioned and the localization algorithms estimate the traveled distance badly.

Chapter 2

Considered Sensors and Robots

In this thesis, different types of wheeled robotic platforms are considered, that are equipped with various types of perception sensors. This chapter gives an overview of them and describes their significant properties that can influence the estimation of the robot pose. Particular properties are then used in the localization techniques to improve the estimation of the robot pose.

2.1 Differentially Driven Robot

This type of the robot is typically equipped with two wheels and a supporting point. The relative coordinate system is typically placed at the center C that is located on the wheel axis, see Fig. 2.1a. This construction is widespread in mobile robot applications due to ability to turn the robot around its center C but it can also rotate around a general center ICC. The angular robot velocity can be determined as

$$\omega = \frac{v_L}{R - \frac{L}{2}} = \frac{v_R}{R + \frac{L}{2}} = \frac{v_R - v_L}{L},$$

where the radius R is

$$R = \frac{L}{2}\frac{v_r + v_L}{v_r - v_L}$$

and the robot linear velocity is

$$v = \omega R = \frac{1}{2}(v_R + v_L).$$

The kinematic model according to the robot frame is

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix}. \tag{2.1}$$

This model is ideal, which means it does not consider a wheel slippage.

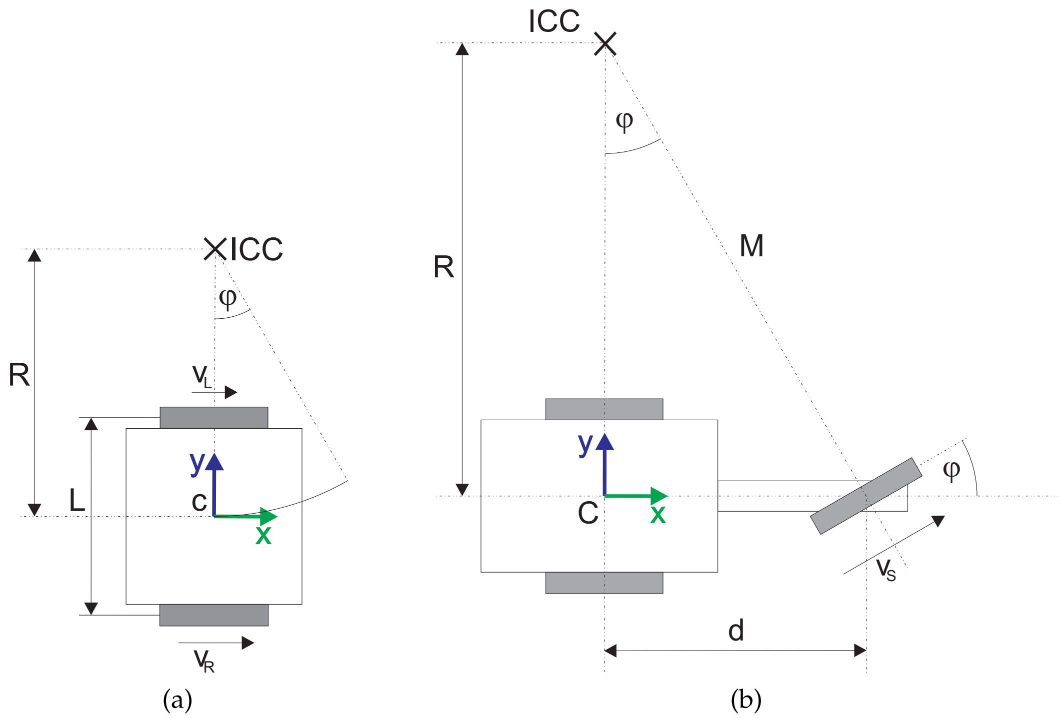


Figure 2.1: The models of robots with differential (a) and ackerman (b) drive.

2.2 Robot with Ackerman Drive

This robot type is also called car-like, see Fig. 2.1b. The robot turning radius depends on the angle φ of the steering wheel

$$R = \frac{d}{\tan(\varphi)}.$$

The angular velocity is determined as

$$\omega = \frac{v_s}{M} = \frac{v_s}{d}\sin(\varphi).$$

The ideal kinematic model according to the robot frame is

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\varphi) \\ \sin(\varphi) \\ \frac{\sin(\varphi)}{d} \end{bmatrix} \begin{bmatrix} v_s \end{bmatrix}. \tag{2.2}$$

2.3 Laser Rangefinder

A laser rangefinder scans its surroundings in a plane with angular interval $\langle\theta_{min}, \theta_{max}\rangle$ and provides distance measurements from the sensor to the detected obstacles. The distances are computed using a laser beam time of flight between transmission and reception of the reflected beam. If none obstacle occurs, the maximum waiting time interval runs over and the maximum distance is returned for such a case. Two methods of determining the distance to an obstacle are as follows.

- **Time of flight measurement** is suitable for measuring distances in hundred of meters due to less accuracy of this method that works as follows. A short light pulse is sent out and a time interval is measured until a reflected light pulse is received. A distance to an obstacle is calculated using an information about velocity of the light. The high velocity of light is a difficulty because a time error $\delta_e = 1$ ns causes a distance error 15 cm; so, precise time measurement is needed.

- **Phase shift method** is used for determination of shorter distances than in the previous case. The improvements of accuracy is done by sinusoidally modulation of an optical power of the laser beam. The phases of the transmitted and reflected beams are compared. The phase shift ω can be determined as $\omega = 2\pi t f_{mod}$, where t is the time of flight and f_{mod} is the modulation frequency. This shows that a higher modulation frequency can improve the distance resolution.

The laser rangefinders are produced by several companies and they differ in many parameters, such as maximal range, precision, resolution, etc. that can influence the localization of mobile robot. Therefore, the laser rangefinders used in this thesis are described in the following section.

2.3.1 Particular Considered Sensors

The considered laser rangefinder is the SICK LMS200 [53], see Fig. 2.2a. The standard device measures distances up to 80 m with a centimeter or millimeter accuracy. It scans environment in a plane with $180°$ range of view with the angular resolution $0.5°$. Beside the robust and heavy LMS200 device, a smaller SICK rangefinder exists, see Fig. 2.2b, which is more suitable for robotic applications with small robots.

(a) (b) (c)

Figure 2.2: The SICK laser rangefinder devices (a,b) [53] and Hokuyo sensor (c) [54].

In addition to the SICK sensors, Hokuyo laser rangefinders [54] are also quite popular in robotic community as the drivers for them are available in robotic systems like Player/Stage [49] or ROS [45], etc. The Hokuyo UTW-30LX (Fig. 2.2c) is small and lightweight and detects obstacles within a distant range from 0.1 to 30 meters. It scans environment in a plane with $270°$ field of view and angular resolution $0.25°$. The general accuracy of the provided distance measurements is worse comparing to the SICK devices.

2.4 Camera

A camera is a representative of a passive sensors that senses the robot environment and produces a picture from which important features, shapes of objects, materials properties and scene arrangement can be estimated. Detection of these environment properties depends on the type of image sensor that varies according to the sensor technology, the range of sensed spectral bands, image resolution and imaging geometry. In this thesis, images from the Malaga dataset [3] are used, which has been captured by the color camera AVT Marlin F-131C, see Fig.2.3 with a CCD sensor. The quality of captured images (i.e., the quality of input sets for localization methods) depends on the image resolution that is 1280×1024. Moreover, it is also influenced by the way how the environment is captured into the camera projection plan. This can be mathematically described by the pinhole camera model.

Figure 2.3: AVT Marlin F-131C camera [55].

2.4.1 Pinhole Camera Model

The pinhole camera model describes how a scene feature $F = [x, y, z]$, which is defined in the camera coordinate system, is projected onto a point $P = [u, v]$ in the image plane, see Fig. 2.4. Without considering any lens distortions, the projection is defined as

$$\begin{bmatrix} uw \\ vw \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad (2.3)$$

where w is the scale factor, f_x, f_y stand for the focal lengths in units of pixels and c_x, c_y are the coordinates of the image center. The main lens distortions are radial and tangential and they produce a displacement of the projected image. They can be corrected using Brown's distortion model

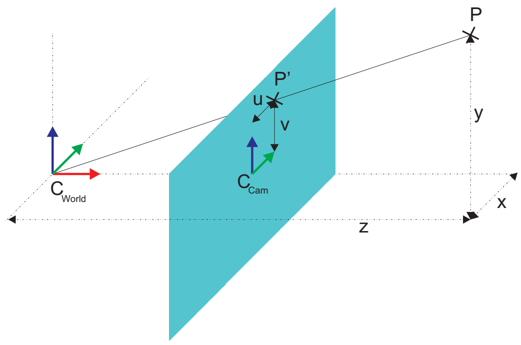


Figure 2.4: The camera pinhole model.

$$\begin{aligned} u' &= u + u_r + u_t \\ v' &= v + v_r + v_t, \end{aligned} \tag{2.4}$$

where

$$\begin{array}{rcl} u_r &=& u(k_1 r^2 + k_2 r^4 + \cdots) \\ v_r &=& v(k_1 r^2 + k_2 r^4 + \cdots) \\ u_t &=& (2p_1 uv + p_2(r^2 + 2u^2))(1 + p_3 r^2 + \cdots) \\ v_t &=& (2p_2 uv + p_1(r^2 + 2v^2))(1 + p_3 r^2 + \cdots) \end{array}$$

and $r = \sqrt{u^2 + v^2}$ and $[k_1 \ k_2 \ \cdots]$ are radial and $[p_1 \ p_2 \ \cdots]$ tangential distortion coefficients.

The camera pinhole model creates a $2D$ image features from $3D$ environment observations. When using a camera for estimation of the mobile robot pose in $\mathbb{R}^3$, the input features sets for localization techniques have to also be in $\mathbb{R}^3$; thus, the recovering of a feature depth has to be solved.

2.4.2 Recovering a Feature Depth

The position of the features in two directions can be directly read from the captured scene but the depth information has to be determined by an additional method. Although cameras that produce 3D images exist, they are expensive [50], and therefore, the depth is determined from a more images of the same scene in this thesis. A widespread solution of the full estimation of depth information is usage of two cameras in a stereo pair, which capture the same scene [11]. The third solution is “ego-motion” that refers to the problem of camera localization from a sequence of pictures from one camera [27]. In this thesis, the second approach used and the method is described in Section 3.3.

Chapter 3

Perception Process

The perception process consists of three parts. The first part is sensing of the environment. Then, important features, such as corners or blobs, are extracted and associated to corresponding pairs in the second part. Finally, the found feature pairs are used as input datasets in localization techniques. In this thesis, the laser rangefinder and color stereo camera are utilized as the perception sensors. The laser rangefinder produces a set of measured features, which can be used directly or only important features can be extracted. In this chapter, the *relevance filter* [20] extraction method is described. It reduces the number of measured features according to the relevance measurement. When using a camera, important features are found as salient objects by image processing detectors [22] and they are represented by descriptor, e.g., SIFT, SURF, RIFF, BRIEF, ORB etc. [43]. In Section 3.2, a brief introduction to feature detection is given and the selected *Speeded Up Robust Features* (SURF) [1] descriptor is discussed.

3.1 Detection of Features in Laser Rangefinder measurements

When using a laser rangefinder as a perception sensor, the amount of captured features can be reduced by using the relevance filter [20]. This filter removes the unnecessary features and only important features remain, see Fig. 3.1. It works as follows:

1. If the feature set contains more than three features, then the relevance $K(f_i)$ is determined for each feature f_i in the set as

$$K(f_i) = |f_{i-1}, f_i| + |f_i, f_{i+1}| - |f_{i-1}, f_{i+1}|,$$

   where $f_{i-1},\ f_{i+1}$ are neigbours of feature f_i, $i \in \langle 2, N-1 \rangle$. N is the size of the feature set. Otherwise, the filtration procedure is terminated.

2. The minimal $K(f_{min})$ is found over all relevances $K(f)$. If the found value is bigger than the chosen threshold, the feature f_{min} is discarded and the whole process is repeated.

3. If $K(f_{min})$ is smaller than the chosen threshold or the feature set contains less than four features, the algorithm terminates.

This method is suitable for feature sets that are captured in a structured environment because the features corresponding to the straight parts are discarded and only features corresponding to corners are preserved. The corners are more unambiguous than the features on the straight parts (e.g., walls), therefore this filtration improves the data association, see chapter 7.

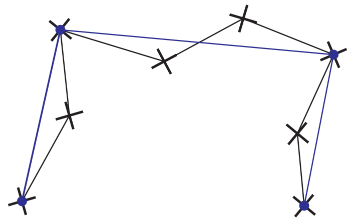


Figure 3.1: The original measured points (black) and the preserved points (blue) after relevance filtering.

3.2 Detection of Features in Camera Measurements

Feature detectors locate a feature in an image. The most of the existing detectors can be divided into corner or region detectors. The found feature is represented by its descriptor. The requested properties of an ideal descriptor are robustness to occlusions and background clutter and invariance to many kinds of variations, geometric and photometric transformations. The definition of these properties are given in the following section together with an introduction of basic functions used in image processing.

3.2.1 Basic Image Functions

Consider an image $I(p)$, where $p = [x,\ y]$ is a pixel. The differences of image are I_x and I_y with respect to x, y respectively. A Gaussian kernel with a local scale parameter σ is defined as

$$g(p,\sigma) = \left(\frac{1}{2\pi\sigma} e^{-\dfrac{p^T p}{2\sigma}} \right).$$

Scale Space

Detectors have to find features at different scales, because the search of correspondences often requires feature comparison in images that are captured at different scales. The linear scale space of image $I(p)$ is a serie of $L(p,\sigma)$, which is determined by smoothing an image $I(p)$ with a Gaussian $g(p,\sigma)$ using different scales σ, see Eq.3.1.

$$L(p,\sigma) = I(p) * g(p,\sigma) = \int I(p-q)g(q,\sigma)dq. \tag{3.1}$$

Harris Matrix

The Harris Matrix represents the gradient information, i.e., a matrix of partial derivatives. The eigenvalues of the Harris matrix determine if a point is a corner or not. The Harris matrix is defined as

$$A(p) = \begin{bmatrix} I_x(p) \\ I_y(p) \end{bmatrix} \begin{bmatrix} I_x(p) \\ I_y(p) \end{bmatrix}^T = \begin{bmatrix} I_x^2(p) & I_x(p)I_y(p) \\ I_y(p)I_x(p) & I_y^2(p) \end{bmatrix}. \tag{3.2}$$

Hessian Matrix

The Hessian matrix $H(p,\sigma)$ at the pixel p at the scale σ is defined as follows

$$H(p,\sigma) = \begin{bmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{bmatrix}, \tag{3.3}$$

where $L_{xx}(p,\sigma)$ are the second partial derivatives of $L(p,\sigma)$ with respect to x and $L_{yy}(p,\sigma)$, $L_{xy}(p,\sigma)$, $L_{yx}(p,\sigma)$ are defined in a similar way.

Laplacian Function

The Laplacian function is invariant to rotation in a given image $I(p)$. It is defined as

$$\triangle L(p,\sigma) = L_{xx}(p,\sigma) + L_{yy}(p,\sigma), \tag{3.4}$$

i.e., it is a trace of the Hessian matrix H.

Geometric Transformations

Each pixel p_1 can be mapped to its corresponding pixel p_2 by a geometric transformation. Different transformations as translation, reflection, rotation, skew, scale, etc. can be expressed by the function

$$p_2 = Ap_1 + b, \tag{3.5}$$

where A is the transformation matrix and b is the translation matrix.

3.2.2 Detectors

The one of the first detectors introduced in 1977 is Moravec's corner detector [31], which finds the local maximum of minimum intensity changes. Harris point that this detector is anisotropic, noisy and sensitive to edges; therefore, he proposed Harris corner detector [13], which is also used as a building block of modern detectors. This detector uses the Harris matrix and its eigenvalues to detect corners. However, it is not scale invariant; so, Mikolajczyk and Schmid [30] proposed new detector, which uses scale-adapted Harris measure or the determinant of the Hessian matrix to select the location and the Laplacian to select the scale. This method is robust, scale-invariant and it has a high *repeatability*. The repeatability expresses the reliability of a detector to find the same physical salient objects under different viewing conditions.

The scale invariant property is important for detectors to be used in localization problems. Lowe proposed the *Scale Invariant Feature Transform* (SIFT) [24] that approximates the Laplacian of Gaussians (LoG) by a Difference of Gaussians (DoG) filter. Using this approximation, the detector is faster than previous detectors. The Fast-Hessian detector, which is proposed by the authors of *Speeded-Up Feature Transform* descriptor [1], improves the detection speed even more by using the box filters instead of DoG. From the published comparison [22], the Hessian-based detectors are more stable and repeatable than the Harris-based. Moreover, when the determinant of the Hessian matrix is used instead of its trace (the Laplacian), it improves the feature detections. The Harris corner, SIFT and Fast-Hessian detectors are important in the evolution of detectors, therefore there are described more detaily in the following paragraphs.

Harris corner detector

The Harris corner detector [13] is based on the Harris matrix, see Section 3.2.1, which is computed for each pixel p in an image $I(p)$. The eigenvalues $\lambda_1, \ \lambda_2$ of this matrix determines three possible locations of the pixel.

- $\lambda_1 \approx 0$ & $\lambda_2 \approx 0$: the pixel is in an uniform intensity region;
- $\lambda_1 \approx 0$ & $\lambda_2 \gg 0$: the pixel is on an edge;
- $\lambda_1 > 0$ & $\lambda_2 > 0$: the pixel is a corner.

A computation of the matrix eigenvalues is computationally expensive. That is why the authors propose the Harris corner metric

$$m_h = \lambda_1\lambda_2 + \kappa(\lambda_1 + \lambda_2)^2 = det(A) - \kappa\, \mathrm{trace}(A)^2,$$

where κ is a tunable sensitivity parameter. Using this metric, the pixel location is determined as follows:

- if m_h is small, then the pixel is in an uniform intensity region;
- if $m_h < 0$, then the pixel is on an edge;
- if $m_h > 0$, then the pixel is a corner.

The detected features are invariant to the rotation, but they are not invariant to the scaling.

Scale Invariant Feature Transform (SIFT)

This method is a region detector [22] and it approximates the Laplacian of the Gaussians (LoG) by the Difference of the Gaussian (DoG). This idea is proposed by Crowley and Parker in [8], where they select the scale-space extrema in a serie of DoG images by a convolution of the image with the DoG functions. Based on this method, Lowe [24] propose the SIFT detector designed to handle the scale invariant problem. The detection consists of three steps:

1. *The scale-space extrema detection* is the same as the Crowley and Parker method.

2. *The keypoint localization* discards the low contrast candidate points, that are sensitive to noise. This is done using the function

$$L(z_e) = L(z) + \frac{1}{2}\frac{\delta^T L}{\delta z} z_e,$$

   where

$$z_e = -\left(\frac{\delta^2 L}{\delta z^2}\right)^{-1} \frac{\delta L}{\delta z}$$

   and $z = [x \ y \ \sigma]$. The points with $L(z_e) > 0.03$ are preserved. Then, points along an edge are discarded too. These points are detected using the ration

$$r = \frac{lambda_1}{\lambda_2} > 10,$$

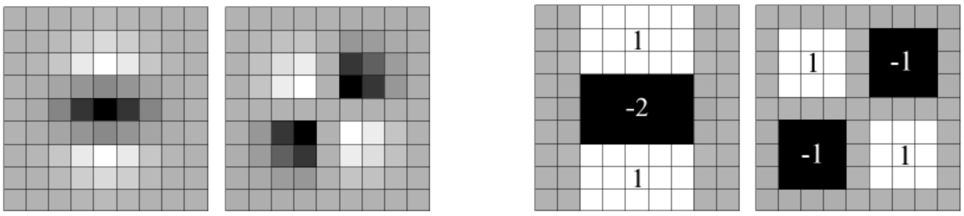
   where $\lambda_1, \ \lambda_2$ are eigenvalues of the Hessian matrix. This two filtering processes improve the stability of features matching methods.

3. *The orientation is assigned* to the preserved keypoints with these sub-steps.

   - The magnitude and orientation of all points in the circular region of the found extreme are calculated.
   - The magnitude is smoothed by a Gaussian window.
   - The histogram with 36 bins is obtained by an accumulation of these magnitudes. The magnitude is assigned to the bin using the corresponding orientation.
   - The orientation corresponding to the maximum of histogram and the orientation corresponding to the local maximum, whose value is above $80\,\%$ of the global maximum, are selected.
   - For each selected orientation, a correction is made. The parabola is fitted to the histogram values of the selected orientation and its two neigbours. Then, the original orientation is replaced by the peak of the parabola.
   - The keypoint is created in the region, which orientation is the same as the selected one.

Fast-Hessian detector

The Fast-Hessian detector proposed by authors of the SURF descriptor [1] is based on the Hessian matrix because it has a good accuracy. The Gaussians used in the definition of the Hessian matrix, see Eq. 3.3, are optimal for scale-space analysis, but they have to be discretised and cropped, see Fig. 3.2a. Authors proposed a novel approximation for the Hessian matrix using box filters, see Fig. 3.2b, which approximates the second order Gaussian derivatives. Their advantage is that they can be determined at a very low computational cost using integral images. Therefore, the calculation time is independent on the filter size.



(a) (b)

Figure 3.2: An example of the discretised and cropped Gaussian for the second order derivative $L_{yy}(p,\sigma)$, $L_{xy}(p,\sigma)$ for $\sigma = 1.2$ (a) and the proposed approximation D_{yy}, D_{xy} (b), where the gray regions are equal to zero [1].

The algorithm works as follows:

1. *Integral image* $I_{\sum}(p)$ at a pixel $p = [x,\ y]$ is determined by equation

$$I_{\sum}(p) = \sum_{i=0}^{i\leq x} \sum_{j=0}^{j\leq y} I(i,j).$$

   The sum of the intensities $\sum$ over a rectangular area in a image $I(p)$ can be determined by computed integral images as

$$\sum = I_{\sum}(A) - I_{\sum}(B) - I_{\sum}(C) + I_{\sum}(D),$$

   where $A,\ B,\ C,\ D$ are corners of an area, see Fig. 3.3. This computation of $\sum$ is independent on area size, which positively influence the computing speed.

2. *The determinant of Hessian matrix is approximated* by the equation

$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2,$$

   where D_{xx}, D_{yy}, D_{xy} are box filters, w is the relative weight of the filter response because the Gaussian kernels and approximated kernels have a different energy. The weight w depends on the scale σ, but authors observed

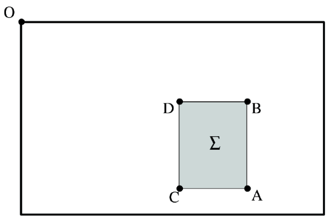


Figure 3.3: The calculation of the sum of the intensities inside the rectangular region [1].

that the impact is insignificant; so, they proposed to use $w = 0.9$ that corresponds to the box filter with $\sigma = 1.2$. The approximated determinant represents the blob response in the image pixel p, it is stored and used to detected important features in the next step.

3. The Fast-Hessian detector is a *new approach to the scale-space*, see Eq. 3.1. The box filters of any size can be applied directly on the original image at exactly the same speed, that is in contrast with the original approach, where the same filter have to by iteratively apply to the output of the previous filtered layer. Therefore, the scale-space is determined by up-scaling the filter size rather than iteratively reducing the image size [1], see Fig. 3.4. This approach improves the computational requirements, moreover it preserve the high-frequency image components, because an image is not sampled.

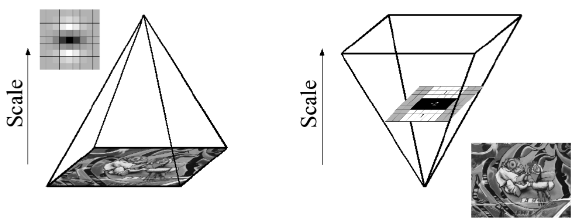


Figure 3.4: The origin "pyramid" approach (left) for scale-space determination, where an image is downsampled, and the proposed approach, where the size of box filter is changed [1].

4. *The important features* are found using the non-maximum suppression method introduced by Neubeck and Van Gool [34] and an interpolation of the maxima of Hessian matrix by method proposed by Brown and Lowe [4].

3.2.3 Descriptors

The features extracted by a detector are represented using a descriptor of the feature surroundings. The size and properties of the descriptor influence the behaviour of other methods, e.g., data association. Many types of descriptors are proposed [22] and one of the first descriptors use the local derivatives [19]. Schmidt and Mohr [35] extend the local derivatives as the local gray-value invariants for an image retrieval. These descriptors are outperformed by the SIFT descriptor proposed by Lowe [24]. The SIFT descriptor computes a 3D histogram of local oriented gradients around the detected feature and creates a vector with 128 values.

Various methods based on the SIFT descriptor exist. For example Ke and Sukthankar [18] apply a principal component analysis (PCA) on the gradient image around the detected feature to simplify the SIFT descriptor. They create the PCA-SIFT descriptor with 36 values, which leads to faster processing of the found features in other algorithms, e.g., features matching. But it is less distinctive than the original SIFT and the PCA slows down the determination of the feature descriptor. Lazebnik [21] proposed the RIFT (rotation invariant feature transform) descriptor to improve the rotation invariance of the descriptor. The disadvantage of SIFT is the high dimensionality of the descriptor that slows down the matching process; thus, authors of SURF [1] proposed a descriptor with 64 values that is based on the idea of SIFT. The SURF descriptor describes the intensity content within the interest point neighborhood too, but it uses a distribution of the first order Haar wavelet responses in x and y directions rather than the gradients. In this thesis, the SURF descriptor is used for description of visual features in camera images. It is based on the SIFT descriptor, therefore a more detailed description of the SURF and SIFT descriptors are presented in the following paragraphs.

Scale Invariant Feature Transform (SIFT)

The SIFT descriptor [24] is created by the first gradient magnitude and orientation in the neighbourhood of the detected feature z_k, which have the scale σ_k and rotation Θ_k. The descriptor contains 16 orientation subhistograms (see Fig. 3.5), where each subhistogram consists of 8 bins; thus, SIFT descriptor contains 128 values for each detected feature z_k. The determination of the descriptor is as fol-

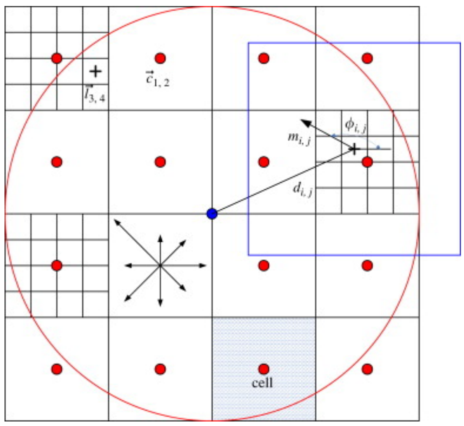


Figure 3.5: The SIFT descriptor [22].

lows:

1. The center τ_{ij} is generated for each cell, see Fig. 3.5.

2. The locations l_{ij} are generated for each cell.

3. The orientation bin φ with 8 values for the histogram is generated.

4. The centers and locations are transformed using the rotation matrix $R(\Theta_k)$ and translation z_k.

5. The gradient magnitude m_{ij} and the orientation ϕ_{ij} of the location l_{ij} are sampled around the feature z_k in the scale σ_k.

6. The x-coordinate weighting vector $\vec{w}^x_{ij} \in R^{16}$ of the location l_{ij} is determined according to the equation

$$\vec{w}^x_{ij} = [max(1 - |\tau_{mn_x} - l_{ij_x}|/4),\ 0],$$

where $m, n \in \langle 1, 4\rangle$. The y-coordinate weighting vector $\vec{w}^y_{ij} \in R^{16}$ is determined by the same equation, but y-coordinates τ_{mn_y} and l_{ij_y} are used. The local weighting vector is determined as

$$\vec{w}^l_{ij} = [w^x_{ij}(\vec{m,n}) \times w^y_{ij}(\vec{m,n})].$$

This step is repeated 8 times, because each cell have 8 orientations.

7. The orientation weighting vector $\vec{w}^0_{ij} \in R^8$ is determined as

$$\vec{w}^0_{ij} = [max(1 - 4|\vec{\theta}|/\pi,\ 0],$$

where

$$\vec{\theta} = \mathrm{mod}(\phi_{ij} - \Theta_k - \varphi + \pi, 2\pi) - \pi.$$

The vector $\vec{w}^0_{ij}$ is determined for each cell.

8. The 128 dimensional histogram is computed as

$$h_k = \sum^{1} 6_{ij=1}[\vec{w}^l_{ij}(k) \cdot \vec{w}^0_{ij}(k)]_{k \in \langle 1,128\rangle} \cdot w^G_{ij} m_{ij},$$

where w^G_{ij} is the Gaussian weighting factor.

9. The histogram is normalized.

Speeded-Up Robust Feature (SURF)

The SURF descriptor [1] describes the distribution of the intensity content as SIFT, but it is built on the distribution of the first order Haar wavelet response, which reduces the computation time and increase the robustness. The descriptor is created in the following steps.

1. *The orientation* of the detected feature f is identified as follows. First, the Haar wavelet responses in x and y directions is determined within a circular area with the radius $r = 6\sigma$, where σ is the scale at which the feature f is detected, and the center of the area is at the detected feature f. These wavelet responses are weighted with a Gaussian with the variance 2σ. Then, the local orientation vector is determined by summing the horizontal and vertical responses within a sliding window, see Fig. 3.6. The orientation of the detected features is the longest of the local orientations.

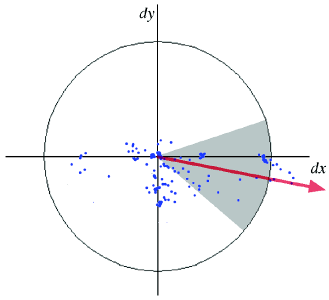


Figure 3.6: A gray sliding window used to detect the dominant orientation of the Haar wavelet responses [1].

2. *The descriptor* is determined using the sums of Haar wavelet responses too. First, the square areas centered at the feature f are created with orientation determined in the previous step. The window is explicitly used with the size 20σ and it is divided into 16 square sub-regions, which preserve important spatial information. In the next step, the Haar wavelet responses in the horizontal direction d_x and in the vertical direction d_y are computed at 5×5 regularly spaced points at each sub-region. These responses are weighted using the Gaussian with the variance 3.3σ (σ is the scale at which the feature f is detected) to increase the robustness while geometric transformations or features errors occur. The sub-region descriptor is determined as

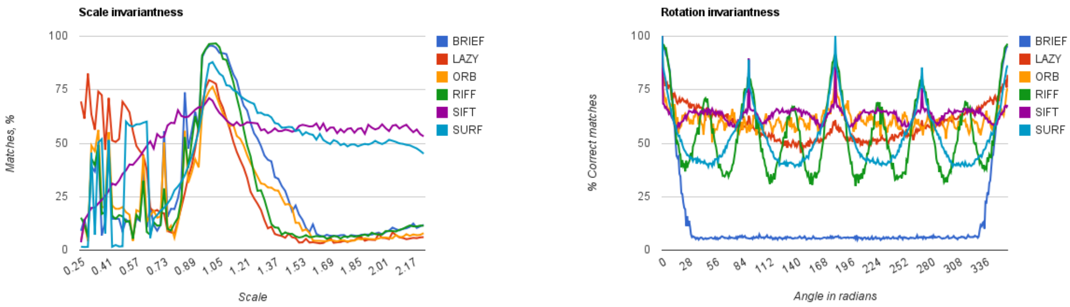
$$v = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|].$$

Finally, the SURF 64 dimensional descriptor is created by determining descriptor v for each sub-region.

3.2.4 Selection of Feature Detection Method

Authors of SURF [1] proves that their proposed Fast-Hessian detector is faster than others existing detectors at that time. An independent comparison of descriptors behaviour is available at [43]. The described SIFT and SURF descriptors and others are compared in a scale and rotation invariant test, see Fig. 3.7. The metric is the percentage of correct matches between the features from the source image and the features from the transformed source image.

The SURF sensitivity to odd multiples of $\pi/4$ can be seen in Fig. 3.7b. It is caused by a discretisation of the Gaussian and it is a weak point for all Hessian-based detectors. The SURF and SIFT descriptors are the best from the tested descriptors in the scale-invariant test, see Fig. 3.7a. Based on these evaluations, the SURF detection (detector and descriptor) is used in this thesis.



(a) (b)

Figure 3.7: Results of the scale (a) and rotation (b) invariant test for various descriptors [43].

3.3 Depth Estimation from Stereo Images

The stereo images are taken by a general stereoscopic camera system that is shown in Fig. 3.8. Each camera is defined by its position $\vec{C}$, optical axis $\vec{A}$ and image plane created by the vectors $\vec{H}, \vec{V}$. If the cameras are placed at the same y-coordinate and their axes are parallel, the simplified model can be used, see Fig. 3.9. This configuration is used in this thesis. The distance B between the cam-

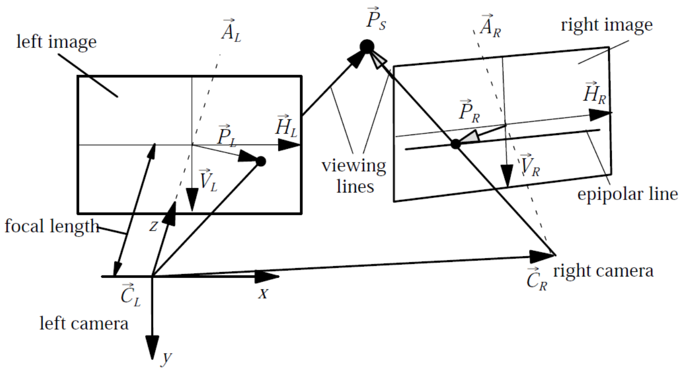


Figure 3.8: The model of the stereoscopic camera system [11].

eras can be expressed as

$$B = B_1 + B_2 = D\tan(\varphi_1) + D\tan(\varphi_2);$$

thus, the depth D of a detected feature can be determined as

$$D = \frac{B}{\tan(\varphi_1) + \tan(\varphi_2)}.$$

Using the trigonometry functions and $\varphi_p = \frac{\varphi_0}{2}$, it holds that

$$\begin{aligned} \tan(\varphi_1) &= \frac{2x_1 \tan(\varphi_p)}{x_0}, \\ \tan(\varphi_2) &= \frac{-2x_2 \tan(\varphi_p)}{x_0}. \end{aligned}$$

Finally, the depth D is

$$D = \frac{Bx_0}{2\tan(\varphi_p)(x_1 - x_2)}. \tag{3.6}$$

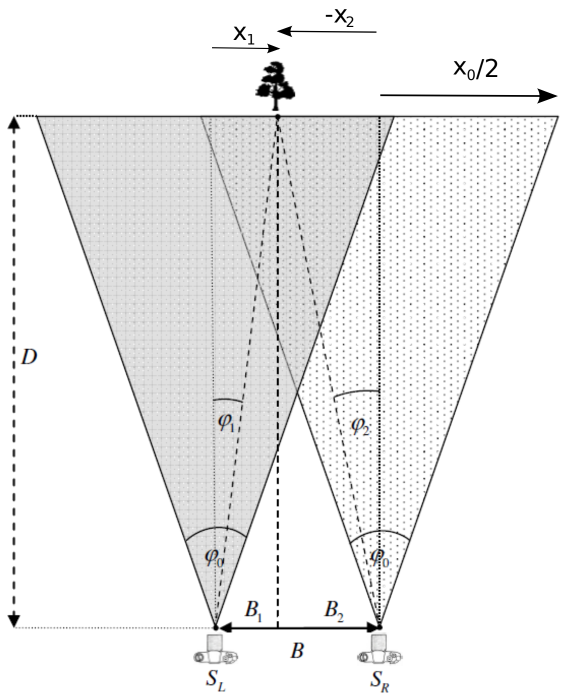


Figure 3.9: The simplified model of the stereoscopic camera system [17].

Depth information is obtained by a triangulation of corresponding image features with known stereoscopic camera parameters. Therefore, the coordinate difference between the corresponding image points, called disparity, has to be estimated.

3.4 Data Association

Methods of data association can be divided into two groups according to the amount of features that are requested to associate [6]. The **individual measurement data association** methods associate a single captured feature from one set with the appropriate feature from the other sets. The features in these sets are measured independently. The **batch data association** methods associate a batch of features that are sensed at one time instant. This situation arises when the used sensor takes a measurement of the environment as a single snap or the scan frequency is faster than the robot's dynamics. The batch measurements are given, for example, from a laser rangefinder or a camera, i.e., the types of sensors used in this thesis.

The batch association has some special properties in contrast to the individual association. The first is the *greedy mutual exclusion*, it means that no two features can be associated with the same feature within a given batch. The second property is the possibility that a feature pair can be compared within the entire batch.

The data association is an independent research area and many different methods exist [5]. This section is focused only on the *nearest neighbour association* method and the additional filtering procedure that is proposed here.

3.4.1 New Proposed Association Method

During the research of LEL performance, it was observed that the LEL significantly depends on the quality of the corresponding pairs, see Section 8.3 for details. Also, it was proved that the standard *nearest neighbour* method has provided insufficient precision of LEL, see Section 8.4. Therefore, the new extension of *nearest neighbour* method is proposed here.

Having two features $A = [a_1,\ a_2,\ \ldots,\ a_N]$ and $B = [b_1,\ b_2\ \ldots,\ b_M]$ for which each b_j is associated to the particular feature a_i if

1. their descriptors are identical. Such a case cannot ever exist because the images are always influenced by a noise. Therefore, the similar descriptors are evaluated using the Euclidean distance between feature descriptors;

2. they have the same sign of the Laplacian, as proposed in [1];

3. and a feature b_k with the similar descriptor does not exist to guarantee the greedy mutual exclusion.

Figure 3.10: The corresponding pairs between left and right camera, the red pairs are the outlying pairs.

The proposed extension consists of the following steps.

1. The smaller set is chosen as a reference, the features from the bigger set are assigned to the reference features. Without loss of generality, it is supposed that $N > M$ in the next steps of the association procedure.

2. For each feature a_i from the set A, the distances to all features from the set B are determined. Thus, the matrix of the distances D with size $N \times M$ is obtained.

3. The matrix D is normalized.

4. The similarity coefficient $e \in \langle 0, 1 \rangle$ is chosen. The features with the Euclidean distance equal or smaller than a given e are considered as corresponding pairs.

5. For each feature a_i, the amount of the corresponding candidates are determined using the coefficient e. These amounts are stored in the matrix C with the size $N \times 1$.

6. The amount $\sum_a$ of the features a_i with only one corresponding candidate is computed.

7. The new similarity coefficient e is chosen to maximize the $\sum_a$ and the procedure (from the step 4) is repeated. The procedure ends when the maximum of $\sum_a$ is found for the similarity coefficient e_{max}.

Finally, the corresponding pairs $[a_i,\ b_j]$ are determined using the coefficient e_{max}. The feature b_j is associated with the feature a_i if the distance $D(i, j)$ is equal or smaller than e_{max} and any other distance $D(i, (1 \ldots M)/j)$ is not equal or smaller than e_{max}.

The quality of the corresponding pairs can be improved by using the second property of the batch data association; so, comparing all found pairs together. This allows to throw away corresponding pairs, which are outlying. An outlying pair is determined using differences of their positions. In Fig. 3.10, an example of the outlying correspondence pairs can be seen (in red). This new proposed approach is compared with the standard *nearest neigbour* method in Section 8.4, where the influence to the mobile robot localization is verified as well.

Chapter 4

Least Entropy-Like Localization Technique

The Least Entropy-Like (LEL) [7] estimator is designed to find parameters of a transformation between two different input sets with robust behaviour for sets corrupted by outliers. The idea is based on rewarding the presence of majority low relative errors and penalizing a minority of large ones, i.e., finding such a cost function that is able to "globally" measure the residual dispersion and rewards evenly distributed residuals. These requirements formulated by the authors request properties of the cost function that can be fulfilled by the definition of the entropy. The formulated cost function has the same mathematical properties as the entropy, but the proposed method does not exploit the stochastic or information theoretic meaning of the entropy [16]. Therefore, the authors named the technique *Entropy-like*.

4.1 The Entropy-Like Estimator

The LEL estimator is deduced using a standard normalized discrete entropy function, which is defined as

$$H = -\frac{1}{N}\sum_{i=1}^{N} \lambda_i \log \lambda_i .$$

Regarding possible errors in the input data, the entropy H has an advantage that datasets, including few states with high probability, have a much lower entropy than datasets including states with approximately same probability. Motivated by this fact, the LEL estimator is designed by Eq. 4.1 to estimate such a transformation $\mathbb{T}$ that maximizes probability of function λ_i, which minimizes the entropy function.

$$\mathbb{T}_{LEL} = \operatorname{argmin}_{\mathbb{T}} H \tag{4.1}$$

For finding the transformation $\mathbb{T}$, the function λ can be defined as

$$\lambda_{ij} = \frac{r_{ij}^2}{D},$$

where

$$D = ||r||^2 = \sum_{i=1}^{N} \sum_{j=1}^{M} r_{ij}^2$$

is the squared 2-norm of the estimation error, M is the space dimensions and N is the number of points in the given datasets. The residuals

$$r = Q - R_{ini}P + T_{ini}$$

are computed using the initial estimations R_{ini} and T_{ini}. The entropy-like function used in the LEL estimator can be compute as

$$H = -\tfrac{1}{log(NM)} \textstyle\sum_{i=1}^{N} \sum_{j=1}^{M} \lambda_{ij} log\lambda_{ij} \quad if\ D \neq 0. \tag{4.2}$$

The regular function H is not define if $D = 0$ and the authors of [7] defined it as

$$H = 0 \quad for\ D = 0. \tag{4.3}$$

The new entropy-like function H defined by Eq. 4.2 and Eq. 4.3 has all properties as the standard normalized entropy function. The properties are

$$\begin{aligned} H &\in [0, 1] \\ H &= 0 \text{ iff} \begin{cases} r_i = 0 \,\forall\, i \in [1, N] \\ or \\ \exists!\ i^* :\ r_{i^*} \neq 0 \text{ and } r_i = 0 \,\forall\, i \neq i^* \end{cases} \\ H &= 1 \text{ iff } r_i^2 = r_j^2 \neq 0 \,\forall\, i, j \in [1, N]. \end{aligned}$$

The relative squared residual λ_{ij} has the following properties

$$\lambda_{ij} \in [0, 1],$$
$$\sum_{i=1}^{N} \sum_{j=1}^{M} \lambda_{ij} = 1.$$

4.2 Initialization

The proposed entropy-like function is nonlinear with possible several local minima. Moreover, it has been shown that the existence of the global minimum cannot be guaranteed. Therefore, the performance of the LEL estimator depends on the initial estimation of the transformation. The authors of LEL use the Horn algorithm due to its low computational requirements. Here, it should be noted that when using datasets from a laser rangefinder, Horn method is not a proper function, see discussion in Section 4.2.1.

4.2.1 Horn Algorithm

Horn algorithm [14] estimates the transformation $\mathbb{T}$ directly from datasets and does not iterate, which is its main advantage. It is defined for three dimensional space but modified algorithm for two dimensional space also exists. Here, the origin version is described. The main idea is that the input datasets P and Q are referred by their centroids

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i,\ \bar{Q} = \frac{1}{N}\sum_{i=1}^{N} Q_i.$$

The algorithm minimizes the mean square objective function

$$E(\omega, T) = \frac{1}{N}\sum_{i=1}^{N} \|RQ_i + T - P_i\|^2,$$

where N is the number of points in each dataset. The transformation is

$$\bar{P} = sR\bar{Q} + T,$$

where s is changing of a scale. The rotation matrix R is determined using the following steps.

1. The cross-covariance matrix $\sum_{PQ}$ is given by

$$\sum_{PQ} = \frac{1}{N}\sum_{i=1}^{N}\left[(Q_i - \bar{Q})(P_i - \bar{P})^T\right].$$

2. A matrix A is constructed as

$$A = \sum_{PQ} - \sum_{PQ}^{T}.$$

3. The column vector $\Delta = [A_{23}A_{31}A_{12}]^T$ is formed using cyclic components of the matrix A.

4. The symmetric matrix N is formulated as

$$N = \begin{bmatrix} \mathrm{trace}(\sum_{PQ}) & \Delta^T \\ \Delta & \sum_{PQ} + \sum_{PQ}^T - \mathrm{trace}(\sum_{PQ})I_3 \end{bmatrix},$$

where I_3 is the identity matrix with the size three.

5. An eigenvector $\bar{q} = \begin{bmatrix} q_0 & q1 & q2 & q3 \end{bmatrix}$, which corresponds to the maximum eigenvalue of the matrix N is selected. Horn [14] proves that this vector is a quaternion, an another expression of the rotation.

The scale is then determined as

$$s = \left(\frac{\sum_{i=1}^{n} \parallel \bar{Q}_i \parallel}{\sum_{i=1}^{n} \parallel \bar{P}_i \parallel}\right)^{\frac{1}{2}} \tag{4.4}$$

and the translation is

$$T = \bar{P} - sR\bar{Q}. \tag{4.5}$$

Data sensitivity of Horn algorithm

Horn method is sensitive to the input data as it is shown in the following example, which is ilustrated in Fig. 4.1. Let the robot be placed at the position P_p and the laser rangefinder make the scan P of the robot surroundings. Then, the robot rotates around its center about a small angle and a new scan Q is captured. It can be observed that the robot position is the same but centroids of the scans are significantly different, therefore Horn method estimates the translation incorrectly. This model situation is an excessive example and the behaviour of Horn algorithm can be improved by carefully preprocessing of the input datasets but it costs the computing time and the method becomes slower.

Figure 4.1: The robot rotates from P to Q, the centroids of captured datasets are different while the positions of the robot are identical.

4.3 Optimization

The authors of the LEL method use the **Levenberg-Marquardt** [28] optimization algorithm (LMA) to find a minimum of nonlinear function H over a space of parameters defined by $\mathbb{T}$. However, the LMA finds only a local minimum, not the global. The LMA interpolates between the *Gauss-Newton method* and *Steepest Descent*, which estimates the new parameter vector as

$$\mathbb{T}_{i+1} = \mathbb{T}_i - \mu \nabla H(\mathbb{T}).$$

The convergence of the *Steepest Descent* can take a long time for complex functions because a small constant steps are used to correct detection of minima. This behavior can be improved using the second order information. The process of the estimation of new parameter $\mathbb{T} = \mathbb{T}_0 + \delta$ is an iterative procedure and it consists of the following steps.

1. The function $H(\mathbb{T} + \delta)$ is approximated by its linearization

   $$H(\mathbb{T} + \delta) = H(\mathbb{T}) + J\delta,$$

   where J is the Jacobian matrix.

2. The parameter vector $\mathbb{T} + \delta$ is estimated to minimize the function $H(\mathbb{T})$; so, the ideal state is

   $$H(\mathbb{T}) + J\delta = 0.$$

3. Multiplying the above equation by J^T it can be rewritten as

$$-\mathbb{H}\delta = J^T H(\mathbb{T}), \tag{4.6}$$

where $\mathbb{H} = J^T J$ stands for approximation of the Hessian matrix.

4. The new parameter vector can be estimated as

$$\mathbb{T}_{i+1} = \mathbb{T}_i - \mathbb{H}^{-1} d,$$

where $d = J^T H(\mathbb{T}_i)$ is the average error gradient. Using this definition, the *Steepest Descent* function can be rewritten as

$$\mathbb{T}_{i+1} = \mathbb{T}_i - \mu d.$$

5. Levenberg's contribution is "blending" these two equations, the "damped" attribute ν is added to Eq. 4.6

$$-(\mathbb{H} + \nu I)\delta = J^T H(\mathbb{T}). \tag{4.7}$$

The positive damping vector ν is adjusted at each iteration. The function H is estimated for the proposed parameter vector $\mathbb{T}$. If the function value is decreased, then the step is accepted and ν is decreased and the Levenberg's method described by Eq. 4.7 gets closer to the *Gauss–Newton* algorithm. If the function value is increased, the step is retracted and ν is increased getting the Levenberg's method closer to the *Steepest Descent*.

6. The proposed method has a disadvantage for large values of ν, because the method is closer to the *Steepest Descent* and the convergence can be slow in the direction of small gradient. Marquardt proposed improvements based on the Hessian matrix, where larger movements are made along directions of the smaller gradient. Therefore, Marquardt replaced the identity matrix I with the diagonal matrix consisting of the diagonal elements of $\mathbb{H}$, proposing the final Levenberg-Marquardt optimization method

$$-(\mathbb{H} + \nu \operatorname{diag}[\mathbb{H}])\delta = J^T H(\mathbb{T}). \tag{4.8}$$

The Levenberg-Marquardt optimization method described by Eq. 4.8 estimates the parameter vector $\mathbb{T}$ independently, but from the robot's kinematics model (Section 2), it can be seen that the transformation parameters are not independent. Therefore, an idea to improve the robot pose estimation is to include the robot's kinematic constrains to the optimization method. In Chapter 8, several experiments are made and it is observed that the influence of the parameters independence is insignificant.

4.4 Parametrization of the 3D Rotation Matrix

When solving the nonlinear optimization problem, the used parametrization of unknown variables has to be *fair*. It means that such a parametrization does not cause more numerical sensitivity that one involved in the problem itself [15]. The straightforward representation of the rotation matrix are Euler angles. They are not fair, because they cause a nonlinear characteristic to the Jacobians in the optimization process. Moreover they could be numerically unstable. Another parametrization are quaternions, which are fair in this sense [15]. The quaternion is defined as

$$\vec{q} = q_0 + q_1 i + q_2 j + q_3 k,$$

where q_0 is the real part and q_1, q_2, q_3 are the imaginary parts. It holds that $|\vec{q}| = 1$ if the quaternion $\vec{q}$ represents rotation. The rotation matrix can be then defined as

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}.$$

The quaternion is defined by four elements, but it has only 3 DOF and this is problem when using the unconstrained nonlinear optimization technique like the Levenberg-Marquardt method. The naïve solution is to optimize only three imaginary parts and then the real part can be determined as

$$q_0 = \pm\sqrt{(1 - q_1^2 - q_2^2 - q_3^2)}.$$

During the optimization, it must be assured that the radicand is always positive, which cannot be guaranteed during Levenberg-Marquardt method; therefore, this approach is not suitable. The authors of [36] propose a robust solution, where the quaternion is transformed to a three dimensional vector. The idea of the proposed approach is based on the fact that all unit quaternions lie on the unit sphere in $\mathbb{R}^4$ and for such a quaternion there always exists the tangential hyperplane Ψ. The initial estimation of the rotation $\vec{h_0}$ and the desired estimation $\vec{h_z}$ lie on the great circle[1] of the sphere. The goal is to describe the distance of $\vec{h_z}$ from $\vec{h_0}$ and the direction on the great circle using only three independent parameters. Such a description can be made by a vector $\vec{v}$ that lies on the hyperplane Ψ because the hyperplane is a subspace of $\mathbb{R}^4$; thus, vectors in this plane can be represented in $\mathbb{R}^3$ with the origin at $\vec{h_0}$.

4.4.1 Base of the Tangential Hyperplane

The hyperplane Ψ tangential to $\vec{h_0}$ is defined by all points $\vec{x} \in \mathbb{R}^4$ satisfying

$$\vec{h_0}^T(\vec{x} - \vec{x_k}) = 0,$$

[1]The great circle is the intersection of the sphere and a plane, which passes through the center point of the sphere.

where $\vec{x_k}$ is a point lying in the hyperplane. When choosing $\vec{x_k} = \vec{h_0}$, the equation can be rewritten as

$$\vec{h_0}^T \vec{x} = 1.$$

The hyperplane in the normal form is defined as

$$n_1x_1 + n_2x_2 + n_3x_3 + n_4x_4 - 1 = 0,$$

where $\vec{h_0} = (n_1, n_2, n_3, n_4)$. This hyperplane can be defined by the base

$$B = [b_1,\ b_2,\ b_3,\ b_4].$$

This base is determined as follows. The vector b_1 is created as parallel to $\vec{h_0}$ with the unit size. The other base vectors are created using Gram-Schmidt method [46], they are orthogonal to each other and they have unit size; so, the base B is orthogonal too. The estimated vector $\vec{v} = [0,\ \mu,\ \nu,\ \sigma]$ has always the first coordinate equal to zero. The vector $\vec{v}_0$ corresponding to the initial quaternion $\vec{h_0}$ is

$$\vec{v}_0 = [0,\ 0,\ 0,\ 0].$$

4.4.2 Computing the Resulting Quaternion

The estimated vector $\vec{v}$ has to be normalized

$$\vec{v_N} = \frac{\vec{v}}{|\vec{v}|}.$$

Then, the resulting quaternion $\vec{h_z}$ is determined using $\vec{h_0}$ and $\vec{v_N}$. The quaternion

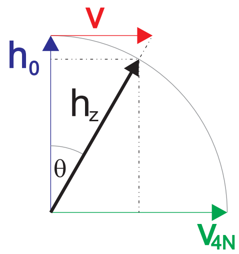


Figure 4.2: Resulting quaternion.

$\vec{h_z}$ lies on the "great circle" that is defined by the intersection of the sphere with the 2D plane defined by $\vec{h_0}$ and $\vec{v_N}$. Such a quaternion can be computed as

$$\vec{x} = \mu\vec{h_0} + \nu\vec{v_N}.$$

The quaternion $\vec{v}$ defines the direction of $\vec{h_z}$ and the distance from $\vec{h_0}$, see Fig. 4.2. Thus, the resulting quaternion is determined as

$$\vec{h_z} = \cos(\Theta)\vec{h_0} + \sin(\Theta)\vec{v_N}, \tag{4.9}$$

where $\Theta = \mathrm{atan}(\vec{v})$.

This approach has the following advantages.

1. The unit-norm constrain is avoided.
2. The three variables of vector $\vec{v}$ can be changed freely by an optimization algorithm.
3. The resulting quaternion has always unit norm.

In Section 8, this approach is compared with the rotation representation by Euler angles.

Chapter 5

Other Localization Techniques

Three main approaches for estimating transformation parameters $\mathbb{T}$ between two robot poses using scans of its surrounding exist. The first is known as the **scan-to-scan matching** method in which the transformation is estimated using only the actual scan and the previous one. In contrast, the **scan-to-map matching** method uses the actual scan and the current map of the robot operational environment. This map can be pre-prepared or the robot can build it simultaneously while localizes itself according to this map, which is a problem known as the **simultaneous localization and mapping (SLAM)**. This chapter describes the widespread algorithms for the matching problems based on the ICP algorithm and its extensions, RANSAC algorithm and gives a brief introduction to SLAM approaches.

5.1 Matching Problems

The *scan-to-scan matching* method uses only two following scans for estimation of the robot transformation $\mathbb{T}$, where the estimated position differs from the true robot transformation with an error δ at each such estimation. So, if the robot is localized using the *scan-to-scan* method, the errors are cumulated along the whole trajectory, and therefore, the difference between estimated and ground truth pose is bigger at each step. The idea behind the *scan-to-map matching* method is to find the transformation $\mathbb{T}$ using the actual scan and a map of the robot environment. The error between estimated and true pose still exists but it is not cumulated due to the map as reference. The total difference between the global and ground truth pose is smaller than using *scan-to-scan* matching method.

5.1.1 Iterative Closest Point

The *Iterative closest point* (ICP) procedure is a commonly used for scan-to-scan matching [2]. The main idea is based on minimizing the square error

$$E(\varphi, T) = \sum_{i=1}^{k} |RQ'_i + T - P_i|^2,$$

where Q' is the corresponding set to the input set Q. The algorithm works as follows.

1. A given input estimation of matrices R_{ini} and T_{ini} are used to determine a new dataset of the initially transformed measurements

$$Q_1 = R_{ini}Q + T_{ini}.$$

2. The distances between each point from the set P and all points from Q_1 are calculated. The pair with the smallest distances is found and the points of the pair are marked as the corresponding points. The corresponding point from the set Q_1 is added to the set Q'. Not necessary all points from the dataset Q must be used and some points can be used more times, see Fig. 5.1.

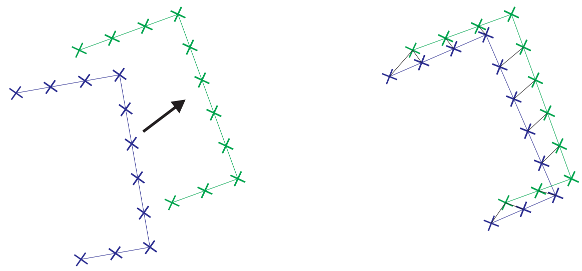


Figure 5.1: An example of corresponding points between two datasets.

3. The parameters of the transformation $\mathbb{T}$ are estimated using

$$\begin{aligned} \varphi &= \arctan \tfrac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}} \\ T &= \begin{bmatrix} \overline{x}' \\ \overline{y}' \end{bmatrix} - R_\omega \begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} \overline{x} &= \frac{1}{k}\sum_{i=1}^{k} x_i(t), & \overline{y} &= \frac{1}{k}\sum_{i=1}^{k} y_i(t), \\ \overline{x}' &= \frac{1}{k}\sum_{i=1}^{k} x_i(t+1), & \overline{y}' &= \frac{1}{k}\sum_{i=1}^{k} y_i(t+1), \\ S_{xx'} &= \sum_{i=1}^{k}(x_i - \overline{x})(x_i(t+1) - \overline{x}'), & S_{yy'} &= \sum_{i=1}^{k}(y_i - \overline{y})(y_i(t+1) - \overline{y}'), \\ S_{xy'} &= \sum_{i=1}^{k}(x_i - \overline{y})(y_i(t+1) - \overline{y}'), & S_{yx'} &= \sum_{i=1}^{k}(y_i - \overline{x})(x_i(t+1) - \overline{x}'). \end{aligned}$$

4. The estimated transformation parameters are used to determine the square error. If the error is smaller than the given threshold, the estimation process ends. Otherwise, the algorithm continues with the first step using the newly estimated parameters as initial values.

One disadvantage of the ICP algorithm is that it estimates the rotation of the robot with a significant error due to the correspondences found by the closest-point rule, which contains a little information about the rotation. An iterative matching range point has been proposed [26] to improve the estimation of the rotation.

5.1.2 Iterative Dual Correspondences

The main idea of the *Iterative dual correspondences* algorithm (IDC) [26] is to combine the Iterative Closest Point (ICP) and Iterative Matching Range Point (IMRP) in a single algorithm providing their advantages. The IMRP [26] works in a similar way as the ICP algorithm, but it uses a different rule for searching the corresponding points. Consider the datasets Q and Q' that can be described by the transformation matrices R and T:

$$P' = RP + T. \tag{5.1}$$

The idea of IMRP is to ignore the translation T and then it holds that $|P| \approx |P'|$. The vector v is from the origin to the point P and it has angle Θ. The vector v' is from the origin to the point P' and it has angle Θ'. It holds that the angles are related by equation $\Theta' = \Theta + \varphi$. The IDC algorithm uses a translation estimated by the ICP and a rotation estimated by the IMRP and produces a better solution than each algorithm individually. Moreover, it has the same stability as the ICP and it has identical convergence as the IMRP.

5.1.3 Random Sample Consensus

A method called *Random Sample Consensus* (RANSAC) [12] estimates parameters of a mathematical model based on a random selection of few representative points from an input dataset. The method is an iterative procedure that is terminated if the value of the selected criterion E is smaller than a given threshold. The criterion describes how the model M fits to the input data D. For example, the criterion can be defined as

$$E = \sum_{i=1}^{N} |d_i - m_i|,$$

where d_i is an input point, m the model point and N is the total amount of points in the dataset D and also in M.

The RANSAC algorithm can be used to fit a 2D line to an input set of points, where some of the points approximately fit the line (such points are called inliers) and some of them are so-called outliers, i.e., points corresponding to noise or incorrect measurements. The advantage of RANSAC is that it fits only inliers to a model of line unlike the least squares method that fits the line to all points from the set.

The RANSAC method is used to estimate the transformation $\mathbb{T}$. The algorithm works as follows. First, n pairs of points are randomly selected from the input sets P and Q, in this case $n = 2$, and the points are used to solve Eq. (1.1). Then, the determined transformation is applied to all points from the set Q and the criterion

E is computed again. If it is lower than a given threshold and lower than the last estimation, the new transformation parameters are saved. This procedure is repeated k-times.

5.2 Simultaneous Localization and Mapping

The simultaneous localization and mapping (SLAM) technique is used by autonomous robots to localize itself according to the map that is simultaneously created using world observations and the robot pose. Using terminology from Section 1, SLAM is problem of recovering a model of the robot world $\mathcal{M}$ and the sequence of robot locations X_T from the odometry U_T and measured features Z_T [37]. Two main forms of SLAM problem exist, the first is the **online SLAM problem** where only the current robot pose x_t and the map $\mathcal{M}$ are estimated based on the odometry measurements U_T and features observations of the environment Z_T,

$$p(x_t, \mathcal{M}|Z_T, U_T).$$

Algorithms solving the online SLAM are usually incremental and can process one data item at a time. The second one is the **full SLAM problem** that computes the posterior of the whole robot trajectory X_T and the map $\mathcal{M}$. This is defined as

$$p(X_T, \mathcal{M}|Z_T, U_T).$$

SLAM algorithms use two other models to solve these problems. The robot kinematic model describes how the next robot position x_t can be estimated using the previous position x_{t-1} and odometry measurement u_t. This model can be determined by the probability distribution

$$p(x_t|x_{t-1}, u_t).$$

The second model describes the relation between scan z_t of the environment, the true map $\mathcal{M}$ and the robot current position x_t. The model is

$$p(z_t|x_t, \mathcal{M}).$$

5.2.1 Taxonomy

Several SLAM based approaches and algorithms can be found in literature. The particular methods are focused on specific issues of the problem and they can be divided according to the following taxonomy.

- **Metric or topological map**

  A metric map contains geometrical information about relations of the map features. A topological map describes only relations between the map features.

- **Feature-based or volumetric map**

  Algorithms that work with a feature-based map, extract from scans only some features that are significant for them, for example corners in the case of the indoor environment. The volumetric map is a high dimensional representation that allows realistic reconstruction of the environment.

- **Known or unknown correspondences**

  When adding a landmark to the map, it has to be decided if the landmark is new or if it is already in the map, i.e., if the new landmark corresponds to the existing one. Some SLAM techniques suppose that correspondences are known while others do not. The problem of estimating the correspondence is one of the most difficult problems in SLAM [37].

- **Small or large uncertainty**

  If the robot visits an identical place twice while it operates in the environment, the robot position and map are made more precise. This situation is known as the loop closing problem and it is an advantage for localization algorithms. The uncertainty before the loop closing may be large and SLAM techniques have to handle with it.

- **Static or dynamic environment**

  Objects are not moving in static environment over time and the localization methods often evaluate a dynamic effect as measurement outlier.

- **Passive or active SLAM**

  Passive SLAM algorithms only observe the environment and some other algorithm controls the robot motion. Active SLAM algorithms control the robot motion for the purpose of an accurate map.

Specification of the thesis approaches

The feature-based metric map is used in this thesis for cases, where it is supposed that the correspondences are unknown. The both scenarios with small and large uncertainties are tested in the dynamic environments. All algorithms are passive, which means that the robot is navigated by another function.

5.2.2 Three Main SLAM Approaches

This subsection presents a brief description of three main SLAM approaches from which others methods are derived. The historically first is based on the Extended Kalman filter, the second approach is based on graph-based optimization techniques and it is often used for the full SLAM problems. The third uses particle filters and it is a popular method for the online SLAM problem.

Extended Kalman Filter (EKF)

The idea of EKF SLAM is based on a single state vector μ for estimating the robot and landmarks positions and the covariance matrix $\sum$ for representation of the uncertainty in these estimates [38]. While the robot is moving, these entities are updated using the Extended Kalman filter. When a new feature is observed, a new state is added to the system state vector μ. The disadvantage of EKF SLAM is that the size of the covariance matrix grows quadratically with the number of the observed features; so, it is not suitable for spaces with many landmarks. EKF SLAM assumes metrical, feature-based map with known correspondences.

Graph-based Optimization Technique

This SLAM technique constructs a graph where nodes are robot locations and map features. The arcs are between two consecutive robot positions and between robot positions and sensed features where some features can be sensed from more robot positions, but each node is connected only to few other nodes. The robot and features positions are then determined through nonlinear sparse optimization [25]. The advantage of this method is that the required memory is linear in contrast to EKF SLAM methods; so, they can work with high-dimensional maps.

Particle Methods

In particle filters methods a probability distribution is represented as a set of particles that are hypothesis of the true state of robot or landmark [23]. Each particle is represented by the mean vector μ and variances $\sum$. The particle SLAM method computes the probability of the new measurement that are compared with the actual sensed scans. Based on this comparison, particles are weighted. A higher weight is given to the particles whose prediction match the measurements. In the next step, the particles are re-sampled and only particles with a high weight are selected. For the new particle set, the mean vector μ and variances $\sum$ are updated.

Chapter 6

Evaluation of the Localization Techniques

The LEL algorithm is applied in two different scenarios in order to verify the LEL's behaviour using input datasets corrupted by outliers. The evaluation of the behaviour is evaluated based on the *precision*, *repeatability* and *quality* of the robot pose estimation that are defined in Section 6.1. The performance of LEL is compared with the ICP and RANSAC algorithms.

In the first scenario defined in Section 6.2, a mobile robot with the differential drive is equipped with a laser rangefinder and it operates in an indoor environment. Two different experimental datasets are made within this scenario. In the first case, the robot movements and sensors perception are simulated in the Player/Stage framework while in the second case, real measurements (from the Radish database) are used. The robot is localized in $\mathbb{R}^2$; thus, the transformation

$$\mathbb{T} = [dx,\ dy,\ \varphi]$$

is estimated using the *scan-to-scan* matching method. In the second scenario defined in Section 6.3, the Malaga dataset [3] is used. A mobile robot with the ackerman drive is equipped with a pair of cameras configured to capture stereo images. The robot operates in an outdoor environment and it is localized in $\mathbb{R}^3$ by estimating the robot transformation

$$\mathbb{T} = [dx,\ dy,\ dz,\ \varphi,\ \beta,\ \gamma].$$

using the *scan-to-scan* matching again.

6.1 Terminology

The behaviour of the localization techniques is evaluated by measuring of the *precision*, *repeatability* and *quality* of the robot *path*. In this thesis, the term path means the discretized form of a continuous real robot path because particular measurements are captured according to the single robot pose. So, the robot path is a sequence of the robot poses, as defined in Section 1. The localization methods are evaluated using n trials. For each trial, the robot is navigated along a predefined trajectory, while its pose is available using a reference localization system providing the *ground truth*. The error d between the ground truth p_g and estimated p_e positions on the path is

$$d(i) = |p_g(i) - p_e(i)|,$$

where i is the index of the position at the particular path in trial j and $p_g(i),\ p_e(i)$ are vectors. The precision of the localization $\bar{d}$ is then determined as the average value of error $d(f)$ of the final robot position f during n trials according to Eq. (6.1).

$$\bar{d} = \frac{1}{n}\sum_{j=1}^{n} d(f) \tag{6.1}$$

The repeatability is computed as the sampled standard deviation s_n by Eq. (6.2).

$$s_n = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(d_j - \bar{d})^2} \tag{6.2}$$

The quality of the robot localization is determined using the normalized sum of the errors d, defined as

$$q = \frac{1}{l}\sum_{i=1}^{m} d(i), \tag{6.3}$$

where l is a length of the robot path.

6.2 Indoor Scenario

Two different indoor datasets are used to evaluate the LEL behaviour, the simulated datasets from the Player/Stage framework and the real measurements from the Radish database. The Player [49] is a network server based framework for robot control, which provides an unified interface to the robot's actuators and sensors. Many different robotic hardware are supported. Moreover, the drivers for new hardware can be added. The Player acts as a server and an user is its client that communicates with the Player using TCP sockets. The Stage [49] is a 2D simulator of mobile robots, their sensors and objects in the environment.

The Radish database is an open project for robotic community that provides datasets useful in robotics research and applications. The motivation for creating such a database is facilitation of the development, evaluation and comparison of robotics algorithms. These datasets can be created and used by anyone. They include logs of odometry, laser and sonar data taken from real or simulated robots in different environment conditions. The dataset *usc-sal200-021120* is used in this thesis. Here we would like to thank Andrew Howard for providing this data. The Pioneer2DX [44] robot was used to create the dataset, see Fig. 6.1. The robot has stepper motors that provide a precise odometry values. The robot was equipped with the SICK LMS 200 rangefinder that was mounted 8 cm forward

Figure 6.1: The robot Pioneer2DX used to capture the indoor dataset [47].

from the center of the robot. The robot was teleoperated by a human operator during an exploratory tour in the university building. The maximum range of the laser rangefinder was set to 8 meters; therefore, ends of the university corridors were not detected. The dataset contains the maximum range measurements that are not corresponding to any real feature in the environment; thus, they have to be extracted out from the dataset.

The used data association algorithm is the same as described in Section 3.4, but the Euclidean distance is determined using only the coordinates of the measured features.

6.3 Outdoor Scenario

The Malaga dataset [3] is considered to verify the LEL behaviour in the localization of the mobile robot in real outdoor environments. This dataset includes raw pictures from the stereo cameras and raw measurements from SICK rangefinders. In addition, its main advantage is that it includes centimeter-accuracy ground truth robot path. The main motivation of this dataset is the possibility to evaluate different robotic approaches and compare their properties in a realistic outdoor scenario. The Malaga dataset can be used in visual SLAM or visual odometry approaches, where a robot can be localized in $\mathbb{R}^2$ or $\mathbb{R}^3$.

6.3.1 Parameters of Vehicle and Sensors

An electric buggy was used to captured the data, see Fig.6.2. The main advantage of the electric buggy over a car with a combustion engine is that it avoids inherent vibrations. The buggy was equipped with twelve sensors. Two Hokuyo UTW-30LX placed in the front and in the rear of the vehicle, respectively. The SICK laser rangefinder LMS-200 is placed in the front and it is configured to measure the maximum distance up to 80 meters with the accuracy of 40 mm. Two additional SICK laser rangefinders LMS-221 are placed at each side of the vehicle to measure the robot surroundings up to 32 meters with the accuracy 5 mm. Two CCD color cameras AVT Marlin F-131C are placed at the front of the vehicle. They are distant 0.857 m from themselves and their optical axes are parallel and pointing forward. Images are captured at 7.5 fps with the resolution of 1024×768 pixels.

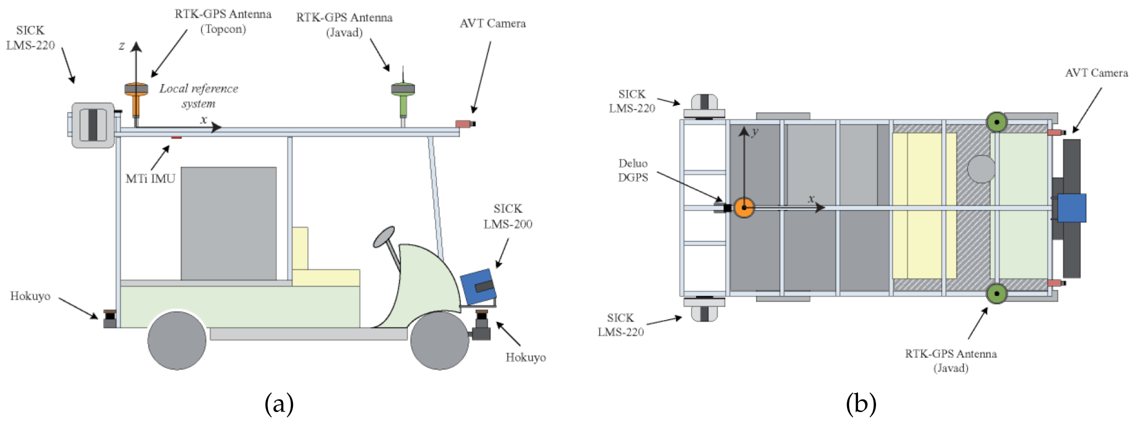


Figure 6.2: The electric buggy with the ackerman drive used to capture the Malaga dataset [3].

The vehicle is fitted out with an inertial measurement unit (IMU) and GPS devices for determination of the precise ground truth. The IMU consists of gyroscopes, accelerometers and magnetometers, which are combined using an Extended Kalman Filter to provide 3D orientation data at the maximal frequency 100 Hz. This unit has the accuracy $2.0°$ with the angular resolution $0.05°$ during the vehicle motions.

The Differential GPS (DGPS) and Real-Time-Kinematics (RTK) devices are used too. The DGPS overcomes the major limitations of a regular GPS system by using a static reference station with a known position. The differential corrections are sent to the unit attached to the vehicle via a radio signal. The static reference stations covers a large area (in order of kilometers). The DGPS system allows the vehicle pose estimation in tens of centimeters. In contrast, RTK devices provide a centimeter level of the accuracy by using the reference station corrections and carrier phase of the GPS satellite. Moreover, the reference station can be moved; so, the RTK devices can be placed closer to the vehicle, which would improve the precision of the pose estimation. The vehicle is equipped with one DGPS unit and three RTK-GPS units.

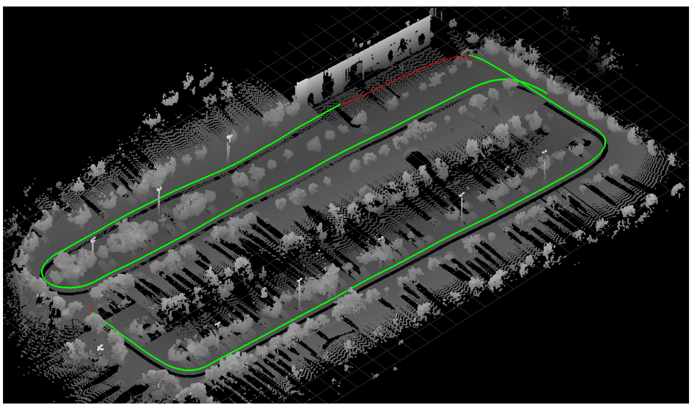


Figure 6.3: The points cloud of the used Malaga dataset. The green points denote the ground truth part and approximated path is in red [3].

6.3.2 Features Extraction

The dataset includes raw perception measurements, estimated traveled path of the sensors and ground truth path for the robot's center. In some areas, the paths are not determined due to loss of the GPS signal, see. Fig. 6.3. The captured raw images from left and right cameras have to be preprocessed in the following steps in order to determine 3D feature set. This set is then used as an input dataset in localization techniques. The preprocessing has been made with Matlab software [48] without any special optimization, which are not part of this thesis and do not influence the evaluation.

1. The open SURF detector for Matlab [41] is used to find important features in the left and right images, see Fig.6.4, which are captured at the same time instant.

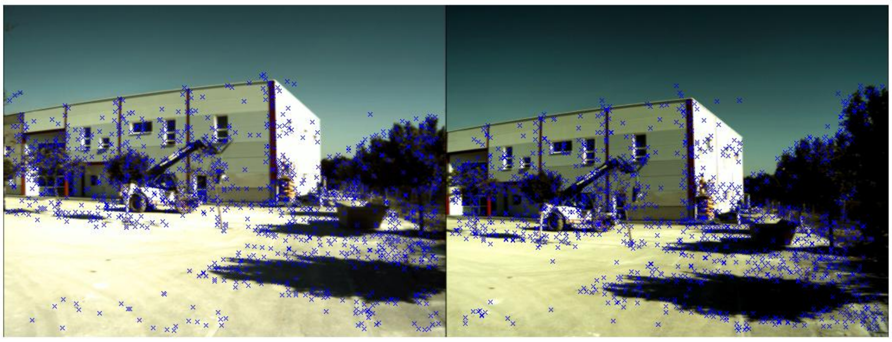


Figure 6.4: Important features found by the SURF detection in the left and right images.

2. The best corresponding features are found based on the proposed algorithm described in Section 3.4, see Fig.6.5.

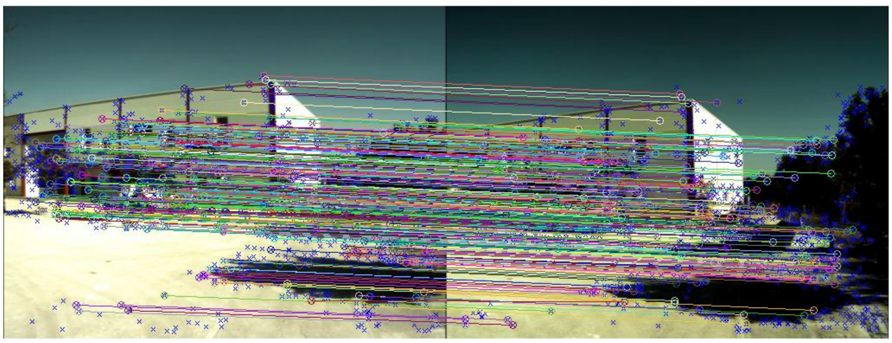


Figure 6.5: An example of the best corresponding points between the left and right cameras.

3. Using the found corresponding pairs of features, the depth of percepted features is determined, see Section 3.3. Without loss of a generality, the depth is assigned to the features from the left camera, see Fig. 6.6; thus, these features are defined in $\mathbb{R}^3$. The features have coordinates according to the image coordinate systems. Therefore, they are transformed to the robot coordinate system by
$$F' = R_r(R_cF + T_c) + T_r,$$
where F, F' are features according to the image and robot coordinate system, respectively. $R_c,\ T_c$ are rotation and translation matrices between the image coordinate system and camera coordinate system, R_r, T_r are transformation matrices between the camera and robot coordinate systems.

Figure 6.6: A visualization of the estimated depths, the blue points are in the foreground, green points are in the middle part and red points are in the background.

4. The best correspondences are determined between features F' extracted from the images captured in two consecutive robot poses, see Fig.6.7.

Figure 6.7: An example of the best corresponding points between the images captured in two consecutive robot poses.

Chapter 7

Indoor Experiments

Performance of the ICP, IDC, RANSAC and LEL method is evaluated and compared in an indoor scenario using the following experiments. In Section 7.1, the emphasis is put on the algorithm behaviour for input datasets corrupted by different levels of outliers. The input datasets are generated in the Matlab software [48] that enables to change the number of outliers and their parameters while other parameters of the datasets are fixed. It is expected that the precision and quality of the ICP and IDC will be worse for increasing number of the outliers. On the other hand, outliers should not have a significant influence to LEL and RANSAC. In this experiment, the Least-square (LS) method is considered in the comparison because it is used by the RANSAC method for estimation of the transformation between randomly selected features from the input dataset.

In Section 7.2, the algorithms are employed in the *scan-to-scan* matching problem, i.e., in the estimation problem of the robot pose transformation $\mathbb{T} = [x,\ y,\ \varphi]$ in $\mathbb{R}^2$ using two consecutive laser scans P and Q acquired at the time instants t_1 and t_2 ($t_2 > t_1$). The input datasets are simulated using the Player/Stage framework [49] that allows to control noise of sensor parameters and the robot environment parameters. It is expected that all evaluated algorithms estimate the transformation parameters with a small errors; so, they could be used for the robot localization. This evaluation is then repeated using real data from the experiments in a structured indoor environment, see Section 7.3. Finally, a summary of the evaluation is given in Section 7.4.

7.1 Influence of the outliers

In this proposed experiment, the algorithm robustness to the presence of outliers is evaluated. The number of outliers and their parameters are controlled, while other parameters of the input sets are fixed. The input sets are simulated as two laser scans that measure features located on a straight line, e.g., a wall. The equidistant placement of the features on the line is considered here, in spite of the fact that features scanned by a laser rangefinder do not have equidistant distances.

The dataset P is created as a sampled line

$$y = ax + b$$

with the parameters $a = 4.5,\ b = 3.1$. The dataset Q is then created by a transformation of the set P with $\mathbb{T} = [1.00; 2.00; 0.52]$. The total number of the points in the datasets is $N = 100$. In both cases, the samples of the lines are perturbed by a noise variable e that is drawn as a random number with the uniform distribution in the interval $\langle -1.5y; 1.5y \rangle$, where y refers to the y-coordinate of the sample. The ICP, LS and LEL estimation techniques require an initial guess of the estimated transformation, and therefore, the initial value of $\mathbb{T}$ has been selected as $\mathbb{T}_{ini} = 1.1\mathbb{T} = [\ 1.1 \quad 2.2 \quad 0.57\ ]$. The RANSAC algorithm is initialized by the following parameters.

- The minimum number of the data points required to fit the model is $n = 2$ due to the localization of the robot in $\mathbb{R}^2$.
- A threshold value for determining if a data point fits a model is chosen to $tr = 10$.
- The model fits well to the data if the number of points for which the error is bellow the threshold tr is greater or equal to $0.6N$.

The influence of the outliers to the performance of the transformation estimation $\mathbb{T}$ has been evaluated in three cases with different numbers of the outliers.

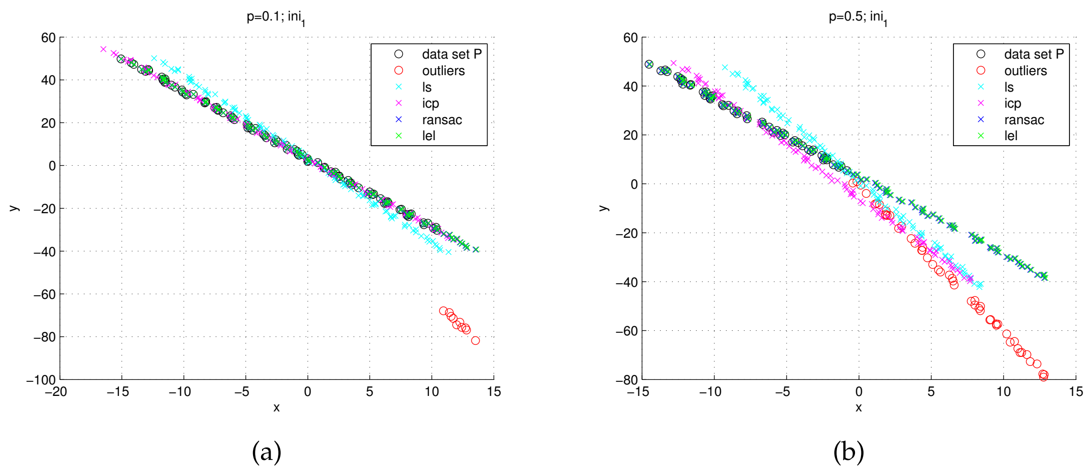
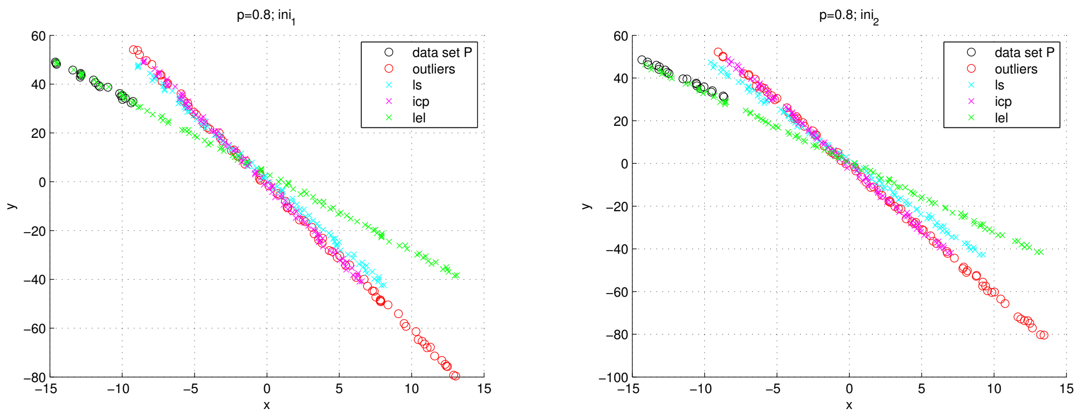


Figure 7.1: The comparison of the algorithm performances for the input sets corrupted by: a) $10\,\%$ of outliers, b) $50\,\%$ of outliers.

In the first case, $10\,\%$ of samples in the set Q are replaced by outlying values, which are samples of a different linear function with the parameters $[a,\ b] = [-6,\ \ -1]$. Then, the sets P and Q are used by the estimation techniques to estimate the transformation parameters $\mathbb{T}_e$. These estimations are compared using sets $\hat{P}_k$ that are created by the transformation of the original set P with the estimated parameters $\mathbb{T}_e$. The sets P and $\hat{P}_k$ are then displayed in Fig. 7.1a. In the ideal situation, the correct points from the set P overlap the points from $\hat{P}_k$. In the next case, the number of outliers is chosen as the boundary value $50\,\%$. The LEL and RANSAC methods still provide a suitable estimation, but the ICP and *IDC* approaches are significantly influenced by such a high number of the outliers, see Fig. 7.1b.

As the third situation, $80\,\%$ of outliers is chosen. From Fig. 7.2a, it can be observed that the ICP estimates a dominant transformation of the outliers regardless the correct transformation has been used for the initialization. Contrary, LEL still gives a satisfiable estimation. The experiment with $p = 0.8$ has been repeated with a worse initial transformation than in the previous case. An estimation given by LS is used as the initial transformation here. It can be observed that LEL still provide a good estimation, see Fig. 7.2b.



(a) (b)

Figure 7.2: The comparison of the algorithm performances for input sets corrupted by $80\,\%$ of outliers and different initial conditions.

The best performance in this evaluation provides the LEL and RANSAC estimation methods; so, the robustness of LEL to outliers is verified.

7.2 Simulated Input Datasets

In this evaluation, the Player/Stage framework is used to simulate a robot motion and sensoric measurements. A non-holonomic robot with a differential drive is simulated with odometry measurements influenced by different levels of noise. The robot is equipped with a laser rangefinder to scan the robot surrounding environment, the rangefinder is placed at the robot center and its orientation is forward looking. The rangefinder has $180°$ scanning area with $\delta = 0.5°$ difference between two neighboring measurements and the maximal range 8 m is influenced by noise with the amplitude 0.01 m. In this section, the noise is drawn from the normal distribution in all cases.

7.2.1 Squared Robot Path

In the first case, the robot is navigated along a rectangular trajectory in an indoor structured environment, where dimensions of the environment are intentionally chosen as 10×10 m to guarantee that the robot's rangefinder measures the boundaries of the environment. The maximal translation velocity is $v_{t_{max}} = 0.1$ ms^{-1} and the maximal rotation velocity is $v_{r_{max}} = 3°\mathrm{s}^{-1} = 0.0524$ rad/s. The robot odometry is influenced by a noise with the amplitude $x = 0.05$ m, $y = 0.05$ m and $\varphi = 0.5° = 0.0087$ rad. The noise parameters are intentionally chosen to significantly disturb the odometry measurements because it is expected that evaluated algorithms estimate the robot trajectory more accurately than the odometry. The odometry and rangefinder measurements are recorded with the frequency 1 Hz.

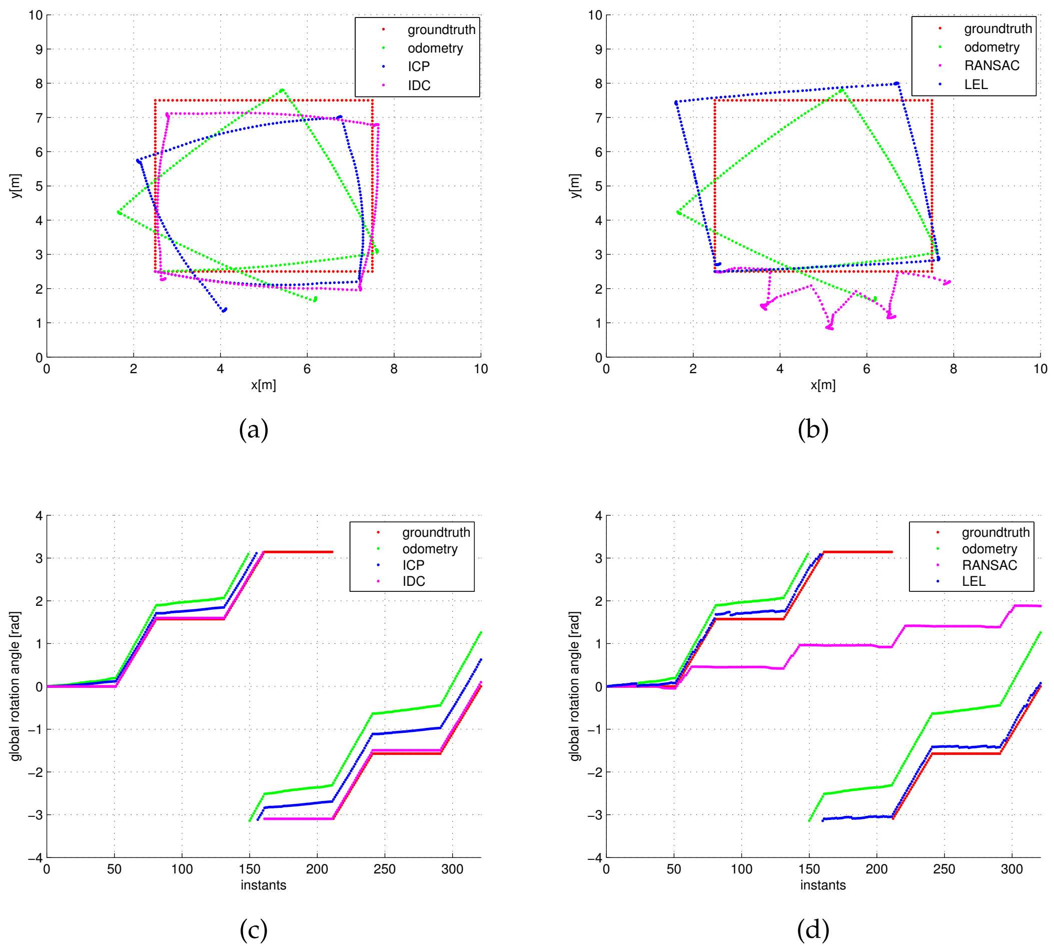
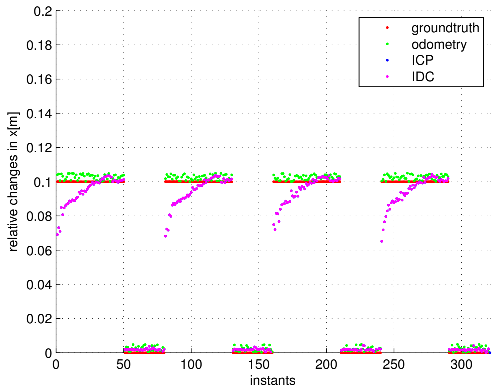


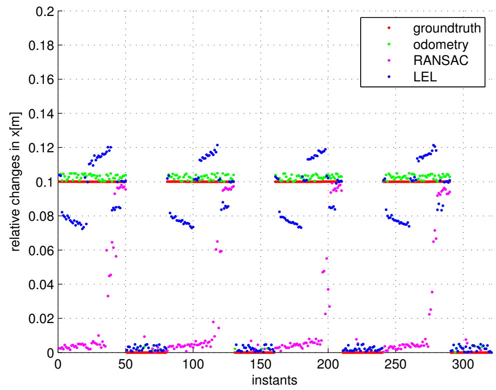
Figure 7.3: The ground truth, odometry and estimated global robot paths (a,b) and heading (c,d) for the squared path.

The evaluated algorithms use the odometry for initialization and they estimate the robot pose using scans of the environment. A visualization of the estimated robot positions can be seen in Fig. 7.3a and Fig. 7.3b. Regarding the figures, the IDC algorithm provides the most accurate estimation of the robot position along the traveled path. The LEL algorithm provides a more accurate estimation of the robot path than the ICP algorithm. Moreover, it is also a more precise than the odometry values. In contrast, RANSAC gives the worst estimation at all. The ICP, IDC, LEL determinate correctly situations when the robot is rotating, as can be seen in the visualization of the robot heading estimations in Fig. 7.3d, RANSAC does not.

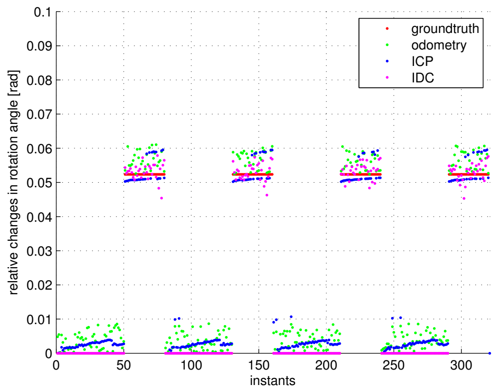
The global robot path is determined using the estimated relative transformations $\mathbb{T}_e$; so, the precision and quality of the robot pose estimation depends on the accurate estimation of each parameter $\mathbb{T}_e$. The estimated relative translation dx and relative angle $d\varphi$ are plotted in Fig. 7.4. It can be observed that the less accuracy of the LEL and ICP is caused by errors in the estimations of the relative translation and rotation. The RANSAC method estimates all relative parameters incorrectly.



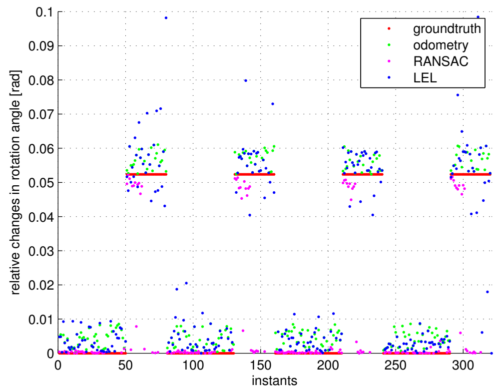
(a)



(b)



(c)



(d)

Figure 7.4: The ground truth, odometry and estimated parameters of the relative translation (a,b) and rotation dφ (c,d) for a squared robot path.

Table 7.1: Algorithm properties

Algorithm	$\bar{d}$ [m]	s_n [m]	q [m]	T [s]
ICP	2.14	0.13	19.01	241
IDC	0.21	0.08	9.15	256
LEL	1.28	0.12	12.85	84
RANSAC	8.78	9.05	83.72	609

An overall performance of the algorithms is evaluated in ten trials for different input sets. This experiment is done because the algorithms depend on the initialization and amount of outliers, as it has been shown in Section 7.1. Thus, in each trial, the input set is generated with different noisy values. Then, the performance metrics (the precision, repeatability and quality of the localization) according to Section 6.1 are computed from the trials and the results are depicted in Table 7.1. The column T denotes the average number of the required computational time to

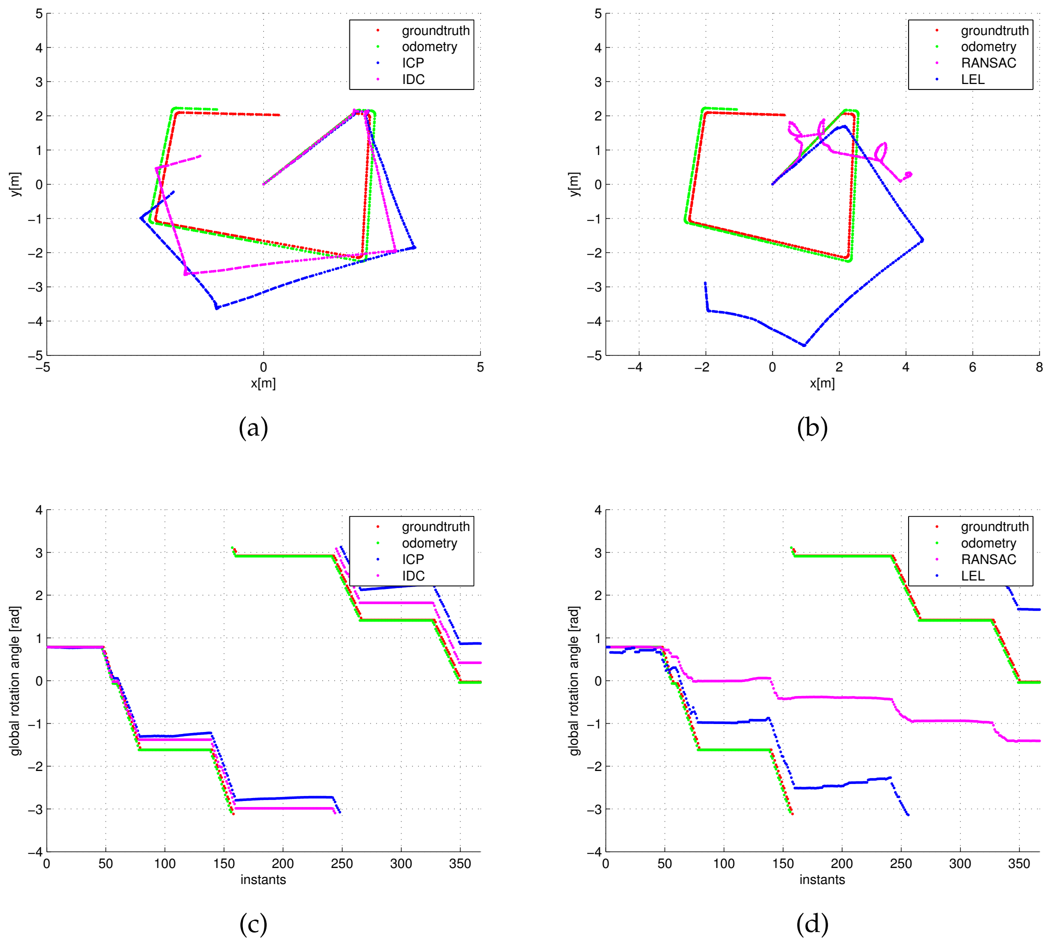


Figure 7.5: The ground truth, odometry and estimated global robot paths (a,b) and headings (c,d) for the general path.

perform a single trial using a standard laptop with CPU running at 2 GHz and 1.5 GB RAM. Here, it is worth to mention that the LEL estimation technique is the least computational intensive, which is one of its benefits over the other methods.

7.2.2 General Robot Path

This scenario is also simulated using the Player/Stage framework with the same robot and sensor but the robot is navigated to obtain measurements from the whole environment, i.e., the robot is employed in the exploration task. This navigation is made using an example code from the Stage framework and it is chosen because it produces a general robot path. The odometry is influenced by the noise with the parameters $x = y = 0.005$ m, $\varphi = 0.005$ rad. The maximal translation speed of the robot is $v_{t_{max}} = 0.075$ ms^{-1} and the maximal rotation speed is $v_{r_{max}} = 0.1$ rads^{-1}.

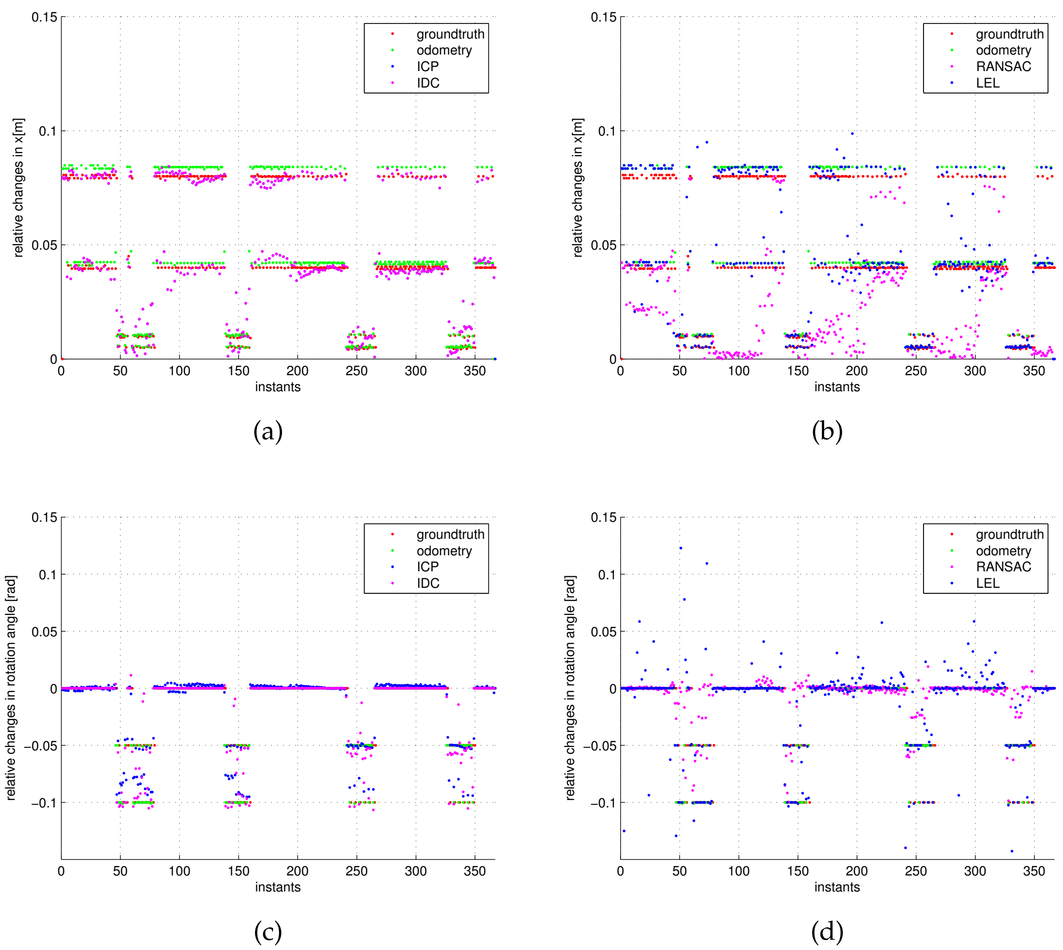
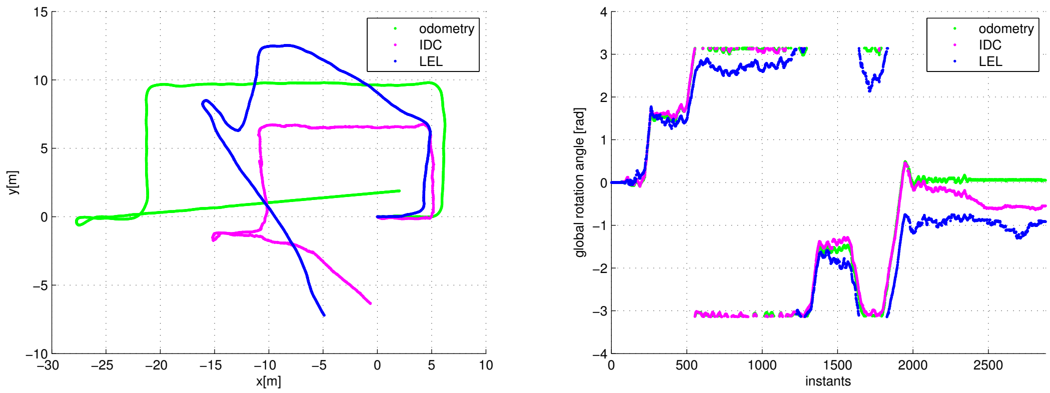


Figure 7.6: The ground truth, odometry and estimated parameters of the relative translation (a,b) and rotation dφ (c,d) for the general robot path.

The estimated global poses and headings of the robot are visualized in Fig. 7.5. According to the presented results, the IDC and RANSAC algorithms have the same performance as in the previous scenario; thus, the IDC algorithm provides the best result and RANSAC is the worst evaluated estimation technique. The LEL estimation is influenced by the significant incorrect estimation of the relative parameters during the first robot turn, see plots in Fig. 7.6 of the relative estimations for a detail.

7.3 Real Robot Environment

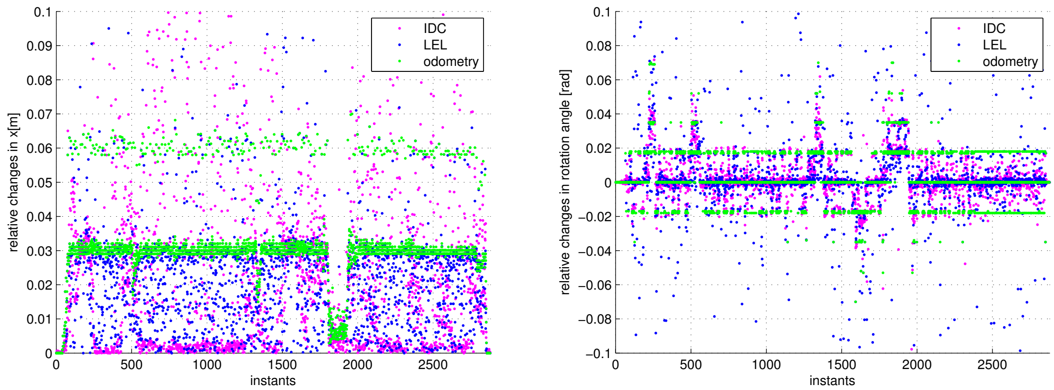
The dataset *usc-sal200-021120* from the Radish database has been selected to evaluate the performance of the studied estimation techniques employed in a real robot localization problem. The environment used is a university building with several long corridors, where the issue described in Section 1.2 can occur.



(a) (b)

Figure 7.7: The odometry and estimated global robot paths (a) and heading (b) for a robot path in a real environment.

Only the LEL and IDC estimation techniques are compared in this scenario. The IDC is chosen because it provides the best performance in the previous evaluations. In Fig. 7.7a, the global estimated positions are visualized together with the precise odometry measurement. It can be observed that both algorithms suffers from the problem of the long corridors. This situation is observable also in Fig. 7.8a, e.g., for the area between the time instants $500 - 1200$. Regarding this area, it can be seen that the estimations correspond to the odometry values but they are influenced by a low amount of significant features in the input dataset; so, the estimated relative translation is smaller than the odometry value. The LEL algorithm has the same drawbacks as in the previous experiments, i.e., it estimates the rotation less accurately than the IDC, see Fig. 7.7b and Fig. 7.8b.



(a) (b)

Figure 7.8: The odometry and estimated parameters of the relative translation (a) and rotation dφ (b) for a robot path in the real environment.

7.4 Summary

In this chapter, the LEL performance is evaluated and compared with the other localization methods in scenarios representing a localization problem in an indoor environment. In the first case, the LEL robustness to outliers is shown. The RANSAC method provides a sufficient performance only for input datasets corrupted up to $50\,\%$ of outliers. The IDC and ICP methods are significantly influenced by outliers. Then, the ICP, IDC, LEL and RANSAC algorithms are evaluated in the mobile robot localization task using the *scan-to-scan* approach for three different robot environments.

In the first and second environments, the Player/Stage framework is used to obtain measurements from the odometry and the laser rangefinder while the robot is navigated along a squared and general paths. In this evaluation, the IDC algorithm provides the best performance. The LEL algorithm provides a competitive performance to the ICP, but it is significantly faster. The RANSAC algorithm is the slowest algorithm. Moreover, it provides the worst performance. In the third environment, the real measurements are used from the Radish database and the performance of the IDC and LEL algorithms are compared. Both algorithms are influenced by a low amount of the significant measurements in the input datasets, that is caused by the problem of long corridors, which is described in Section 1.2. The IDC provides a better estimation of the rotation parameter than LEL.

Based on these evaluations, the LEL performance in the mobile robot localization task is promising, but it has to be improved. For example, only important features can be extracted from the input datasets and corresponding measurements can be associated by a better method to increase precision of the estimated parameters. Such improvements have been implemented and evaluated for a outdoor localization scenarios described in the next chapter.

Chapter 8

Outdoor Experiments

Five different sets of experiments have been performed to evaluate the LEL behaviour in an outdoor scenario. In the first experiment, simulated input datasets are used to observe the LEL behaviour for datasets corrupted by different levels of outliers. Moreover, an influence of the initial estimation of the transformation parameters and the influence of the rotation matrix parametrization are verified in this experiment.

In the second type of experiments, the robot localization task is simulated, it means that the dataset with features from different robot poses and odometry is generated. The estimations of relative transformation parameters are then used to determine the global robot path. These two experiments are similar to the experiments that are performed by the authors of LEL. In these experiments, the important fact is that corresponding pairs are known. The quality of the correspondences influences the LEL behaviour, as it is shown in the third experiment.

Finally, the LEL performance is verified using the real measurements from the Malaga dataset [3], where features and correspondences have to be determined. The proposed data association method, see Section 3.4, is verified in the last experiment.

8.1 Influence of the Dataset Parameters

The input datasets $P,\ Q$ for the localization methods are generated to simulate sets of features measured by a robot. The dataset P is generated as a plain perpendicular to the z-axis with random values drawn from the uniform distribution within the interval $z \in \langle 3.6, 4.4 \rangle$. The new dataset Q representing the next measurements to be aligned to the previous dataset is created according to known transformation matrices R_0 and T_0 by Eq. 8.1.

$$Q = R_0^{-1}(P - T_0) \tag{8.1}$$

Then, $\mathbb{O}\,\%$ of features in dataset Q are replaced with outlier values, see Fig. 8.1. The transformation $\mathbb{T}_0$ with parameters $R_0,\ T_0$ is the ground truth value. In this section, all random values, which represent noisy measurements, are drawn from the uniform distribution.

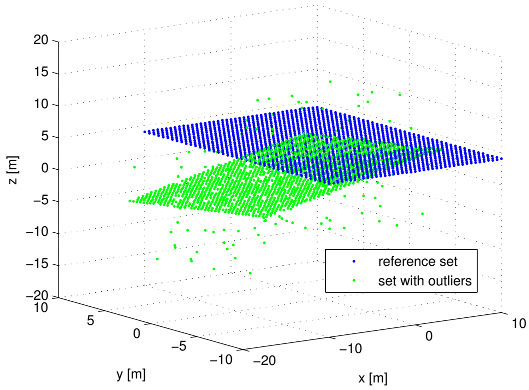


Figure 8.1: An example of the generated datasets P (in blue) and Q with added noise and outliers (in green).

8.1.1 Influence of the Outliers

The influence of the outliers is evaluated in 50 trials for 50 datasets Q_10 that are generated by Eq. 8.1 and $10\,\%$ of randomly selected features are replaced by outliers with $z \in \langle -20z_f, 20z_f \rangle$, where z_f is the value of the selected feature. The initial transformation is generated using the ground truth transformation $\mathbb{T}_0$ with added noise from the interval $\langle -0.1\mathbb{T}_0,\ 0.1\mathbb{T}_0 \rangle$.

Then, the LEL method estimates the transformation between the sets P and Q_{10}. These estimations are compared with the ground truth transformation $\mathbb{T}_0$. The results show that the estimated transformation differs from $\mathbb{T}_0$ only in order of 10^{-5} m; so, the LEL estimation can be considered as sufficiently precise for the localization. This experiment is repeated with $20\,\%$ and $50\,\%$ levels of outliers. The differences between the estimated and ground truth transformations are in order of 10^{-4} m. From these results, it can be observed that the LEL algorithm is robust to the presence of outliers.

8.1.2 Influence of the Initial Transformation

The entropy-like function H is a non-linear function, where several local minima exist. Moreover, it depends on six variables of the transformation $\mathbb{T}$ in the case of outdoor localization scenario; thus, the found local minimum depends on the initial transformation parameters as well. In this experiment, a set of 50 initial transformations $\mathbb{T}_p$ is generated using $\mathbb{T}_0$ with the added noise drawn from the standard uniform distribution on the interval $\langle -p\mathbb{T}_0,\ p\mathbb{T}_0 \rangle$, for values $p \in \{0.1,\ 0.2,\ 0.5\}$. For each initial transformation, the LEL estimation is determined using the input sets $P,\ Q$. The differences between $\mathbb{T}_0$ and the estimated transformation are determined and the average differences are shown in Tab. 8.1. It can be observed that the estimation error increases with a bigger uncertainty of the initial transformation parameters.

The differences in Tab. 8.1 seem to be insignificant. However, the impact of the noisy initial transformation can be better observed in the robot localization task,

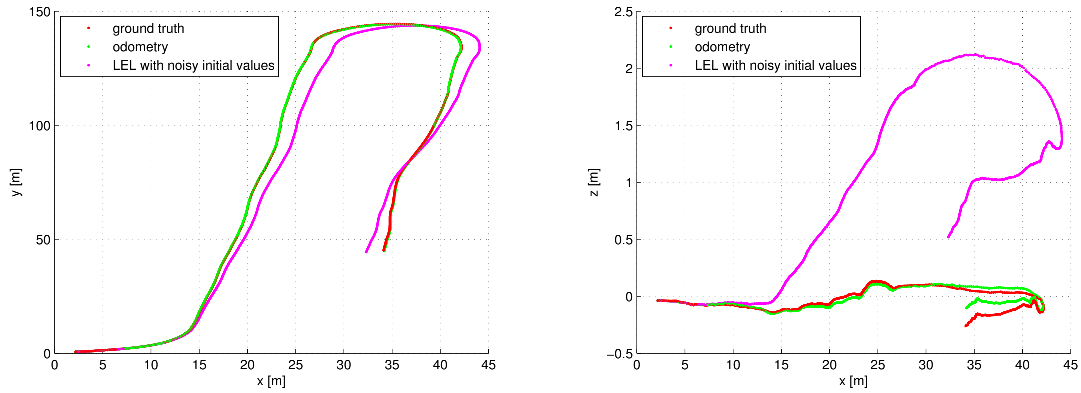
Table 8.1: The average differences between the ground truth and estimated transformation for different noise levels of the initial transformation.

	dx [mm]	dy [mm]	dz [mm]	φ [10^{-3}rad]	β [10^{-3}rad]	γ [10^{-3}rad]
$d_{0.05}$	0.51	0.78	0.82	0.21	0.11	0.09
$d_{0.10}$	0.52	0.81	1.00	0.35	0.15	0.16
$d_{0.25}$	1.12	2.02	1.96	0.93	0.42	0.38

where the differences are cumulated. Therefore, the impact to the global robot pose increases. To observe this behaviour, a new experiment is arranged as follows.

1. The input initial set P_0 contains extracted features from an image from the Malaga dataset.
2. The set Q_0 is then generated using Eq. 8.1, where the transformation matrices are created using relative ground truth changes from the Malaga dataset.
3. The set Q_{0_n} is created by adding the noise and outliers to the set Q_0.
4. The initial transformation parameters are generated by adding a noise value from the interval $\langle -0.1\mathbb{T}_0, 0.1\mathbb{T}_0 \rangle$ to the ground truth value $\mathbb{T}_0$.
5. Then, the sets $P_0,\ Q_{0_n}$ are used as inputs for the LEL algorithm to estimate the relative transformation.
6. In the next step, P_1 is equal to Q_0 and the set Q_{1_n} is generated by transforming of P_1 and by adding noise and outliers.
7. This generation is repeated to get N pairs of $P_j,\ Q_{j_n}$, where the important fact is that the features from the both datasets P_j and Q_{j_n} have perfect and known correspondences.
8. Finally, the relative transformations are used to determine the robot global path, see Fig. 8.2.

The estimated global robot path from the LEL technique is compared with the ground truth and odometry path, see Fig. 8.2. The odometry path is determined using the noisy initial parameters of $\mathbb{T}$ from the step 4 of the process above. In Fig. 8.2a, xy-view of the paths is shown, where the LEL estimated path reflected the ground truth. The difference between them seem to be caused by a cumulation of the errors due to the used *scan-to-scan* matching approach. However, the influence of the noisy initial values is better observable in xz-view in Fig. 8.2b, where the estimated LEL path is significantly influenced by the initial guess. Thus, the precondition of the difficulties caused by the initial values are verified here. The LEL's dependence on the initial values is one of its disadvantage that should be resolved in a future work.



(a) (b)

Figure 8.2: Influence of the noisy initial transformation parameters (odometry) to the LEL algorithm in the robot localization task.

8.1.3 Influence of Rotation Representation

The representations of the rotation matrix R by *Euler angles* and by *quaternions* are compared in two different cases. In the first case, the transformation

$$\mathbb{T}_1 = [0.1217,\ 0.2178,\ 0.3616,\ 0.0166,\ 0.0011,\ 0.0008]$$

between sets P and Q is chosen to simulate the relative robot changes. It means that the transformation contains small values, especially angles are $\sim 0.5°$. The non-linearity of goniometric functions $\sin(\alpha),\ \cos(\alpha)$ is insignificant for a small angle α. However, the considered angles are intentionally chosen much bigger to test significant non-linearity of the goniometric functions in the second case using the transformation

$$\mathbb{T}_2 = [0.1217,\ 0.2178,\ 0.3616,\ \pi/3,\ \pi/4,\ \pi/6].$$

The performance of the LEL estimation is compared according to the ground truth transformation using the both approaches of the representations of the rotations.

In both cases, 50 trials with different noise in the initial values are considered. The used terminology is:

- The LEL estimation via Euler angles is $\mathbb{T}_e$, via quaternions $\mathbb{T}_q$.
- The Euler difference is $d_e = \mathbb{T}_e - \mathbb{T}_K$, where $\mathbb{T}_K$ reffers to $\mathbb{T}_1$ or $\mathbb{T}_2$.
- The quaternion difference is $d_q = \mathbb{T}_q - \mathbb{T}_K$.
- The average value of differences $\bar{d_a}$ is determined as

$$\bar{d_a} = \frac{1}{N}\sum_{i=1}^{N}(d_e - d_q),$$

where N is the number of trials.

In the first case, the values of $\bar{d_a}$ are only in order of 10^{-6}; thus, the impact of the rotation representation is insignificant for small angles. In the second case, the differences d_e are bigger than differences d_q. The average value is

$$d_a = [0.67,\ 0.80,\ 1.39,\ 0.07,\ 0.05,\ 0.03],$$

where the translation difference is in meters and the rotation difference is in radians.

The summary of this verification is as follows:

1. The optimization of non-linear least-entropy like function H is a more precise for the representation of the rotation matrix via quaternions for angles $\alpha \gg 0.5°$.

2. The optimization is equally precise for angles $\alpha \sim 0.5°$.

3. The optimization process using the representation via Euler angles is faster than via quaternions.

8.2 Localization of Mobile Robot with Known Correspondences

The datasets $P,\ Q$ are generated using the same method as in the simulation of the robot localization tasks presented in Section 8.1.2. The odometry is generated from the ground truth transformation $\mathbb{T}_0$ with added noise drawn from the uniform distribution on the interval $\langle -0.25\mathbb{T}_0, 0.25\mathbb{T}_0 \rangle$. The noise amplitude is intentionally chosen to be small to reduce influence of the initial value.

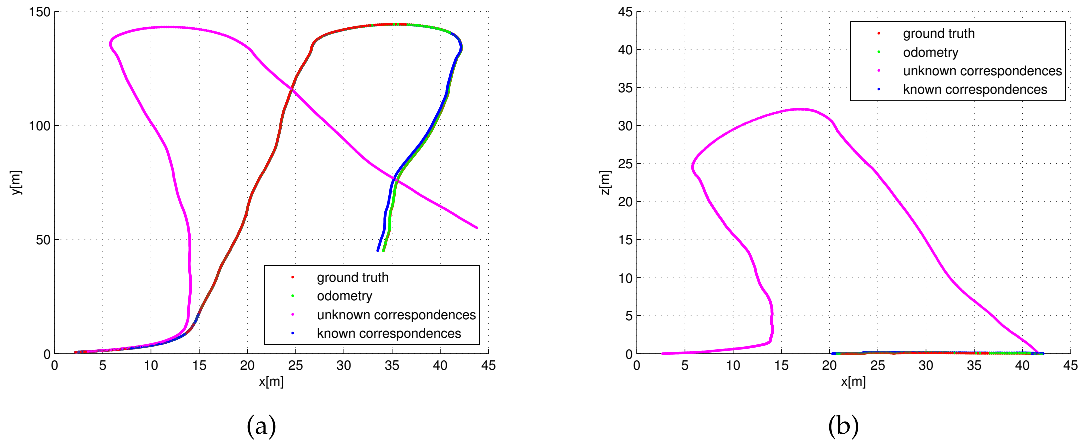
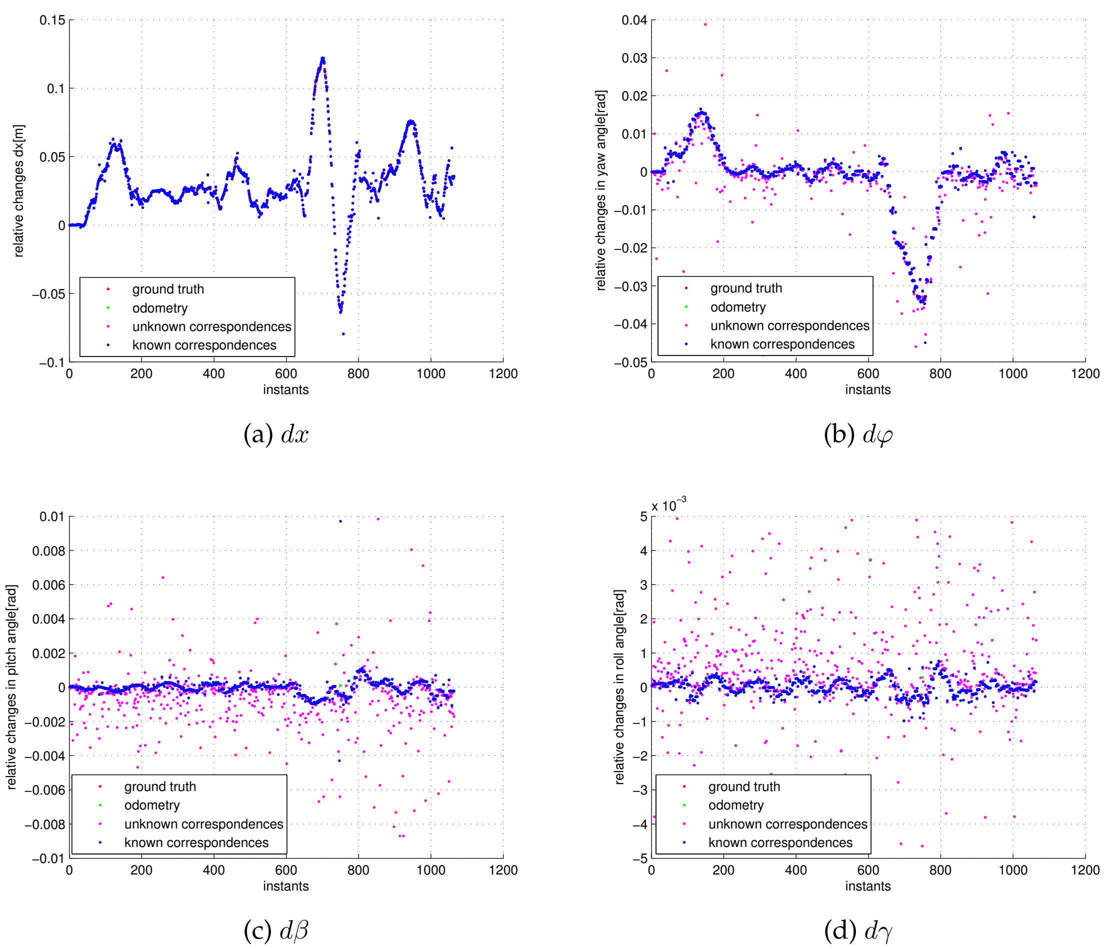


Figure 8.3: The LEL estimation of the robot global path using known (blue) and unknown (magenta) correspondences.

It can be observed that the LEL estimation is quite precise when known correspondences are used, see blue path that is overlapping with the ground truth path in Fig.8.3.

8.3 Localization of Mobile Robot with Unknown Correspondences

The LEL behaviour is verified here in the situation when the correspondences between features in the input datasets $P,\ Q$ are unknown. The exactly same datasets $P,\ Q$ as in the previous experiment are used; however, the data association algorithm from Section 3.4 is used to determine corresponding pairs of the feature descriptors. Therefore, the descriptors D of features in the dataset Q are influenced as follows.



(a) dx

(b) $d\varphi$

(c) $d\beta$

(d) $d\gamma$

Figure 8.4: The LEL estimation of the relative changes using known (blue) and unknown (magenta) correspondences. The LEL estimations using known correspondences overlap the odometry and ground truth values.

1. The noise drawn from the uniform distribution on the interval $\langle -0.1D, 0.1D \rangle$ is added to descriptors D. The noise is intentionally significant to cause problematic data association.

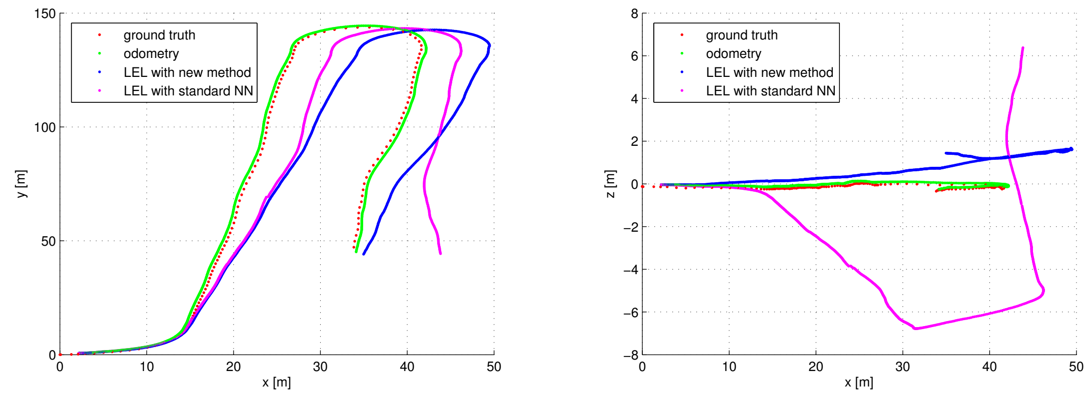
2. The features in dataset Q are randomly permuted.

The new dataset Q_a is created as the best corresponding features from Q to the set P using the data association algorithm.

The LEL estimations using the datasets $P,\ Q_a$ are used to determine the global robot path, see magenta path in Fig. 8.3. It can be seen that wrong corresponding pairs significantly influence the LEL behaviour. When the relative changes are compared, it can be observed that the estimation of the translations $dx,\ dy,\ dz$ is not influenced by a wrong correspondence matching, see Fig. 8.4a (the LEL estimations overlap the odometry and ground truth values). However, the estimation of the rotation angles is significantly influenced by wrong correspondences matching, see Fig. 8.4b, 8.4c, 8.4d.

8.4 Comparison of Data Association Methods

The significant influence of correspondence matching has been shown, and therefore, the new proposed data association methods, see Section 3.4 is compared with the standard Nearest Neighbour approach (NN). This comparison is made by using two different sets of the corresponding pairs $\mathbb{C}_{NN}$ and $\mathbb{C}_P$ as the input datasets for the LEL algorithm. These sets are generated as follows.



(a) (b)

Figure 8.5: The global paths using the new proposed data association method (blue) and Nearest Neigbour method without any filtration (magenta).

1. The input datasets $P,\ Q$ that were generated in the previous case are used.

2. The set $\mathbb{C}_{NN}$ is created by the corresponding features from Q to P that are determined using the minimal Euclidean distance in the NN and no extra filtration is made.

3. The set $\mathbb{C}_P$ is created by the corresponding features from Q to P that are determined by the new proposed method.

According to Fig. 8.5, the LEL algorithm provides better estimations of robot path using the corresponding pairs $\mathbb{C}_P$ than with $\mathbb{C}_{NN}$. Therefore, this method is used in the final experiment.

8.5 Localization of Mobile Robot with Unknown Correspondences in the Real Environment

In this final experiment, the localization method based on the LEL estimation techniques is applied in the real outdoor environment. The input datasets are obtained from the Malaga set PARKING 0L by the method described in Section 6.3.2. The authors of the Malaga set provide the estimated path for the left camera; so, these values are used as an initial transformation for the LEL estimation and they are refered as odometry in the figures. Based on the previous evaluation, the representation of the rotation matrix R by the Euler angles is chosen because it is faster than the parametrization via quaternions and the influence of the goniometric functions is insignificant due to small angle values.

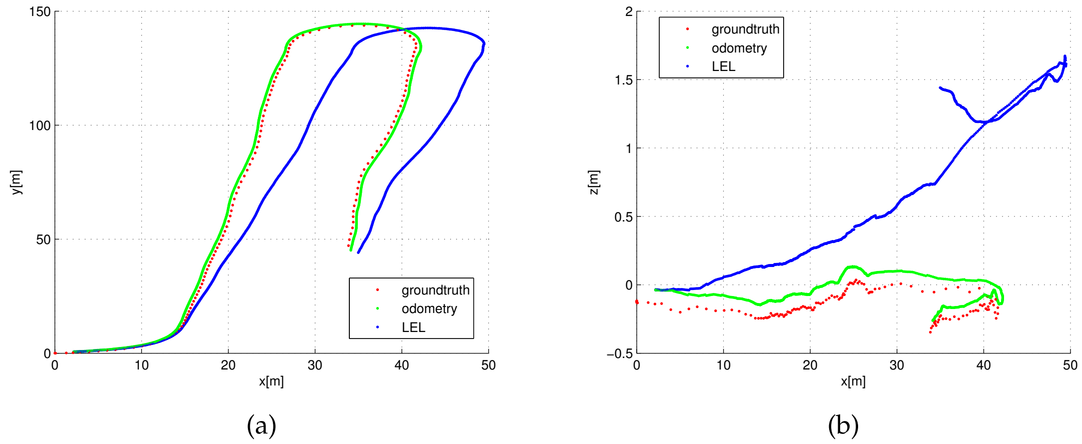


Figure 8.6: The global robot paths in xy-view (a) and xz-view (b).

The comparison of the global paths (sequences of the robot poses) is shown in Fig. 8.6. The constant difference between the odometry and ground truth paths is caused by the fact that the ground truth path is related to the center of the robot, but the odometry path is related to the center of the left camera. Regarding these figures, an altitude of the robot is significantly incorrect. This error is quite surprising; thus, the implementation of algorithm has been checked for a possible error. It has been found out that the error in the robot's altitude is really caused by the significant errors in the estimation of the angles, see the global yaw, pitch and roll angles in Fig. 8.7.

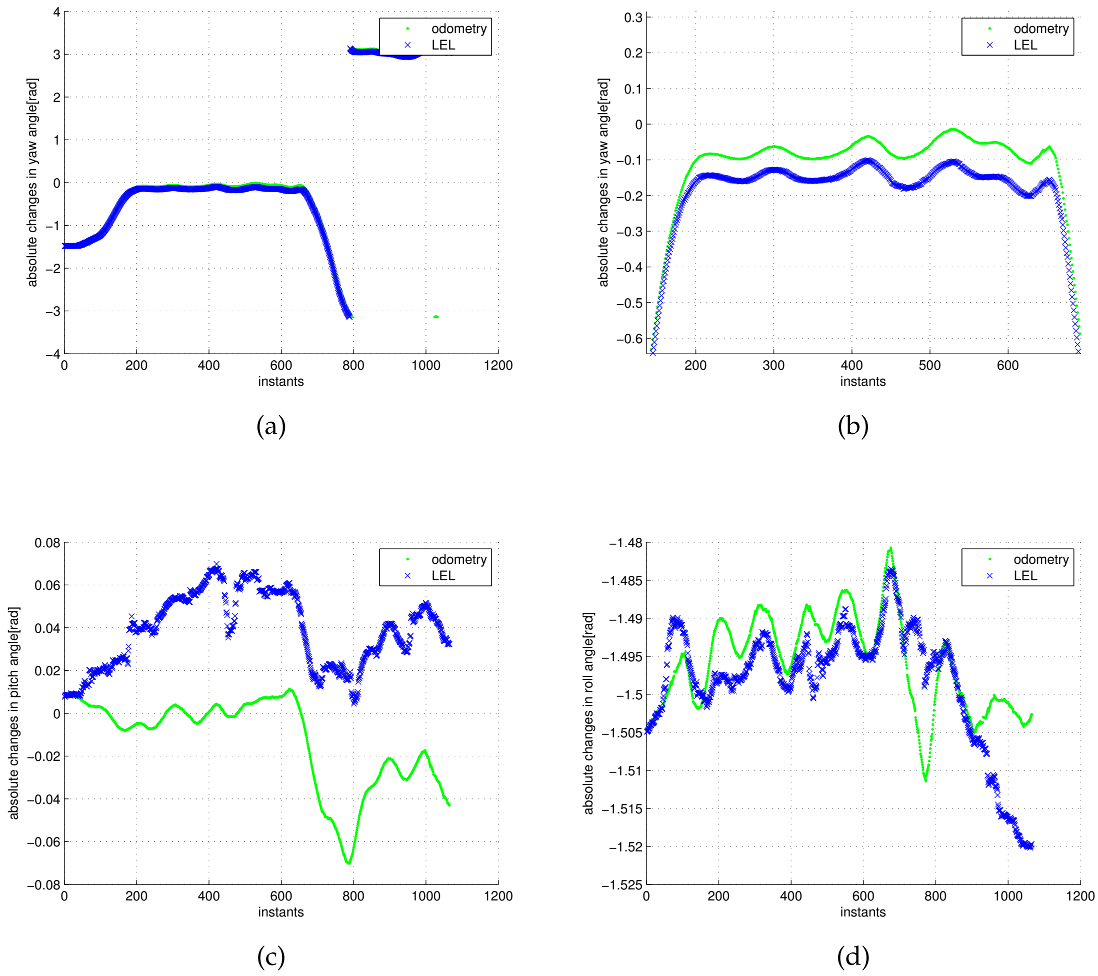
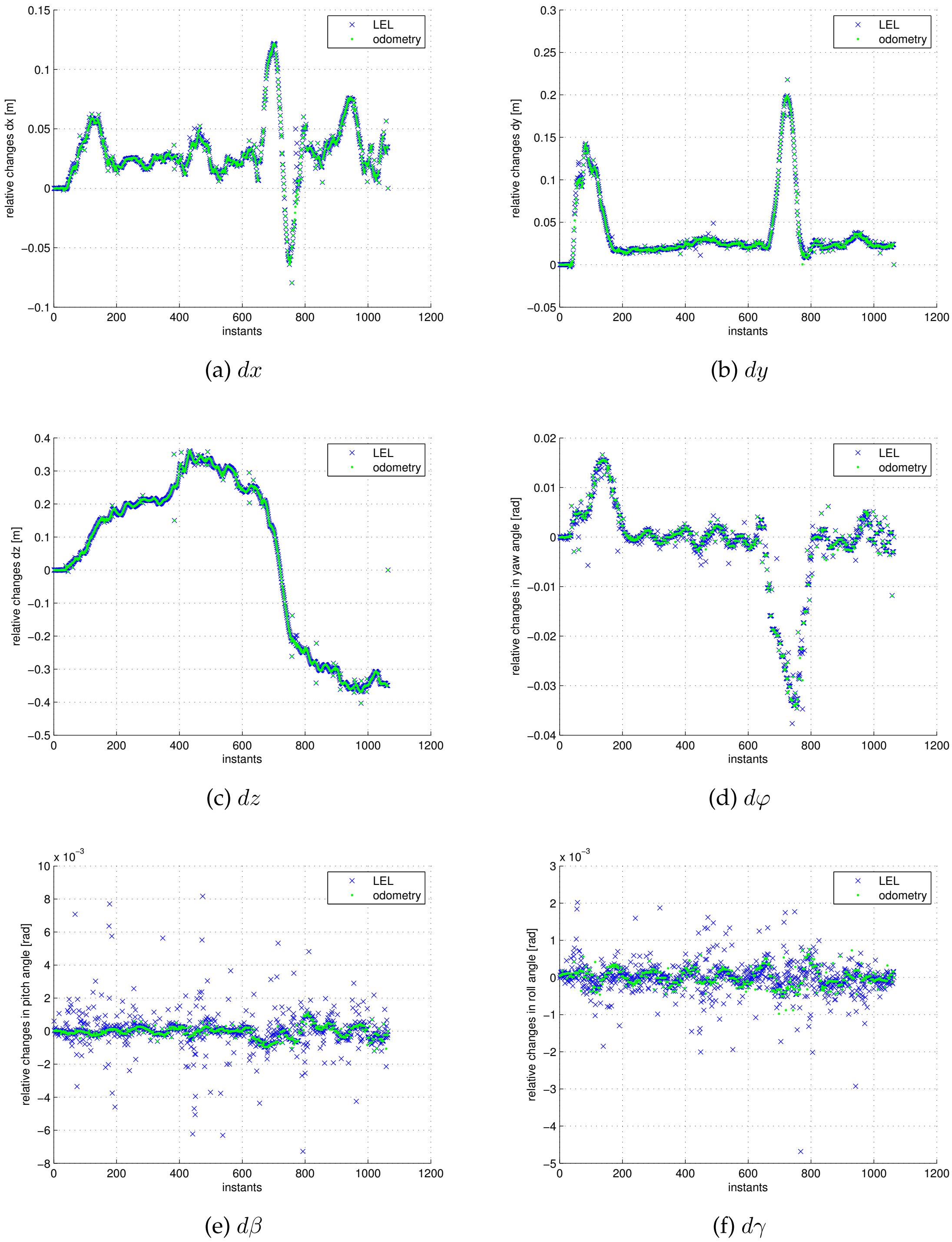


Figure 8.7: The global robot yaw(a,b), pitch(c) and roll(d) angles.

The robot is moving on the rough asphalt surface, which influences the rotation angles. The robot is heading forward by x-axis, see Fig. 6.2. The pitch angle β is a rotation around y-axis of the robot; thus, the pitch angle influence if the cameras (placed in the front part of the robot) are heading towards the ground or towards the sky. The significant pitch change can be seen in Fig. 8.7c between $600 - 800$ instants. The roll angle γ is a rotation around x-axis and it is plotted in Fig. 8.7d.

The precision of the LEL estimation for each parameter of the transformation $\mathbb{T} = [dx,\ dy,\ dy,\ \varphi,\ \beta,\ \gamma]$ can be compared from Fig. 8.8. In the shown sub-figures, the ground truth values are not plotted because the ground truth path contains smaller amount of the robot poses than the odometry in the same path section; so, the comparison of the relative increments is not possible. The odometry values are precise in this dataset; therefore, the LEL estimated parameters are compared according to them. The LEL estimation of the translation is precise, the blue (LEL) and green (odometry) graphs are overlapping with small differences. The LEL estimation of the rotation is influenced by the precision of the data association.



(a) dx (b) dy

(c) dz (d) $d\varphi$

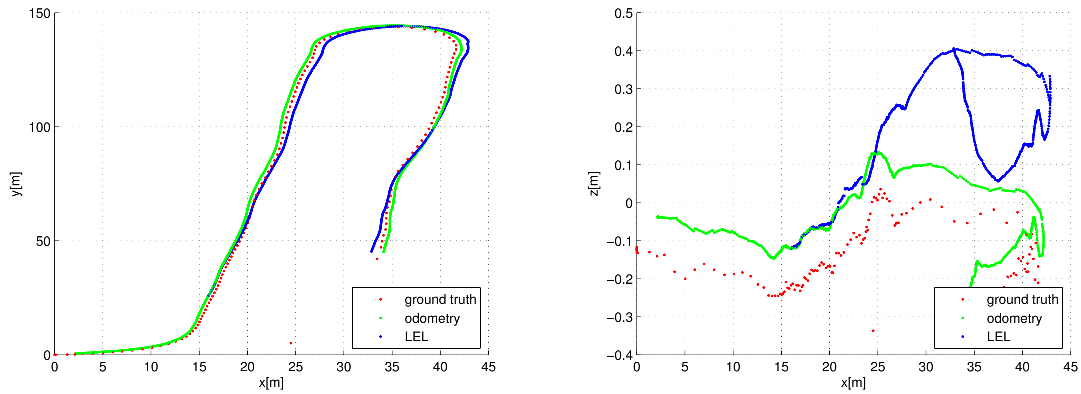
(e) $d\beta$ (f) $d\gamma$

Figure 8.8: The relative changes of the robot pose.

The precision of the global estimated path is also influenced by the fact that the LEL algorithm is used in the *scan-to-scan* approach. Therefore, the small errors in the estimation cause that the difference between the ground truth and estimated global pose increases. The odometry values are quite precise here; so, the influence of the noisy initial transformation is insignificant in this experiment. However, the unknown correspondences cause errors in the estimation of the transformation parameters. The influence of the data association is discussed with new findings in the following section.

8.5.1 Influence of Data Association

It can be observed from Fig. 8.6a that the differences between the estimated and the ground truth paths become significantly greater from the location $[x,\ y] = [15,\ 10]$, which approximately corresponds to the $120th$ instant. If the input images are evaluated for the instants $100 - 200$, the moving large object is found. The features detected on this object influence the data association and consecutively also the LEL estimation. Therefore, a new experiment is proposed. The same dataset from the Malaga set is used, but the starting robot pose is considered at the location $[x,\ y] = [16,\ 25]$; so, the images with the moving object are skipped. In this case, the LEL estimates a better global robot path than in the previous experiment, see Fig. 8.9. This experiment shows that the LEL behaviour significantly depends on the correct data association.



(a) (b)

Figure 8.9: The global robot paths in xy-view (a) and xz-view (b) with the robot initial position at $[16,\ 25]$ m.

Regarding the figures, the robot pose is estimated by LEL with less precision from the location $[x,\ y] = [25,\ 100]$ that corresponds to the $400th$ instant. The images are evaluated again and it is observed that the robot is crossing the empty crossroads at this location; so, only a few distant features are detected.

8.6 Summary

The Least-Entropy Like (LEL) method is proposed to be invariant to outliers in the input datasets. This property is validated in this thesis. The differences between the LEL estimation and the ground truth transformation are insignificant if datasets are corrupted by $10\,\%,\ 20\,\%,\ 50\,\%$ outliers. The impact of the initial transformation is verified as well. Regarding the presented results, the performance of LEL depends on the quality of the initial transformation and it has to be chosen carefully.

The influence of the parametrization of the rotation matrix to the optimization is tested too. It is shown that the representations via Euler angles and via quaternions have the same behavior for small changes of the angle ($\alpha \sim 0.5°$). The quaternion representation provides a more precise estimation than the Euler angles for angle changes $\alpha \gg 0.5°$. In the localization of a mobile robot, the changes of the angle are small, and therefore, the representation using the Euler angles is sufficient. Moreover, the optimization process with parametrization via Euler angles is faster than via quaternions.

The authors of LEL support behaviour of the localization technique only by experiments where known exact correspondences are used [7]. This thesis extend these experiments and provides additional study of the LEL behaviour for the cases where perfect correspondence are known as well as unknown. The presented results shows that LEL estimates the transformation precisely using known correspondences. On the other hand, unknown correspondences significantly influence the LEL estimation. Therefore, the new data association method is proposed in this thesis to reduce the influence of wrong corresponding pairs to the estimation. The comparison of the standard NN method and the new proposed method is verified and the results indicate that the new proposed method significantly improves performance of the LEL estimation.

Finally, the LEL is verified in a scenario within a real outdoor environment. Such an evaluation has not yet been done (to the best of our knowledge), and therefore, it provides a more realistic expectations about the LEL performance in practical deployment of the method in the localization task of a mobile robot. The quality of the LEL estimation in this experiment (in the sense of proposed terminology in Section 6.1) is $q = 16.52$. The error is caused by a wrong correspondences due to a large moving object. The experiment without data from moving object is done and the quality of the LEL estimation is $q = 2.6$; so, it is better than in the previous case. The robot travels along the 220 m long path, the error of the final estimated pose and ground truth is only 0.92 m in xy plain and 0.65 m in xz plain, which represents 0.5% of the total length of the traveled path.

Regarding the results, the proposed localization method using the Least-Entropy Like estimation technique and data association method with filtration seems to be an appropriated for the mobile robot localization task in an outdoor environment. The LEL estimations can be significantly improved by filtration of features detected on moving objects.

Chapter 9

Conclusion

In this thesis, a problem of mobile robot localization is studied. In particular, the thesis is focused on an evaluation of the new *Least Entropy-like* (LEL) estimation technique employed in finding parameters of the transformation describing a robot motion; hence, an estimation of the robot global pose. The parameters of the transformation are determined from two sets of observations of the robot's surroundings acquired by a robot's sensor system at two consecutive time instants.

The LEL estimation method has been proposed in 2009 [16] and its application in the mobile robot localization task has been proposed relatively recently in 2011 [7]. The LEL estimation has been designed to be robust to outliers, and therefore, it seems to be a suitable technique for solving the localization problem. However, it has not been widely used in robotics due to its short history, and therefore, the main goal of this thesis is to evaluate the performance of the LEL method in various scenarios of the mobile robot localization problem, especially in outdoor environments.

Moreover, the LEL estimation technique is compared with other well-established estimation methods (i.e., already considered in mobile robotics), such as methods based on the ICP or RANSAC. A summary of the gained experiences and achieved results during solving the diploma thesis are described in the next subsections together with an overview of the developed localization system. Finally, in Section 9.2 a possible future work is proposed.

9.1 Summary of the Gained Experiences and Achieved Results

First, I acquainted myself with the new proposed LEL technique for parameters estimation and its application in the mobile robot localization task. The idea of LEL is based on rewarding the presence of majority of low relative errors and penalizing a minority of large ones, i.e., finding such a cost function that is able to "globally" measure the residual dispersion and to reward evenly distributed residuals. The requested properties of the cost function can be fulfilled by the definition of the entropy. I have implemented the LEL estimation algorithm in Matlab software and then, I have verified the implementation correctness by repeating the same experiments as the authors of LEL have done. I have proposed the first experiments to verify the robustness to the presence of outliers, see Section 7.1. The achieved results confirm the robustness of the algorithm to this kind of noise.

During the work on the thesis, we have observed that the application of the LEL algorithm in a mobile robot navigation using the rangefinder sensors is a more demanding task than we have expected at the beginning of the work. This complication was caused by following difficulties. The first difficulty was caused by specific properties of the observations captured by a laser rangefinder. As explained in Section 4.2.1, the authors of LEL have used the Horn method for an initial estimation of the transformation's parameters, but this method estimates the parameters incorrectly when a laser rangefinder is used.

The second difficulty was caused by poor experiments that have been done by the authors of LEL. They have used only simulated data for which correspondences between observations are exactly known. The correspondences significantly influence the LEL behaviour as it is shown in Section 8.3. If the datasets from a laser rangefinder are used, the correspondences are unknown; so, the LEL performance using these datasets cannot be ever precise as for using known correspondences. Due to the time constraints caused by these difficulties, we decided to consider the mobile robot localization rather than on the mobile robot navigation task, see Section 1.2. This allows us to be focused on issues of the localization method and estimations technique itself.

A summary of the findings for indoor localization scenarios

I have studied different approaches to solve the mobile robot localization problem that use sets of observations sensed by the robot's exteroceptors. Namely, I have compared *scan-to-scan* and *scan-to-map* matching approaches in Section 5.1 with simultaneous localization and mapping (SLAM) technique described in Section 5.2. The scan-to-scan matching approach estimates the robot transformation using only two consecutive sets. In contrast, the *scan-to-map matching* approach estimates the transformation by matching the actual scan to the current map of the robot operational environment. This map can be pre-prepared or the robot can build it simultaneously while localizes itself according to this map, which is a problem known as SLAM.

Regarding SLAM, I acquainted myself with three main approaches: 1) the Extended Kalman filter (EKF) approach; 2) the graph-based optimization techniques; and 3) the particle filters approach. Here, it is worth to mention that an usage of SLAM approaches for the mobile robot localization is not the main goal of this thesis and SLAM is included mainly to provide a comparison with the studied localization approaches. Regarding the localization approaches, I have decided to focus on the *scan-to-scan* matching approach, because I assumed, the robot can operate in a completely unknown environment. Besides, the data association has less computational complexity than in the *scan-to-map* approach.

Regarding the particular evaluated localization methods, I have chosen the *Iterative Closest Point* (ICP), its extension *Iterative Dual Correspondences* (IDC) and the *Random Sample Consensus* (RANSAC) methods (Section 5.1) for a comparison with the new LEL estimation technique. I have proposed the indoor scenario for evaluation of all methods, where the laser rangefinder was used as the perception sensor.

Three different experiments have been performed to evaluate the LEL performance in the mobile robot localization in an indoor scenario. Two experiments were simulated in the Player/Stage framework where the robot was navigated along the square path (Section 7.2.1) and in an exploration task (Section 7.2.2). The third experiment was made using the real datasets from the Radish database (Section 7.3). The IDC algorithm provides the best estimation of the robot path from the tested algorithms, LEL provides better estimation than the ICP algorithm.

A summary of the findings for outdoor localization scenarios

In the second part of my diploma thesis, I have studied the localization problem using a visual features in outdoor environments. Here, six parameters of the robot pose are estimated; so, the observations have to be obtained in $\mathbb{R}^3$. For this purpose, a stereoscopic system containing two cameras is considered. I acquainted myself with the problem of the depth estimation from stereo images and I have decided to use an approach that determines the disparity between left and right images for an observation. The depth is then determined using this disparity and geometric information about the stereoscopic system. The observations (also called features) for disparity determination are extracted from the images using the computer vision features detection methods. I have decided to used the *Speeded-Up Robust Feature*(SURF) descriptor and its *Fast-Hessian* detector, because it outperforms all other studied detection methods, see Section 3.2.4.

The detected features from the right and left images have to be associated to the corresponding pairs for the correct disparity determination. Also, the features from two consecutive robot poses have to be associated for the estimation process. Therefore, I studied the data association problem and I have chosen the Nearest-Neigbour association method because it is used in many localization methods. In addition, I have proposed a new extension of the Nearest-Neigbour association method, which is based on the determination of the minimal distance criterion that maximizes the number of strong corresponding pairs, see Section 3.4. Moreover, the corresponding pairs are then compared based on the geometric properties of the whole batch and wrong correspondences are filtered out, which improves the performance of the LEL estimation technique noticeably, see Section 8.4.

The performance of LEL depends not only on the quality of the correspondences, but also on the initial estimation of the transformation parameters and on the used optimization algorithm. The initial estimation of the transformation parameters influences the found minimum during the Levenberg-Marquardt optimization because the optimized Entropy-like function has several local minima. Moreover, the optimization process is influenced by the representation of the rotation matrix, see Section 4.4, where parametrization using Euler angles and quaternions are compared. The sensitivity of LEL to the initial transformation and to the rotation matrix representation has been verified. In Section 8.1.2, it is shown that the initial transformation significantly influence the LEL estimation and the parametrization using the quaternions provides a better performance than the Euler angles, see Section 8.1.3.

Finally, I have proposed the outdoor scenario for verification of the developed localization system based on LEL. In this scenario, real measurements from the Malaga dataset are used. The authors of LEL tested the performance only in the situation where features correspondences were exactly known. Therefore, I have proposed the first experiment to observe the influence of the error in the correspondence association, see Section 8.3. The experiment showed that influence of the data association is significant, which motives me to propose a new data association method. A comparison (Section 8.4) of the new method with the original Nearest-Neighbour method shows that the new method significantly improves the estimation of the whole LEL estimator.

Finally, the developed localization system has been applied in the visual localization task in the real outdoor environment using the Malaga dataset. It has been found out that a large moving object seen at the beginning of the robot path cause a wrong data association, which consecutively leads to incorrect estimations of the robot's angles. If features on the moving object are not considered in the data association, the performance is significantly improve, see Section 8.5.1. Based on the achieved results, the overall behaviour of the localization method using the *scan-to-scan* matching (where errors are cumulated along the path) can be summarized that the system based on the LEL technique can be used for the mobile robot localization with an appropriate and careful data association.

An overview of the developed localization system for outdoor environments is summarized in the following steps:

1. The stereoscopic camera system is used to obtain images of the robot surrounding environment.

2. The important features are extracted from images using the SURF detection method.

3. The features from left and right images captured at the same robot pose are associated by the new proposed method.

4. The depth of features is determined using the disparity between corresponding features and the geometric properties of the used stereoscopic system; so, the features are defined in $\mathbb{R}^3$.

5. These features from two consecutive robot poses are associated to corresponding pairs.

6. These pairs and an initial estimation of the transformation parameters are used in Levenberg-Marquardt optimization technique that finds the local minimum of the Least-Entropy Like estimation function.

7. These estimations of transformation parameters are used to determine the global robot pose.

9.2 Future work

Regarding the achieved results, the localization system based on the Least-Entropy Like function seems to be promising for mobile robot localization in an outdoor environment. It has robust behaviour while the input datasets are corrupted by outliers, which is an important property for correct parameters estimation. Nevertheless, it significantly depends on an initial estimation of the transformation parameters and on the correct data association. The LEL method has provided the smallest time complexity from the verified algorithms, but it is not still sufficient for online localization. Thus, a future extension may be oriented to reduce these influences.

The influence of the initial parameters can be improved by estimating the Least-Entropy Like function using several rough estimations of the initial parameters. Then, the most probably estimation can be selected based on the minimal value of the entropy function. The robustness to data association can be improved by adding a method for detection of moving objects.

Another possible extension is an application of the proposed localization technique within the *scan-to-map* approach, where the estimations errors are not cumulated along the robot path.

Bibliography

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[2] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.

[3] José-Luis Blanco, Francisco-Angel Moreno, and Javier González. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4):327–351, November 2009.

[4] Matthew Brown and David Lowe. Invariant features from interest point groups. In *British Machine Vision Conference*, pages 656–665, 2002.

[5] Henrik I. Christensen and Gregory D. Hager. Sensing and estimation. In *Springer Handbook of Robotics*, pages 87–107. Springer, 2008.

[6] Aron J. Cooper. A comparison of data association techniques for simultaneous localization and mapping. Master's thesis, Massachusetts Institute of Technology, United States, 2005.

[7] Francesco Di Corato, Lorenzo Pollini, Mario Innocenti, and Giovanni Indiveri. An entropy-like approach to vision based autonomous navigation. In *ICRA*, pages 1640–1645. IEEE, 2011.

[8] James L. Crowley and Alice C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(2):156–170, February 1984.

[9] Frank Dellaert. A sample of Monte Carlo methods in robotics and vision. In *Proceedings of the ISM International Symposium on the Science of Modeling - The 30th Anniversary of the Information Criterion (AIC)*, pages 211–222, Yokohama; Japan, 2003.

[10] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo Localization for mobile robots. In *ICRA*, 1999.

[11] Lutz Falkenhagen. Depth estimation from stereoscopic image pairs assuming piecewise continuos surfaces. In *Image Processing for Broadcast and Video Production*, pages 115–127. Springer, 1994.

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[13] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[14] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[15] Joachim Hornegger and Carlo Tomasi. Representation issues in the ML estimation of camera motion. In *ICCV*, pages 640–647, 1999.

[16] Giovanni Indiveri. An entropy-like estimator for robust parameter identification. *Entropy*, 11(4):560–585, 2009.

[17] D. Vrancic J. Mrovlje. Distance measuring based on stereoscopic pictures. In *Proceedings of 9th International PhD Workshop on Systems and Control: Young Generation Viewpoint*, 2008.

[18] Yan Ke and Rahul Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, CVPR'04, pages 506–513, Washington, DC, USA, 2004. IEEE Computer Society.

[19] J. J. Koenderink and A. J. van Doom. Representation of local geometry in the visual system. *Biol. Cybern.*, 55(6):367–375, March 1987.

[20] Longin Jan Latecki and Rolf Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Comput. Vis. Image Underst.*, 73(3):441–454, March 1999.

[21] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1265–1278, August 2005.

[22] Jing Li and Nigel M. Allinson. A comprehensive review of current local features for computer vision. *Neurocomput.*, 71(10-12):1771–1787, June 2008.

[23] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

[24] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[25] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[26] Feng Lu and Evangelos Milios. Robot pose estimation in unknown environments by matching 2D range scans. *J. Intell. Robotics Syst.*, 18(3):249–275, March 1997.

[27] B. Rousso M. Irani and S. Peleg. Recovery of ego-motion using image stabilization. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 454–460, 1994.

[28] Donald W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[29] Rainer MAUTZ. Combination of indoor and outdoor positioning. In *1st International Conference on Machine Control and Guidance*, pages 79–88, 2008.

[30] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, October 2004.

[31] Hans Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, page 584, August 1977.

[32] Anastasios Mourikis. *Characterization and optimization of the accuracy of mobile robot localization*. PhD thesis, Department of Computer Science and Engineering, University of Minnesota, 2008.

[33] Rudy Negenborn. Robot localization and kalman filters - on finding your position in a noisy world. Master's thesis, Institute of Information and Computing Sciences, Utrecht University, 2003.

[34] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, ICPR '06, pages 850–855, Washington, DC, USA, 2006. IEEE Computer Society.

[35] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5):530 –535, may 1997.

[36] Jochen Schmidt and Heinrich Niemann. Using quaternions for parametrizing 3-D rotations in unconstrained nonlinear optimization. In *Vision, Modeling, and Visualization*, pages 399–406. AKA/IOS Press, 2001.

[37] John J. Leonard Sebastian Thrun. Simultaneous localization and mapping. In *Springer Handbook of Robotics*, pages 871–889. Springer, 2008.

[38] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainly. *International Journal of Robotics Research*, 5(4):56–68, December 1986.

[39] Global positioning system. `http://www.inventeksys.com/gps-facts/`.

[40] Vicon - motion capture system. `http://www.vicon.com/`.

[41] Open SURF method for Matlab. http://www.mathworks.com/matlabcentral/fileexchange/28300-opensurf-including-image-warp.

[42] Glonass - russian navigation system. http://www.glonass-ianc.rsa.ru/en/.

[43] Comparison test of features descriptors. http://computer-vision-talks.com/2011/08/feature-descriptor-comparison-report/.

[44] Pioneer robot. http://www.mobilerobots.com/researchrobots/PioneerP3DX.aspx.

[45] Robot Operating System. http://www.ros.org/wiki/.

[46] Gram-Schmidt orthogonal method. http://mathworld.wolfram.com/Gram-SchmidtOrthonormalization.html.

[47] Pioneer robot. http://www-lar.deis.unibo.it/equipments/p2dx/index.html.

[48] Matlab software,. http://www.mathworks.com/.

[49] Player/ stage. http://playerstage.sourceforge.net/.

[50] 3D camera. http://www.sick.com/group/EN/home/products/product_portfolio/vision/Pages/3d_cameras.aspx.

[51] A robotic cleaner. http://www.generation5.org/content/2005/sim2005.asp?Print=1.

[52] Adam - an automative industry robot. http://www.roboticstrends.com/industry_manufacturing/article/adam_mobile_robot_is_proven_tire_handling_solution.

[53] SICK - laser rangefinder. http://www.sick.com.

[54] Hokuyo - laser rangefinder. http://www.hokuyo-aut.jp/.

[55] Camera AVT Marlin F-131C. http://www.alliedvisiontec.com/.

CD Content

The CD is attached to the printed version of this work containing the text of the diploma thesis in a PDF format, source codes of thesis in $\LaTeX$ format and source codes of localization techniques. In following table the directory structure on the CD is described.

Table 1: Directory structure on the CD

Directory	File Description
`src`	source codes of localization techniques
`doc`	source codes of diploma thesis
`thesis.pdf`	text of diploma thesis