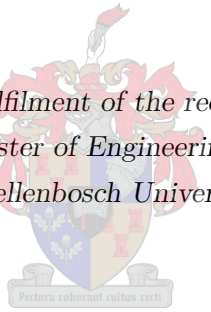
Probabilistic Outlier Removal for Stereo Visual Odometry

by

Alexander Chiu



Thesis presented in partial fulfilment of the requirements for the degree of
Master of Engineering
at Stellenbosch University

Supervisors:

Prof T. Jones Dr C.E. van Daalen
Department of Electrical and Electronic Engineering

March 2017

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature: A. Chiu .....

Date: March 2017 .....

Abstract

The field of autonomous navigation is currently receiving significant attention from researchers in both academia and industry. With an end goal of fully autonomous vehicle systems, an increased effort is being made to develop systems that are more efficient, reliable and safe than human-controlled vehicles. Furthermore, the low cost and compact nature of cameras have led to an increased interest in vision-based navigation techniques. Despite their popularity, measurements obtained from cameras are often noisy and contaminated with *outliers*. A critical requirement for consistent and reliable autonomous navigation is the ability to identify and remove these outliers when measurements are highly uncertain. The focus of the research presented in this thesis is therefore on effective and efficient *outlier removal*.

Many existing outlier removal methods are limited in their ability to handle datasets that are contaminated by a significant number of outliers in real-time. Furthermore, many of the current techniques perform inconsistently in the presence of high measurement noise. This thesis proposes methods for probabilistic outlier removal in a robust, real-time visual odometry framework. No assumptions are made about the vehicle motion or the environment, thereby keeping the research in a general form and allowing it to be applied to a wide variety of applications.

The first part of this thesis details the modelling of sensor measurements obtained from a camera pair. The mapping process from 3D space to image space is described mathematically and the concept of triangulating matched image features is presented. Stereo measurements are modelled as random variables that are assumed to be normally distributed in image coordinates. Two techniques used for uncertainty propagation, linearisation and the unscented transform, are investigated. The results of experiments, performed on synthetic datasets, are presented and show that the unscented transform outperforms linearisation when used to approximate the distributions of reconstructed, 3D features.

The second part of this thesis presents the development of a novel outlier removal technique, which is reliable and efficient. Instead of performing outlier removal with the standard hypothesise-and-verify approach of RANSAC, a novel mechanism is developed that uses a probabilistic measure of shape similarity to identify sets of points containing outliers. The measure of shape similarity is based on inherent spatial constraints, and is combined with an adaptive sampling approach to determine the probability of individual points being outliers. This novel approach is compared against a state-of-the-art RANSAC technique, where experiments indicate that the proposed method is more efficient and leads to more consistent motion estimation results. The novel outlier removal approach is also incorporated into a robust visual odometry pipeline that is tested on both synthetic and practical datasets. The results obtained from visual odometry experiments indicate that the proposed method is significantly faster than RANSAC, making it viable for real-time applications, and reliable for outlier removal even when measurements are highly uncertain.

Opsomming

Die area van outonome navigasie kry tans vele aandag van navorsers in akademie en in die bedryf. Met 'n einddoel van volledige outonome navigasie voertuigstelsels, word 'n verhoogde poging gemaak om stelsels te ontwerp wat meer effektief, betroubaar en veiliger is as menslik beheerde voertuie. Verder, die lae prys en kompakte struktuur van kameras het gelei tot 'n verhoogde belangstelling in visie gebaseerde navigasie tegnieke. Ten spyte van hierdie gewildheid, is kamera metings gewoonlik ruiserig en besoedel met *uitskieters*. 'n Kritiese vereiste vir konsekwente en betroubare outonome navigasie is die vermoë om uitskieters te kan identifiseer en verwyder as die metings hoogs onseker is. Die fokus van die navorsing wat in hierdie tesis aangebied sal word is dus op effektiewe en doeltreffende *uitskieterverwydering*.

Talle bestaande uitskieterverwydermetodes is beperk in hulle vermoë om datastelle besoedel met vele uitskieters intyds te kan hanteer. Verder, talle van die huidige tegnieke tree inkonsekwent in die teenwoordigheid van hoë ruis op. Hierdie tesis stel metodes voor vir waarskynliksheid-verwydering van uitskieters in 'n kragtige, intydse, visuele verplasingsmeter raamwerk. Geen aannames word gemaak oor die voertuig se beweging of die omgewing nie. Die navorsing word dus algemeen gehou en laat toe om toegepas te word op verskillende toepassings.

Die eerste gedeelte van hierdie tesis verduidelik die modellering van sensor metings geneem van 'n kamera paar. Die karteringsproses van 3D ruimte na beeld ruimte word wiskundig verduidelik en die konsep van triangulasie van ooreenstemmende beeldkenmerke word aangebied. Stereometings word gebruik as toevalsveranderlikes wat aanvaar word as normaal versprei in die beeld koördinate. Twee tegnieke wat gebruik word vir onsekerheid vooruitskatting, 'n lineariseringsmetode en die sigmapunt-transformasie, word ondersoek. Die resultate van eksperimente wat uitgevoer is op sintetiese datastelle word aangebied, en dit wys dat die sigmapunt-transformasie beter funksioneer as die lineariseringsmetode wanneer dit gebruik word om die verspreiding van gerekonstrueerde, 3D kenmerke te benader.

Die tweede gedeelte van hierdie tesis bied die ontwikkeling van 'n nuwe uitskieterverwyderingsmetode, wat betroubaar en doeltreffend is aan. In plaas van uitskieters te verwyder met RANSAC se standaard tegniek van hipotetiseer-en-verifieer, word 'n nuwe meganisme ontwikkel wat vorm ooreenkoms meet om stelle punte wat uitskieters bevat te identifiseer. Die meting van vorm ooreenkoms is gebaseer op ingebore ruimtelike beperkings en word gekombineer met aanpasbare monstering om die waarskynlikheid van sekere punte om uitskieters te wees te bepaal. Hierdie nuwe benadering word vergelyk teen RANSAC waar eksperimente wys dat die voorgestelde metode meer doeltreffend is en lei tot meer konsekwente resultate. Die nuwe uitskieterverwyderingsmetode is ook opgeneem in 'n kragtige visuele verplasingsmeter wat getoets is met beide sintetiese en praktiese datastelle. Die resultate wat behaal is van die visuele verplasingsmeter eksperimente dui aan dat die voorgestelde metode aansienlik vinniger is as RANSAC, wat dit haalbaar maak vir intydse toepassings, en betroubaar is vir uitskieterverywydering al is die metings hoogs onseker.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations and Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
AGV	Autonomous Ground Vehicle
AUV	Autonomous Underwater Vehicle
DATMO	Detection and Tracking of Moving Objects
DoF	Degrees-of-Freedom
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
GPS	Global Positioning System
IMU	Inertial Measurement Unit
KL	Kullback-Leibler
KLT	Kanade-Lucas-Tomasi
MLESAC	Maximum Likelihood Estimation Sample and Consensus
PnP	Perspective-from-n-Points
PORUS	Probabilistic Outlier Removal Using Shapes
PROSAC	Progressive Sample and Consensus
PTAM	Parallel Tracking and Mapping
RANSAC	Random Sample and Consensus
SfM	Structure-from-Motion
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localisation and Mapping
SSD	Sum of Squared Differences
SURF	Speeded-Up Robust Features
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter

Notation

x	Scalar
X	Axis or random variable
$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix

$\mathcal{X} = \{x_1, \ldots, x_n\}$	Set
$\mathcal{X} = \langle x_i \rangle_{i=1}^{n}$	Sequence
$\mathcal{X}^{[i]}$	The i^{th} element in sequence $\mathcal{X}$
$\hat{x}$	Estimate of x
$\bar{x}$	Average of x
σ_x^2	Variance of x
$\boldsymbol{\mu}_{\mathbf{x}}$	Mean of $\mathbf{x}$
$\boldsymbol{\Sigma}_{\mathbf{x}}$	Covariance of $\mathbf{x}$
$\mathbf{J}_f$	Jacobian of f
$H(X)$	Entropy of X
$P(X = x)$	Probability of event x
$p(x)$	Probability density function of X

Coordinate systems

X, Y	Image axes
X_L, Y_L	Left image axes
X_R, Y_R	Right image axes
X_c, Y_c, Z_c	Camera axes
X_r, Y_r, Z_r	Robot axes
X_w, Y_w, Z_w	World axes

Variables

b	Baseline
f	Focal length
c_x	x-offset of principal point
c_y	y-offset of principal point
$\mathbf{P}_c$	Camera projection matrix
$\mathbf{K}$	Camera calibration matrix
$\mathbf{R}$	Rotation matrix
$\mathbf{t}$	Translation vector
$\mathbf{P}_k$	Pose of robot at time k
$\mathbf{T}_{k,k-1}$	Relative pose from time $k-1$ to time k
$\mathbf{z}_k$	Observation at time k
σ	Noise level
Δ	Threshold
η	Probability of success
ϵ	Inlier ratio
m	Number of model dimensions
N	Number of points
k_r	Number of RANSAC iterations
$\mathcal{I}$	Set of inliers/inlier indices
$I_{G,u}$	Information gain of action u

α	True positive rate
β	False positive rate
M_{good}	Number of accurate model estimates

Acknowledgements

"Sometimes our light goes out but is blown into flame by another human being. Each of us owes deepest thanks to those who have rekindled this light."

— Albert Schweitzer

I firmly believe that I would not have been able to get through the past two years if it were not for the support and encouragement of those around me. I would therefore like to take the time to thank several people for contributing towards this project.

First, I would like to express my sincere gratitude to my two supervisors, Dr Corné van Daalen and Prof. Thomas Jones. They managed to survive a multitude of topic changes along the way, while providing continuous support and guidance that have made this research possible. They have been patient and shared so much knowledge whilst allowing me the freedom to work in my own way.

A big thank you must also go out to everyone in the Electronic Systems Laboratory, past and present, who have made my postgraduate studies such a memorable time in my life. Each and every outing, procrastination session and heated debate (both related and unrelated to research) at the coffee machine was thoroughly enjoyed.

A special thank you goes out to old friends, who have shared this journey with me for many years, and to new friends who I have made along the way. Your friendship is greatly appreciated.

Finally, I must express my appreciation to my parents for providing me with unfailing support (especially during the last push) and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Chapter 1

Introduction

Autonomous navigation is currently a major topic of interest for researchers across a multitude of industries and fields. Over the past few years, there has been an increased effort to develop fully autonomous vehicle systems capable of operating in generic environments with minimal human intervention. Competitions like the DARPA Urban Challenge [1] have become famous for showcasing state-of-the-art autonomous systems to the public. Interest from industry is also at an all-time high; Google's self-driving cars [2] have completed over 3 000 000 km of fully autonomous driving on public roads and Tesla Motors' AutoPilot [3] system, which provides semi-autonomous driver assistance, is standard on several of their publicly-sold vehicles. The idea of autonomous systems is gaining traction with the general population; however, these systems are often still dependent on accurate, pre-determined maps of the environment. Furthermore, many existing systems are only capable of operating in regulated environments [4]. Truly autonomous navigation systems have therefore not been achieved and research in the field remains of high interest.

Camera systems are often used as primary sensors in autonomous navigation projects due to their lightweight and relatively low cost. Autonomous navigation platforms that make use of cameras include self-driving cars [2, 3] and other autonomous ground vehicles (AGVs) [5, 6], unmanned aerial vehicles (UAVs) [7–9], as well as autonomous underwater vehicles (AUVs) [10]. However, despite their popularity, camera measurements can be highly uncertain. Moreover, outliers – statistical anomalies that are inconsistent with system and measurement models – are not uncommon when working with visual measurements. Consequently, computer vision techniques used in autonomous navigation frameworks must be robust to both highly uncertain measurements and outliers. Noisy inlier measurements can be robustly handled by modelling their uncertainty, but this requires outliers to be removed first. However, it is difficult to distinguish between inliers and outliers when measurements are highly uncertain. The focus of this thesis is therefore on robust outlier removal for vision-based autonomous navigation pipelines.

This chapter provides a motivation for the work performed in this thesis. It begins with a brief overview of autonomous navigation and introduces several of the sub-systems present in an autonomous navigation framework. The focus then shifts towards vision-based localisation techniques and the rationale behind the use of visual sensors for autonomous navigation. A literature study is performed on a widely-used localisation technique known as visual odometry. Thereafter, outliers in the context of computer vision are discussed, followed by a basic review of outlier removal techniques implemented in modern visual odometry applications. After the relevant literature has been discussed, the objectives of this research are formally defined. Lastly, an overview of this thesis is provided, which details the key results and findings of the remaining chapters.

1.1 Overview of Autonomous Navigation

Autonomous navigation can be described as the process of guiding a robot[1] through a known or unknown environment, from a start to a goal location, with little or no human intervention. This involves identifying

[1]The term *robot* is preferred over vehicle as it encapsulates both physical and virtual artificial agents.

obstructions that may be encountered, and controlling the position of the robot relative to the environment. A further requirement is the ability of the robot to move through the environment safely, and within the bounds of some acceptable performance metric. From these requirements, the task of autonomous navigation can be broken down into several interconnected sub-problems, which are discussed in this section.

The design of an autonomous navigation system can be formulated in several ways. One such design, describing a generalised autonomous navigation framework, is provided in Figure 1.1 and is based on a framework first proposed in the thesis of Van Daalen [4]. This is the structure used in autonomous navigation research performed within the Electronic Systems Laboratory of Stellenbosch University and is also assumed in this thesis. There are three main modules required for fully autonomous navigation [11],

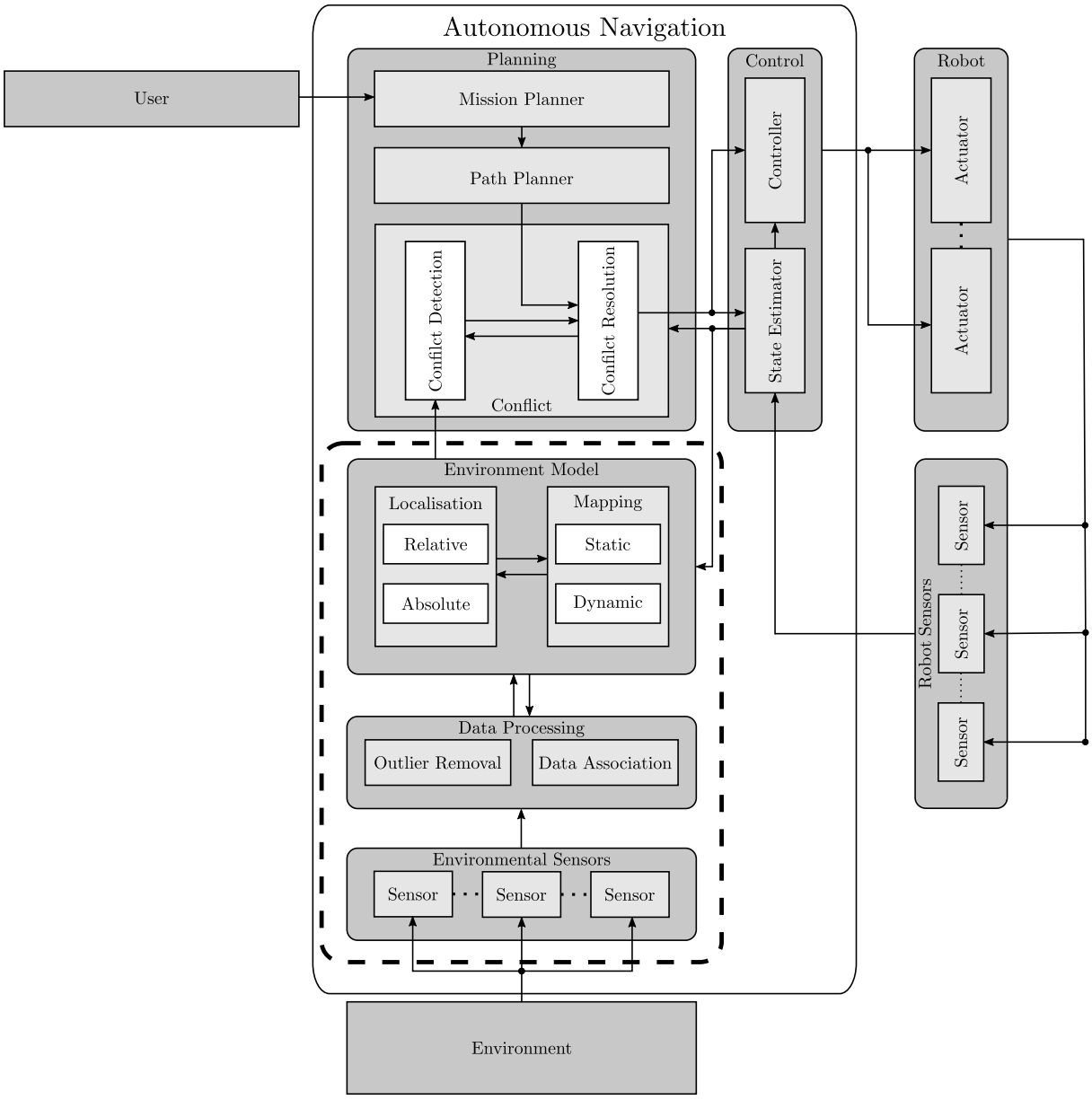


Figure 1.1: A generalised, autonomous navigation framework adapted from the thesis of Van Daalen [4]. The scope of the research performed in this thesis is outlined (dashed).

namely, environmental modelling, planning and control. Additional components in the system include sensors, which measure both robot and environmental states, and actuators mounted to the robot frame. Environmental sensors are exteroceptive sensors, such as range or vision sensors, that capture

measurements of objects relative to the robot. In contrast, robot sensors are proprioceptive sensors, such as a global positioning system (GPS) or wheel encoders, that measure internal states of the robot. Considering that multiple environmental sensors may be capturing measurements of a single object in the environment, uncertainty exists with regards to the correspondence of measurements and environmental objects. It is therefore necessary to process the measurements obtained from the sensors before they are incorporated into an autonomous navigation system during a stage known as data association. As already mentioned, sensor readings may contain statistical anomalies; it is therefore necessary to perform outlier removal to identify and remove incorrect data associations. A brief overview of the main components in an autonomous navigation system is given as follows.

Environment Modelling. The environment modelling module is responsible for the spatial representation of the robot and its surroundings. This task usually consists of two components, namely, localisation and mapping. Mapping is the process of integrating measurements of the environment from multiple sensors into a common representation and fusing sensor measurements with any pre-existing map information of the environment. Localisation, on the other hand, is concerned with determining the relative pose of the robot in the reference frame of the environment using the available sensors. Although separate problems, these two components are largely interdependent and are often solved together in a framework known as simultaneous localisation and mapping (SLAM) [12, 13]. Additionally, environments which contain non-static objects require the mapping module to distinguish between static and dynamic objects. This requires the implementation of a detection and tracking of moving objects (DATMO) [14, 15] process within the environment modelling module.

Planning. The planning module determines the actions a robot should perform to accomplish some predetermined task. This process consists of several stages – first, a mission planner interprets input from a human user to construct a set of high-level objectives for the robot. Secondly, a path planner generates a route, usually in the form of waypoints, for the robot to follow in order to achieve the objectives. A conflict avoidance module operates in conjunction with the path planner to ensure that the chosen path is free of predicted conflict or collisions with any objects in the environment. Within the conflict avoidance module, a conflict detection system uses the environmental model and the current estimate of the robot states to calculate the probability of collisions for the predicted robot trajectory [4]. Conflict resolution is then performed to generate an updated path that reduces the probability of conflict, while still guiding the robot to the next waypoint as determined by the path planner.

Control. The control module is responsible for accurately steering the robot along the trajectory generated by the planning module. Estimates of the robot states, obtained from on-board sensors, are passed to a controller unit. The controller considers the dynamics of the robot and generates control commands that are executed by the robot's actuators.

The performance of the autonomous navigation framework as a whole is highly dependent on accurate robot localisation [16]. This is motivated by the fact that both the planning and control modules require accurate estimates of the robot's pose in order to avoid conflict and move through the environment. The relationship between localisation and mapping, however, is more complex. Accurate maps are critical for localising correctly, while accurate localisation is a pre-requisite for high quality maps. Robust and accurate localisation will therefore be the primary focus of this work while path planning and control fall outside the scope of this project (outlined in Figure 1.1).

1.2 Visual Navigation

A multitude of sensors have been incorporated into systems attempting to solve the autonomous navigation problem with varying levels of success. Solutions have been proposed which make use of dead-reckoning

sensors, such as encoders [17] and inertial measurements units (IMUs) [18], as well as relative sensors such as radar [19] and lidar [20], or an absolute sensor in the form of a GPS [21]. In recent years, however, the use of visual sensors for navigation purposes has increased markedly.

The popularity of vision sensors has risen for several reasons. Camera systems are relatively low-cost sensors in comparison to lidars and radars [22], as well as compact and lightweight, making them suitable for use on small-scale robotic systems. Reliable data association is also difficult when working with radar and lidar measurements as features in the environment are not easily recognisable and distinguishable [23]. Dead-reckoning sensors are cheap and have high frame rates, but suffer from significant accumulative drift issues [24]. Furthermore, the amount of information captured in a single time step is far greater for cameras than other sensors [25]. Vision sensors are also well suited for operating in indoor environments as opposed to systems dependent on GPS, which rely on line-of-sight communication with satellites.

Several configurations of camera systems have been used successfully in visual navigation frameworks. Se et al. [26] developed a SLAM system which used a stereo camera pair to track visual features detected in the environment. Their system was capable of concurrently estimating a robot's pose and generating a sparse 3D map of the environment; however, the system was unable to perform global localisation – that is, the robot could not localise itself using the map when picked up and placed in arbitrary positions. Several other projects have also implemented stereo systems successfully in their solutions to the SLAM problem [22, 27, 28]. The vSLAM algorithm, proposed by Karlsson et al. [23], uses a factorised variation of the SLAM formulation [29] to build a map from visual landmarks captured using a single camera. As opposed to the work of Se et al. [26], vSLAM is able to recover when the robot is moved to an arbitrary position. Other attempts at using a single camera for localisation include a parallel tracking and mapping (PTAM) implementation by Klein and Murray [30] and a corner-based SLAM system for indoor navigation by Celik et al. [31]. RGB-D cameras [32], sensors that return both image and depth information, have also been used for autonomous navigation purposes [33–35].

As mentioned previously, one of the major advantages of visual sensors is the large amount of information captured in a single time step. However, this also leads to a high computational cost when processing data – a major concern of visual SLAM frameworks. The total number of features grows quickly when working with vision sensors making the map difficult to maintain. For this reason, the majority of visual SLAM solutions are restricted to small-scale environments [36], although recent work has been aimed at supporting larger environments [37, 38]. This concern was identified by Davison et al. [25] in their MonoSLAM algorithm, a real-time SLAM solution capable of reconstructing the trajectory of a moving monocular camera. Davison et al. [25] stated that for systems which require real-time localisation for other components, such as planning and control, it is more important to accurately obtain the pose of the robot, rather than a detailed map of the environment. Their approach therefore used a sparse map of landmarks to allow for accurate localisation in real-time.

An alternative approach to SLAM is to estimate the ego-motion of the robot directly without maintaining a map. This is known as visual odometry and will be discussed in detail in Section 1.2.1. Visual odometry and visual SLAM are both techniques which try to obtain consistent robot pose estimates as the robot moves – that is, the robot trajectory. However, despite this similarity, their core philosophies are different. Visual SLAM is more concerned with obtaining a globally consistent map, while visual odometry only maps features locally in order to localise accurately. This simplification allows visual odometry to reconstruct the robot's trajectory incrementally, and makes the technique more adept at running in real-time. The work in this thesis will therefore focus on accurate localisation, akin to the work of Davison et al. [25], using visual odometry rather than performing SLAM.

1.2.1 Visual Odometry

The task of estimating the ego-motion of a robot using only visual input is known as visual odometry. This term was popularised by Nistér et al. [5] and inspired by wheel odometry which measures wheel

rotations with encoders to estimate the trajectory of wheeled robots. Conceptually, visual odometry can be viewed as a specification of the structure-from-motion (SfM) [39] problem, and is based on the measurement of incremental changes in pose. SfM determines the 3D reconstruction of both the scene and camera poses from sequential or unordered image sets, whereas visual odometry ignores the structure of the environment, and focuses on estimating camera poses sequentially in real-time as new images become available.

The first investigations into ego-motion estimation of robotic vehicles using cameras were initiated in the early 1980's. Moravec [40] worked on one of the earliest vision-based localisation projects which made use of a single camera and sliding rail connected to a rover that captured "stereo" images in a stop-and-go fashion. The pipeline for motion estimation proposed by Moravec – detecting corners in an image, matching features between different views and calculating the relative body transformation across viewpoints – was the first of its kind and has remained largely unchanged in subsequent visual odometry implementations. The work done by Moravec was further developed by Matthies [41] to incorporate stereo cameras. A camera pair was used to estimate the 3D positions of landmarks[2] in the field of view of a robot, as well as to estimate the motion of the robot as it moved through an unknown environment. Over the past two decades, visual odometry has attracted considerable attention from researchers in the autonomous navigation and robotics community [42–44]. Notably, visual odometry has even been used in applications such as the Mars Rover [45, 46]. The interest in visual odometry stems from the high accuracy of estimated trajectories compared to those obtained from wheel encoders, which suffer from slippage in uneven terrain [46]. Furthermore, the use of visual odometry is not limited to wheeled robots and can be used to estimate 6 degrees-of-freedom (DoF) motion. This allows the trajectories of robots with more complex motions to be estimated [7, 47, 48]. Moving away from existing literature, a generalised visual odometry problem statement is now formulated.

Consider a robot moving through an unknown environment along some trajectory as depicted in Figure 1.2. It is assumed that the robot in question is equipped with a body-fixed stereo camera system and is capable of 6DoF motion. The pose of the robot, $\mathbf{P}_k$, describes the rotation and translation of the body-fixed robot coordinate system (denoted by subscript r) in the world coordinate system (denoted by subscript w) at a discrete time step, k. The relative pose between robot poses at successive time steps is expressed as a rigid body transformation [36],

$$\mathbf{T}_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}_{3\times 1} & 1 \end{bmatrix}, \tag{1.1}$$

where $\mathbf{R}_{k,k-1} \in SO(3)$ and $\mathbf{t}_{k,k-1} \in \mathbb{R}^{3\times 1}$ describe the relative rotation and translation between poses respectively, such that $\mathbf{P}_k = \mathbf{P}_{k-1}\mathbf{T}_{k,k-1}$. Now consider, a camera system rigidly connected to the robot that captures images as the robot moves through the environment from different viewpoints. If $\mathbf{T}_{k,k-1}$ can be determined from successive viewpoints, the pose of the robot relative to an initial coordinate frame can be reconstructed by concatenating relative transformations. The current pose $\mathbf{P}_n$ of the robot is calculated by

$$\mathbf{P}_n = \mathbf{P}_0 \times \mathbf{T}_{1,0} \times \mathbf{T}_{2,1} \times \ldots \times \mathbf{T}_{n,n-1}, \tag{1.2}$$

where $\mathbf{P}_0$ is the initial pose relative to world coordinates, which can be arbitrarily chosen, and the full trajectory of the robot is described by the sequence, $\mathcal{P} = \langle \mathbf{P}_i \rangle_{i=0}^{n}$. It is important to note from Equation 1.1 that errors incurred during the estimation of individual transformations, $\mathbf{T}_{k,k-1}$, accumulate over the trajectory of the robot. This is known as motion drift and is an important consideration when performing visual odometry.

Looking at the ego-motion estimation problem presented, it is now possible to develop a visual odometry pipeline capable of solving the problem at hand. A standard visual odometry solution can therefore be described in terms of three main components.

[2] *Landmarks* are stationary features in the environment which are easily identifiable and distinct.

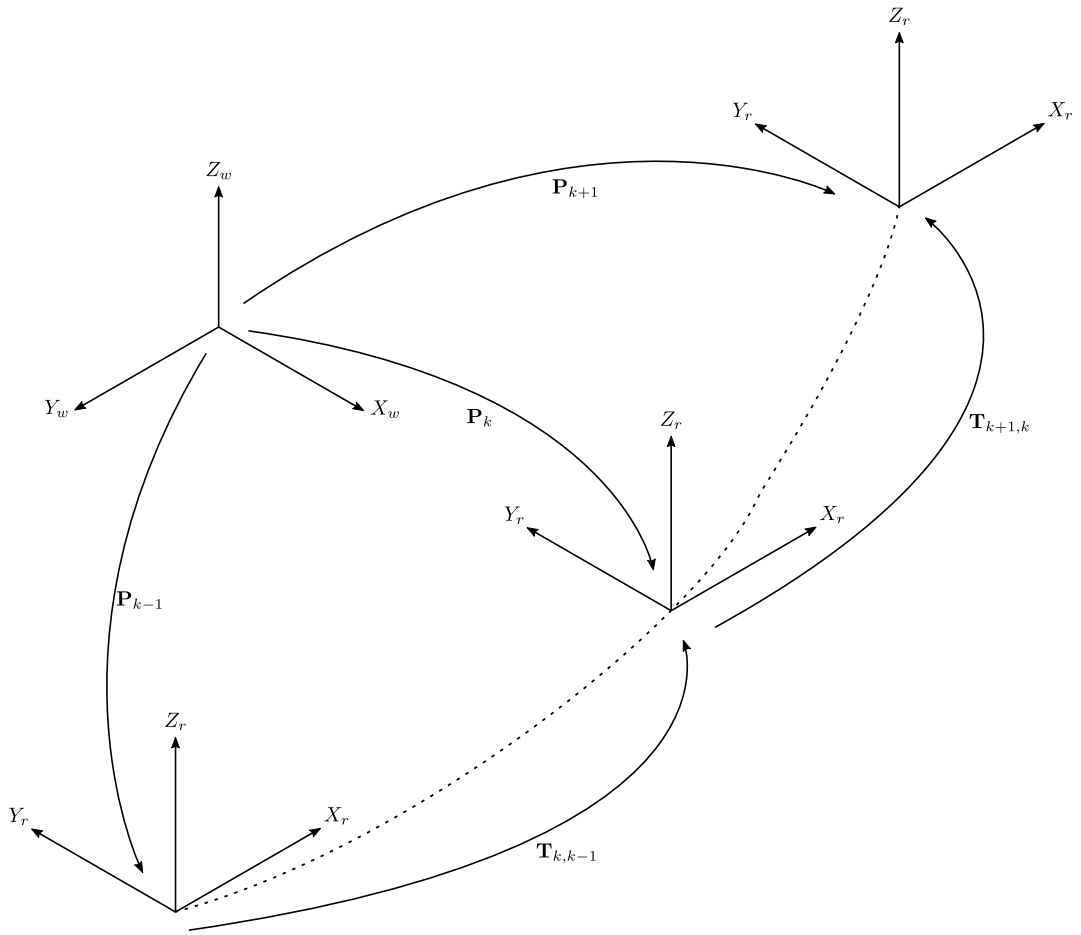


Figure 1.2: Graphical representation of the visual odometry problem for a robot moving along an abritary path (dashed line). The relative pose (rigid body transformations) between successive time steps, $\mathbf{T}_{k,k-1}$, is determined by tracking visual features across viewpoints. Transformations are accumulated over time to obtain the pose of the robot, $\mathbf{P}_k$, in the world coordinate system. Adapted from *Visual Odometry: Tutorial* by Scaramuzza and Fraundorfer [36].

Feature Detection. The first stage of the visual odometry pipeline determines easily identifiable features in image frames. These prominent points allow 3D landmarks to be calculated and used to estimate the trajectory of the robot. Several properties have been identified in literature that determine the quality of features [49–51]. One such property of image features is repeatability – that is, the same feature should be identified across several images of the same scene. Furthermore, each landmark should be distinct enough to construct unique feature descriptions, and accurately localised in image space. Lastly, the detection of features should be efficient enough for use in time-critical applications.

There is no shortage of literature available on various feature detection techniques used in autonomous navigation frameworks. Point features, such as corners and blobs[3], are usually preferred over edges and contours due to a greater accuracy in feature location measurement [52]. Moravec [40] proposed one of the earliest corner detectors in his visual odometry framework. Corners were identified by determining the sum of squared differences (SSD) of image patches that were shifted in a number of directions, and isolating points of interest. The Harris corner detector [53] extended the work done by Moravec by considering the differentials of corner scores with respect to direction, rather than simply using shifted patches. Other notable corner detectors are the features from accelerated segment test (FAST) detector by Rosten and Drummond [54], which is known for high computational efficiency, as well as Shi-Tomasi features [55], which were proposed as an improvement to the original Harris corner detector.

Two of the more popular blob detectors are the scale invariant feature transform (SIFT) [56] and

[3] *Corners* describe points of intersection between two or more lines whereas *blobs* are small regions of an image which are distinct.

speeded-up robust features (SURF) [57]. The SIFT detector convolutes a difference of Gaussian operator with images at various scales and identifies regions of interest as local optima in scale and space. Although extremely robust to changes in scale and viewpoints, SIFT suffers from high computational costs. SURF was developed as an extension to SIFT to overcome the computational complexity of SIFT by using box filters to approximate Gaussian operators [52]. Each blob is then converted into a descriptor that captures properties, such as the appearance, orientation and gradients of the blob, which enables direct comparison between identified features.

The use of corner and blob detectors are application dependent; corner detection is more computationally efficient, but less distinctive, while blobs are more readily identified under large scale changes. The correct choice of detector is therefore often dependent on the environment. Detailed overviews of the aforementioned corner and blob detectors, that compare performance and accuracy in the context of SfM, are available in literature [52, 58, 59].

Feature Correspondences. Once features have been identified in an image frame, the next step in the pipeline identifies corresponding features in other frames. There are two approaches that can be followed to match features between image frames. The first approach is feature tracking – that is, features are identified in a single frame, and then tracked by searching for the corresponding features locally in another frame. In the second approach, feature matching, points of interest are identified independently across multiple frames, and matched according to descriptor similarity scores.

Feature tracking is generally only applicable where the motion between successive image frames is small. If the relative transformation is large, appearance deformation occurs and features are not as easily tracked. The early visual odometry frameworks [40, 41] were developed for small-scale environments where small relative motions were assumed and feature tracking was implemented using correlation in conjunction with prior knowledge of the robot's motion. The Kanade-Lucas-Tomasi (KLT) [55, 60] tracker was developed to allow for the tracking of features over larger motions by incorporating an affine transformation model; however, feature matching is often preferred in modern visual odometry frameworks [6].

In the feature matching approach, features identified across multiple frames are matched by comparing the descriptors of each feature. This is more suitable for finding corresponding features when large motions have occurred between viewpoints. Possible feature matches are scored in terms of some metric such as SSD or a Euclidean distance [52]. The naïve feature matching approach is quadratic in complexity as each feature is individually compared against all others; however, the process can be made more efficient by introducing constraints to the matching process – that is, reduce the search space for possible matches by only searching in regions where the corresponding features are expected [61]. Another concern is that the matching process is not necessarily one-to-one. A single feature in one frame may be matched to several features in a second image frame; therefore, it is necessary to perform a mutual consistency check where only corresponding features that are mutually matched are considered.

Feature correspondences are important for two reasons; first, matched features allow the position of 3D landmarks to be determined in stereo applications and secondly, the relative motion between image frames are determined from sets of matched features (or their respective 3D landmarks).

Motion Estimation. In the final stage of the visual odometry pipeline, the robot motion between successive viewpoints is estimated. The goal is to determine the relative rotation and translation from sets of feature correspondences across viewpoints. These transformations are then concatenated to recover the full trajectory of the robot. It should be noted that sets of features may be described in terms of their image locations or 3D coordinates, and that different motion estimation approaches are applicable to each case.

The first approach focuses on estimating camera motion from 2D-to-2D feature correspondences – that is, features in both viewpoints are described in terms of their respective image coordinates. The geometric relationship, up to a scale factor, between two images is described by the essential matrix [62, p. 257],

which can be computed directly from 2D-to-2D point correspondences. Several well-known techniques have been proposed to calculate the essential matrix such as Longuet-Higgins' eight-point algorithm [63] and the five-point algorithm proposed by Nistér [64]. Once the essential matrix has been calculated, the relative rotation and translation can be determined [62, pp. 258–259]. Four different transformations are obtained in general; however, the correct transformation can be identified as only one of the solutions produces 3D points that exist in front of both cameras. The scale factor is then calculated from a set of triangulated features, after which the relative translation is rescaled accordingly [52].

A second approach assumes features are described by 3D coordinates in their respective viewpoints. The relative motion is determined by calculating the transformation that aligns the sets of 3D points from each viewpoint, which is also known as the absolute orientation problem. The transformation is chosen such that the Euclidean distance error between the two sets of points is minimised. There are several existing solutions to the absolute orientation problem that have been proposed [65–67] in literature. As opposed to the 2D-to-2D case, 3D features are already described in an absolute scale; hence, the full trajectory can be determined by direct concatenation of the estimated relative transformations.

The final motion estimation approach is based on 3D-to-2D feature correspondences. Instead of minimising the Euclidean distance as in the absolute orientation problem, the relative transformation is computed such that the image re-projection error is a minimum, more commonly known as the perspective-from-n-points (PnP) problem. Notably, Nistér et al. [5] concluded that PnP results in more accurate motion estimates than absolute orientation due to large uncertainties in depth of triangulated points. Different solutions to the PnP problem can be found in existing literature, such as the direct linear transformation algorithm [62, pp. 88–93] or the perspective-from-three-points (P3P) method which solves the minimal case [68]. This concludes the discussion on visual odometry and the focus now moves to outliers and their effect on accurate motion estimation.

1.2.2 Outliers in Computer Vision

The visual odometry pipeline discussed so far has assumed that features are identified and matched between viewpoints without any errors. Furthermore, motion estimation is performed on all matched features, and accurate estimates of the robot trajectory are obtained. In practical implementations, however, this assumption is rarely valid, and observations are usually contaminated with unpredictable and erroneous measurements known as outliers. The motion estimation techniques described earlier are extremely sensitive to outliers and their presence leads to incorrect results. Outlier removal is therefore a necessary addition to visual odometry frameworks.

Before outlier *removal* is discussed, it is important to consider what exactly is meant by outliers in the context of computer vision. Hawkings [69] provides a generalized and intuitive definition of outliers as follows,

> "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."

To put it differently, outliers can be viewed as observations which are anomalous with an underlying model that the majority of the other data points are consistent with. In visual odometry, the underlying model is the motion of the camera estimated from a set of matched features, and outliers are feature points that are inconsistent with this camera motion. Figure 1.3 depicts a set of putative matches that contains outliers. The distinction between inliers (purple square markers) and outliers (blue circle markers) should be clear in this example.

Errors made during the feature matching stage result in incorrect data associations, which lead to observations that are inconsistent with the motion of the camera between viewpoints. It is generally accepted that these incorrect matches cannot be avoided during the feature matching stage [71, 72]. Depending on the complexity of the mathematical models used to determine and match features, effects

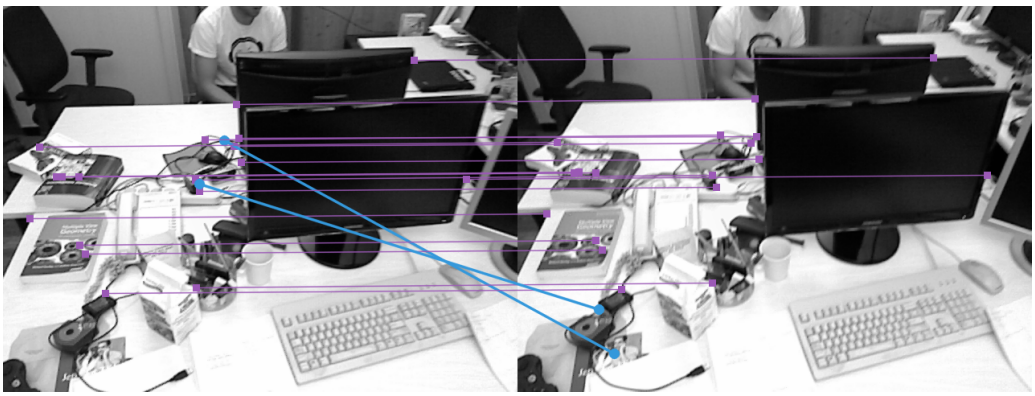


Figure 1.3: Illustration of outliers (mismatched features). Outliers, depicted by blue circle markers, are introduced during feature matching and are inconsistent with the camera motion between viewpoints. Image frames are part of the *TUM RGB-D Benchmark*[4] [70].

such as image noise, blur and partial occlusions may not be considered [52], which can cause erroneous matches. Additional phenomena such as depth discontinuities and repeated patterns in structured scenes may also cause mismatches [71]. Furthermore, features found on dynamic objects in the environment are also inconsistent with the motion of the camera. If these outliers are not removed, the model parameters determined during motion estimation will be incorrect.

1.2.3 Robust Motion Estimation

The problem of estimating parameters in the presence of outliers and measurement noise is known as robust estimation. A popular solution to this problem attempts to distinguish between inliers and outliers so that optimal parameter estimation is performed on inlier points only – that is, outlier removal is performed before parameters are estimated. The difficulty with outlier removal lies in the fact that there is no general mathematical representation for what constitutes an outlier. Moreover, a perfect separation of inliers and outliers generally requires a model (described by the parameters being estimated) of the data points' underlying structure. A simple illustration of this chicken-and-egg problem, given by Stewart [73], explains that it is impossible to determine which points are consistent with a line without first knowing what the line parameters are. These complications and the importance of robust estimation have lead to increased interest in outlier removal techniques.

A widely-used robust estimator is the random sample and consensus (RANSAC) algorithm introduced by Fischler and Bolles [68]. RANSAC applies an iterative, hypothesise-and-verify approach (outlined in Algorithm 1.1) and has been established as the standard method for dealing with outliers introduced by mismatched features. RANSAC and variations of the RANSAC algorithm are commonly implemented in

Algorithm 1.1 RANSAC

```
function RANSAC
    for k_r iterations do
        Randomly sample m points
        Estimate model parameters from sampled points
        Determine the number of points consistent with estimated model
    end for
    return Model with the largest number of consistent points
end function
```

autonomous navigation pipelines with various model parametrisations. Nistér et al. [5] pioneered the use of RANSAC, in combination with the efficient five-point solver of Nistér [64], to perform visual odometry

[4] Available for download at https://vision.in.tum.de/rgbd/dataset/freiburg1/rgbd_dataset_freiburg1_xyz.tgz.

in the presence of outliers. This five-point RANSAC algorithm was also used by Lhuillier [74] to estimate the motion of a catadioptric camera. A single-point-RANSAC visual odometry framework, specific to road vehicles, was proposed and evaluated against five-point RANSAC in a paper by Scaramuzza [75]. A recent visual odometry framework by Kitt et al. [43] uses RANSAC with a re-projection error model to remove outliers and Naroditsky et al. [76] determined the pose of a moving monocular camera using a four-point PnP solver in conjunction with a RANSAC approach. Other notable uses of RANSAC include a RANSAC-based motion estimation stage implemented on the Mars Rover [77] and a robust stereo SLAM algorithm by Bellavia et al. [78]. The popularity of RANSAC stems from its simplicity of implementation and ability to tolerate large numbers of outliers [79]. However, the number of iterations required by RANSAC grows exponentially [52] with the number of outliers, which leads to increased computational costs. In autonomous navigation frameworks, real-time performance is essential, which means that the computational load associated with outlier removal should not be high. Several adaptations to the RANSAC algorithm [71] have been proposed to increase efficiency as well as robustness, and will be discussed in detail in Chapter 4.

RANSAC is the standard approach used in most visual odometry frameworks to handle outliers; however, several alternative outlier removal techniques have been proposed. A basic outlier rejection method was implemented by Badino and Kanade [80], which iteratively removes features that have residual errors larger than a certain threshold, for robust SfM. Although simple and computationally efficient, this method assumes that motion estimates determined in the presence of outliers are approximately correct, and is therefore only applicable to systems where the number of outliers is low. Instead of filtering outliers, a different formulation of the robust motion estimation problem by Howard [6], focuses on *inlier detection*. The approach is based on the work of Hirschmuller et al. [81] and uses underlying geometric constraints of 3D landmarks to identify inliers. Sets of features that are mutually consistent with each other are identified prior to motion estimation. This make the technique extremely robust to outliers, however, an optimal solution is NP-complete [6], and approximations must be made for real-time applications. Furthermore, the geometric constraints used are deterministic in nature, and do not account for measurement uncertainty. For these reasons, RANSAC-based approaches remain the more popular choice and will therefore play a major role in the investigations of this thesis.

1.3 Research Objectives

This thesis details the development, analysis and verification of a novel outlier removal approach for vision-based autonomous navigation applications. The specific problem of robust motion estimation in the presence of highly uncertain measurements and large numbers of outliers is addressed. Accurate localisation is prioritised over the creation of a detailed environmental map, and the proposed method is implemented and verified in conjunction with a visual odometry framework. A review of available literature (Section 1.2.2) indicates that RANSAC-based outlier removal techniques are almost exclusively used in the field. This work aims to investigate alternatives for robust outlier removal, which directly address the limitations of RANSAC-based approaches. The research objectives are summarised as follows.

1. The primary objective of this thesis is the development of a computationally efficient and robust outlier removal method for visual odometry pipelines. This technique should be capable of removing outliers reliably in a real-time system suffering from uncertain measurements and a high number of outliers.

2. A secondary objective is investigating methods for environmental sensor noise propagation in autonomous navigation applications. The methods should also be computationally efficient and provide accurate representations of measurement distributions, which support the goal of robust motion estimation in highly uncertain conditions.

In terms of the autonomous navigation framework presented in Figure 1.1, the scope of this project does not include path planning and control modules. Furthermore, it is assumed that no robot sensors are available, and that a stereo camera pair is the only environmental sensor. Sensor fusion is therefore also excluded from the scope of this thesis.

1.4 Overview of Thesis

Chapter 1, the opening chapter of this thesis, introduces relevant background knowledge related to autonomous vehicle systems. The concept of autonomous navigation is defined with the aid of a generalised framework. Thereafter, a brief review of vision-based navigation systems is performed before a visual odometry pipeline is proposed. Outliers in the context of vision-based systems are discussed, which is followed by a review of outlier removal techniques. The research objectives are formally stated before concluding with an overview of the thesis. The remaining chapters are divided into two parts as follows:

Part I. The first part of this thesis consists of two chapters that focus on the modelling of environmental sensor measurements. The work performed in these two chapters is related to the secondary objective of this research project, and is supplementary to the later chapters.

Chapter 2 presents a mathematical model for a single camera system, and also details the geometric relationship between point correspondences across multiple views. The concept of epipolar constraints is introduced as well as the process of image rectification and its application to feature matching. Thereafter, it is shown how the position of 3D points relative to the camera system are calculated from matched image points using triangulation.

Measurement uncertainty, in the context of stereo 3D reconstruction, is discussed in Chapter 3. Matched features are modelled as Gaussian random variables which undergo non-linear transformations; two techniques, linearisation and the unscented transform, are investigated for approximating the distribution of processed measurements. Thereafter, the two techniques are compared experimentally with synthetic stereo data under various conditions, where it is shown that linearisation performs notably worse than the unscented transform in conditions of high uncertainty.

Part II. The second part of this thesis is concerned with the development and evaluation of a novel outlier removal method for visual odometry. This part consists of four chapters and forms the primary contribution of this research.

Chapter 4 provides a detailed review of existing outlier removal techniques. RANSAC is discussed in detail with a focus on limitations of the algorithm in terms of robustness and efficiency. Thereafter, several improved versions of RANSAC and an alternative to hypothesise-and-verify techniques are described.

A robust visual odometry pipeline, based on a probabilistic RANSAC approach, is implemented in Chapter 5. A similarity measure, proposed by Brink et al. [82], is derived and extended to that of a probabilistic distance. Techniques used for determining absolute orientation and motion refinement are also presented. Thereafter, the discussion moves towards the generation of synthetic visual odometry datasets for experimental purposes. The chapter ends with a comparison of the unscented transform and linearisation when incorporated into robust visual odometry frameworks with highly uncertain measurements. It is shown that visual odometry using the unscented transform performs significantly better than a visual odometry framework that implements linearisation for uncertainty propagation.

A novel outlier removal approach is developed in Chapter 6 that attempts to address the limitations of the standard RANSAC algorithm and makes use of inherent shape information to distinguish between inliers and outliers. The proposed algorithm is first presented conceptually, and then developed theoretically in a general formulation of the relative transformation problem. Thereafter, a probabilistic measure of shape similarity is developed for visual odometry applications, as well as an adaptive sampling technique

based on decision theory principles. The chapter ends with discussion of experimental results that show the proposed technique is a viable approach for outlier removal.

In Chapter 7, the novel outlier removal approach is compared against the probabilistic RANSAC method of Chapter 5. Experimental results indicate that the proposed technique outperforms RANSAC when datasets are highly contaminated with outliers and perturbed with high levels of measurement noise. The viability of the proposed technique for visual odometry is further verified by performing robust motion estimation on synthetic and practical visual odometry datasets. The proposed technique is shown to obtain more consistent trajectory reconstructions, with significantly reduced execution times, for both simulated and practical experiments.

The final chapter of this thesis, Chapter 8, presents a brief summary of the research outcomes and important results, as well as an overview of the contributions made. The chapter concludes with a discussion of possible future work.

Part I

Environmental Sensor Modelling

Chapter 2

Camera Geometry

As discussed in the introductory chapter, the work contained in this thesis forms part of a visual odometry framework. It is assumed that the only environmental sensors available are a set of stereo cameras. Hence, it is necessary to understand how 3D information, relating to a robot's motion and objects in the environment, is obtained from a series of sequential images. A logical starting point is therefore an analysis of camera geometry, as well as the relationship between 3D points and their corresponding 2D image coordinates.

A camera is a device which captures light rays reflected off of objects in the environment and performs a projective mapping from $\mathbb{R}^3$ (world coordinates) to $\mathbb{R}^2$ (image coordinates) space. A mathematical representation of this mapping process – a camera model – is a matrix that contains various parameters describing characteristics of the camera. Computer vision and robotic perception applications require this geometric description of the imaging process as it allows the location of objects in the environment to be determined relative to the camera system.

This chapter begins with the analysis and development of a mathematical model that describes a single camera's geometry. A description of various camera parameters is provided, as well as distortion coefficients that model non-linear effects introduced by imperfect lenses. Thereafter, the geometry of a multiple view system is investigated, and the significance of point correspondences across images, as well as how they are used to determine the 3D location of features, is discussed.

2.1 Single View Geometry

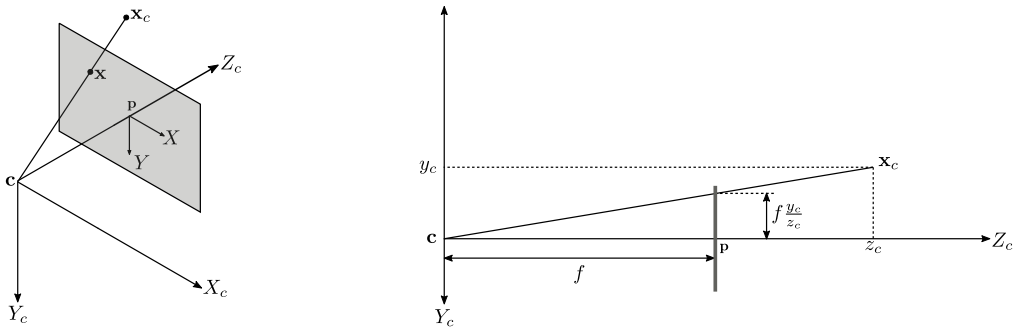
This section introduces core concepts required to model a single perspective camera. A standard pinhole camera model is derived that describes the projection of points in 3D space to a 2D image plane. The standard linear model is extended to include a distortion model, which accounts for non-linearities introduced by imperfect lenses, and a brief overview of various camera parameters is provided.

2.1.1 Pinhole Camera Model

A basic pinhole camera [62, p. 153] is the standard model used in computer vision applications. The centre of projection of the camera is assumed to coincide with the origin of the camera coordinate system, denoted by the subscript c. An image plane exists at some distance, f, along the principal axis (also known as the viewing axis), Z_c, perpendicular to the $X_c - Y_c$ plane. This is shown in Figure 2.1. The point of intersection of the image plane and the principal axis is known as the principal point, and the image coordinate system is denoted by the X and Y axes.

In this camera model, a point in 3D space is mapped to a 2D point on the image plane where a ray, drawn from the 3D point to the centre of projection, meets the image plane. Suppose a 3D point in the camera coordinate system, $\mathbf{x}_c$, described by

$$\mathbf{x}_c = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^\top, \tag{2.1}$$



(a) **(b)**

Figure 2.1: A basic pinhole camera model. (a) The camera centre, **c**, is defined at the origin of a camera coordinate system with an image plane (shaded) that meets the principal axis at the principal point, **p**. A 3D point, $\mathbf{x}_c$, is projected to a 2D image coordinate, **x**. (b) $Y_c - Z_c$ plane view of pinhole camera model. Adapted from Hartley and Zisserman [62, Fig. 6.1].

is projected onto the image plane. If the origin of the image coordinate system is assumed to be at the principal point, the image coordinates, **x**, are calculated by

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} x & y & z \end{bmatrix}^\top \\ &= \begin{bmatrix} f\frac{x_c}{z_c} & f\frac{y_c}{z_c} & f \end{bmatrix}^\top .\end{aligned} \tag{2.2}$$

A more general projection, which compensates for an offset in principal point, is given by

$$\mathbf{x} = \begin{bmatrix} f\frac{x_c}{z_c} + c_x & f\frac{y_c}{z_c} + c_y & f \end{bmatrix}^\top , \tag{2.3}$$

where c_x and c_y are the coordinates of the principal point in the image coordinate system. The projection in Equation 2.3 can be expressed as a linear mapping if the 3D point and image point are described in the form of homogeneous coordinates [62, p. 2] such that

$$\tilde{\mathbf{x}} = \begin{bmatrix} u & v & w \end{bmatrix}^\top , \tag{2.4}$$

where $x = \frac{u}{w}$ and $y = \frac{v}{w}$ and tilde notation denotes a homogeneous vector. Equation 2.3 therefore simplifies to

$$\begin{aligned}\tilde{\mathbf{x}} &= \mathbf{P}_c \tilde{\mathbf{x}}_c \\ \begin{bmatrix} u \\ v \\ w \end{bmatrix} &= \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix},\end{aligned} \tag{2.5}$$

where $\mathbf{P}_c$ is the camera projection matrix. Writing $\mathbf{P}_c$ in the form,

$$\mathbf{P}_c = \mathbf{K}[\mathbf{I}_{3\times 3}|\mathbf{0}_{3\times 1}], \tag{2.6}$$

allows the definition of a new matrix, **K**, known as the camera calibration matrix such that

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.7}$$

where α_x and α_y are the focal lengths in pixel dimensions along the X and Y axes respectively and s is the skew parameter. The form of Equation 2.6 is important as it emphasises the fact that the camera

centre is assumed to be located at the origin of the camera coordinate system, with the viewing axis of the camera pointing in the direction of the Z_c axis.

In general, a point in space will not be expressed in the camera coordinate frame. Instead, points are usually expressed relative to the robot. An additional coordinate axis system is therefore defined, denoted with a subscript r, which describes the robot axes. The two coordinate frames are related by some rotation and translation such that a point in the robot frame, $\mathbf{x}_r$, is transformed to the camera coordinate frame by

$$\begin{aligned} \mathbf{x}_c &= \mathbf{R}(\mathbf{x}_r - \mathbf{c}_r) \\ &= \mathbf{R}\mathbf{x}_r + \mathbf{t}, \end{aligned} \tag{2.8}$$

where $\mathbf{c}_r$ represents the coordinates of the camera centre relative to the robot coordinate frame, $\mathbf{R} \in SO(3)$ and $\mathbf{t} = -\mathbf{R}\mathbf{c}_r$. It follows, making use of homogeneous vectors, that

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix}. \tag{2.9}$$

Combining Equations 2.5 and 2.9, the projection from the robot frame to image coordinates is described by

$$\tilde{\mathbf{x}} = \mathbf{P}_c \tilde{\mathbf{x}}_r, \tag{2.10}$$

where $\mathbf{P}_c = \mathbf{K}[\mathbf{R}|\mathbf{t}]$. The generalised pinhole camera model has eleven degrees of freedom, five of which are contained in $\mathbf{K}$ and relate to the internal properties of the camera. The elements of $\mathbf{K}$ are known as intrinsic parameters, and are determined during camera calibration [83]. The remaining six parameters are contained in $\mathbf{R}$ and $\mathbf{t}$, which relate the camera coordinate system to the robot coordinate frame, and are therefore called the extrinsic parameters of the camera.

2.1.2 Lens Distortion

Thus far it has been assumed that the linear mapping described in Section 2.1.1 accurately models the projection of points in 3D space to 2D image coordinates. In the derivation of the pinhole camera model, an assumption was made that a 3D point, its corresponding image point and the camera centre are co-linear – that is, rays of light maintain a straight trajectory when passing through the camera lens. This assumption is violated in practice when using physical cameras and lenses, resulting in distorted image measurements. In this section two forms of lens distortion, namely, radial and tangential distortion, are discussed and modelled.

2.1.2.1 Radial Distortion

A significant source of deviation arises from an effect known as radial distortion [62, p. 189]. The curvature of the camera lens causes light rays entering further from the centre of the lens to be refracted more than those passing through closer to the optical centre [84, p. 375]. As a result, straight scene lines, which would be projected to straight image lines with an ideal lens, are now imaged as curved lines. The effect of radial distortion is depicted in Figure 2.2. Each image point undergoes a radial displacement proportional to its distance from the centre of radial distortion[1], consequently, the distortion is more prominent near the edges of the image.

The effect of radial distortion is mitigated by applying a corrective transformation to each pixel coordinate. This undistorts image measurements to those that would have been obtained if a perfectly

[1]The centre of radial distortion is usually taken as the principal point, however, they may differ slightly and can be estimated during camera calibration.

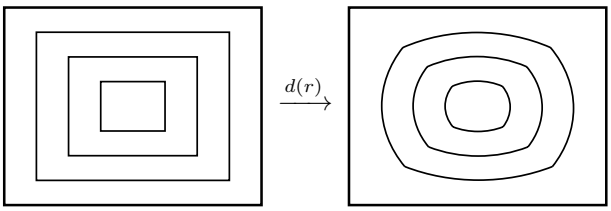


Figure 2.2: The effect of radial distortion. Radial distortion results in straight scene lines being projected to curved lines in the image frame. The distortion function, $d(r)$, becomes more prominent as the radius increases. Adapted from Hartley and Zisserman [62, Fig. 7.5].

linear lens was used. The corrected image coordinates are calculated by,

$$x_u = d_x + d(r)(x_d - d_x) \tag{2.11}$$

and

$$y_u = d_y + d(r)(y_d - d_y), \tag{2.12}$$

where $[x_d, y_d]^\top$ are distorted image measurements, the centre of radial distortion is described by $[d_x, d_y]^\top$ and the estimated, undistorted image measurements are given by $[x_u, y_u]^\top$. The distortion factor, $d(r)$, is a function of the radius, $r = \sqrt{(x_d - d_x)^2 + (y_d - d_y)^2}$. The function $d(r)$ is different for each lens; however, in practice it is sufficient to model the distortion with a third-order Taylor series approximation,

$$d(r) \approx 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3, \tag{2.13}$$

where κ_1, κ_2 and κ_3 are radial distortion parameters estimated during camera calibration [83].

In general, the undistorted coordinates, x_u and y_u, are not integer values. For this reason, Equation 2.12 is not used directly when constructing undistorted images. Instead, all x_u and y_u pixels in an undistorted image plane are mapped to non-integer x_d and y_d pixel locations. The values of these non-integer image coordinates are then interpolated, and mapped to their respective undistorted pixel locations.

2.1.2.2 Tangential Distortion

An additional source of distortion in physical cameras is in the form of tangential distortion. This is usually introduced during the manufacturing process when the imaging plane is not perfectly parallel to the camera lens. As a result, image points are displaced elliptically depending on the location and radius of the point in question [84, p. 376]. Tangential distortion is corrected by applying corrective transformations proposed by Brown [85],

$$x_u = x_d + [2p_1 y_d + p_2(r^2 + 2x_d^2)] \tag{2.14}$$

and

$$y_u = y_d + [2p_2 x_d + p_1(r^2 + 2y_d^2)], \tag{2.15}$$

where $[x_d, y_d]^\top$, $[x_u, y_u]^\top$ and r are defined in Equation 2.12. The coefficients, p_1 and p_2, are tangential distortion parameters determined during camera calibration [83]. The inverse mapping process described in Section 2.1.2.1 is applied verbatim.

2.2 Multiple View Geometry

This section focuses on the geometry present in multiple view camera systems. The projection properties of the pinhole camera model, derived in Section 2.1.1, result in certain geometric constraints for corresponding points across viewpoints. Different views are obtained from sequential frames from a single camera, or alternatively, captured simultaneously from a set of stereo cameras. Without a loss of generality, this section will only focus on the stereo camera setup as derivations can be used verbatim for the single camera case.

2.2.1 Epipolar Geometry

Epipolar geometry describes the projective geometry across two camera views [62, p. 239]. Figure 2.3 illustrates a 3D point, $\mathbf{x}_c$, projected to two image points, $\mathbf{x}_L$ and $\mathbf{x}_R$, by means of a stereo pair depicted by their camera centres, $\mathbf{c}_L$ and $\mathbf{c}_R$. The straight line joining the two camera centres is known as the baseline, B. The camera centres, image coordinates and $\mathbf{x}_c$ are coplanar to what is known as the epipolar plane. This plane, denoted by π, introduces constraints to the location of corresponding points in different viewpoints. Suppose for the stereo rig in Figure 2.3 that the coordinates of a single image point, $\mathbf{x}_L$, are

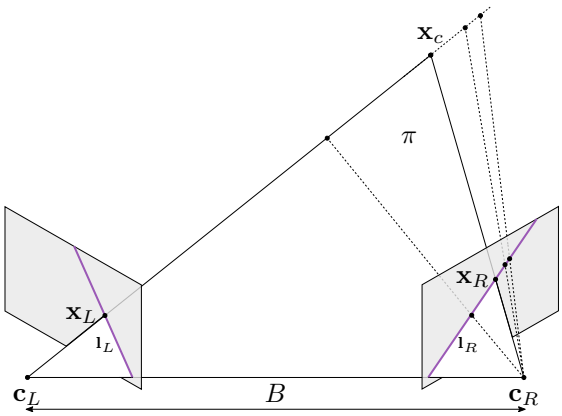


Figure 2.3: Epipolar geometry of stereo pair with camera centres indicated by $\mathbf{c}_L$ and $\mathbf{c}_R$. The 3D point, $\mathbf{x}_c$, corresponding image points, $\mathbf{x}_L$ and $\mathbf{x}_R$, and camera centres lie in a common plane, π. The projections of rays from $\mathbf{c}_L$ and $\mathbf{c}_R$ to $\mathbf{x}_c$, on the left and right image planes, are given as $\mathbf{l}_L$ and $\mathbf{l}_R$ respectively. Adapted from Hartley and Zisserman [62, Figure 9.1].

given but $\mathbf{x}_c$ and $\mathbf{x}_R$ are unknown. The baseline, and ray from $\mathbf{c}_L$ to $\mathbf{x}_L$ determine the epipolar plane, π. The 3D point, $\mathbf{x}_c$, must lie along this ray and the corresponding ray from $\mathbf{c}_R$ to $\mathbf{x}_R$ must also exist on π. Consequently, $\mathbf{x}_R$ must lie on the line, $\mathbf{l}_R$. This line in the right image frame is known as the epipolar line of $\mathbf{x}_L$, and by the same token there exists an epipolar line in the left image frame, $\mathbf{l}_L$. This is significant, as a search for corresponding points is now limited to epipolar lines instead of the full image plane. The points of intersection between the baseline and the image planes are called epipoles, and are equivalent to the projections of the camera centres in another view.

The constraints introduced by epipolar geometry are represented in an algebraic form by a matrix, $\mathbf{F}$, called the fundamental matrix. It has already been stated for the camera system in Figure 2.3, that there is a relationship between an image point and epipolar lines in other image views. In *Multiple View Geometry* [62, pp. 242-243] it is shown geometrically that this relationship is linear and given by

$$\mathbf{l}_R = \mathbf{F}\mathbf{x}_L, \tag{2.16}$$

where $\mathbf{F}$ is a 3×3 homogeneous matrix with a rank of two. Furthermore, it can be shown that $\mathbf{F}$ satisfies the condition,

$$\mathbf{x}_R^\top \mathbf{F}\mathbf{x}_L = 0, \tag{2.17}$$

for a pair of matched points between image frames. This relationship allows $\mathbf{F}$ to be determined from a set of matched features (at least seven matches), without reference to the individual camera matrices. The essential matrix, mentioned in Section 1.2.1, is calculated from

$$\mathbf{E} = \mathbf{K}^\top \mathbf{F}\mathbf{K}, \tag{2.18}$$

where $\mathbf{K}$ is the camera calibration matrix[2] defined in Equation 2.7. The essential matrix is a specialisation of the fundamental matrix that assumes normalized image coordinates. This assumption introduces

[2]It is assumed that the two cameras have the same calibration matrix.

additional constraints and reduces the degrees of freedom, but requires that the cameras have been calibrated and that $\mathbf{K}$ is known. Once the essential matrix has been obtained, the relative pose of the cameras between viewpoints can be determined [62, pp. 258-259].

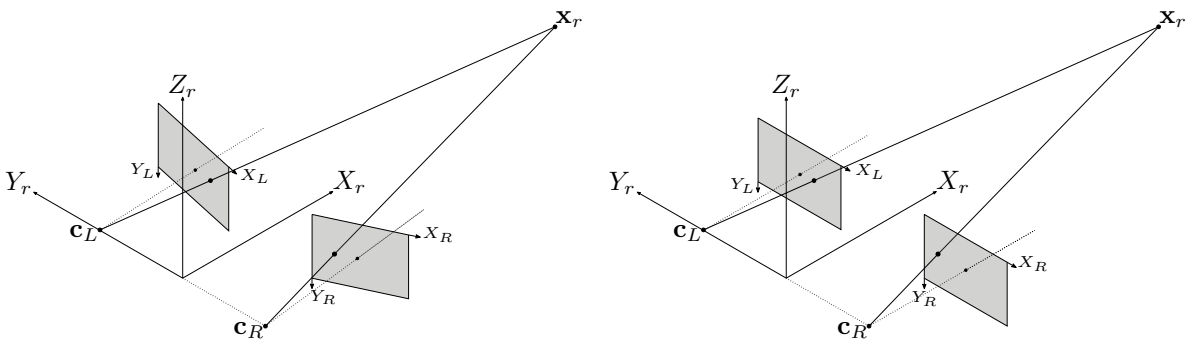
2.2.2 Rectification of Stereo Images

In the previous section, it was shown how the relative pose between two cameras could be determined from a set of 2D-to-2D correspondences. As mentioned in Section 1.2.1, there are two alternative motion estimation approaches that make use of 3D-to-3D and 3D-to-2D feature correspondences. It is therefore necessary to calculate the 3D position of a feature given its image coordinates. A process known as image rectification is usually performed before 3D reconstruction takes place. The goal of image rectification is to transform the image planes so that they are co-planar and parallel to the baseline of the camera system.

Consider the unrectified stereo camera pair of Figure 2.4a with projection matrices,

$$\begin{aligned}\mathbf{P}_L &= \mathbf{K}_L\mathbf{R}_L[\mathbf{I}|-\mathbf{c}_L]\\ \mathbf{P}_R &= \mathbf{K}_R\mathbf{R}_R[\mathbf{I}|-\mathbf{c}_R],\end{aligned} \tag{2.19}$$

describing their pose relative to a robot coordinate frame. Projective transformations, $\mathbf{H}_L$ and $\mathbf{H}_R$, are required such that the epipolar lines of the two image planes are transformed in a way which makes them parallel to the baseline as in Figure 2.4b. The resulting rectified camera pair can then be expressed as



(a) **(b)**

Figure 2.4: Image rectification of stereo camera pair. (a) An unrectified stereo camera pair. (b) The resulting camera system after image rectification has been performed, where the epipolar lines of both image planes are parallel to the baseline.

$$\begin{aligned}\mathbf{P}_L &= \mathbf{K}_{\text{rect}}\mathbf{R}_{\text{rect}}[\mathbf{I}|-\mathbf{c}_L]\\ \mathbf{P}_R &= \mathbf{K}_{\text{rect}}\mathbf{R}_{\text{rect}}[\mathbf{I}|-\mathbf{c}_R],\end{aligned} \tag{2.20}$$

where both cameras are modelled as having the same camera calibration matrix, $\mathbf{K}_{\text{rect}}$ and the same orientation described by $\mathbf{R}_{\text{rect}}$. The choice of $\mathbf{K}_{\text{rect}}$ is arbitrary, and taken to be the average of $\mathbf{K}_L$ and $\mathbf{K}_R$ such that

$$\mathbf{K}_{\text{rect}} = \frac{1}{2}(\mathbf{K}_L + \mathbf{K}_R). \tag{2.21}$$

As shown by Brink et al. [86], each row of $\mathbf{R}_{\text{rect}}$ can be determined separately. The first row of $\mathbf{R}_{\text{rect}}$, which is equivalent to the vector pointing in the direction of the transformed X-axis, must be parallel to the baseline such that,

$$\mathbf{r}_1 = \frac{\mathbf{c}_R - \mathbf{c}_L}{\|\mathbf{c}_R - \mathbf{c}_L\|}. \tag{2.22}$$

Furthermore, the transformed Y-axis is orthogonal to $\mathbf{r}_1$ and is chosen to be orthogonal to the viewing axis of the left camera [51],

$$\mathbf{r}_2 = \frac{\mathbf{u} \times \mathbf{r}_1}{\|\mathbf{u} \times \mathbf{r}_1\|}, \tag{2.23}$$

where $\mathbf{u}$ is the unit vector in the direction of the left camera's viewing axis. The third row must be orthogonal to both $\mathbf{r}_1$ and $\mathbf{r}_2$ such that,

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2, \tag{2.24}$$

where $\mathbf{R}_{\text{rect}}$ is given by

$$\mathbf{R}_{\text{rect}} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}^\top. \tag{2.25}$$

It follows that the projective transformations are given such that,

$$\begin{aligned} \mathbf{H}_L &= \mathbf{K}_{\text{rect}} \mathbf{R}_{\text{rect}} \mathbf{R}_L^\top \mathbf{K}_L^{-1} \\ \mathbf{H}_R &= \mathbf{K}_{\text{rect}} \mathbf{R}_{\text{rect}} \mathbf{R}_R^\top \mathbf{K}_R^{-1}. \end{aligned} \tag{2.26}$$

The rectified image planes are now determined by re-sampling image points by

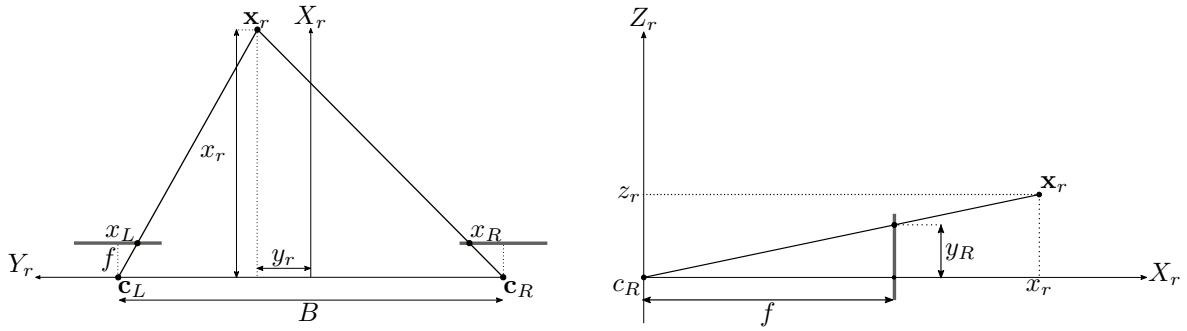
$$\begin{aligned} \mathbf{x}_L^{\text{rect}} &= \mathbf{H}_L \mathbf{x}_L \\ \mathbf{x}_R^{\text{rect}} &= \mathbf{H}_R \mathbf{x}_R, \end{aligned} \tag{2.27}$$

where $\mathbf{x}_L^{\text{rect}}$ and $\mathbf{x}_R^{\text{rect}}$ are the rectified image points of the left and right image plane respectively. Consequently, disparities in pixel coordinates between the two image frames exist in the X-direction only, with no disparity in the Y-direction. The advantages of working with a rectified stereo pair are two-fold. First, searching for point correspondences is more efficient due to the simplified epipolar structure and secondly, rectifying images considerably simplifies the triangulation process, which is discussed in the next section.

2.3 Triangulation

Given a set of point correspondences across two images and the projection matrices of the two cameras, it is possible to reconstruct a 3D scene point from corresponding image points via triangulation. This is necessary for motion estimation approaches that make use of 3D-to-3D and 3D-to-2D feature correspondences.

Triangulation can be viewed geometrically as back projecting each image point and determining the intersection point of the two rays. This process is simplified when working with a rectified camera pair. A top-down and side view of a rectified, stereo-view system is shown in Figure 2.5. The stereo pair are assumed to be placed on Y_r (centred on Z_r) of the robot coordinate system, with the viewing axis of each camera pointing in the direction of X_r. The projection of a three dimensional feature, $\mathbf{x}_r$, to the left and



(a) **(b)**

Figure 2.5: Geometry of rectified, stereo camera coordinate system. (a) Top down view of stereo pair geometry. (b) Side view of stereo pair geometry. Adapted from Brink [51, Figure 3.6].

right image frames, corresponds to image coordinates $[x_L, y_L]^\top$ and $[x_R, y_R]^\top$ respectively. Triangulation

is used to reconstruct $\mathbf{x}_r$ in the robot coordinate frame using a set of matched image features. From Figure 2.5a the following relationships,

$$\begin{aligned} \frac{f}{x_L} &= \frac{x_r}{\frac{B}{2} - y_r}, \\ \frac{f}{-x_R} &= \frac{x_r}{\frac{B}{2} + y_r}, \end{aligned} \tag{2.28}$$

are obtained from the use of similar triangles. Solving for x_r and y_r simultaneously gives,

$$\begin{aligned} x_r &= \frac{fB}{x_L - x_R}, \\ y_r &= \frac{-x_L B}{x_L - x_R} + \frac{B}{2}. \end{aligned} \tag{2.29}$$

Likewise, using Figure 2.5b, an expression for z_r is obtained,

$$z_r = \frac{-yB}{x_L - x_R}, \tag{2.30}$$

where $y = y_L = y_R$ due to rectification. These equations can be expanded to include offset principal points [51], c_x and c_y, where $\mathbf{x}_r$ is reconstructed from matched stereo coordinates,

$$\mathbf{x} = \begin{bmatrix} x_L & y_L & x_R & y_R \end{bmatrix}^\top, \tag{2.31}$$

such that,

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \mathbf{f}_{\text{tri}}(\mathbf{x}) = \begin{bmatrix} \frac{fB}{x_L - x_R} \\ \frac{-B(x_L - c_x)}{x_L - x_R} + \frac{B}{2} \\ \frac{B(2c_y - (y_L + y_R))}{2(x_L - x_R)} \end{bmatrix}, \tag{2.32}$$

with $\mathbf{f}_{\text{tri}}$ denoting the triangulation function.

2.4 Chapter Summary

The relationship between points in 3D space and their respective projection to image coordinates is critical for computer vision and robotic perception applications. A mathematical camera model was therefore introduced that describes the projective geometry of a single camera. An overview of various camera parameters and sources of lens distortion was also provided. Thereafter, the camera system was extended to that of a stereo pair. The concept of epipolar geometry was presented, as well as the process of image rectification. Triangulation equations were derived that allow the position of 3D features to be determined from their respective stereo image measurements. It is important to note that the effect of measurement uncertainty has so far not been considered. This will be the focus of Chapter 3.

Chapter 3

Measurement Uncertainty

In Chapter 2, the relationship between points in 3D space and their corresponding image coordinates was discussed. It was shown that the position of 3D features could be determined from the image projections of two, as long as the camera matrices relating to each viewpoint were known. Up to this point, however, the effect of measurement uncertainty has been ignored and the derivation of triangulation equations in Section 2.3 have assumed perfect knowledge of image coordinates and camera parameters. Unfortunately, this is a poor assumption for practical systems. Measurements captured from images are corrupted by sensor noise, and used in various computer vision methods such as pose estimation or SfM, and consequently, errors are propagated through the system.

In this chapter, the effect of measurement uncertainty[1] is no longer disregarded. Instead, measured image coordinates are modelled as continuous, normally distributed random variables that are passed through non-linear transformations. As a result, processed measurements are represented by transformed random variables with their respective uncertainties. Two commonly used techniques for uncertainty propagation, applicable to non-linear transformations of continuous random variables, form part of the extended Kalman filter [87] (EKF) and the unscented Kalman filter [88] respectively. The EKF linearises the transformation about the original mean, while the UKF employs a sampling technique known as the unscented transform. In this chapter, the accuracies of linearisation and the unscented transform are investigated when used for approximating the distributions of 3D reconstructed features. This investigation forms one of the contributions of this thesis.

The chapter begins with an investigation into the non-linear transformation of Gaussian random variables. Two commonly used techniques for the non-linear transformation of Gaussian random variables, namely, the standard approach of linearisation and the unscented transform, are discussed. Experiments are performed with synthetic stereo vision data to compare the accuracy of the discussed techniques and evaluate their viability for uncertainty propagation in stereo vision applications.

3.1 Non-linear Transformation of Gaussian Random Variables

Suppose $\mathbf{x}$ is an n-length vector of jointly Gaussian random variables,

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}), \tag{3.1}$$

and a non-linear function, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ exists such that

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_m(\mathbf{x}) \end{bmatrix}^\top, \tag{3.2}$$

where $\mathbf{y}$ is a vector of transformed random variables. Due to the non-linear nature of $\mathbf{f}(\,\cdot\,)$, a closed-form solution for the transformed distribution of $\mathbf{y}$ is not guaranteed to exist. For tractability, a simplifying

[1] It is assumed that the camera parameters are known to a much higher degree of accuracy than image coordinates, therefore uncertainty is only introduced by the measurement noise of image points and that there are no errors in projection matrices. This assumption is also made by Hartley and Zisserman in *Multiple View Geometry* [62, p. 310].

assumption is therefore made to model the resulting distribution as Gaussian such that,

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{y}, \boldsymbol{\Sigma}_\mathbf{y}). \tag{3.3}$$

It is therefore necessary to efficiently and accurately determine the second-order statistics of $\mathbf{y}$, namely the transformed mean and covariance, $\boldsymbol{\mu}_\mathbf{y}$ and $\boldsymbol{\Sigma}_\mathbf{y}$.

3.1.1 Linearisation via Taylor Series Expansion

Linearisation is a popular technique used to approximate distributions of transformed continuous random variables. This approach involves linearising the non-linear transformation about some point (usually taken as the mean of the input distribution), and is motivated by the fact that a linear transformation of a Gaussian random variable results in a random variable which is also Gaussian in nature [89, p. 159]. The second-order statistics, required to describe the Gaussian approximation of the transformed distribution, are then determined from closed-form computations. From the perspective of computational efficiency, this is a key advantage of linearisation [90, p. 56].

The closed-form computations, required to calculate the second-order statistics of the transformed distribution, consist of two calculations. First, the non-linear transformation stated in Equation 3.2 can be approximated by a first-order Taylor Series expansion [91],

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\boldsymbol{\mu}_\mathbf{x}) + \mathbf{J}_\mathbf{f} \cdot (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}). \tag{3.4}$$

In Equation 3.4, the $m \times n$ matrix, $\mathbf{J}_\mathbf{f}$, is the Jacobian of the non-linear mapping evaluated at the mean, $\boldsymbol{\mu}_\mathbf{x}$, such that

$$\mathbf{J}_\mathbf{f} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\boldsymbol{\mu}_\mathbf{x}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}\right|_{\mathbf{x}=\boldsymbol{\mu}_\mathbf{x}}. \tag{3.5}$$

In the second stage of calculations, the estimated mean and covariance of the transformed distribution are simply calculated by a linear transformation of a Gaussian variable such that

$$\boldsymbol{\mu}_\mathbf{y} = \mathbf{f}(\boldsymbol{\mu}_\mathbf{x}) \tag{3.6}$$

and

$$\boldsymbol{\Sigma}_\mathbf{y} = \mathbf{J}_\mathbf{f} \boldsymbol{\Sigma}_\mathbf{x} \mathbf{J}_\mathbf{f}^\top. \tag{3.7}$$

Figure 3.1 illustrates the basic concepts of linearisation applied to the non-linear transformation of a one-dimensional Gaussian random variable, $p(x) \sim \mathcal{N}(\mu_x, \sigma_x)$. In this example, a linear approximation of

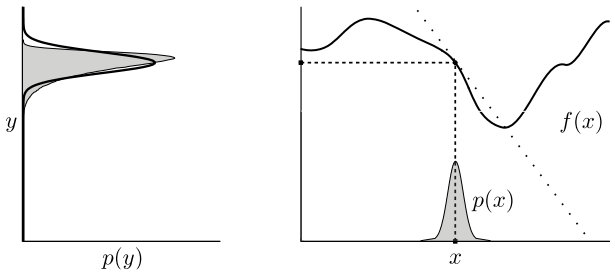


Figure 3.1: Illustration of the non-linear transformation of a Gaussian random variable using linearisation. A tangent to $f(x)$ is calculated at the mean of $p(x)$. The transformed random variable, $p(y)$, is then obtained by passing $p(x)$ through the linearly approximated function. The shaded area represents the actual distribution of $p(y)$ with the Gaussian approximation indicated by the solid line. Adapted from *Probabilistic Robotics* by Thrun et al. [90, Figure 3.8].

$f(x)$ is determined by calculating the tangent (dotted line) of the function at μ_x. The actual distribution

of the transformed random variable, $p(y)$, is represented by the shaded area. A Gaussian approximation, indicated by the solid line, of this transformed distribution is determined by passing $p(x)$ through the linear approximation of $f(x)$.

In the field of robotics, linearisation has become a standard practice as part of the extended Kalman filter (EKF) [87]. The EKF is a well-known state estimator, which makes use of this principle to simplify non-linear models, and allows standard Kalman filter techniques to be used in conjunction with linearised systems. Several EKF implementations have been applied with great success [22, 92]; however, limitations of the linear approximation have been identified. Most notably, are the approximation errors incurred when transformations are not locally linear – that is, when the point of linearisation coincides with a highly non-linear region of $\mathbf{f}(\,\cdot\,)$. Furthermore, the uncertainty of the random variable being transformed also plays a role in the accuracy of the linear approximation. Prior distributions with high uncertainty are therefore more affected by non-linearities which leads to larger approximation errors.

3.1.2 Unscented Transform

The unscented transform [93] is a sample-based technique used to approximate the second-order statistics of a continuous random variable undergoing a non-linear transformation. It was first described by Julier and Uhlmann [88] in the framework of the unscented Kalman filter (UKF).

A number of weighted samples, known as sigma points, that describe second-order statistics are deterministically selected from the prior distribution at $\boldsymbol{\mu}_{\mathbf{x}}$ and symmetrically about the main axes of $\boldsymbol{\Sigma}_{\mathbf{x}}$. The non-linear transformation is applied to each sigma point, after which a Gaussian distribution is fit to the transformed sigma points. For an n-dimensional distribution, $2n+1$ points are required to describe the second order statistics of the transformed distribution, where the sigma points, $\mathcal{X}$, are determined by

$$\mathcal{X}^{[i]} = \begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, & \text{for } i = 0 \\ \boldsymbol{\mu}_{\mathbf{x}} + \left(\sqrt{(n+\lambda)\boldsymbol{\Sigma}_{\mathbf{x}}}\right)_{i,} & \text{for } i = 1 \dots n \\ \boldsymbol{\mu}_{\mathbf{x}} - \left(\sqrt{(n+\lambda)\boldsymbol{\Sigma}_{\mathbf{x}}}\right)_{i-n,} & \text{for } i = n+1 \dots 2n, \end{cases} \tag{3.8}$$

where λ is determined by

$$\lambda = \alpha^2(n+\kappa) - n, \tag{3.9}$$

and the notation, $\mathcal{X}^{[i]}$, refers to the i^{th} element in $\mathcal{X}$. The notation $(\mathbf{B})_i$ means that the i^{th} column of matrix $\mathbf{B}$ is selected. The parameters, α and κ, affect the spread of sigma points around the mean. The square root of the covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{x}}$, is determined using Cholesky decomposition [94, pp. 96–98].

Each sigma point has two associated weights used to determine the mean and covariance of the transformed distribution. These weights are calculated as

$$\mathcal{W}_m^{[i]} = \begin{cases} \frac{\lambda}{n+\lambda} & \text{for } i = 0 \\ \frac{1}{2(n+\lambda)} & \text{for } i = 1 \dots 2n \end{cases} \tag{3.10}$$

and

$$\mathcal{W}_c^{[i]} = \begin{cases} \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) & \text{for } i = 0 \\ \frac{1}{2(n+\lambda)} & \text{for } i = 1 \dots 2n, \end{cases} \tag{3.11}$$

where $\mathcal{W}_m^{[i]}$ and $\mathcal{W}_c^{[i]}$ are mean and covariance weights corresponding to the i^{th} sigma point and β incorporates prior knowledge of the distribution type. The transformed sigma points, $\mathcal{Y}$, are then calculated by applying the non-linear transformation, $\mathbf{f}(\,\cdot\,)$, to each sigma point,

$$\mathcal{Y}^{[i]} = \mathbf{f}(\mathcal{X}^{[i]}), \tag{3.12}$$

where $\mathbf{f}(\,\cdot\,)$ is as defined in Equation 3.2. The transformed sigma points are then used to calculate approximate second-order statistics of $\mathbf{y}$ such that,

$$\boldsymbol{\mu}_{\mathbf{y}} = \sum_{i=0}^{2n} \mathcal{W}_m^{[i]} \mathcal{Y}^{[i]} \tag{3.13}$$

and

$$\boldsymbol{\Sigma}_{\mathbf{y}} = \sum_{i=0}^{2n} \mathcal{W}_c^{[i]} (\mathcal{Y}^{[i]} - \boldsymbol{\mu}_{\mathbf{y}})(\mathcal{Y}^{[i]} - \boldsymbol{\mu}_{\mathbf{y}})^\top. \tag{3.14}$$

An illustration of the unscented transform, applied to the same one-dimensional transformation example of Section 3.1.1, is shown in Figure 3.2. Once again, the actual distribution of the transformed

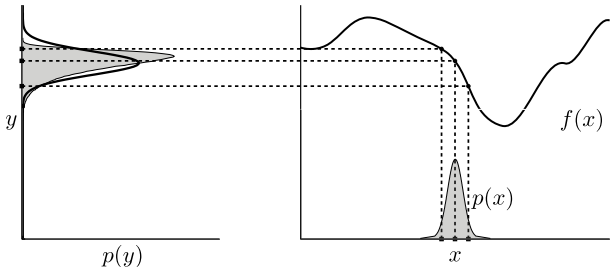


Figure 3.2: Illustration of the non-linear transformation of a Gaussian random variable using the unscented transform. Sigma points are deterministically sampled from $p(x)$ and passed through the non-linear transformation, $f(x)$. A normal distribution is then fit to the transformed sigma points to obtain a Gaussian estimate of $p(y)$. The actual transformed distribution (shaded) is shown for reference. Adapted from *Probabilistic Robotics* by Thrun et al. [90, Figure. 3.8].

random variable is given by the shaded area; however, instead of approximating $p(y)$ through a linear approximation of $p(x)$, three sigma points are selected that capture the mean and variance of $p(x)$. These sigma points are then passed through $f(x)$, and a Gaussian distribution is fit to the transformed sigma points.

There are several characteristics of the unscented transform that make it an attractive alternative to linearisation for uncertainty propagation. First, the unscented transform does not make any assumptions with regards to local linearity of the non-linear transformation; therefore, it remains applicable to even highly non-linear functions. Secondly, no calculation of Jacobians is required. This simplifies implementation, and also allows the unscented transform to be applied to non-linear functions that are not easily differentiable. Furthermore, the unscented transform has the same asymptotic complexity as linearisation with a constant time penalty [90], hence, the sampling technique is still highly efficient. A disadvantage of the unscented transform is the need for scaling parameters that require tuning. It will now be discussed how these techniques can be used for the propagation of sensor uncertainty in stereo vision applications.

3.2 Propagation of Stereo Vision Uncertainty

So far a lot of effort has been spent describing techniques used to approximate transformed measurement distributions. The reader might wonder why this is is significant. Several probabilistic computer vision techniques [82, 95, 96] make use of transformed uncertainties to obtain improved results over deterministic techniques and it is therefore critical that the approximations are representative of the actual transformed distribution. In this section, the propagation of uncertainty will be investigated for 3D feature reconstruction via triangulation of stereo measurements.

The goal is to construct a sensor model for stereo cameras so that the distribution of processed measurements can be determined and used for probabilistic calculations in later stages of a computer vision pipeline. A sensor model is probabilistic representation of the stochastic process that relates a true state, $\mathbf{x}$, to the measured values of the state, $\mathbf{z}$, obtained from a sensor,

$$p(\mathbf{z}|\mathbf{x}). \tag{3.15}$$

By applying a non-linear transform to $p(\mathbf{z}|\mathbf{x})$, the distribution of a processed measurement is then described by

$$p(\mathbf{y}|\mathbf{x}), \tag{3.16}$$

where $\mathbf{y}$ is a processed measurement.

Consider a stereo camera pair with the geometry of Figure 2.5, where a single 3D feature, $\mathbf{x}_r$, is projected to the left and right image planes. A stereo measurement is described as matched image coordinates from the left and right image planes and modelled as a Gaussian random variable such that,

$$p(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mathbf{z}, \mathbf{\Sigma}_\mathbf{x}), \tag{3.17}$$

where $\mathbf{x} = \begin{bmatrix} x_L & y_L & x_R & y_R \end{bmatrix}^\top$ is a vector of the true image projections of $\mathbf{x}_r$ on the left and right image planes and $\mathbf{z}$ is a measurement of $\mathbf{x}$. In general, errors occur in image coordinates in the location of matched features in two images due to digitization errors and matching errors [97]. It is therefore common to assume that features in the images are contaminated by additive, zero-mean Gaussian noise as stated by Hartley and Sturm [98]. Furthermore, the sources of error are assumed to be independent in the X and Y directions [99], therefore the covariance, $\mathbf{\Sigma}_\mathbf{x}$, is taken as a diagonal matrix,

$$\mathbf{\Sigma}_\mathbf{x} = \begin{bmatrix} \sigma_{x_L}^2 & 0 & 0 & 0 \\ 0 & \sigma_{y_L}^2 & 0 & 0 \\ 0 & 0 & \sigma_{x_R}^2 & 0 \\ 0 & 0 & 0 & \sigma_{y_R}^2 \end{bmatrix}. \tag{3.18}$$

The standard deviations of the image coordinates in the left and right image frames, stated in Equation 3.18, are determined experimentally during camera calibration.

A mapping from image coordinates to a 3D point in the robot coordinate frame was derived in Equation 2.32. The reconstructed 3D feature, $\mathbf{y}$, is determined from triangulation of a stereo measurement such that,

$$\mathbf{y} = \mathbf{f}_{\text{tri}}(\mathbf{z}) = \begin{bmatrix} \hat{x}_r & \hat{y}_r & \hat{z}_r \end{bmatrix}^\top, \tag{3.19}$$

where

$$\mathbf{f}_{\text{tri}}(\mathbf{z}) = \begin{bmatrix} \frac{fB}{\hat{x}_L - \hat{x}_R} \\ \frac{-B(\hat{x}_L - c_x)}{\hat{x}_L - \hat{x}_R} + \frac{B}{2} \\ \frac{B(2c_y - (\hat{y}_L + \hat{y}_R))}{2(\hat{x}_L - \hat{x}_R)} \end{bmatrix}. \tag{3.20}$$

It should be noted that triangulation is now performed on measurements, instead of the actual projected image points as implied in the previous chapter. The reconstructed 3D feature, $\mathbf{y}$, therefore contains estimated coordinates, indicated by hat notation, of the 3D feature.

This triangulation function is non-linear which complicates the uncertainty propagation process as the resulting joint distribution of processed measurements is not guaranteed to be Gaussian. A simplifying assumption is made to model the reconstructed 3D feature distribution as Gaussian such that,

$$p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{y}, \mathbf{\Sigma}_\mathbf{y}). \tag{3.21}$$

This allows the techniques described in Section 3.1 to be used for approximating $\boldsymbol{\mu}_\mathbf{y}$ and $\mathbf{\Sigma}_\mathbf{y}$.

As mentioned in Section 3.1.1, the non-linear transformation of Equation 3.20 can be approximated with a first-order Taylor series expansion. The linearisation approximation requires the Jacobian, $\mathbf{J}_{\mathbf{f}_{\text{tri}}}$, of the triangulation function given by

$$\mathbf{J}_{\mathbf{f}_{\text{tri}}} = \frac{\partial \mathbf{f}_{\text{tri}}}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\mathbf{z}} = \begin{bmatrix} \frac{-fB}{(\hat{x}_L - \hat{x}_r)^2} & 0 & \frac{fB}{(\hat{x}_L - \hat{x}_R)^2} & 0 \\ \frac{-B(c_x - \hat{x}_R)}{(\hat{x}_L - \hat{x}_R)^2} & 0 & \frac{B(c_x - \hat{x}_L)}{(\hat{x}_L - \hat{x}_r)^2} & 0 \\ \frac{-B(2c_y - (\hat{y}_L + \hat{y}_R))}{2(\hat{x}_L - \hat{x}_R)^2} & \frac{-B}{2(\hat{x}_L - \hat{x}_R)} & \frac{B(2c_y - (\hat{y}_L + \hat{y}_R))}{2(\hat{x}_L - \hat{x}_R)^2} & \frac{B}{2(\hat{x}_L - \hat{x}_R)} \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{z},} \tag{3.22}$$

where $\mathbf{J}_{\mathbf{f}_{\text{tri}}}$ is evaluated at the point of linearisation – that is, the measurement, $\mathbf{z}$. The estimated covariance of the transformed measurement distribution is then determined by

$$\boldsymbol{\Sigma}_{\mathbf{y}} = \mathbf{J}_{\mathbf{f}_{\text{tri}}} \boldsymbol{\Sigma}_{\mathbf{x}} \mathbf{J}_{\mathbf{f}_{\text{tri}}}^{\top} \tag{3.23}$$

and the transformed mean is simply,

$$\boldsymbol{\mu}_{\mathbf{y}} = \mathbf{f}_{\text{tri}}(\mathbf{z}). \tag{3.24}$$

The unscented transform requires nine sigma points to capture the second order statistics of the four dimensional prior distribution. These sigma points are determined from Equation 3.8. The choice of parameter, $\beta = 2$, is selected for Gaussian optimality [100], $\kappa = 0$ is chosen to satisfy the heuristic $(n + \kappa = 3)$ recommended by Julier et al. [93], and α is chosen empirically. The transformed sigma points are calculated by performing triangulation on each sigma point and then used to determine $\boldsymbol{\mu}_{\mathbf{y}}$ and $\boldsymbol{\Sigma}_{\mathbf{y}}$ from Equations 3.13 and 3.14 respectively.

3.3 Verification of Approximated Distributions

Monte Carlo simulations were performed to verify the assumption of Gaussianity for reconstructed, 3D feature distributions. A simulated stereo camera pair with the rectified geometry of Figure 2.5 was assumed with camera parameters as indicated in Table 3.1. Simulated 3D features were generated along the X_r

Table 3.1: Camera parameters for simulated stereo pair.

f_x (px)	f_y (px)	c_x (px)	c_y (px)	B (m)
500.0	500.0	500.0	250.0	1.0

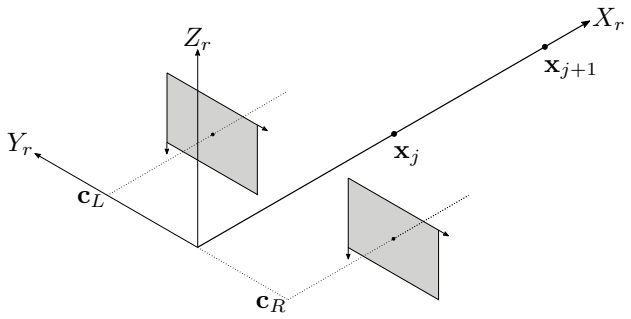


Figure 3.3: Experimental setup used to verify approximated 3D feature distributions. Simulated 3D features are generated at various depths along X_r. Two 3D features, $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$, are shown.

axis of the robot coordinate frame at several depths from the camera pair, as illustrated in Figure 3.3, and projected to image coordinates. Synthetic measurements were produced by contaminating each stereo correspondence with additive noise randomly sampled from a zero-mean, normal distribution with a diagonal covariance,

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \sigma^2 \mathbf{I}_{4\times 4}, \tag{3.25}$$

where $\sigma = 1.0$ was chosen.

A total of 1000 noisy measurements were generated and passed through the triangulation function to illustrate the true measurement distribution in the robot coordinate frame. The transformed Monte Carlo samples of two experiments, for distances of 5.0 m and 20.0 m, are depicted in Figures 3.4 and 3.5 as grey (shaded) dots respectively. The 2σ contours of the approximated distributions obtained from linearisation and the unscented transform are also shown as well as the true and approximated means.

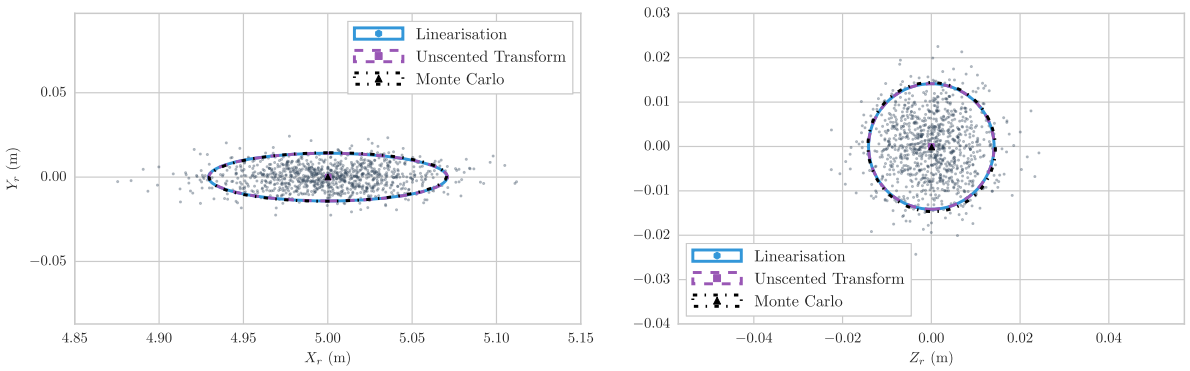


Figure 3.4: 2σ contour plots and means of linearisation and unscented transform approximations for at a distance of 5.0 m from the stereo camera pair. (a) $X_r - Y_r$ projection of 3D feature distribution. (b) $Z_r - Y_r$ projection of 3D feature distribution.

Figures 3.4a and 3.4b illustrate the validity of the Gaussian assumption used to describe uncertainty in the robot coordinate frame at a distance of 5.0 m. The estimated means and 2σ contours obtained from both approximation techniques are consistent with the results of the transformed Monte Carlo samples. As expected, the uncertainty in the viewing axis, X_r, is much larger than that of the tangential axes, Y_r and Z_r. This is consistent with a major concern when performing reconstruction from stereo correspondences, namely, highly uncertain depth measurements [62, p. 321].

This concern is further justified when working with distant reconstructed features such as in Figures 3.5a and 3.5b. The assumption of Gaussianity is less representative of the true distribution at further distances.

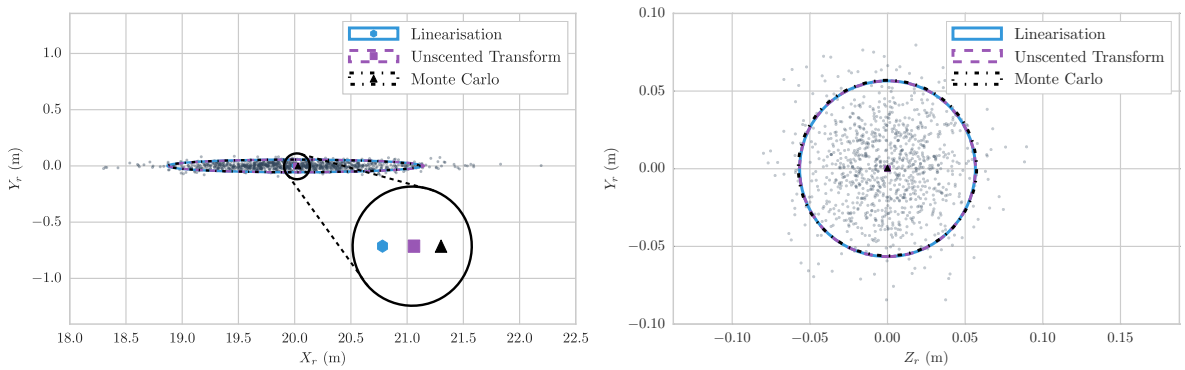


Figure 3.5: $2\,\sigma$ contour plots and means of linearisation and unscented transform approximations for a distance of 20.0 m from the stereo camera pair. (a) $X_r - Y_r$ projection of 3D feature distribution. (b) $Z_r - Y_r$ projection of 3D feature distribution.

Figure 3.5a demonstrates this for a point at 20 m; the true distribution is no longer symmetric with more measurements occurring on the far side of the mean along X_r resulting in a heavy-tailed distribution. The Gaussian approximations assume symmetry and therefore do not fully capture the uncertainty of the measurement and also result in a biased mean estimate as shown by the zoomed in portion of Figure 3.5a. Although not significantly so, the estimated mean from the unscented transform is closer to the true mean than that of linearisation. The effect of approximation errors introduced by linearisation is more pronounced at 50 m as shown in Figure 3.6. Here the difference in approximated mean is significant and suggests that the unscented transform may be better suited when only distant features are available. This warrants further investigation into the accuracy of the approximated distributions under highly uncertain conditions.

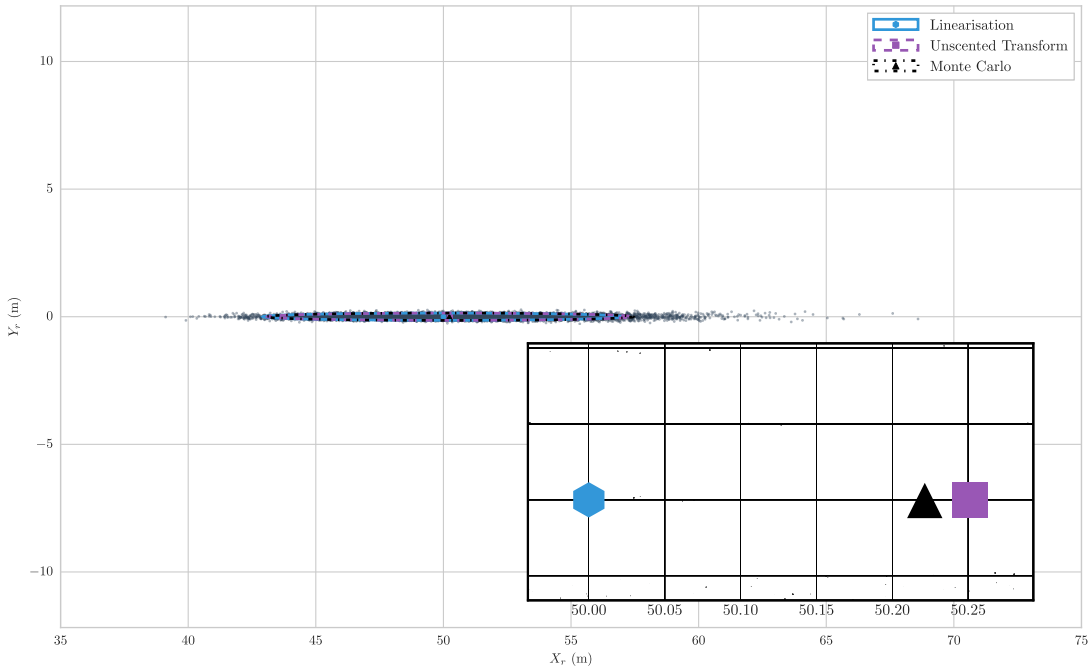


Figure 3.6: $X_r - Y_r$ projection of 3D feature distribution for a distance of 50.0 m from the stereo camera pair. Subplot indicates magnified area around approximated means.

3.4 Relative Entropy

The results of Section 3.3 indicated the validity of the Gaussian approximations made; however, several approximation errors were noted. Moreover, the qualitative results were not conclusive in comparing the unscented transform and linearisation in terms of accuracy. A quantitative evaluation of the accuracy of the approximated distributions obtained from linearisation and the unscented transform is therefore required.

The Kullback-Leibler (KL) divergence [101] is a measure of similarity between two probability distributions, P and Q, where P is usually taken as the true distribution and Q is some approximation of P. In this sense, the KL divergence is representative of the relative entropy across the two distributions – that is, a measure of the loss of information when using Q instead of P. When P and Q are d-dimensional, continuous distributions, this relative entropy is determined by

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{\mathbb{R}^d} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x} \geq 0, \tag{3.26}$$

where $p(\mathbf{x})$ and $q(\mathbf{x})$ are probability density functions of P and Q respectively, and $D_{\mathrm{KL}}(P \parallel Q)$ is the KL divergence of P with respect to Q. Intuitively, as $D_{\mathrm{KL}}(P \parallel Q)$ is only zero when $P = Q$ and increases as Q diverges from P, the KL divergence can be used as a distance metric when evaluating the similarities of probability distributions; however, it should be noted from Equation 3.26, that $D_{\mathrm{KL}}(P \parallel Q) \neq D_{\mathrm{KL}}(Q \parallel P)$ and thus is not a true metric due to asymmetry.

Determining the KL divergence requires knowledge of the two probability distributions, P and Q. However, in many applications an analytical description of the true prior distribution, P, is not available. It is therefore useful to approximate the KL divergence without the need for closed-form expression for P. A method for estimating the KL divergence for continuous random variables from independent and identically distributed (i.i.d) samples was proposed by Wang et al. [102] and further investigated by Pérez-Cruz [103].

Consider n i.i.d samples that are generated from $p(\mathbf{x})$ such that $\mathcal{X}_p = \langle \mathbf{x}_p^{[i]} \rangle_{i=1}^n$ where $\mathbf{x}_p^{[i]} \in \mathbb{R}^d$. Likewise, m i.i.d samples originating from $q(\mathbf{x})$ are given by $\mathcal{X}_q = \langle \mathbf{x}_q^{[i]} \rangle_{i=1}^n$ such that $\mathbf{x}_q^{[i]} \in \mathbb{R}^d$. It was

shown that an estimate of the KL divergence is given by

$$\hat{D}_{\mathrm{KL}}(P \parallel Q) = \frac{1}{n}\sum_{i=1}^{n} \frac{\hat{p}(\mathbf{x}_p^{[i]})}{\hat{q}(\mathbf{x}_p^{[i]})}, \tag{3.27}$$

where $\hat{p}(\mathbf{x}_p^{[i]})$ and $\hat{q}(\mathbf{x}_p^{[i]})$ are nearest neighbour density estimates [104] of $p(\mathbf{x})$ and $q(\mathbf{x})$ such that,

$$\begin{aligned} \hat{p}(\mathbf{x}_p^{[i]}) &= \frac{k}{(n-1)} \frac{\Gamma(0.5d+1)}{\pi^{0.5d} r_k(\mathbf{x}_p^{[i]})^d} \\ \hat{q}(\mathbf{x}_p^{[i]}) &= \frac{k}{m} \frac{\Gamma(0.5d+1)}{\pi^{0.5d} s_k(\mathbf{x}_p^{[i]})^d}. \end{aligned} \tag{3.28}$$

In Equation 3.28, $s_k(\mathbf{x}_p^{[i]})$ and $r_k(\mathbf{x}_p^{[i]})$ are the Euclidean distances in $\mathbb{R}^d$ from $\mathbf{x}_p^{[i]}$ to the k^{th} nearest neighbour of $\mathbf{x}_p^{[i]}$ in $\mathcal{X}_q$ and $\mathcal{X}_p \setminus \mathbf{x}_p^{[i]}$ respectively and $\pi^{0.5d}/\Gamma(0.5d+1)$ is the volume of a unit-hypersphere in $\mathbb{R}^d$. Equation 3.27 simplifies to

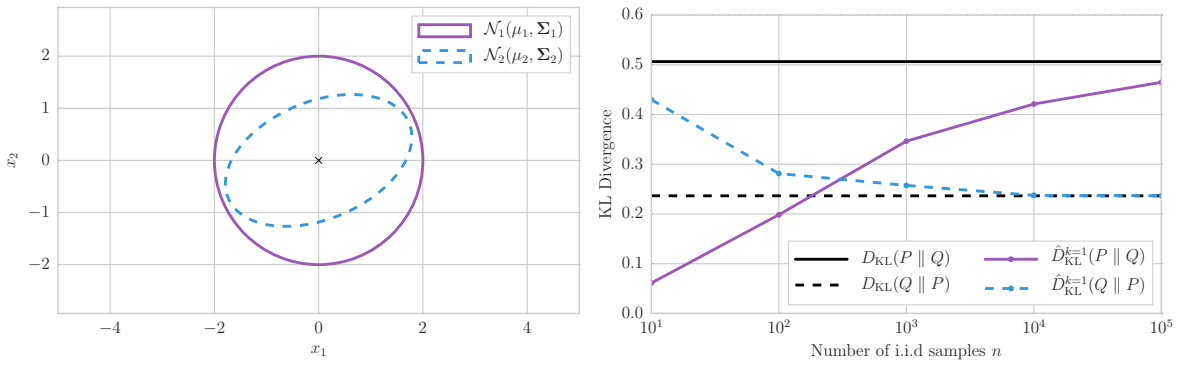
$$\hat{D}_{\mathrm{KL}}(P \parallel Q) = \frac{d}{n}\sum_{i=1}^{n} \ln \frac{s_k(\mathbf{x}_p^{[i]})}{r_k(\mathbf{x}_p^{[i]})} + \ln \frac{m}{n-1}, \tag{3.29}$$

which allows an estimate of the KL divergence to be calculated without determining the underlying distributions. It should be noted that Equation 3.29 is incorrect in the paper by Pérez-Cruz [103] and has been amended above.

An experiment was performed to verify that Equation 3.29 converges to the true KL divergence. Consider two multivariate Gaussian density distributions, $\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, shown in Figure 3.7a with means and covariances as follows:

$$\begin{aligned} \boldsymbol{\mu}_1 &= \boldsymbol{\mu}_2 = \mathbf{0}_{2\times 1} \\ \boldsymbol{\Sigma}_1 &= \mathbf{I}_{2\times 2} \\ \boldsymbol{\Sigma}_2 &= \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.4 \end{bmatrix}. \end{aligned} \tag{3.30}$$

Here, $\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ is taken as the true distribution, $p(\mathbf{x})$, and $\mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ is taken as the approximated



(a) **(b)**

Figure 3.7: Verification of nearest neighbour KL divergence estimation for 2D Gaussian random variables. (a) The $2\,\sigma$ contour plots of $p(\mathbf{x})$ (solid) and $q(\mathbf{x})$ (dashed). (b) The estimated KL divergences calculated using a k^{th} nearest neighbour estimator, $k = 1$, as a function of the sample size ($n = m$).

distribution, $q(\mathbf{x})$. We wish to determine the relative entropy when Q is used instead of P. The KL divergence is determined from Equation 3.26, which can be expressed as,

$$D_{\mathrm{KL}}(P \parallel Q) = \frac{1}{2}\left(\mathrm{Tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - d + \ln \frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right), \tag{3.31}$$

when both P and Q are d-dimensional, multivariate normal distributions [105] where $\mathrm{Tr}(\cdot)$ is the trace of a matrix. In Figure 3.7b, the true KL divergence, $D_{\mathrm{KL}}(P \parallel Q)$, is shown alongside the k^{th} $(k = 1)$ nearest neighbour estimate of the divergence, $\hat{D}_{\mathrm{KL}}^{k=1}(P \parallel Q)$. Several simulations were performed while varying the number of i.i.d samples $(n = m)$ generated for each distribution. The curves shown represent the mean values of 50 independent experiments performed and for completeness, the converse divergence, $D_{\mathrm{KL}}(Q \parallel P)$, and its respective estimate are also shown. As can be seen in Figure 3.7b, the estimate, $\hat{D}_{\mathrm{KL}}^{k=1}(P \parallel Q)$ converges to the true KL divergence as $n \to \infty$. Likewise, the estimate of $D_{\mathrm{KL}}(Q \parallel P)$ also converges to the correct divergence value obtained from Equation 3.31. These results illustrate the viability of the nearest neighbour estimator for determining relative entropy when an analytical description of the approximated distribution is not available.

3.5 Experimental Results

In this section the approximated distributions obtained from linearisation and the unscented transform are evaluated quantitatively. Simulated experiments are performed to compare the accuracy of reconstructed, 3D feature point distributions with synthetic stereo vision data using the same experimental setup as Section 3.3. In the first experiment, the KL divergence estimator discussed in Section 3.4 is used to evaluate the similarity of the two approximated distributions in relation to the true, underlying 3D point distribution. In the second experiment, the closed-form KL divergence of the approximations relative to best-fit Gaussian distributions are determined. The results of 100 independent experiments are shown in Figures 3.8 and 3.9. Error bars show the standard deviations of the simulation results and shading represents the 95% confidence interval[2] of the mean result for each experiment. These two experiments are now discussed in more detail.

3.5.1 Relative Entropy from Actual Distribution

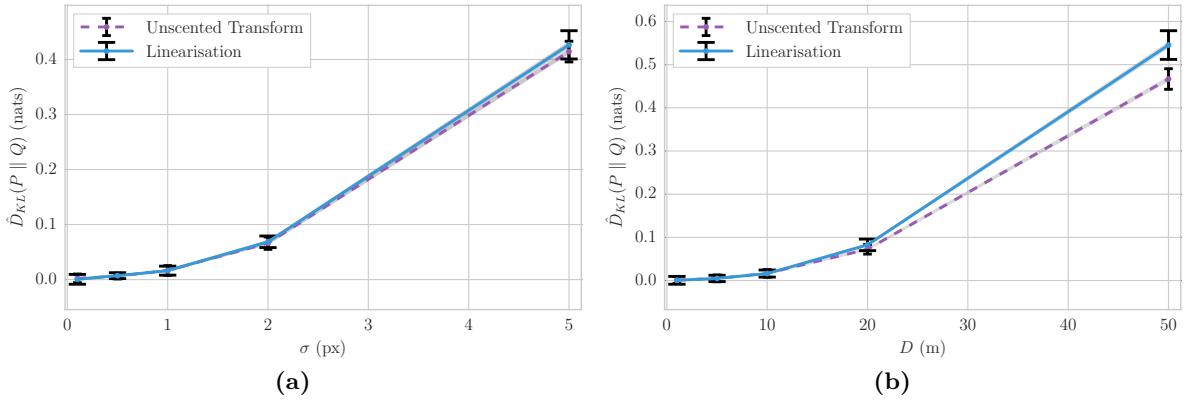
In the first experiment, linearisation and the unscented transform are used to approximate the distribution of a reconstructed, 3D point when performing triangulation on synthetic stereo vision data. The experimental setup depicted in Figure 3.3 was repeated. A total of 10 000 simulated stereo measurements were generated by projecting the 3D feature to image coordinates and contaminated with additive noise randomly sampled from a zero-mean normal distribution. For the sake of simplicity, it was assumed that the covariance of the additive noise in the image coordinates is a diagonal matrix with a single variance, σ^2, across all noise components such that,

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \sigma^2 \mathbf{I}_{4\times 4}. \tag{3.32}$$

Several simulations were then performed where the distance to the sensor and the noise level, σ, were varied. For simulations where the distance to the sensor was varied, the noise level was kept constant at $\sigma = 1.0\,\mathrm{px}$. Simulations were also performed where the distance of the 3D feature from the origin of the robot coordinate system was kept constant at 10 m while the noise level was varied. As there is no closed-form expression for the actual distribution of the reconstructed feature point, an approximated version of the KL divergence, stated in Equation 3.27, is used.

The KL divergence results obtained from synthetic stereo vision data are shown in Figure 3.8. Figure 3.8a illustrates that linearisation and the unscented transform perform similarly when the noise level is varied. As the noise level is increased from $\sigma = 0.1\,\mathrm{px}$ to $\sigma = 5.0\,\mathrm{px}$, the approximated distributions become less representative of the underlying distribution as indicated by greater KL divergences. However, the difference in KL divergence of the two approximation techniques is insignificant even at a noise level of $\sigma = 5.0\,\mathrm{px}$. A significant effect seen in Figure 3.8b, is how both approximation techniques perform poorly when only distant features are available. The unscented transform, however, clearly outperforms

[2] Confidence intervals are detailed in Appendix A.1.



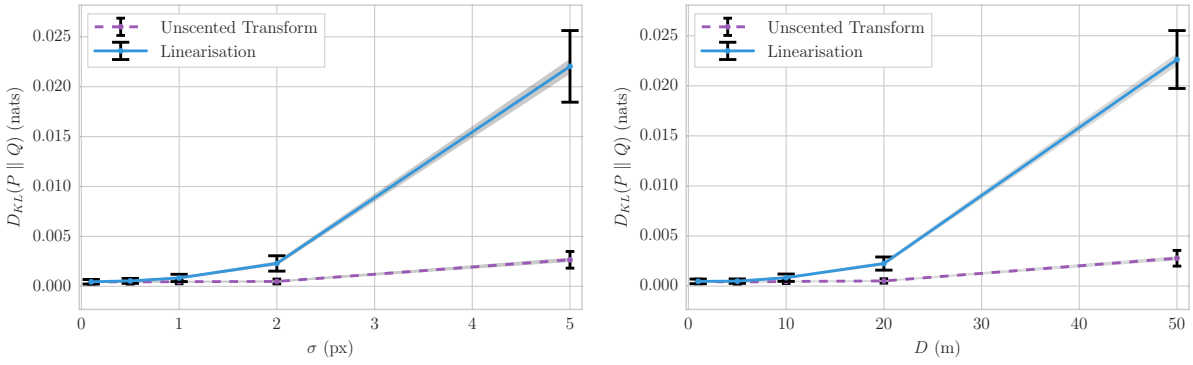
(a) (b)

Figure 3.8: Relative entropy of linearisation and the unscented transform approximations from actual 3D feature distribution. (a) The approximated KL divergence as a function of noise level for a constant depth of 10 m. (b) The approximated KL divergence as a function of distance from stereo camera pair for a constant noise level of $\sigma = 1.0$ px.

linearisation. The approximations are equivalent up to a range of 20 m, after which linearisation performs notably worse than the unscented transform.

3.5.2 Relative Entropy from Best Fit Gaussian Distribution

The sources of the divergence seen in Section 3.5.1 are twofold; first, the Gaussian assumption itself, and secondly, the method used to approximate the second-order statistics describing the Gaussian assumption. It is therefore useful to evaluate the accuracy of the unscented transform and linearisation with the effect of the Gaussian assumption removed. This is done by obtaining the best Gaussian fit of the actual distribution by using moment matching [106] and comparing linearisation and the unscented transform using the closed-form KL divergence in Equation 3.26. Figure 3.9a shows the result of varying the noise



(a) (b)

Figure 3.9: Relative entropy of linearisation and the unscented transform approximations from best fit Gaussian approximation. (a) The KL divergence of approximated distributions as a function of noise level for a constant depth of 10 m. (b) The approximated KL divergence as a function of distance from stereo camera pair for a constant noise level of $\sigma = 1.0$ px.

level while keeping the depth of the 3D feature constant at 10 m. Quite clearly, under these conditions, the unscented transform results in significantly better approximations of the second order statistics for reconstructed 3D features as shown by the minor divergence from the best fit Gaussian distribution. The results of varying the depth while keeping a constant noise level of $\sigma = 1.0$ px are shown in Figure 3.9b. Once again, it can be seen that the unscented transform significantly outperforms linearisation.

3.6 Chapter Summary

The effect of measurement uncertainty with regards to 3D reconstruction was considered in this chapter. Stereo measurements were modelled as Gaussian random variables which undergo a non-linear transformation. Two techniques, linearisation and the unscented transform, were investigated and used to approximate the distributions of reconstructed 3D features. The techniques were compared using the KL divergence and synthetic stereo datasets for several noise levels and depths. Linearisation was shown to perform notably worse than the unscented transform; an increase in accuracy was especially apparent when using the unscented transform for features far away from the camera system where measurements are highly uncertain.

The difference in accuracy of the approximated distributions can be attributed to the mechanisms of the two approaches. Linearisation approximates the non-linear transformation as a linear function at the mean, which is a poor approximation when measurements are highly uncertain and distant. Both techniques suffer from approximation errors when compared to the actual distribution, as was shown in Section 3.5.1, with the unscented transform only performing slightly better. However, the unscented transform was significantly more accurate than linearisation when the effect of the Gaussian assumption was removed. Since the Gaussian assumption is necessary for the purpose of tractability, it can be concluded that the unscented transform is better suited for noise propagation in 3D reconstruction purposes. The discussion now shifts to Part II of this work which is concerned with a novel outlier removal framework.

Part II

Outlier Removal Framework

Chapter 4

Overview of Outlier Removal

A brief introduction to outliers and outlier removal in the context of computer vision was provided in Chapter 1. Outliers were defined as feature matches that are inconsistent with the motion of the camera system between successive viewpoints, and the importance of outlier removal for robust motion estimation was discussed. Several outlier removal methods were highlighted in Section 1.2.3 where the most commonly implemented approach for visual navigation systems was identified as RANSAC.

A detailed overview of state-of-the-art outlier removal techniques is the focus of this chapter. A number of recent research efforts have attempted to improve the performance of the standard RANSAC algorithm. Several of the proposed methods have focused on optimising the sampling strategy of RANSAC, while others have modified the way in which hypotheses are generated and verified [79]. These adapted RANSAC techniques and alternative outlier removal methods are analysed in this chapter to highlight the current difficulties of outlier removal. A novel mechanism for outlier removal is then developed (Chapter 6), inspired by some of the concepts discussed in this chapter, in an attempt to resolve these difficulties.

This chapter begins by introducing the standard RANSAC algorithm where the mechanism and the limitations of the approach are discussed. Thereafter, several extensions to the RANSAC algorithm are detailed which address limitations in terms of robustness and efficiency. The chapter concludes with a discussion of an alternative formulation of the outlier removal problem known as inlier detection.

4.1 Standard RANSAC

The RANSAC algorithm, proposed by Fischler and Bolles [68], is a well known technique used for the estimation of model parameters in the presence of outliers. The framework is based on an iterative, hypothesise-and-verify approach which is robust to contaminated[1] observations and has become a standard in computer vision applications.

4.1.1 RANSAC Approach

The standard RANSAC framework consists of two stages which are performed iteratively on a set of observations; namely a hypothesise stage and a verification stage. The first stage instantiates a model from a set of randomly selected points, where the cardinality of the sampled set is equal to the smallest number of points required to determine the model's parameters. This is also known as a minimal set. In the second stage of the algorithm, a measure of consensus is determined for each hypothesised model by determining the number of observations that are consistent with the model. Consistency is usually defined in terms of some predefined error threshold. After several iterations of these two steps, the model with the highest support is returned. A breakdown of the RANSAC approach is given in Algorithm 4.1.

It is important to note that RANSAC is a non-deterministic algorithm due to the random sampling of points and therefore a correct model estimate is not guaranteed. The probability of obtaining a correct

[1] A *contaminated* set of observations contains one or more outliers.

Algorithm 4.1 Standard RANSAC

Input:

$\mathcal{Z}$	Sequence of m-dimensional observations: $\langle \mathbf{z}^{[i]} \rangle_{i=0}^{N}$
Δ	Threshold for determining model consistency
η	Probability of success
ϵ	Fraction of inliers

Output:

θ_M	Model estimate with largest support
$\mathcal{I}$	Set of inlier observations consistent with θ_M

function RANSAC($\mathcal{Z}, \Delta, \eta, \epsilon$)
  $\mathcal{I} \leftarrow \emptyset$
  $k_r \leftarrow$ RequiredIterations(η, ϵ, m) ▷ Equation 4.1
  for k_r iterations **do**
    $\mathcal{S} \leftarrow$ RandomSample($\mathcal{Z}, m$) ▷ $\|\mathcal{S}\| = m$
    $\hat{\theta}_{k_r} \leftarrow$ HypothesiseModel($\mathcal{S}$)
    $\mathcal{I}_{k_r} \leftarrow$VerfiyModel($\hat{\theta}_{k_r}, \mathcal{Z}, \Delta$)
    if $|\mathcal{I}_{k_r}| > |\mathcal{I}|$ **then**
      $\theta_M \leftarrow \hat{\theta}_{k_r}$
      $\mathcal{I} \leftarrow \mathcal{I}_{k_r}$
    end if
  end for
  return $\mathcal{I}, \theta_M$
end function

model estimate is increased by iteratively performing the hypothesise-and-verify stages. The probability of finding a correct model, η, is equivalent to randomly selecting an uncontaminated minimal subset of points from the set of observations – that is, sampling a set of m observations, which does not contain outliers, from $\mathcal{Z}$. Assuming that points are sampled independently and following the derivation of Fischler and Bolles [68], the number of iterations, k_r, required to satisfy η is determined by

$$k_r = \frac{\ln(1-\eta)}{\ln(1-\epsilon^m)}, \tag{4.1}$$

where ϵ is the probability of sampling a single inlier (also known as the inlier ratio) from $\mathcal{Z}$.

The model parameters with the largest support[2], θ_M, after k_r iterations have been performed are then used as the estimated model. Alternatively, observations identified as inliers from the largest consensus set, $\mathcal{I}$, can be used to estimate a refined model – that is, model parameters are re-estimated from a non-minimal set of inlier by performing outlier removal.

4.1.2 Limitations of RANSAC

Two implicit assumptions are made by the standard RANSAC framework when determining the number of iterations as shown in Equation 4.1. These assumptions are not necessarily valid in practical implementations of the algorithm and therefore need to be addressed. First, an assumption is made that model parameters generated from an inlier set are consistent with all other inliers in the dataset. Secondly, it is assumed that a model instantiated from a set of observations containing at least one outlier, results in poor support. This means that correct and erroneous models are distinguishable by evaluating the support of the model.

The first assumption is not valid when observations are corrupted by significant levels of noise. Models instantiated from noisy inlier samples are not guaranteed to be consistent with all other inliers. The number of effective inliers is therefore reduced and as a result, the number of iterations required by

[2]The *support* of a model is defined as the number of observations which are consistent with the model.

RANSAC is increased. This was demonstrated by Tordoff and Murray [107] where it was shown that the number of iterations determined from Equation 4.1 is optimistic and obtaining accurate[3] estimates in the presence of noisy observations may require the number of iterations to be higher by an order of magnitude or more. Furthermore, degenerate configurations of sampled points may generate estimated models with large support even if the samples are contaminated with outliers [71, 95] – violating the second assumption.

It has already been mentioned in Section 1.2.2 that one of the main reasons for RANSAC's popularity stems from its ability to handle large numbers of outliers [79]. However, the associated computational cost is significant for high levels of contamination. The number of RANSAC iterations required for different values of m and ϵ are shown in Table 4.1. Quite clearly, the number of iterations grows dramatically as

Table 4.1: Number of RANSAC iterations required to instantiate a correct model with $\eta = 0.95$ for different model complexities, m, and inlier ratios, ϵ.

	ϵ			
m	0.9	0.7	0.5	0.3
2	2	5	11	32
3	3	8	23	109
5	4	17	95	1.23×10^3
7	5	35	382	1.37×10^4

contamination level increases, which limits the use of RANSAC for real-time systems where high numbers of outliers may be encountered. Furthermore, from Table 4.1, it can be seen that the number of points needed to estimate the model also affect the required iterations. Minimal model parametrisations are therefore of high interest as they lead to more efficient implementations.

4.2 Extensions to RANSAC

There are several aspects of the standard RANSAC algorithm which have been extended for improved efficiency and robustness. Proposed methods which focus on robust measures of consensus are discussed in Section 4.2.1. Section 4.2.2 details attempts to optimise model verification. Techniques for improved hypothesis generation are presented in Section 4.2.3 and Section 4.2.4 presents alternative sampling strategies.

4.2.1 Consensus Measure

The standard RANSAC algorithm evaluates model support by determining the number of observations with an error[4] below a certain threshold. The choice of this threshold is therefore critical as a threshold set too high will result in outliers not being rejected correctly and consequently, poor estimates. Several robust measures of consensus have been proposed and are discussed in this section.

4.2.1.1 MLESAC

An extension of the RANSAC framework, dubbed maximum likelihood estimation sample consensus (MLESAC) and proposed by Torr and Zisserman [108], makes use of a probabilistic cost function as opposed to cardinality of the inlier set as a measure of consensus. Instead of simply maximizing the support for a model, random sampling is used to maximise the likelihood of hypothesised models. The

[3] An *accurate* estimate is defined as a model which is consistent with at least 75% of the inlier observations in the dataset. This definition is synonymous with that of a "good model estimate" used by Tordoff and Murray [107].

[4] A re-projection error or Sampson distance is generally used as an error function [62, pp. 94–100].

residual error [62, pp. 133-134] of the i^{th} point, $e_{\text{res}}^{[i]}$, given the current estimate of model parameters, $\hat{\theta}_M$, is modelled as a mixture of Gaussian and uniformly distributed random variables, such that,

$$p(e_{\text{res}}^{[i]}|\hat{\theta}_M) = \epsilon \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-e_{\text{res}}^2}{2\sigma^2}} + (1-\epsilon)\frac{1}{v}. \tag{4.2}$$

The parameter, ϵ, is the probability of a single observation being an inlier, σ is the standard deviation of Gaussian noise in image coordinates, and v is the width of the search window in which outliers can occur. It should be noted that Torr and Zisserman [108] assume that the probability of an observation being an inlier is the same across all points. An initial estimate of the fraction of inliers is used for ϵ, after which the expectation maximisation (EM) [109] algorithm is used to recursively estimate ϵ for each hypothesis. Hypothesised models are generated in the same manner as standard RANSAC by randomly sampling minimal sets and fitting a model to the sampled points. During verification, the best model is chosen by minimising the negative log likelihood,

$$-L = -\sum_{i=0}^{n} \log p(e_{\text{res}}^{[i]}|\hat{\theta}_M), \tag{4.3}$$

over a number of iterations as determined from Equation 4.1.

Torr and Zisserman [108] conducted a comparison of MLESAC and RANSAC for fundamental matrix estimation on both synthetic and real images. It was shown that MLESAC produced results equal to or superior to that of RANSAC at the expense of additional computational overheads. However, as pointed out by Tordoff and Murray [107], MLESAC suffers from the same limitations as standard RANSAC where the number of iterations required to obtain an accurate estimate is significantly higher than predicted by Equation 4.1.

4.2.1.2 Probabilistic Consensus Measure

A probabilistic outlier removal method based on the standard RANSAC algorithm was formulated by Brink et al. [82]. The proposed technique determines consensus in 3D coordinates, similar to the work of Dubbelman et al. [110], by calculating a probabilistic measure of how well two, time-corresponding 3D points align after motion. The use of 3D landmarks over image features is motivated by the availability of 3D point estimates in SLAM frameworks and by the more efficient model parametrisation when compared to essential and fundamental matrix RANSAC approaches. Furthermore, with the probabilistic consensus measure, measurement uncertainty is incorporated as opposed to the naïve approach of using the Euclidean distance between 3D points. The proposed RANSAC framework was compared against standard fundamental matrix RANSAC for SLAM by Brink et al. [82]. A significant improvement in accuracy was achieved by probabilistic RANSAC as well as a faster execution time.

4.2.2 Hypotheses Verification

Robust methods for consensus measure were discussed in the previous section. The focus now shifts to improving the way in which hypotheses are verified.

As discussed previously, the efficiency of the RANSAC algorithm is heavily dependent on the number of outliers in the dataset as well as the total number of observations. For high levels of contamination, a large number of erroneous models are verified against all observations, and consequently, significant portions of computation time are spent verifying models which are inconsistent with the majority of observations in the dataset. Several efforts have attempted to optimise the verification stage by reducing the amount of time spent evaluating erroneous models.

4.2.2.1 R-RANSAC

Matas and Chum [111] proposed an optimised version of RANSAC, known as randomised RANSAC (R-RANSAC), which incorporates the use of randomised hypothesis evaluation to filter out incorrect

hypotheses early in the verification stage. As a result, fewer models are verified against the full observation set and the overall execution time is reduced.

The speed increase of R-RANSAC is achieved by introducing a statistical test, which is only performed on a small subset of points, d out of N observations where $d \ll N$, as a pre-verification step. Hypothesised models are then only verified against the full set of observations when the statistical test is passed. This introduces two intermediary probabilities, γ and ζ, which describe the probability of a uncontaminated and contaminated sample set passing and failing the pre-verification test respectively. Consequently, Equation 4.1 must be adjusted,

$$k = \frac{\ln(1-\eta)}{\ln(1-\gamma\epsilon^m)}, \tag{4.4}$$

to compensate for uncontaminated sample sets which are rejected by the statistical test. It should be noted from Equation 4.4 that the inclusion of γ has increased the number of iterations required to guarantee η. At first, this might appear counter-intuitive as the goal is to reduce the computation time of RANSAC. However, as the majority of incorrect models are now rejected before the verification step, the average number of observations evaluated per hypothesised model is now reduced [111]. R-RANSAC was implemented with a $T_{d,d}$ pre-test[5] [71, 111], on both synthetic and real world datasets for epipolar geometry estimation. Significant speed increases were obtained for both narrow and wide baseline stereo pairs.

4.2.2.2 Optimal Randomised RANSAC

An optimal randomised, hypothesis evaluation approach was proposed by Chum and Matas [112] as an extension to R-RANSAC – dubbed WaldSAC. The verification stage of the RANSAC algorithm is formulated as an optimisation problem which minimises the number of verifications performed when evaluating hypothesised models. A sequential probability ratio test (SPRT), proposed by Wald [113], describes a cumulative likelihood ratio, λ_d, determined from d samples. If λ_d exceeds a certain threshold, A, the hypothesised model is rejected, otherwise the model is accepted if $\lambda_d < A$ for all N observations. The choice of A is important as a smaller threshold value means fewer correspondences are verified per model at the expense of rejecting potentially accurate models. An optimal choice of A can be derived if the inlier ratio, ϵ, and the probability of a point being consistent with a poor model, δ, are known [112]. In practice, however, these parameters are unknown and need to be estimated iteratively. In much the same way as the original R-RANSAC implementation, simulated and practical experiments were performed for epipolar geometry estimation[112]. WaldSAC was shown to be significantly faster than standard RANSAC and slightly faster than R-RANSAC with the $T_{d,d}$ test.

4.2.3 Hypotheses Generation

The previous section focused on improved versions of RANSAC that optimised the way in which models were verified. This section discusses methods that improve the generation of hypotheses.

Hypotheses are generated by randomly sampling minimal subsets from the observation set and fitting a model to the sampled points. As pointed out in Section 4.1.2, models instantiated from noisy samples are not always consistent with all inlier points. Moreover, the number of iterations required must be increased, which limits the use of RANSAC for real-time applications. Several improvements to the way in which hypotheses are generated have been proposed to increase efficiency and robustness.

4.2.3.1 Lo-RANSAC

The Lo-RANSAC algorithm [114] aims to obtain a more robust model fit when generating hypotheses. This particular variation of RANSAC directly addresses the incorrect assumption that model parameters

[5] In the $T_{d,d}$ test, the hypothesised model is validated against d sampled points, and the test is passed if all d points are consistent with the model. A value of $d = 1$ is recommended by Matas and Chum [111].

determined from an outlier-free minimal set are consistent with all inliers by incorporating a local optimisation step performed on select hypotheses. An inner RANSAC loop, which is run for a fixed number of iterations, is performed on inlier sets corresponding to the best current model estimate. As a result, a higher quality model is instantiated at the expense of increased computational costs. The additional computational costs are negated by the fact that consensus scores are improved more rapidly than with the standard RANSAC algorithm and the outer RANSAC loop therefore requires fewer iterations to guarantee an accurate model estimate [79].

4.2.3.2 Preemptive RANSAC

Nistér [115] proposed the preemptive RANSAC algorithm which is targeted at real-time SfM applications. A fixed number of hypotheses are generated and scored in parallel. This approach is significantly different than the techniques discussed so far. Instead of employing a "depth-first" approach [79], where a hypothesis is completely evaluated before progressing to further hypotheses, a "breadth-first" approach is employed where several models are instantiated and evaluated simultaneously. Each hypotheses is verified on randomly sampled points according to a cost function and the worst performing models are rejected iteratively until a single hypothesis survives.

The preemptive RANSAC formulation is specifically suited for applications where real-time operation is essential. The algorithm is efficient and has a constant runtime; however, it is prone to fail if the fraction of inliers is too low as the generated hypotheses may not contain an uncontaminated model.

4.2.4 Sampling Strategy

In the standard RANSAC framework, samples are drawn uniformly from the observation set. This assumes that all observations are equal and that every observation has the same prior probability of being an inlier; however, this assumption is naïve as all observations are not equal – for example, certain feature matches are more likely to be inliers than others depending on their match scores. In many implementations, information is often available that allow observations to be scored according to how likely they are of being an inlier and several approaches take advantage of this fact to generate better sample sets.

4.2.4.1 Guided-sampling

An improvement to the MLESAC framework was proposed by Tordoff and Murray [107] based on the guided sampling of observations. They argued that an observation more likely to be an inlier should be sampled more often and that information is inherently available in the feature matching process that allows matched features to be ordered in terms of quality. By guiding the sampling process to select higher quality data points more frequently, the number of iterations required by RANSAC is reduced.

As opposed to the original MLESAC algorithm, Tordoff and Murray [107] proposed the use of individual probabilities, $\epsilon^{[i]}$, for each point in Equation 4.2 such that,

$$p(e_{\text{res}}^{[i]}|\hat{\theta}_M) = \epsilon^{[i]} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-e_{\text{res}}^2}{2\sigma^2}} + (1-\epsilon^{[i]})\frac{1}{v}. \tag{4.5}$$

Furthermore, it was shown that it is possible to estimate $\epsilon^{[i]}$ in stereo matching applications using the number of potential matched features and the correlation scores of the matches. The uniform sampling of points from the observation set is then replaced with a Monte Carlo sampling approach according to $\epsilon^{[i]}$. Consequently, point correspondences that have a higher prior probability of being an inlier are sampled more often and as a result, the number of iterations required to guarantee a certain confidence was reduced by an order of magnitude [79].

4.2.4.2 PROSAC

Chum and Matas [72] proposed an improved hypothesise-and-verify technique known as progressive sample and consensus (PROSAC), which takes advantage of a linear ordering of the observation set to achieve increased computational performance. The mechanism of PROSAC is almost identical to that of RANSAC except that the points used to determine hypotheses are not sampled from the full observation set. Instead samples are drawn from a growing subset which only contains the most likely inlier points. This is similar to the guided sampling approach of Tordoff and Murray [107], which takes advantage of the fact that better hypotheses are generated from samples that are more likely to be inliers. However, there are two important differences between the two approaches. First, PROSAC does not explicitly require estimates of the probabilities, $\epsilon^{[i]}$. Furthermore, PROSAC dynamically adjusts the sampling strategy based on the results of each sampling iteration. At worst, PROSAC performs equivalently to standard RANSAC, although generally far fewer iterations are required to find an accurate model estimate [71, 82].

4.3 Inlier Detection

An alternative formulation of the outlier removal problem, described as inlier detection, was detailed by Hirschmuller et al. [81] and extended by Howard [6]. Instead of randomly sampling the observation set and rejecting potential outliers, inlier detection attempts to construct a set of mutually consistent observations by exploiting the rigidity of a scenes between time steps. The relationship between two sequences of 3D points in a previous and current camera frame, $\mathcal{X}_{k-1} = \langle \mathbf{x}_{k-1}^{[i]} \rangle_{i=1}^{N}$ and $\mathcal{X}_k = \langle \mathbf{x}_k^{[i]} \rangle_{i=1}^{N}$, is described by a rigid body transform consisting of a rotation and translation such that,

$$\mathbf{x}_k^{[i]} = \mathbf{R}\mathbf{x}_{k-1}^{[i]} + \mathbf{t}. \tag{4.6}$$

One of the properties of a rigid body transformation is that the relative Euclidean distance between two points, denoted by i and j, and their corresponding transformed points remains constant. This introduces the constraint,

$$\left\| \mathbf{x}_k^{[i]} - \mathbf{x}_k^{[j]} \right\| - \left\| \mathbf{x}_{k-1}^{[i]} - \mathbf{x}_{k-1}^{[j]} \right\| < \Delta_e, \tag{4.7}$$

where Δ_e is a user defined error threshold to incorporate measurement noise. In the case where this inequality does not hold, it can be inferred that one of the two 3D features is an outlier and therefore should not be used to determine the relative camera motion.

To aid in the identification of a mutually consistent inlier set, a consistency matrix, $\mathbf{W}$, is constructed for all pairwise combinations of the 3D points. The element W_{ij} is assigned a value of one if the constraint is satisfied and a zero otherwise. Finding the largest set of mutually consistent points is equivalent to finding the maximum clique contained in a graph with adjacency matrix $\mathbf{W}$. Although extremely robust to outliers, the complexity of populating $\mathbf{W}$ is quadratic and the maximum clique problem is NP-complete [6]. This limits the use of the technique in real-time systems, however, a sub-optimal algorithm is proposed by Howard [6].

4.4 Chapter Summary

The RANSAC algorithm, a standard for outlier removal in computer vision applications, was presented in this chapter. Limitations were discussed in terms robustness to noise as well as poor efficiency when datasets are highly contaminated.

Several adaptations of the RANSAC algorithm were then discussed. MLESAC [108] and the probabilistic framework of Brink et al. [82] provide robust techniques for calculating model support. Two methods, R-RANSAC [111] and WaldSAC [112], optimise model verification by incorporating a pre-verification stage where poor hypotheses are rejected early. As a result the average execution time is reduced. The Lo-RANSAC [114] makes use of an inner RANSAC loop as a local optimisation step for hypotheses

generation with an increased robustness to noisy measurements at the expense of additional computational costs. The highly efficient preemptive RANSAC [115] algorithm generates and scores a fixed number of hypotheses from random samples in a parallelised framework. Although capable of running in real time, the algorithm performs unreliably when the observation set is highly contaminated. The guided sampling [107] and PROSAC [71] approaches both employ adapted sampling strategies that select points with a higher probability of being an inlier more often, thereby reducing the number of iterations required by RANSAC.

An alternative to RANSAC-based outlier removal was also presented. An inlier detection approach proposed by Hirschmuller et al. [81] and Howard [6] attempts to find the largest mutually consistent set of inliers by employing geometric constraints. Although robust to outliers, an optimal solution to the maximal clique problem is intractable for real-time solutions.

From the aforementioned outlier removal techniques, two classes of solutions appear to exist, namely, hypothesise-and-verify approaches which employ random sampling and a deterministic technique approach used to identify inlier sets. In this thesis, a novel outlier removal approach is developed (Chapter 6) which does not fall into either of these classes and addresses the limitations of RANSAC. The probabilistic RANSAC framework developed by Brink et al. [82] is chosen as the benchmark against which the proposed technique will be compared. This decision is based on the fact that a new mechanism is being proposed and a fair comparison requires the technique to be contrasted with the standard RANSAC approach. The framework of Brink et al. [82] is efficient, due to its minimal model parametrisation, as well as robust to measurement noise. Furthermore, it is simple to implement, makes use of 3D features which are readily available in visual odometry systems and makes no further assumptions about the data. The next chapter focuses on the development and analysis of a probabilistic RANSAC framework incorporated into a visual odometry pipeline.

Chapter 5

Robust Visual Odometry

In the previous chapter, a brief overview of several outlier removal techniques was provided. The focus now shifts to the development of a robust visual odometry pipeline that employs the probabilistic RANSAC framework proposed by Brink et al. [82]. This will then act as the benchmark for comparison against the novel outlier removal technique developed in the next chapter.

Visual odometry attempts to estimate the ego-motion of a robot from a sequence of images; however, it has already been mentioned that mismatched features are unavoidable in practical vision systems. Least square estimation is extremely sensitive to these outliers and outlier removal is therefore required to achieve robust motion estimation. A standard RANSAC approach, with some adaptations, is developed in conjunction with a simulated visual odometry system. Furthermore, a motion model is developed to generate randomised robot trajectories as well as synthetic stereo datasets. These synthetically generated datasets allow experiments to be performed under various conditions such as high noise levels or where measurements are highly contaminated with outliers.

This chapter begins with an outline of the stereo visual odometry pipeline assumed, before moving on to a detailed overview and analysis of the probabilistic RANSAC framework implemented. Thereafter, the process of motion estimation refinement, the final stage of the visual odometry pipeline, is presented. The motion model and simulated robot system used to generate synthetic datasets are then discussed. The chapter concludes with experimental results of robust visual odometry experiments performed on synthetic datasets.

5.1 Stereo Visual Odometry Pipeline

Visual odometry is defined as the process of estimating the ego-motion of a robot from a set of stereo images captured at successive time steps. In this section, a robust stereo visual odometry pipeline is developed. The pipeline is largely unchanged from the original formulation by Moravec [40], discussed in Section 1.2.1, consisting of feature detection, feature matching and relative transformation estimation. The only difference is the inclusion of an outlier removal method, discussed in Section 1.2.3, to allow for robust motion estimation.

An outline of the steps performed in a stereo visual odometry framework is given below.

1. Identify and match features in the left and right image planes at the previous time step, $k-1$, such that
$$\mathcal{F}_{k-1} = \left\langle \mathbf{z}_{k-1}^{[i]} \right\rangle_{i=1}^{N}, \tag{5.1}$$
where $\mathbf{z}_{k-1}^{[i]} = \begin{bmatrix} \hat{x}_L^{[i]} & \hat{y}_L^{[i]} & \hat{x}_R^{[i]} & \hat{y}_R^{[i]} \end{bmatrix}^\top$ is a stereo measurement of a 2D feature identified at time $k-1$ and $\mathcal{F}$ denotes a sequence of image features.

2. Reconstruct 3D landmarks, $\mathbf{x}_{r,k-1}^{[i]}$, via triangulation of stereo features (Equation 2.32),
$$\mathcal{X}_{k-1} = \langle \mathbf{x}_{r,k-1}^{[i]} \rangle_{i=1}^{N}, \tag{5.2}$$

where $\mathbf{x}_{r,k-1}^{[i]} = \mathbf{f}_{\text{tri}}(\mathbf{z}_{k-1}^{[i]})$ and $\mathcal{X}$ denotes a sequence of 3D points.

3. Match the features, $\mathcal{F}_{k-1}$, to their corresponding features at the current time step, $\mathcal{F}_k$, such that elements in sequence $\mathcal{F}_k$ are ordered according to their matches with $\mathcal{F}_{k-1}$.

4. Determine an index set, $\mathcal{I}$, which excludes outlier observations in $\mathcal{F}_{k-1}$ and $\mathcal{F}_k$.

5. Estimate relative transformation parameters, $\mathbf{R}_{k,k-1}$ and $\mathbf{t}_{k,k-1}$, that minimise the image reprojection error of the points, $\langle \mathcal{F}_{k-1}^{[i]} \rangle_{i \in \mathcal{I}}$ and $\langle \mathcal{F}_{k}^{[i]} \rangle_{i \in \mathcal{I}}$.

6. Concatenate the relative transformation with the previous estimate of the robot pose,

$$\mathbf{P}_k = \mathbf{P}_{k-1}\mathbf{T}_{k,k-1}, \tag{5.3}$$

where $\mathbf{P}_k$ is the pose of the robot at time k, and

$$\mathbf{T}_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}_{3\times 1} & 1 \end{bmatrix}. \tag{5.4}$$

The reconstructed robot trajectory up to time k is then described by $\mathcal{P} = \langle \mathbf{P}_i \rangle_{i=0}^{k}$.

For the simulated visual odometry pipeline implemented in this chapter, sequences of matched features are synthetically generated (discussed in Section 5.4). Mechanisms for feature detection and matching are therefore not implemented, although a brief overview of existing methods is provided in Section 1.2.1. The discussion instead begins with a probabilistic RANSAC method (Step 4) which is developed in Section 5.2 and is followed by a motion estimate refinement stage (Step 5) discussed in Section 5.3.

5.2 Probabilistic RANSAC

In this section, details of the probabilistic framework proposed by Brink et al. [82] are presented. A standard hypothesise-and-verify approach is implemented where a probabilistic similarity measure of points in 3D coordinates is used to determine the consensus of hypothesised models. The model used in this RANSAC framework consists of the rotation and translation of 3D features between viewpoints. The rigid body transformation is determined by solving the absolute orientation problem, which only requires three point correspondences as opposed to the seven required by fundamental matrix RANSAC and the five required by essential matrix RANSAC. Fewer iterations are therefore needed and a faster execution time is achieved [22]. A solution to the absolute orientation problem is now discussed, followed by the derivation of a probabilistic similarity measure.

5.2.1 Absolute Orientation

As mentioned in Section 1.2.1, the motion of the camera can be estimated from 3D-to-3D point correspondences by solving the absolute orientation problem. The solutions proposed by Arun et al. [65] and Horn [66] are prone to return incorrect transformation parameters when datasets are highly corrupted [67] with noise. The least squares approach by Umeyama [67] was proposed as a refinement of the methods by Arun et al. [65] and Horn [66], and is therefore implemented instead.

Consider two sequences of n multi-dimensional points, $\mathcal{X} = \langle \mathbf{x}^{[i]} \rangle_{i=1}^{n}$ and $\mathcal{Y} = \langle \mathbf{y}^{[i]} \rangle_{i=1}^{n}$ such that $\mathbf{x}^{[i]}, \mathbf{y}^{[i]} \in \mathbb{R}^d$ and $n \geq d$. Furthermore, assume that the two sets of points are related by an arbitrary similarity transformation,

$$\mathbf{y}^{[i]} = s\mathbf{R}\mathbf{x}^{[i]} + \mathbf{t}, \tag{5.5}$$

where s is a scaling parameter, $\mathbf{R} \in SO(d)$ and $\mathbf{t} \in \mathbb{R}^{d\times 1}$. Following the notation of Umeyama [67], the goal is to minimise the mean squared error function,

$$e^2(\mathbf{R}, \mathbf{t}, s) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{y}^{[i]} - (s\mathbf{R}\mathbf{x}^{[i]} + \mathbf{t})\|^2. \tag{5.6}$$

The minimum value of Equation 5.6, $e^2_{\min}$ is shown to be

$$e^2_{\min} = \sigma^2_{\mathcal{Y}} - \frac{\mathrm{Tr}\left(\mathbf{DS}\right)^2}{\sigma^2_{\mathcal{X}}}, \tag{5.7}$$

where

$$\boldsymbol{\mu}_{\mathcal{X}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{[i]}, \tag{5.8}$$

$$\boldsymbol{\mu}_{\mathcal{Y}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}^{[i]}, \tag{5.9}$$

$$\sigma^2_{\mathcal{X}} = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}^{[i]} - \boldsymbol{\mu}_{\mathcal{X}}\|^2, \tag{5.10}$$

$$\sigma^2_{\mathcal{Y}} = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{y}^{[i]} - \boldsymbol{\mu}_{\mathcal{Y}}\|^2, \tag{5.11}$$

are the respective mean vectors and variances about the mean. The diagonal matrix, $\mathbf{D}$, is determined by first calculating the cross covariance,

$$\boldsymbol{\Sigma}_{\mathcal{XY}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}^{[i]} - \boldsymbol{\mu}_{\mathcal{Y}})(\mathbf{x}^{[i]} - \boldsymbol{\mu}_{\mathcal{X}})^\top \tag{5.12}$$

and then performing singular value decomposition [94, pp. 59–70] to obtain the matrices, $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{V}$ such that,

$$\boldsymbol{\Sigma}_{\mathcal{XY}} = \mathbf{UDV}^\top. \tag{5.13}$$

The choice of $\mathbf{S}$ is determined from,

$$\mathbf{S} = \begin{cases} \left.\begin{cases} \mathbf{I}_{n\times n}, & \text{if } |\mathbf{U}||\mathbf{V}| = 1, \\ \mathrm{diag}(1,1,\ldots,1,-1), & \text{if } |\mathbf{U}||\mathbf{V}| = -1, \end{cases}\right\} & \text{if } \mathrm{rank}(\boldsymbol{\Sigma}_{\mathcal{XY}}) > d-1, \\ \left.\begin{cases} \mathbf{I}_{n\times n}, & \text{if } |\boldsymbol{\Sigma}_{\mathcal{XY}}| \geq 0, \\ \mathrm{diag}(1,1,\ldots,1,-1), & \text{if } |\boldsymbol{\Sigma}_{\mathcal{XY}}| < 0, \end{cases}\right\} & \text{if } \mathrm{rank}(\boldsymbol{\Sigma}_{\mathcal{XY}}) = d-1. \end{cases} \tag{5.14}$$

The estimated transformation parameters are then determined as follows:

$$\hat{\mathbf{R}} = \mathbf{USV}^\top, \tag{5.15}$$

$$\hat{s} = \frac{1}{\sigma^2_{\mathcal{X}}}\mathrm{Tr}(\mathbf{DS}), \tag{5.16}$$

$$\hat{\mathbf{t}} = \boldsymbol{\mu}_{\mathcal{Y}} - \hat{s}\hat{\mathbf{R}}\boldsymbol{\mu}_{\mathcal{X}}. \tag{5.17}$$

So far it has been shown how to estimate the relative transformation parameters from corresponding 3D points. This will be used in the RANSAC framework to generate hypothesised models for robot motion from sampled points. The focus now shifts to a robust method of determining the support for hypothesised models.

5.2.2 Probabilistic Similarity Measure

For each hypothesised model, it is necessary to determine its support – that is, the number of points consistent with the model. Consistency is usually defined in terms of some error metric such as Euclidean distance, which does not consider the nature of the measurement noise. Brink et al. [82] proposed a probabilistic similarity measure for 3D points which allows for a more robust consensus measure.

Consider two measurements, $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$, taken of 3D landmarks, $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$, at consecutive time steps in the robot coordinate frame. Furthermore, assume that the uncertainty of each measurement is described by the covariance matrices, $\boldsymbol{\Sigma}_{k-1}$ and $\boldsymbol{\Sigma}_k$, respectively. Using a similar notation and following

the proof of Brink [51], assuming that $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$ are measurements of the same 3D landmark at different time steps, it is necessary to calculate the following joint probability,

$$P_c = p(\mathbf{z}_{k-1}, \mathbf{z}_k). \tag{5.18}$$

If $\mathbf{x}$ denotes the true 3D location of $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$ in world coordinates, and it is assumed that the prior, $p(\mathbf{x})$, is uniformly distributed over $\mathcal{V} = \mathbb{R}^3$, Equation 5.18 can be re-written as,

$$\begin{aligned} P_c &= \int_{\mathcal{V}} p(\mathbf{z}_{k-1}, \mathbf{z}_k, \mathbf{x}) \mathrm{d}\mathbf{x} \\ &= \int_{\mathcal{V}} p(\mathbf{z}_{k-1}, \mathbf{z}_k | \mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &\propto \int_{\mathcal{V}} p(\mathbf{z}_{k-1}, \mathbf{z}_k | \mathbf{x}) \mathrm{d}\mathbf{x}. \end{aligned} \tag{5.19}$$

Two further assumptions are now made; first, $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$ are assumed to be normally distributed as discussed in Section 3.2 and secondly, $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$ are assumed to be statistically independent since they are taken at different time steps. Consequently,

$$\begin{aligned} P_c &\propto \int_{\mathcal{V}} p(\mathbf{z}_{k-1}|\mathbf{x}) p(\mathbf{z}_k|\mathbf{x}) \mathrm{d}\mathbf{x} \\ &= \int_{\mathcal{V}} \frac{1}{(2\pi)^{\frac{3}{2}} |\mathbf{\Sigma}_{k-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{z}_{k-1})^\top \mathbf{\Sigma}_{k-1}^{-1} (\mathbf{x}-\mathbf{z}_{k-1})} \frac{1}{(2\pi)^{\frac{3}{2}} |\mathbf{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{z}_k)^\top \mathbf{\Sigma}_k^{-1} (\mathbf{x}-\mathbf{z}_k)} \mathrm{d}\mathbf{x} \\ &= \frac{|\mathbf{\Sigma}|^{\frac{1}{2}}}{(2\pi)^{\frac{3}{2}} |\mathbf{\Sigma}_{k-1}|^{\frac{1}{2}} |\mathbf{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}\lambda} \int_{\mathcal{V}} \frac{1}{(2\pi)^{\frac{3}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})} \mathrm{d}\mathbf{x}, \end{aligned} \tag{5.20}$$

where $\mathbf{\Sigma}$, $\boldsymbol{\mu}$ and λ are defined as follows,

$$\mathbf{\Sigma} = (\mathbf{\Sigma}_{k-1}^{-1} + \mathbf{\Sigma}_k^{-1})^{-1} \tag{5.21}$$

$$\boldsymbol{\mu} = \mathbf{\Sigma}^{-1} (\mathbf{\Sigma}_{k-1}^{-1} \mathbf{z}_{k-1} + \mathbf{\Sigma}_k^{-1} \mathbf{z}_k) \tag{5.22}$$

$$\lambda = (\mathbf{z}_{k-1} - \mathbf{z}_k)^\top (\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k)^{-1} (\mathbf{z}_{k-1} - \mathbf{z}_k). \tag{5.23}$$

As the size of $\mathcal{V}$ goes to infinity, the integral in Equation 5.20 tends to one. As a result,

$$\begin{aligned} P_c &\propto \frac{|\mathbf{\Sigma}|^{\frac{1}{2}}}{|\mathbf{\Sigma}_{k-1}|^{\frac{1}{2}} |\mathbf{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{z}_{k-1}-\mathbf{z}_k)^\top (\mathbf{\Sigma}_{k-1}+\mathbf{\Sigma}_k)^{-1} (\mathbf{z}_{k-1}-\mathbf{z}_k)} \\ &= \frac{1}{|\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{z}_{k-1}-\mathbf{z}_k)^\top (\mathbf{\Sigma}_{k-1}+\mathbf{\Sigma}_k)^{-1} (\mathbf{z}_{k-1}-\mathbf{z}_k)}, \end{aligned} \tag{5.24}$$

where constant terms have been ignored. The result in stated in Equation 5.24 is equivalent to the expression obtained and used by Brink et al. [82]; however, further matrix simplifications [116, p. 42] have been performed to obtain λ in the form of Equation 5.23. This form is advantageous as it is cheaper to calculate due to fewer matrix calculations. Furthermore, the negative log likelihood is used such that,

$$\begin{aligned} -\ln(P_c) &= \frac{1}{2}(\mathbf{z}_{k-1} - \mathbf{z}_k)^\top (\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k)^{-1} (\mathbf{z}_{k-1} - \mathbf{z}_k) + \frac{1}{2} \ln(|\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k|) \\ &\propto (\mathbf{z}_{k-1} - \mathbf{z}_k)^\top (\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k)^{-1} (\mathbf{z}_{k-1} - \mathbf{z}_k) + \ln(|\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k|). \end{aligned} \tag{5.25}$$

A similarity measure, D_C, is now defined as

$$D_C = (\mathbf{z}_{k-1} - \mathbf{z}_k)^\top (\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k)^{-1} (\mathbf{z}_{k-1} - \mathbf{z}_k) + \ln(|\mathbf{\Sigma}_{k-1} + \mathbf{\Sigma}_k|), \tag{5.26}$$

which is used to determine consensus in a probabilistic RANSAC framework where D_C can be compared to some threshold.

The similarity measure of Brink et al. [82] has been extended to that of a probabilistic distance[1]. This is a more intuitive way of formulating the problem, as a larger D_C represents 3D point distributions which

are "further" away from each other. Additionally, the negative log likelihood form has the advantage of being more numerically stable for small probabilities [117] and is better suited for minimisation. It should be noted, however, that the expression in Equation 5.26 can be negative and is therefore not a true metric.

5.2.3 Probabilistic RANSAC Algorithm

The absolute orientation solution, discussed in Section 5.2.1, and the probabilistic similarity measure, derived in Section 5.2.2, are now implemented in a standard RANSAC framework. An overview of the probabilistic RANSAC approach, proposed by Brink et al. [82], with an adapted similarity measure is given in Algorithm 5.1.

Algorithm 5.1 Probabilistic RANSAC

Input:

k_r	Number of RANSAC iterations
$\mathcal{X}_{k-1}$	Sequence of 3D points at time $k-1$: $\langle \mathbf{x}_{k-1}^{[i]} \rangle_{i=1}^{N}$
$\mathcal{E}_{k-1}$	Sequence of corresponding covariance matrices at time $k-1$: $\langle \mathbf{\Sigma}_{k-1}^{[i]} \rangle_{i=1}^{N}$
$\mathcal{X}_k$	Sequence of 3D points at time k: $\langle \mathbf{x}_k^{[i]} \rangle_{i=1}^{N}$
$\mathcal{E}_k$	Sequence of corresponding covariance matrices at time k: $\langle \mathbf{\Sigma}_k^{[i]} \rangle_{i=1}^{N}$
Δ_T	Threshold value for determining consensus

Output:

$\mathcal{I}$	Index set of inlier observations consistent with best model

function ProbabilisticRansac($k, \mathcal{X}_{k-1}, \mathcal{E}_{k-1}, \mathcal{X}_k, \mathcal{E}_k, \Delta_T$)
  $\mathcal{I} \leftarrow \emptyset$ ▷ Best inlier index set
  $\mathcal{J} \leftarrow \{1, 2 \dots n\}$ ▷ Full index set
  for k_r iterations **do**
    $\mathcal{C} \leftarrow \emptyset$ ▷ Inlier index set
    $\mathcal{S} \leftarrow$ RandomSample($\mathcal{J}, 3$) ▷ $\|\mathcal{S}\| = 3$
    $[\hat{\mathbf{R}}, \hat{\mathbf{t}}, \hat{s}] \leftarrow$ umeyama($\langle \mathbf{x}_{k-1}^{[i]} \rangle_{i \in \mathcal{S}}, \langle \mathbf{x}_k^{[i]} \rangle_{i \in \mathcal{S}}$) ▷ Equations 5.15 to 5.17
    if $\hat{s} \approx 1$ **then**
      for $i \in \{1, 2, \dots, N\}$ **do**
        $\tilde{\mathbf{x}}_k^{[i]} \leftarrow \hat{\mathbf{R}} \mathbf{x}_{k-1}^{[i]} + \hat{\mathbf{t}}$
        $\tilde{\mathbf{\Sigma}}_k^{[i]} \leftarrow \hat{\mathbf{R}} \mathbf{\Sigma}_k^{[i]} \hat{\mathbf{R}}^\top$
        $D_C \leftarrow$ Similarity($\tilde{\mathbf{x}}_k^{[i]}, \tilde{\mathbf{\Sigma}}_k^{[i]}, \mathbf{x}_k^{[i]}, \mathbf{\Sigma}_k^{[i]}$) ▷ Equation 5.26
        if $D_C < \Delta_T$ **then**
          $\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$
        end if
      end for
      if $|\mathcal{C}| > |\mathcal{I}|$ **then**
        $\mathcal{I} \leftarrow \mathcal{C}$
      end if
    end if
  end for
  return $\mathcal{I}$
end function

The stereo features identified at the previous and the current time-step, $\mathcal{F}_{k-1}$ and $\mathcal{F}_k$, are triangulated to obtain sequences of 3D features, $\mathcal{X}_{k-1}$ and $\mathcal{X}_k$. Furthermore, corresponding sequences of covariance matrices, $\mathcal{E}_{k-1}$ and $\mathcal{E}_k$, describing the uncertainty of each 3D measurement, are determined using the

[1] An interesting result seen here is the resemblance of Equation 5.26 to the Bhattacharyya distance [118] which measures the similarity of two probability distributions.

techniques discussed in Section 3.2. A hypothesise-and-verify process is repeated for k_r iterations where three, 3D feature matches are sampled from $\mathcal{X}_{k-1}$ and $\mathcal{X}_k$ at each iteration (Lines 4–6). Transformation parameters are estimated from the sampled feature matches using the method of Umeyama [67] (Line 7). Scale is not expected to change between time steps, therefore models are rejected immediately if the estimated scale factor, $\hat{s}$, is not approximately equal to one (Line 8). During the verification stage, each 3D point, $\mathbf{x}_{k-1}^{[i]}$ and its corresponding uncertainty, $\mathbf{\Sigma}_{k-1}^{[i]}$, is transformed to the same coordinate frame as the points in $\mathcal{X}_k$ by the estimated parameters, $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$, resulting in $\tilde{\mathbf{x}}_k^{[i]}$ and $\tilde{\mathbf{\Sigma}}_k^{[i]}$ (Lines 10–11). Equation 5.26 is then used to compare the similarity of $\tilde{\mathbf{x}}_k^{[i]}$ and $\mathbf{x}_k^{[i]}$ (Lines 12). Matched features with a similarity, D_C, less than a threshold, Δ_T, are identified as inliers and their indices are added to an index set, $\mathcal{C}$ (Lines 13–14). After each iteration, the size of the current inlier set is compared to the largest consensus set obtained so far, and the larger of the two is maintained (Lines 17–19). The largest inlier index set, $\mathcal{I}$, is returned after k_r iterations have been performed (Line 22).

5.3 Motion Estimation Refinement

The probabilistic RANSAC framework in Algorithm 5.1 returns an index set, $\mathcal{I}$, consistent with the largest consensus set found. Instead of simply using the model parameters obtained from Equations 5.15 to 5.17, a motion estimation refinement step is performed on the complete set of identified inliers. Suppose an inlier index set is returned at time k. A maximum likelihood motion estimate is obtained by minimising the function,

$$\sum_{i \in \mathcal{I}} \| \mathbf{z}_k^{[i]} - \pi(\mathbf{x}_{r,k-1}^{[i]}, \mathbf{K}; \mathbf{R}, \mathbf{t}) \|^2, \tag{5.27}$$

with respect to $\mathbf{R}$ and $\mathbf{t}$, where $\pi(\mathbf{x}_{r,k-1}^{[i]}, \mathbf{K}; \mathbf{R}, \mathbf{t}) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ represents the projection of a 3D point, $\mathbf{x}_{r,k-1}^{[i]}$, to the image plane using the camera projection matrix described in Equation 2.10. Minimising Equation 5.27 is a non-linear minimisation problem and is solved using the Levenberg-Marquardt algorithm [83]. Now that all of the components of a robust visual odometry pipeline have been presented, the focus now shifts to the generation of synthetic datasets, which will be used for evaluation purposes.

5.4 Synthetic Visual Odometry Datasets

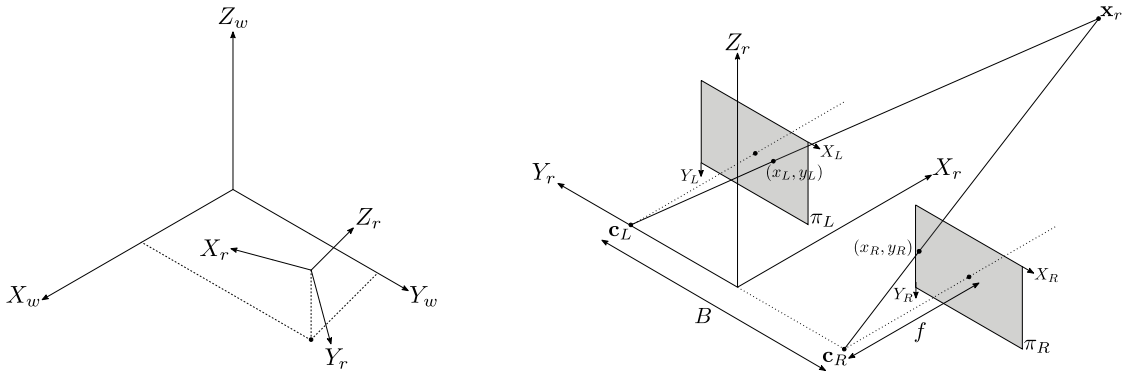
A motion model was implemented to generate random robot trajectories for use in simulated experiments. Instead of deriving a specific model for a wheeled robot driving along a flat surface, a generic motion model, incorporating motion with six degrees of freedom (6DoF), was preferred. As a result, the generated trajectories are not only applicable to wheeled robots, but also to aerial, underwater and humanoid robots.

A "constant velocity, constant angular velocity" motion model, proposed by Davison et al. [25], was implemented[2]. This statistical model assumes that at any given time step, unknown accelerations act on the robot that are consistent with a zero-mean normal distribution. Consequently, large accelerations are unlikely to occur and camera motion is constrained to smooth trajectories. Suppose the state vector, $\mathbf{x}_v$, of the robot is given by

$$\mathbf{x}_v = \begin{bmatrix} \mathbf{r}_w \\ \mathbf{q}_w \\ \mathbf{v}_w \\ \boldsymbol{\omega}_w \end{bmatrix}, \tag{5.28}$$

where $\mathbf{r}_w$ is the position vector of the robot in world coordinates, $\mathbf{q}_w$ is a quaternion describing the orientation of the robot, and the linear and angular velocity vectors are denoted by $\mathbf{v}_w$ and $\boldsymbol{\omega}_w$ respectively. The relationship between the world and robot coordinate frame is shown in Figure 5.1a. During each

[2]It should be noted that the motion model is only used to generate synthetic visual odometry datasets and that the motion estimation pipeline is not dependent on the motion model.



(a) **(b)**

Figure 5.1: Simulated robot coordinate frames. (a) Illustration of robot state vector in world coordinate frame. Thirteen robot states are described by the robot's position vector and orientation quaternion, $\mathbf{r}_w$ and $\mathbf{q}_w$, as well as the linear angular velocities, $\mathbf{v}_w$ and $\boldsymbol{\omega}_w$. (b) A rectified stereo camera pair centered on the robot frame.

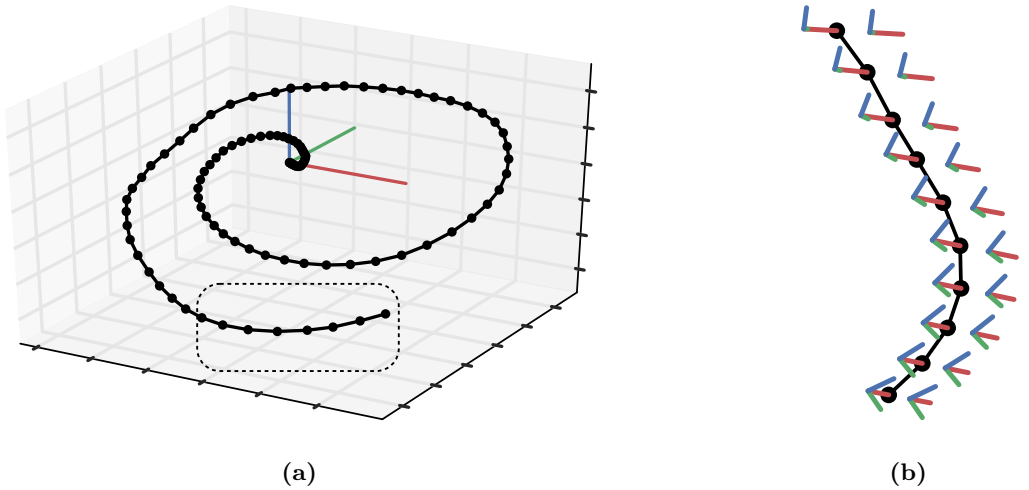
time step, Δt, an impulse of linear and angular velocity is generated from random accelerations such that

$$\mathbf{u} = \begin{bmatrix} \Delta \mathbf{v}_w \\ \Delta \boldsymbol{\omega}_w \end{bmatrix} = \begin{bmatrix} \mathbf{a}_w \Delta t \\ \boldsymbol{\alpha}_w \Delta t \end{bmatrix}, \tag{5.29}$$

where $\mathbf{a}_w \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_a)$, $\boldsymbol{\alpha}_w \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\alpha)$ and $\mathbf{u}$ represents the control input of the robot. It is assumed that the forces responsible for $\mathbf{a}_w$ and $\boldsymbol{\alpha}_w$ are not coupled in any manner, hence, the covariances $\boldsymbol{\Sigma}_a$ and $\boldsymbol{\Sigma}_\alpha$ are diagonal matrices with uncorrelated acceleration components,

$$\boldsymbol{\Sigma}_a = \begin{bmatrix} \sigma_{a_x}^2 & 0 & 0 \\ 0 & \sigma_{a_y}^2 & 0 \\ 0 & 0 & \sigma_{a_z}^2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_\alpha = \begin{bmatrix} \sigma_{\alpha_x}^2 & 0 & 0 \\ 0 & \sigma_{\alpha_y}^2 & 0 \\ 0 & 0 & \sigma_{\alpha_z}^2 \end{bmatrix}. \tag{5.30}$$

The sigma values in Equation 5.30 must be determined empirically in order to approximate the dynamics of a specific robot. An example trajectory generated is shown in Figure 5.2a.



(a) **(b)**

Figure 5.2: An example robot trajectory generated using an acceleration based model. (a) 3D view of the trajectory. (b) A section of the trajectory which has been magnified to illustrate the poses of a stereo pair mounted to the robot. The colours of the axes, $R - G - B$, correspond to the $X_c - Y_c - Z_c$ of each camera.

In addition to the random robot trajectories, 3D landmarks are also randomly generated in the world coordinate frame. These 3D landmarks are then projected to image planes, π_L and π_R, with the stereo

system of Figure 5.1b and parameters of Table 3.1, for each pose along the robot trajectory. The synthetic image features are contaminated with additive measurement noise, sampled from a zero-mean normal distribution with a diagonal covariance,

$$\mathbf{\Sigma}_{\mathbf{x}} = \sigma^2 \mathbf{I}_{4\times 4}, \tag{5.31}$$

to generate stereo measurements for each robot pose. Outliers are then injected so that each pose's measurement set is consistent with an inlier ratio,

$$\epsilon = \frac{\text{number of inliers}}{\text{number of measurements}}. \tag{5.32}$$

Outliers are generated using the error model proposed by Torr and Zisserman [108] (Equation 4.2) – that is, the x-coordinate[3] of randomly sampled features are replaced by a coordinate sampled from the uniform distribution,

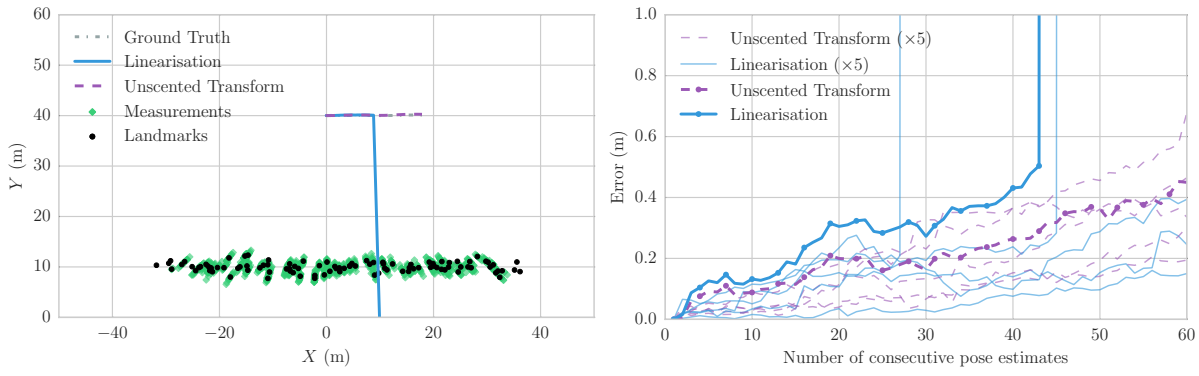
$$p(x_{\text{outlier}}) = \begin{cases} \frac{1}{v} & \text{for } 0 \leq x_{\text{outlier}} \leq v \\ 0 & \text{otherwise,} \end{cases} \tag{5.33}$$

where v is the width of the image plane. The parameters, ϵ and σ, allow synthetic visual odometry datasets to be generated for various inlier ratios and different noise levels. An experiment will now be performed to evaluate the robust visual odometry framework developed in this chapter using one of these synthetic datasets.

5.5 Experimental Results

An experiment was performed that evaluates the performance of probabilistic RANSAC when only distant features, which have high uncertainty, are available. The motivation for this experiment is two-fold. First, it verifies that probabilistic RANSAC is viable in situations where uncertain measurements make identifying outliers difficult and secondly, it illustrates the difference between linearisation and unscented transform when used in a probabilistic visual odometry framework. Emphasis was placed on the effect of using different approximation techniques for noise propagation (Chapter 3) in conjunction with the probabilistic RANSAC framework.

A random robot trajectory, consisting of 60 robot poses, was generated using the motion model described in Section 5.4. This trajectory is shown as ground truth in Figure 5.3a. A total of 500, 3D



(a) (b)

Figure 5.3: Robust motion estimation using probabilistic RANSAC. (a) A simulated vehicle trajectory with distant 3D landmarks as well as their corresponding measurements shown alongside estimated trajectories. An inaccurate estimate of a measurement distribution by linearisation results in outliers not being removed correctly and a poor trajectory reconstruction. (b) Euclidean distance error of estimated trajectories as a function of the number of pose estimates. Results from six simulations are shown (error plots with circle markers correspond to the simulation of (a).

[3] Only x-coordinates are replaced so that outliers still obey the epipolar constraints of a rectified, stereo pair (Section 2.2.2).

landmarks were randomly generated in a planar configuration representing distant 3D features (black dots). These features were projected to 2D stereo image coordinates over 60 viewpoints across the ground truth trajectory and contaminated by additive measurement noise with $\sigma = 1.0\,\text{px}$. Outliers were injected so that each viewpoint's set of feature matches had an inlier ratio of $\epsilon = 0.8$.

The robust visual odometry pipeline described in Section 5.1 was then used to reconstruct the 3D features (green diamonds) and perform motion estimation with the result from a single run shown in Figure 5.3a. Two visual odometry pipelines were executed in parallel, where the mean and covariances used by probabilistic RANSAC were determined using linearisation in one implementation (blue solid line) and the unscented transform in the other (purple dashed line). Due to the non-deterministic sampling of RANSAC, care was taken to ensure that both implementations made use of the same samples for a fair comparison.

As seen in the results of Figure 5.3a, the effect of a poorly approximated measurement distribution is clear. During the course of the trajectory reconstruction, inaccurate measurement distribution approximations using linearisation cause the probabilistic RANSAC method to misidentify outliers as inliers, which results in a poor trajectory estimate. The Euclidean distance error of the estimated trajectories for the above simulation and five additional simulations for the same trajectory are shown in Fig. 5.3b. The errors introduced by linearisation are clearly shown by the three incorrect trajectory reconstructions seen. In contrast, the visual odometry framework using the unscented transform obtained trajectories that do not contain severe errors for all six simulations. This is consistent with the conclusions of Section 3.5 that showed the unscented transform results in more accurate approximations of 3D feature distributions in conditions of high uncertainty.

5.6 Chapter Summary

In this chapter, a probabilistic RANSAC-based outlier removal framework was implemented in conjunction with a stereo visual odometry pipeline for robust motion estimation. This framework forms the benchmark of comparison for the novel outlier removal technique proposed in the next chapter. A technique for solving the absolute orientation problem was discussed and the derivation of a probabilistic similarity measure, proposed by Brink et al. [82], was extended to that of a probabilistic distance. Thereafter, motion estimation refinement was briefly discussed.

A "constant velocity, constant angular velocity" motion model was used to generate random smooth robot trajectories. The robot poses, which make up the trajectory, in conjunction with a simulated stereo pair allow synthetic visual odometry datasets to be constructed with adjustable parameters such as measurement noise and contamination levels.

A synthetic visual odometry dataset was then used to evaluate the performance of the probabilistic RANSAC framework in the presence of highly uncertain measurements. Two implementations of a robust visual odometry pipeline were compared – one making use of linearisation and one making use of the unscented transform. Visual odometry using the unscented transform was shown to significantly outperform visual odometry using linearisation when used for approximating 3D feature distributions under highly uncertain conditions. This is consistent with the results of Section 3.5. The approximation errors introduced by linearisation result in inaccurate approximations of the processed measurement distributions. These inaccuracies cause outliers to be incorrectly identified as inliers, and as a result, poorly reconstructed trajectories are obtained. The visual odometry pipeline implementation with the unscented transform, on the other hand, was able to remove outliers correctly and obtain accurate motion estimates. The unscented transform will therefore be exclusively used for the non-linear propagation of sensor uncertainty throughout the remaining chapters. The focus now shifts to the main contribution of this thesis, namely, a novel outlier removal technique.

Chapter 6

Proposed Outlier Removal Method

In Chapter 4, it was stated that RANSAC-based approaches are by far the most popular outlier removal technique currently implemented in vision-based autonomous navigation applications. However, there are several limitations of the standard RANSAC algorithm that restrict the use of RANSAC when measurements are contaminated with a large number of outliers and high levels of measurement noise. The focus of this chapter is the development of a novel outlier removal technique, which does not have these limitations, as an alternative to RANSAC. This chapter forms the main contribution of this thesis.

Several extensions, discussed in detail in Section 4.2, have been proposed to improve the RANSAC algorithm in terms of efficiency and robustness. One such extension, an efficient and probabilistic RANSAC framework proposed by Brink et al. [82], was implemented in the previous chapter as part of a robust visual odometry pipeline. This pipeline will serve as benchmark against which the proposed outlier removal method will be compared in Chapter 7.

This chapter begins with a conceptual motivation for the proposed outlier removal technique. The well-known problem of fitting a line to a set of 2D points, contaminated with outliers, is used to illustrate RANSAC in parallel with the mechanism of the proposed technique. Thereafter, its applicability to robust visual odometry is briefly discussed and a general formulation of the proposed method is developed theoretically. An adaptive sampling strategy, based on concepts from decision theory, is then presented. The discussion then shifts to outlier removal for visual odometry, where shape parameters and a shape similarity measure are derived specific to the three dimensional problem. The chapter ends with a theoretical comparison of RANSAC and the proposed method in terms of time complexity, as well as experimental results that verify the mechanism of the proposed outlier removal method.

6.1 Shape Based Outlier Removal

A common illustrative example of RANSAC is the fitting of a line to a set of 2D observations contaminated with outliers. This is one of the worked examples provided in the original RANSAC paper by Fischler and Bolles [68]. This simple problem is used as an illustrative example in this section, where the mechanisms of RANSAC and the proposed outlier removal technique are discussed alongside each other, to motivate the use of shape information for outlier removal.

An example dataset of 2D observations is shown in Figure 6.1. It is assumed that the dataset consists of both inliers (shaded dots), which are observations that fit some underlying line model (solid line), and outliers (unshaded dots) which are inconsistent with the line model. An erroneous line model (dashed line), determined from a least squares estimation performed on all observations, demonstrates how outliers corrupt model estimates.

The RANSAC approach, described in Section 4.1, is easily applied to remove outliers in the example of Figure 6.1. Line model parameters are fully described by a pair of points, therefore at least two observations must be randomly sampled in each iteration. A line is fit to the sampled observations and the consensus of the hypothesised model is determined. For the sake of simplicity, a distance threshold is

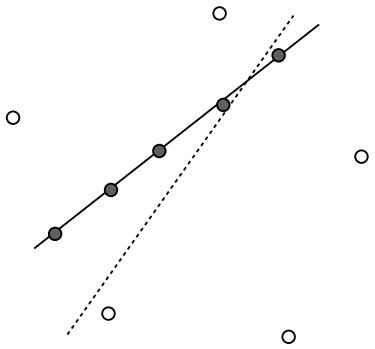


Figure 6.1: Fitting a line to a contaminated dataset. A set of 2D observations, with an underlying line model (solid), consisting of inlier (shaded) and outlier (unshaded) points. A least squares estimation returns an erroneous line model (dashed) in the presence of outliers.

used in this example to determine the consistency of observations with each model. Three iterations of the RANSAC algorithm are shown in Figure 6.2 with sampled observations indicated by dashed circles. Sampled sets which contain at least one outlier result in erroneous model estimates, as seen in Figures 6.2a

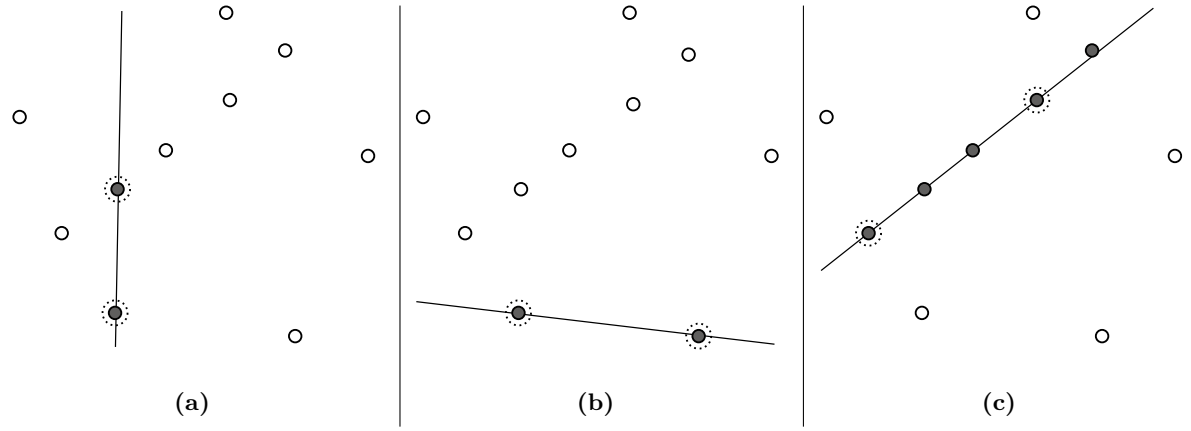


Figure 6.2: Three iterations of RANSAC applied to outlier removal in a 2D line fitting application. Randomly sampled points are indicated by dashed circles. During the verification stage, observations within some error metric of the hypothesised line are identified as inliers (shaded dots). Accurate model estimates will have large support (c), while erroneous model estimates have poor support, (a) and (b).

and 6.2b, and suffer from poor support. However, when an outlier-free set is sampled, as in Figure 6.2c, a model with large support is generated. The observations in the consensus set – that is, the set of identified inliers – which correspond to this well-supported model can then be used to re-estimate an accurate line model.

Moving away from the mechanisms of RANSAC, a novel approach to outlier removal that uses implicit shape information is proposed and motivated. Inspecting Figure 6.1, it should be clear that any non-minimal[1] set of inliers should carry similar shape information – that is, inlier points represent a line. Knowledge of this latent shape information can be exploited to distinguish between inliers and outliers. The proposed technique employs a random sampling approach analogous to that of RANSAC. However, instead of fitting a model to sampled points and determining the support of the model across a number of iterations, sets of points are randomly sampled and their shape is used to determine whether the set is contaminated or not. Sampled points forming contaminated sets are more likely to be outliers. During each iteration of the algorithm, three observations are randomly sampled and a measure of their shape similarity to a line is determined. This similarity measure is used to update the probability of the sampled

[1] In the case of 2D line fitting, a non-minimal set would consist of at least three observations.

points being inliers. Three iterations of the proposed technique are shown in Figure 6.3 with sampled points indicated by dashed circles.

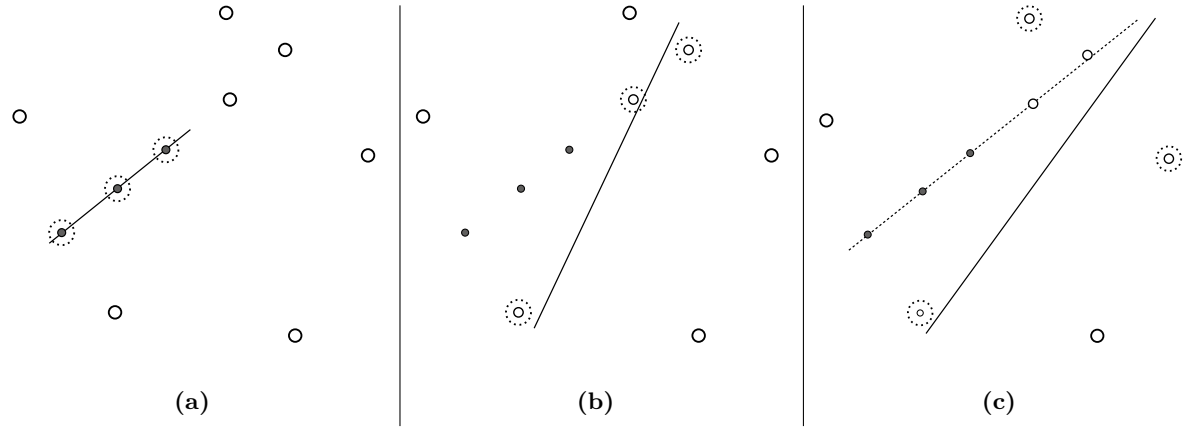


Figure 6.3: The mechanism of the proposed outlier removal technique for the 2D line fitting application. Randomly sampled points are indicated by dashed circles. Initially, all points are assumed to be outliers (empty dots) with high uncertainty (represented by the size of each dot). In subsequent iterations, sets of sampled points that are consistent with a line are more likely to be inliers (a), while those inconsistent with a line are more likely to be outliers, (b) and (c). A model determined from the most likely inliers is shown as a dashed line in (c).

Initially, all points are assumed to be outliers with high uncertainty. This is important as an outlier misidentified as an inlier will lead to erroneous results, where as an inlier misidentified as an outlier is not critical. In Figure 6.3a, three inliers are randomly sampled from the dataset. As should be expected, the inherent shape of inlier points correspond well with a line. These observations are therefore labelled as inliers (shaded dots) with a high probability (low uncertainty is indicated by smaller dots). Likewise, when sampled points do not correspond with a line, as is the case in Figures 6.3b and 6.3c, it can be concluded that the set is likely to contain at least one outlier. Consequently, the individual probabilities of these points being outliers have increased. As opposed to RANSAC, where iterations are performed independently of each other, probabilistic information obtained in one iteration is exploited in subsequent iterations. This makes it possible to become very certain about the states of points which have been sampled more than once, as shown in Figure 6.3c. After several iterations have been performed, a consensus set is constructed from the most likely inliers, and a line model can then be fit to these points as indicated by the dashed line in Figure 6.3c.

One key difference between RANSAC and the proposed technique, which needs to be highlighted, lies in the way observations are sampled during each iteration. As already mentioned in Chapter 4, standard RANSAC assumes that all observations have the same prior probability of being an inlier; therefore, samples are drawn from the observation set uniformly. Conversely, the proposed technique maintains the probabilities of points being inliers or outliers – this means that the sampling strategy can be modified to take advantage of points that have a high probability of being an inlier. Different sampling strategies for the 2D line fitting problem are given in Figure 6.4 to illustrate this concept.

The three examples in Figure 6.4 assume that an inlier set has already been identified as indicated by the shaded dots. Subsequent iterations can then exploit this information when choosing samples in much the same way as the extensions to RANSAC described in Section 4.2.4. In Figure 6.4a, a single inlier is sampled with two uncertain observations. As it is already known that one of the points is an inlier with a high degree of certainty, if the three points do not lie on a line, the remaining two points are more likely to be outliers. Similarly, in Figure 6.4b, two inlier observations and a single uncertain point are sampled; hence, it can be said with great certainty that the single point is an outlier if the three observations do not lie on a line. Figure 6.4c illustrates the case where three uncertain points are sampled. With this

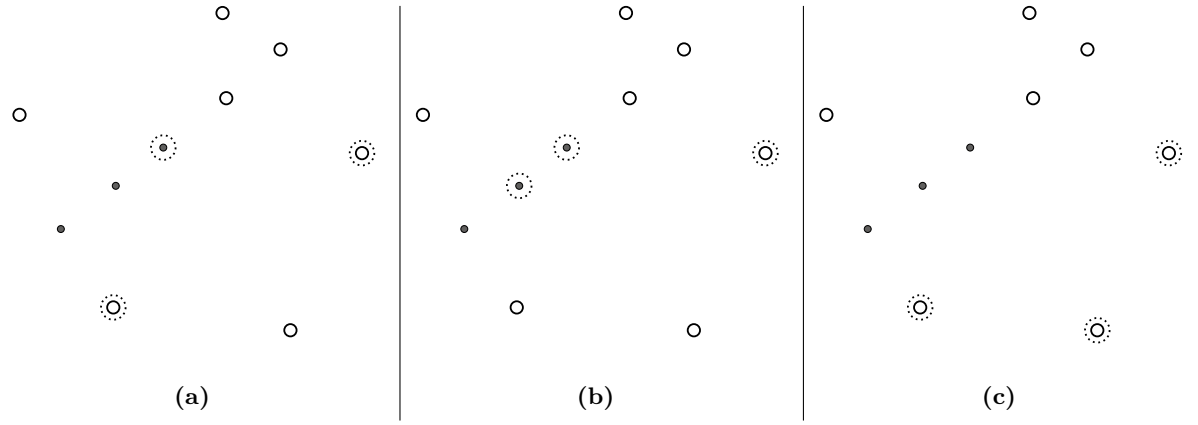


Figure 6.4: Different sampling strategies available after a set of inlier points has been identified. Sampled points are indicated by dashed circles. (a) A single inlier and two uncertain points are sampled. (b) Two inlier points and a single uncertain point are sampled. (c) Three uncertain points are sampled.

sampling strategy, no prior information is exploited and the probabilities of the points are updated in the usual manner.

The proposed technique warrants further investigation as it should be significantly cheaper than RANSAC due to the fact that a consensus set is not determined at each iteration. For high levels of contamination, a large number of erroneous models are verified by RANSAC to determine consensus sets and a significant portion of computation time can therefore be saved by using the proposed technique.

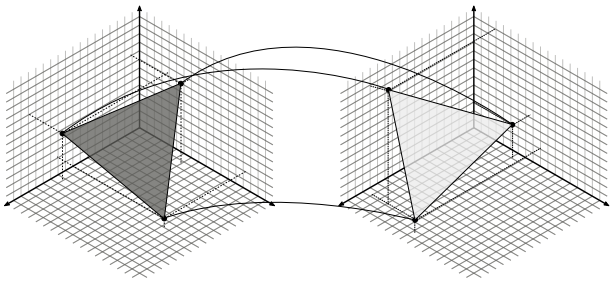
The concepts described in this example form the basis of the novel outlier removal method developed in this thesis. It is shown that latent shape information can be used to distinguish between inliers and outliers in a probabilistic manner. For this reason and maintaining a similar naming convention to RANSAC, the proposed technique will be referred to as PORUS (Probabilistic Outlier Removal Using Shapes) throughout the rest of this thesis. It will now be shown how PORUS can be used for outlier removal in the context of visual odometry.

6.2 PORUS for Visual Odometry

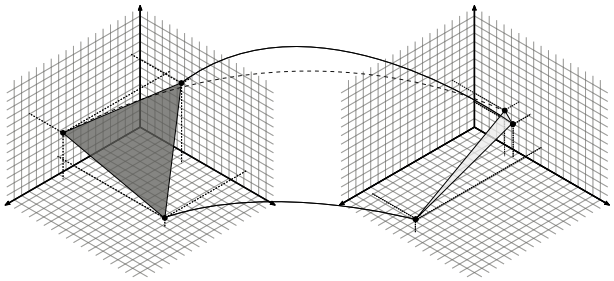
PORUS was explained conceptually in the previous section using the example of fitting a line in the presence of outliers. However, the main focus of this thesis is on robust motion estimation, and therefore it is important to introduce PORUS in the context of visual odometry.

As previously discussed, visual odometry is concerned with the reconstruction of a robot's trajectory by the accumulation of rigid body transformations determined from successive viewpoints. A rigid body transformation, also known as a Euclidean displacement, has the specific property that distances between pairs of points are invariant during transformations [119, p. 1]. As a result, this constraint dictates that an object undergoing a rigid body transformation will not change in shape or size, and more importantly, the shape in $\mathbb{R}^3$ defined by a set of points should be consistent across successive viewpoints. These are the same constraints used by Hirschmuller et al. [81] and Howard [6] to determine mutually consistent sets of inliers as discussed in Section 4.3. Figure 6.5 illustrates the concept of shape consistency between viewpoints. In Figure 6.5a, a triple of 3D points in the first viewpoint (left) forms a triangle. A rigid body transformation is applied to the set of points with the result shown in the second viewpoint (right); importantly, the triangle formed by the transformed triple of points remains unchanged[2]. In Figure 6.5b, the same transformation is applied; however, an outlier is introduced in the second viewpoint as indicated by the dashed line. Quite clearly, a change in shape between the two viewpoints is observed when the observations are contaminated by outliers. Significantly, this change in shape suggests that PORUS may

[2]For the purpose of this example it is assumed that the rigid body transformation is perfect and no noise is present in the system.



(a)



(b)

Figure 6.5: Illustration of PORUS for visual odometry. (a) A triple of three dimensional points, without outliers, undergoing a rigid body transformation. The triangle formed is invariant across viewpoints. (b) The same triple of points undergoing a rigid body transformation where an outlier is introduced in the second viewpoint (dashed line). A clear change in shape is observed.

be used to distinguish between sets of points that contain outliers and those consisting of inlier points only. PORUS will now be developed formally for a general m-dimensional, relative transformation problem before it is used for the specific application of visual odometry.

6.3 Identifying Inliers Using Shape Measurements

A brief overview of the mechanism of PORUS was presented in Section 6.1 and the concept of shape similarity was introduced in Section 6.2. In this section, it will be shown how inliers can be distinguished from outliers using shape measurements from different viewpoints, in a probabilistic manner.

Consider the relative transformation problem illustrated in Figure 6.5, but now imagine a scenario where a sequence of m points in $\mathbb{R}^m$, undergoes an arbitrary rigid body transformation between two viewpoints at consecutive time steps. The arrangement of the points, at each time step, is dependent on a latent shape determined by the structure of the environment. A measured shape can then be obtained from the points in each viewpoint. The shape measurements are dependent on both the latent shape as well as whether the sequence of point matches in each viewpoint contains outliers. A measure of shape similarity can then be determined from the shape measurements of each viewpoint.

Explicitly representing the joint distribution of the above model for all m corresponding points is often intractable. However, there are certain assumed independence properties that can be used to create a simplified representation of the distribution. A proposed Bayesian network[3] for the relative transformation problem is given in Figure 6.6. A random variable is represented by a capital letter, a boldface small letter indicates a vector of random variables, and a particular state of a random variable is given by its corresponding lowercase letter.

[3] A Bayesian network is a structure that describes a joint distribution compactly in a factorised form [120, Chapter 3] by means of a directed acyclic graph (DAG), where each node in the graph represents a random variable and the edges indicate the direct influence of nodes on one another.

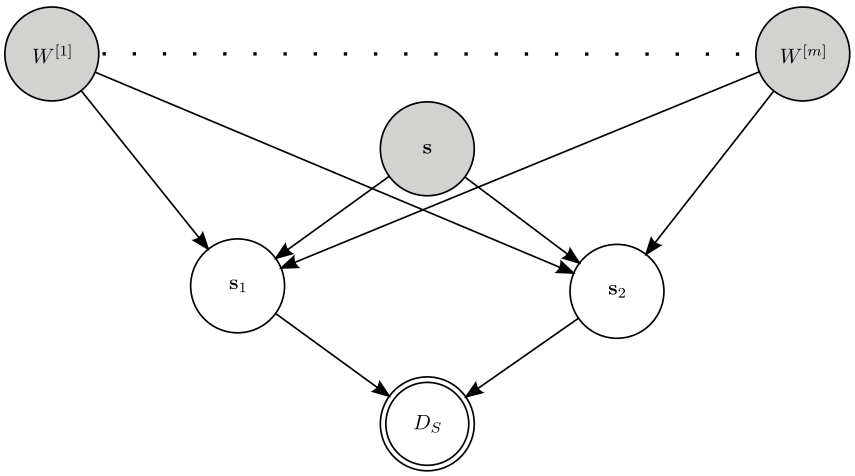


Figure 6.6: Bayesian network of the relative transformation problem. The shaded nodes indicate that $\mathbf{s}$ and $\mathbf{w}$ are latent random variables and the double-line notation represents the fact that D_S is a deterministic function of the random variables, $\mathbf{s}_1$ and $\mathbf{s}_2$.

A latent shape generating points in each viewpoint is described by the random vector, $\mathbf{s}$. Binary random variables, $W^{[i]} \in \{w_O, w_I\}$, describe the probability of the i^{th} matched point pair between viewpoints being an inlier ($W^{[i]} = w_I$) or an outlier ($W^{[i]} = w_O$), where the prior distribution over $w^{[i]}$ is given by,

$$p(w^{[i]}) = \begin{cases} 1-\epsilon & \text{for } w^{[i]} = w_O \\ \epsilon & \text{for } w^{[i]} = w_I, \end{cases} \tag{6.1}$$

and ϵ is the inlier ratio. The random vector, $\mathbf{w}$, contains all the inlier probabilities, $W^{[i]}$. The shape parameters, $\mathbf{s}_1$ and $\mathbf{s}_2$ (detailed in Section 6.5), are determined from the points in each viewpoint, and D_S is a measure of shape similarity, calculated as a deterministic function of $\mathbf{s}_1$ and $\mathbf{s}_2$. It is assumed that D_S is in the form of probabilistic distance – that is, small values indicate high shape similarity and large values represent poor shape similarity. The structure of Figure 6.6 allows the joint distribution to be expressed as product of factors such that,

$$p(d_S, \mathbf{s}, \mathbf{w}, \mathbf{s}_1, \mathbf{s}_2) = p(d_S|\mathbf{s}_1, \mathbf{s}_2)p(\mathbf{s}_1|\mathbf{w}, \mathbf{s})p(\mathbf{s}_2|\mathbf{w}, \mathbf{s})p(\mathbf{s})p(\mathbf{w}). \tag{6.2}$$

The goal of PORUS is to determine the inlier probabilities, $\mathbf{w}$, from measured shapes in two viewpoints. The joint distribution in Equation 6.2 describes this relationship; however, it remains complex and can be simplified further. First, the latent shape, $\mathbf{s}$, is not inferred from the shape measurements to reduce computational requirements. Instead only the inlier probabilities are inferred. Marginalising over the latent shape therefore results in

$$\begin{aligned} p(d_S, \mathbf{w}, \mathbf{s}_1, \mathbf{s}_2) &= \int_{\mathbf{s}} p(d_S, \mathbf{s}, \mathbf{w}, \mathbf{s}_1, \mathbf{s}_2)\mathrm{d}\mathbf{s} \\ &= p(d_S|\mathbf{s}_1, \mathbf{s}_2)p(\mathbf{w}) \int_{\mathbf{s}} p(\mathbf{s}_1|\mathbf{w}, \mathbf{s})p(\mathbf{s}_2|\mathbf{w}, \mathbf{s})p(\mathbf{s})\mathrm{d}\mathbf{s} \\ &= p(d_S|\mathbf{s}_1, \mathbf{s}_2)p(\mathbf{w})p(\mathbf{s}_1, \mathbf{s}_2|\mathbf{w}), \end{aligned} \tag{6.3}$$

Here it is clearly shown that the shape measurements in each viewpoint are dependent on the inlier probabilities of each point. A further simplification can be made if it is assumed that the measurements are only used to calculate the shape similarity, and are then discarded. The random variable, $D_S = \mathbf{f}_{\text{shape}}(\mathbf{s}_1, \mathbf{s}_2)$, then encapsulates $\mathbf{s}_1$ and $\mathbf{s}_2$ such that,

$$\begin{aligned} p(d_S, \mathbf{w}) &= p(d_S|\mathbf{w})p(\mathbf{w}) \\ &= p(d_S|w^{[1]}, \ldots, w^{[m]})p(w^{[1]})p(w^{[2]}) \ldots p(w^{[m]}), \end{aligned} \tag{6.4}$$

where $\mathbf{f}_{\text{shape}}$ is an abritary shape similarity function, and the inlier probabilities of individual matches are assumed to be statistically independent. A simplified Bayesian network is now represented by Figure 6.7, where the shape similarity is dependent on the inlier probabilities of the matches. A consequence of this

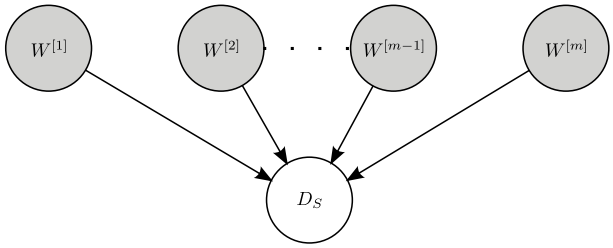


Figure 6.7: Graphical model illustrating the dependency of shape similarity on the presence of outliers.

simplification is that information about $\mathbf{s}_1$ and $\mathbf{s}_2$ is lost and can no longer be inferred from D_S.

In order to infer the inlier probabilities from D_S, it is necessary to determine the likelihood,

$$p(d_S|\mathbf{w}) = \frac{p(d_S, \mathbf{w})}{p(\mathbf{w})}. \tag{6.5}$$

It is then possible to obtain the posterior distribution, $p(\mathbf{w}|d_S)$ by means of Bayes' rule. However, defining $p(d_S|\mathbf{w})$ is not tractable for practical scenarios as it is problem and parameter specific. Instead of working with $p(d_S|\mathbf{w})$ directly, the concepts of inlier/outlier sets and a shape similarity test are introduced to circumvent this issue. An event, I, is defined as a set of matches that consists of inliers only (inlier set), while an event, O, consists of the remaining outcomes where a set of feature matches contains at least one outlier (contaminated sets). A formal definition of I and O is given such that,

$$\begin{aligned} I &= \{(w_I^{[1]}, w_I^{[2]} \ldots, w_I^{[m]})\} \\ O &= \{(w_O^{[1]}, w_I^{[2]} \ldots, w_I^{[m]})\} \cup \{(w_I^{[1]}, w_O^{[2]} \ldots, w_I^{[m]})\} \cup \cdots \cup \{(w_O^{[1]}, w_O^{[2]} \ldots, w_O^{[m]})\}, \end{aligned} \tag{6.6}$$

where $w_I^{[i]}$ is the outcome that the i^{th} match is an inlier and $w_O^{[i]}$ is the outcome that the i^{th} match is an outlier. Furthermore, a shape similarity test is introduced where D_S is compared to a threshold, Δ_S. The test is passed if $D_S < \Delta_S$ and failed otherwise, where a passed test indicates high shape similarity. Intuitively, matches that pass the shape similarity test should have a high probability of being inliers. These two concepts allow intermediary variables, α and β, to be defined that describe the probabilities of an inlier set passing a shape similarity test, as well as a contaminated set passing a shape similarity test respectively,

$$\begin{aligned} \alpha &= P(D_S < \Delta_S|I) \\ \beta &= P(D_S < \Delta_S|O). \end{aligned} \tag{6.7}$$

If $p(d_S|\mathbf{w})$ is defined, it is possible to determine the quantities in Section 6.7; alternatively, they can be determined experimentally. It is therefore assumed that the quantities, α and β, are available for the remainder of the development of PORUS. It should be noted, however, that there are consequences to not using the likelihood, $p(d_S|\mathbf{w})$, directly. By dividing the outcomes into I and O, the link to specific inlier probabilities, $w_O^{[i]}$, is lost. It is now only possible to infer whether a set of matches is contaminated or not, rather than whether specific matches are inliers or outliers.

Assuming the quantities in Section 6.7 are available, and applying Bayes' rule, the probability that a set of matches contains inliers only, given that they have passed a shape similarity test, is given by

$$P(I|D_S < \Delta_S) = \frac{\alpha P_I}{\alpha P_I + \beta P_O}, \tag{6.8}$$

where $P_I = P(I) \approx \epsilon^m$ and $P_O = P(O) = 1 - P_I$. For strict (small) values of Δ_S, it is expected that β will be very small. As a result, $P(I|D_S < \Delta_S)$ should be very close to one, and by the same argument,

$P(O|D_S < \Delta_S) = 1 - P(I|D_S < \Delta_S)$, should be very small – that is, it is expected that false positives[4] are unlikely to occur. This is important, as matches that pass a shape similarity test, can now identified as inliers with high certainty.

Now consider the scenario where there are N matched points from two viewpoints such that $N \gg m$. In order for PORUS to identify points as inliers, m matched points are randomly sampled and evaluated until a set of points pass the shape similarity test. This procedure can be modelled as a geometric distribution [68], where it is shown that the average number of times m points must be sampled, before finding an inlier set that passes the shape similarity test, is given by

$$\bar{k}_P = \frac{1}{\alpha\epsilon^m}, \tag{6.9}$$

with a corresponding standard deviation of

$$\sigma_{k_P} = \frac{\sqrt{1-\alpha\epsilon^m}}{\alpha\epsilon^m}. \tag{6.10}$$

The variable, k_P, is the number of sampling instances required to identify an initial set of inliers using PORUS.

In this section, it was shown that it is possible to identify sets of sampled inliers using a measure of shape similarity across viewpoints. The sampling strategy followed to identify the remaining points, after an initial set of inliers is identified, is now discussed.

6.4 PORUS Sampling Strategy

As conceptually described in Section 6.1, different sampling strategies may be used in PORUS to take advantage of prior knowledge relating to the inlier probabilities of matched point pairs. This section provides a brief overview of basic decision theory concepts that are used to develop an adaptive sampling strategy for PORUS.

6.4.1 Overview of Entropy

Entropy is the measure of uncertainty in a distribution, and is defined formally as,

$$\begin{aligned} H(X) &= E[-\ln p(x)] \\ &= -\int p(x)\ln p(x)\mathrm{d}x, \end{aligned} \tag{6.11}$$

where $H(X)$ is the entropy corresponding to the random variable X with a probability density function, $p(x)$, and $E[\,\cdot\,]$ denotes the expected value operator. A large entropy corresponds to high uncertainty in the state of X and conversely, a small entropy is indicative of high certainty in the state of X. An alternative conceptual representation of entropy is used by Barber [121, pp. 166-167], where $H(X)$ is viewed as,

$$H(X) = -D_{\mathrm{KL}}(P||U) + K \tag{6.12}$$

where $D_{\mathrm{KL}}(P||U)$ is the KL divergence as defined in Equation 3.26, U is a uniform distribution, P corresponds to $p(x)$ and K is a constant. From Equation 6.12, as P tends to a uniform distribution, $D_{\mathrm{KL}}(P||U)$ tends to zero and the entropy increases. Likewise, as P diverges from a uniform distribution, the entropy decreases. This is intuitive as a uniform distribution contains the least amount of information about the state of the random variable, X, and any change in distribution relative to a uniform distribution, results in more information about the state of X. A worked example is now provided in which the entropy of a binary random variable is discussed.

[4] *False positives* are defined as outliers incorrectly identified as inliers, while *false negatives* are defined as inliers incorrectly identified as outliers.

6.4.1.1 Example: Bernoulli Distribution

A Bernoulli distribution is defined as the probability distribution of a random variable that can assume two states; a success state (with probability a) and a failure state (with probability $b = 1 - a$). The probability density function of a binary random variable, X, is given by

$$p(x) = \begin{cases} b = 1 - a & \text{for } x = 0 \\ a & \text{for } x = 1. \end{cases} \tag{6.13}$$

The entropy of the Bernoulli distribution, $H(X)$, calculated from Equation 6.11, is given by

$$\begin{aligned} H(X) &= -\sum_x p(x) \ln p(x) \\ &= -b \ln b - a \ln a. \end{aligned} \tag{6.14}$$

The entropy of $p(x)$ as a function of the probability of success is shown in Figure 6.8. As expected from

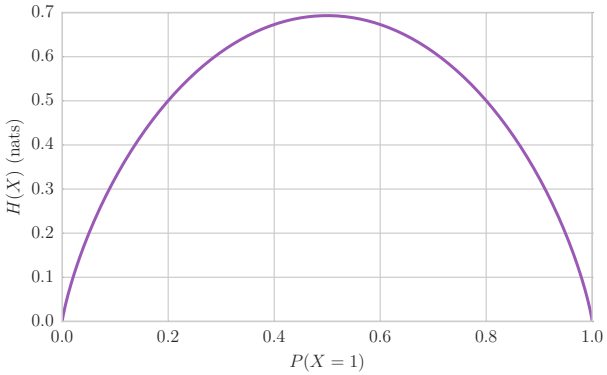


Figure 6.8: Entropy of binary random variable as a function of probability of success.

the definition of entropy, the maximum entropy corresponds to the state of greatest uncertainty when the probability of success and failure are equal (a uniform distribution). As $P(X = 1)$ varies, so does the entropy, where it can be seen that lower values of $H(X)$ correspond an increased certainty in the state of X. If there is no uncertainty in the state, when $P(X = 1) = 0$ or $P(X = 1) = 1$, the entropy is zero.

6.4.1.2 Change in Entropy

The goal of PORUS is to identify inliers and outliers with great certainty. This can be viewed as minimising the entropy in $\mathbf{w}$, the random vector of inlier probabilities. However, it is also required that PORUS is efficient, which means that the uncertainty should be decreased as quickly as possible with each action – that is, the manner in which samples are drawn. The concept of a conditional entropy and information gain is therefore introduced.

The conditional entropy of X, after some action, u, has been performed is given by

$$H(X|u) = -\int p(x|u) \ln p(x|u) \mathrm{d}x. \tag{6.15}$$

Furthermore, the information gain, defined as the expected change in entropy [90, p. 574], associated with each action is calculated as follows,

$$I_{G,u} = H(X) - E[H(X|u)]. \tag{6.16}$$

This allows adaptive sampling to be formulated as a decision theory problem, where the choice of sampling strategy is chosen to maximise the information gain at each sampling instance.

6.4.2 Sampling Strategies

Once an initial set of inliers has been found, there are several possible sampling approaches that can be used to determine whether the remaining matches are inliers or outliers. In this section, an adaptive sampling strategy is developed that attempts to maximise the information gain for each sampling instance. For the development of the adaptive sampling strategy, it is assumed that $\beta = 0$ and consequently, $P(I|D_S < \Delta_S) = 1$. A contaminated set will therefore always fail a shape similarity test. It will be shown in Section 6.7.1 that this assumption is supported by experimental results.

Consider the situation where a set of m inliers has already been identified. At the next sampling instance, m matched points must be selected to perform the next shape similarity test. One possible approach would be to randomly sample $m-1$ inliers, and deterministically select one of the remaining uncertain matches. In this way, the single uncertain match can be identified as an inlier or outlier directly. Alternatively, m matched points, which have not already been identified as inliers, can be randomly sampled. If these samples pass the shape similarity test, all m matched point pairs are identified as inliers. Intuitively, this approach appears to be more efficient as the states of multiple inliers are determined at each sampling instance. This is especially true in situations where there is a low probability of sampling an outlier. However, if the shape similarity test is failed, little information is gained about the states of the matches – that is, it is known that the set of samples is contaminated, but it is not possible to determine exactly which match is an outlier. Additional sampling instances, with a different combination of samples, are then required to determine whether those sampled matches are inliers or outliers with greater certainty. The principle of information gain is therefore used to adaptively change between these two sampling strategies so that the average number of sampling instances required is lowered, thereby reducing the execution time of PORUS.

6.4.2.1 Expected Information Gain

Consider the binary variables, $W^{[i]}$, defined in Section 6.3. Each matched point pair has two possible states, an inlier state and an outlier state. The entropy prior to a shape similarity test, $H(W^{[i]})$, is calculated as

$$\begin{aligned} H(W^{[i]}) &= -\sum_{w^{[i]}} p(w^{[i]}) \log p(w^{[i]}) \\ &= -P(w_O^{[i]}) \log P(w_O^{[i]}) - P(w_I^{[i]}) \log P(w_I^{[i]}) \\ &= -(1-\epsilon) \log (1-\epsilon) - \epsilon \log \epsilon. \end{aligned} \tag{6.17}$$

The expected entropy of $W^{[i]}$, after executing a sampling strategy, u, and observing the result of a shape similarity test, is given by

$$\begin{aligned} E[H(W^{[i]}|u)] &= (\alpha P_I + \beta P_O) H(W^{[i]}|D_S < \Delta_S, u) + (1 - \alpha P_I - \beta P_O) H(w^{[i]}|D_S \geq \Delta_S, u) \\ &= (\alpha P_I) H(W^{[i]}|D_S < \Delta_S, u) + (1 - \alpha P_I) H(W^{[i]}|D_S \geq \Delta_S, u), \end{aligned} \tag{6.18}$$

where P_I and P_O are defined in Equation 6.8 and it is assumed that $\beta = 0$. The expected information gain is then calculated from,

$$I_{G,u} = H(W^{[i]}) - E[H(W^{[i]}|u)]. \tag{6.19}$$

Equation 6.19 is now used to calculate the information gain for two cases, a linear sampling strategy and a greedy sampling approach.

6.4.2.2 Linear Sampling

An overview of PORUS with a linear sampling approach is shown in Algorithm 6.1. Initially, all observations are considered to be outliers and are maintained in an outlier set, $\mathcal{O}$ (Lines 2–3). Random sets of m point

Algorithm 6.1 PORUS with Linear Sampling

Input:

$\mathcal{Z}$ Set of m-dimensional matched point pairs: $\{(\mathbf{z}_1^{[1]}, \mathbf{z}_2^{[1]}), \ldots, (\mathbf{z}_1^{[N]}, \mathbf{z}_2^{[N]})\}$

Output:

$\mathcal{I}$ Set of inliers

$\mathcal{O}$ Set of outliers

1: **function** LINEARSAMPLE($\mathcal{Z}$)
2: $\quad\mathcal{I} \leftarrow \emptyset$
3: $\quad\mathcal{O} \leftarrow \mathcal{Z}$
4: $\quad$**while** $\mathcal{I} = \emptyset$ **do**
5: $\quad\quad\mathcal{S} \leftarrow$ RANDOMSAMPLE($\mathcal{O}, m$) ▷ $\|\mathcal{S}\| = m$
6: $\quad\quad$**if** VERFIYINLIERS($\mathcal{S}$) is true **then** ▷ Shape similarity test
7: $\quad\quad\quad\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{S}$
8: $\quad\quad\quad\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{S}$
9: $\quad\quad$**end if**
10: $\quad$**end while**
11: $\quad$**for** $(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]}) \in \mathcal{O}$ **do**
12: $\quad\quad\mathcal{S} \leftarrow$ RANDOMSAMPLE($\mathcal{I}, m-1$) ▷ $\|\mathcal{S}\| = m-1$
13: $\quad\quad\mathcal{S} \leftarrow \mathcal{S} \cup \{(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]})\}$
14: $\quad\quad$**if** VERFIYINLIERS($\mathcal{S}$) is true **then**
15: $\quad\quad\quad\mathcal{I} \leftarrow \mathcal{I} \cup \{(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]})\}$
16: $\quad\quad\quad\mathcal{O} \leftarrow \mathcal{O} \setminus \{(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]})\}$
17: $\quad\quad$**end if**
18: $\quad$**end for**
19: $\quad$**return** $\mathcal{I}$,$\mathcal{O}$
20: **end function**

pairs are sampled and evaluated until a set of inliers is found (Lines 4–10). Once identified as inliers, the matched point pairs are inserted into the inlier set, $\mathcal{I}$ and removed from $\mathcal{O}$ (Lines 7–8). Thereafter, $m-1$ point pairs are randomly sampled from $\mathcal{I}$ in addition to a single point pair selected from $\mathcal{O}$ (Lines 11–18). Sets of inliers and outliers, are returned once all points in $\mathcal{O}$ have been individually evaluated (Line 19).

Consider the situation where a set of m inliers has already been identified and a linear sampling approach is taken. Furthermore, assume that the inlier ratio in $\mathcal{O}$ is given by ϵ. If a shape similarity test is passed, the single matched point pair from $\mathcal{O}$ is verified as an inlier, and its respective inlier probability distribution is updated to,

$$p(w^{[i]}|D_S < \Delta_S, u_{\text{linear}}) = \begin{cases} 0, & \text{for } w^{[i]} = w_O \\ 1, & \text{for } w^{[i]} = w_I. \end{cases} \tag{6.20}$$

Likewise, if the test is failed,

$$p(w^{[i]}|D_S \geq \Delta_S, u_{\text{linear}}) = \begin{cases} 1, & \text{for } w^{[i]} = w_O \\ 0, & \text{for } w^{[i]} = w_I. \end{cases} \tag{6.21}$$

It should be noted here that linearly sampled inliers which do not pass the shape similarity test are treated as if they are outliers. The entropy of both these distributions is zero and consequently, $E[H(W^{[i]}|u_{\text{linear}})] = 0$. The information gain for a linear point sampling strategy is simply,

$$\begin{aligned} I_{G,\text{linear}} &= H(W^{[i]}) - E[H(W^{[i]}|u_{\text{linear}})] \\ &= -(1-\epsilon)\log(1-\epsilon) - \epsilon\log\epsilon. \end{aligned} \tag{6.22}$$

If the inlier ratio, ϵ, is high, the corresponding entropy, $H(W^{[i]})$ determined from Equation 6.17, is low and consequently, the information gain obtained from the sampling instance is small. Moreover, $I_{G,\text{linear}}$,

is a maximum when $p(w^{[i]})$ is uniformly distributed, so linear sampling is expected to perform well for situations with high uncertainty about the states of matched point pairs.

6.4.2.3 Greedy Sampling

The second technique employs a greedy sampling strategy – that is, m point pairs are sampled from a set of matched points which have not been identified as inliers. This greedy sampling approach is equivalent to repeating the process of identifying an initial inlier set for linear sampling (Lines 5–9 in Algorithm 6.1).

Once again consider the situation where a set of m inliers has already been identified and the inlier ratio in the outlier set, $\mathcal{O}$, is given by ϵ. However, instead of linear sampling, a greedy approach, where m matches are randomly sampled from $\mathcal{O}$, is performed. If the set of matches pass a shape similarity test, all sampled matches are considered inliers and their inlier probabilities are given by

$$p(w^{[i]}|D_S < \Delta_S, u_{\text{greedy}}) = \begin{cases} 0, & \text{for } w^{[i]} = w_O \\ 1, & \text{for } w^{[i]} = w_I, \end{cases} \tag{6.23}$$

which has zero entropy. If the set fails the test, it is not possible to determine the state of individual matched points with certainty. At least one of the matches is an outlier, but it is not clear which one it is. Therefore, the inlier probabilities of all m samples are updated to,

$$p(w^{[i]}|D_S \geq \Delta_S, u_{\text{greedy}}) = \begin{cases} P(w_O^{[i]}|D_S \geq \Delta_S), & \text{for } w^{[i]} = w_O \\ 1 - P(w_O^{[i]}|D_S \geq \Delta_S), & \text{for } w^{[i]} = w_I, \end{cases} \tag{6.24}$$

where

$$\begin{aligned} P(w_O^{[i]}|D_S \geq \Delta_S) &= \frac{P(D_S \geq \Delta_S|w_O^{[i]})P(w_O^{[i]})}{P(D_S \geq \Delta_S)} \\ &= \frac{1-\epsilon}{(1-\alpha)P_I + P_O}. \end{aligned} \tag{6.25}$$

The information gain is then determined by

$$I_{G,\text{greedy}} = m[H(W^{[i]}) - (1 - \alpha P_I)H(W^{[i]}|D_S \geq \Delta_S, u_{\text{greedy}})], \tag{6.26}$$

where the multiplication by m incorporates the fact there is information gain for multiple point pairs. For high values of ϵ and α, Equation 6.26 can be approximated as,

$$I_{G,\text{greedy}} \approx mH(W^{[i]}), \tag{6.27}$$

which is clearly greater than the information gain for the linear sampling approach given in Equation 6.20. This illustrates the benefit of a greedy sampling strategy for low contamination levels.

Point pairs that pass a shape similarity test during greedy sampling are moved to the current inlier set, while matches that fail the test remain in the outlier set, $\mathcal{O}$. The remaining points are now re-sampled and tested in the next sampling instance, however, it is important to note that the inlier probabilities have now changed from their original states. Furthermore, the inlier ratio, ϵ, is no longer valid for the remaining uncertain matches as inliers have been removed from $\mathcal{O}$. Equations 6.22 and 6.26 require ϵ to determine the information gain of each technique, therefore ϵ needs to be re-estimated after every sampling instance.

Suppose that an initial estimate of the inlier ratio is given as ϵ. If m point pairs are sampled and the shape similarity test is passed, a new estimate of the inlier ratio is given by,

$$\hat{\epsilon} = \frac{\epsilon N - m}{N - m}, \tag{6.28}$$

where $N = \|\mathcal{O}\|$. Likewise, if the shape similarity test is failed, $\hat{\epsilon}$ is calculated as

$$\hat{\epsilon} = \frac{(N-m)\epsilon + m(1 - P(w_O^{[i]}|D_S \geq \Delta_S))}{N}, \tag{6.29}$$

where $P(w_O^{[i]}|D_S \geq \Delta_S)$ is determined from Equation 6.25. In Equation 6.29, the estimated inlier ratio is calculated by averaging the inlier probabilities of $N - m$ matched points that were not sampled and the m sampled matches. Furthermore, instead of maintaining the individual inlier probabilities for all matches in $\mathcal{O}$, $\hat{\epsilon}$ is used to update the inlier probabilities of all the remaining matches in $\mathcal{O}$ such that,

$$p(w^{[i]}) = \begin{cases} 1 - \hat{\epsilon} & \text{for } w^{[i]} = w_O \\ \hat{\epsilon} & \text{for } w^{[i]} = w_I. \end{cases} \tag{6.30}$$

Each subsequent sampling instance, and information gain calculation, is therefore performed independently on an outlier set with a given inlier ratio where all matches have the same prior inlier probability. An adaptive sampling approach is now proposed that determines the information gain after each sampling instance to minimise the entropy of the matches.

6.4.3 Proposed Adaptive Sampling Approach

The sampling strategies described in the previous two sections are now combined in an adaptive sampling approach. Before samples are drawn, the expected information gain is calculated for both the linear and greedy sampling strategies using Equations 6.22 and 6.26. Greedy sampling is used as long as the information gain, $I_{G,\text{greedy}}$, is greater than that of $I_{G,\text{linear}}$. Thereafter, linear sampling is performed until all point pairs have been identified as either an inlier or an outlier match.

An overview of the proposed adaptive sampling technique, combining linear and greedy sampling as well as the estimator used for the inlier ratio, ϵ, is shown in Algorithm 6.2. In the same fashion as the linear sampling algorithm, given in Algorithm 6.1, initially all matched point pairs are considered outliers and maintained in $\mathcal{O}$ (Lines 2–3), and an initial inlier set is found by evaluating random sets of m point pairs (Lines 4–10). The inlier ratio in $\mathcal{O}$ is estimated before the information gains of linear and greedy sampling approaches are determined (Lines 11–13). If the information gain of greedy sampling is greater than that of linear sampling, m points pairs are repeatedly sampled and evaluated (Lines 15–19). The inlier ratio is re-estimated after each sampling instance and then used to re-calculate the information gain of each sampling strategy (Lines 20–22). If the information gain of a linear sampling approach is greater than that of greedy sampling, the sampling strategy is adapted and linear sampling is performed on the remaining point pairs in $\mathcal{O}$, and sets of inliers and outliers are returned (Lines 24–32).

It should be intuitive that for high levels of contamination (small ϵ), the first calculation of $I_{G,\text{linear}}$ and $I_{G,\text{greedy}}$ (Lines 11–13) will result in $I_{G,\text{linear}} > I_{G,\text{greedy}}$. This is the worst-case scenario, where adaptive sampling is equivalent to linear sampling. For lower levels of contamination, several instances of greedy sampling would be performed, which should lead to an increase in efficiency by reducing the number of shape similarity tests performed.

So far, a shape similarity test, which has not been defined, has been used to identify inliers. The next section formally defines shape parameters for the 3D case as well as a measure of shape similarity.

6.5 Shape Similarity

So far, PORUS has been analysed for distinguishing between points undergoing rigid body transformations in $\mathbb{R}^m$ and mismatched points (outliers). In Sections 6.3 and 6.4 a shape similarity test was used to distinguish between sets of inliers and contaminated sets. In this section, shape parameters and a similarity test are derived for the specific case of $\mathbb{R}^3$ which will be used for robust visual odometry.

6.5.1 Triangle Shape Parameters

For the three dimensional case, a triple of points is randomly sampled that forms a triangle in space. In order to compare the shape of matched points across viewpoints, it is necessary to obtain a parametric

Algorithm 6.2 PORUS with Adaptive Sampling

Input:

$\mathcal{Z}$ Set of m-dimensional matched point pairs: $\{(\mathbf{z}_1^{[1]}, \mathbf{z}_2^{[1]}), \ldots, (\mathbf{z}_1^{[N]}, \mathbf{z}_2^{[N]})\}$

ϵ Initial estimate for fraction of inliers

Output:

$\mathcal{I}$ Set of inliers

$\mathcal{O}$ Set of outliers

function AdaptiveSample($\mathcal{Z}$)
  $\mathcal{I} \leftarrow \emptyset$
  $\mathcal{O} \leftarrow \mathcal{Z}$
  while $\mathcal{I} = \emptyset$ **do**
    $\mathcal{S} \leftarrow$ RandomSample($\mathcal{O}, m$) ▷ $\|\mathcal{S}\| = m$
    if VerfiyInliers($\mathcal{S}$) is true **then** ▷ Shape similarity test
      $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{S}$
      $\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{S}$
    end if
  end while
  $\hat{\epsilon} \leftarrow$ UpdateInlierRatio($\epsilon, \|O\|$) ▷ Equations 6.28 and 6.29
  $I_{G,\text{linear}} \leftarrow$ InformationGainLinear($\hat{\epsilon}$) ▷ Equation 6.22
  $I_{G,\text{greedy}} \leftarrow$ InformationGainGreedy($\hat{\epsilon}$) ▷ Equation 6.26
  while $I_{G,\text{greedy}} > I_{G,\text{linear}}$ **do** ▷ Greedy Sampling
    $\mathcal{S} \leftarrow$ RandomSample($\mathcal{O}, m$) ▷ $\|\mathcal{S}\| = m$
    if VerfiyInliers($\mathcal{S}$) is true **then** ▷ Shape similarity test
      $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{S}$
      $\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{S}$
    end if
    $\hat{\epsilon} \leftarrow$ UpdateInlierRatio($\hat{\epsilon}, \|O\|$) ▷ Equations 6.28 and 6.29
    $I_{G,\text{linear}} \leftarrow$ InformationGainLinear($\hat{\epsilon}$) ▷ Equation 6.22
    $I_{G,\text{greedy}} \leftarrow$ InformationGainGreedy($\hat{\epsilon}$) ▷ Equation 6.26
  end while
  for $(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]}) \in \mathcal{O}$ **do** ▷ Linear Sampling
    $\mathcal{S} \leftarrow$ RandomSample($\mathcal{I}, m-1$) ▷ $\|\mathcal{S}\| = m-1$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]})\}$
    if VerfiyInliers($\mathcal{S}$) is true **then** ▷ Shape similarity test
      $\mathcal{I} \leftarrow \mathcal{I} \cup \{(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]})\}$
      $\mathcal{O} \leftarrow \mathcal{O} \setminus \{(\mathbf{z}_1^{[i]}, \mathbf{z}_2^{[i]})\}$
    end if
  end for
  return $\mathcal{I}$,$\mathcal{O}$
end function

representation of a triangle. There are several ways to parameterise a triangle; however, in this section a parametric representation is obtained by defining three distances that completely describe the shape and size of a triangle. This avoids the use of any angles which would add severe non-linearities.

As an example, three points described by their position vectors in the robot coordinate frame $\mathbf{x}_r^{[i]}$, $\mathbf{x}_r^{[j]}$ and $\mathbf{x}_r^{[k]}$, are arranged in a triangle Δ_{ijk}. The vector from the first point to the second point is denoted by $\mathbf{r}_{ij} = \mathbf{x}_r^{[j]} - \mathbf{x}_r^{[i]}$ and similarly, the vector from the first to the third point is denoted by $\mathbf{r}_{ik} = \mathbf{x}_r^{[k]} - \mathbf{x}_r^{[i]}$. The corresponding unit vectors are given by $\mathbf{u}_{ij}$ and $\mathbf{u}_{ik}$ respectively. The triangular shape parameters

are then described by

$$\begin{bmatrix} d_{ij} \\ c_{ik\|ij} \\ c_{ik\perp ij} \end{bmatrix} = \mathbf{f}_T \left(\begin{bmatrix} \mathbf{x}_r^{[i]} \\ \mathbf{x}_r^{[j]} \\ \mathbf{x}_r^{[k]} \end{bmatrix} \right) = \begin{bmatrix} \|\mathbf{r}_{ij}\| \\ \mathbf{r}_{ik} \cdot \mathbf{u}_{ij} \\ \sqrt{(\mathbf{r}_{ik})^2 - (\mathbf{r}_{ik} \cdot \mathbf{u}_{ij})^2} \end{bmatrix} \tag{6.31}$$

where d_{ij} is the distance between i and j, $c_{ik\|ij}$ is the component of $\boldsymbol{r}_{ik}$ in the direction $\boldsymbol{r}_{ij}$ and $c_{ik\perp ij}$ is the component of $\boldsymbol{r}_{ik}$ perpendicular to $\boldsymbol{r}_{ij}$. These parameters are illustrated in Figure 6.9 where the three points are denoted by their indices. These parameters, which describe the observed shape of points in a

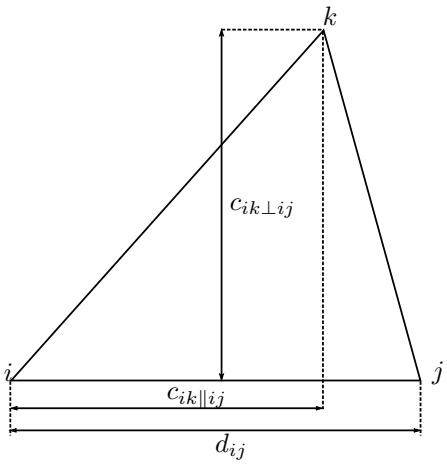


Figure 6.9: Illustration of triangle shape parameters. Three distances, d_{ij}, $c_{ik\|ij}$ and $c_{ik\perp ij}$, parameterise the shape formed by a triple of 3D points.

single viewpoint, are used to determine the change in shape across viewpoints for PORUS. A method for quantifying change in shape is now discussed.

6.5.2 Measure of Shape Similarity

In the previous subsection, shape parameters were derived for visual odometry. The next step requires that a measure of shape similarity between viewpoints be determined.

Suppose there is a vector of shape parameters, $\mathbf{s}_{k-1}$, describing the observed shape in a viewpoint. Likewise, a second vector of shape parameters, $\mathbf{s}_k$, describes the corresponding observed shape in a second viewpoint. In a perfect world, these observed shapes would be identical. In practice, due to the nature of measurement noise, the observed shapes will be different and even more so if the points are contaminated with outliers. It is therefore necessary to capture the similarity of shapes with a quantitative measure, D_S.

Each triangle is described by three 3D feature points, $\mathbf{x}_{r,k}^{[i]}$, which are determined by performing triangulation (Equation 2.32) on stereo measurements,

$$\mathbf{z}_k^{[i]} = \begin{bmatrix} \hat{x}_L^{[i]} & \hat{y}_L^{[i]} & \hat{x}_R^{[i]} & \hat{y}_R^{[i]} \end{bmatrix}^\top. \tag{6.32}$$

The triple of 3D features is then used to determine shape parameters by applying Equation 6.31. This complex relationship is described by

$$\mathbf{s}_k = \mathbf{f}_T \circ \mathbf{f}_{\text{tri}} \left(\begin{bmatrix} \mathbf{z}_k^{[1]} \\ \mathbf{z}_k^{[2]} \\ \mathbf{z}_k^{[3]} \end{bmatrix} \right) \tag{6.33}$$

where $\mathbf{f}_T \circ \mathbf{f}_{\text{tri}}$ is a composite function that maps image coordinates to triangular shape parameters.

It is decided to model $\mathbf{s}_{k-1}$ and $\mathbf{s}_k$ as Gaussian random variables, $\mathbf{s}_{k-1} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{s}_{k-1}}, \boldsymbol{\Sigma}_{\mathbf{s}_{k-1}})$ and $\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{s}_k}, \boldsymbol{\Sigma}_{\mathbf{s}_k})$. As discussed in Section 3.2, the measurement noise in the image coordinate system

is modelled as jointly Gaussian; however, the transformation from image coordinates to triangle shape parameters is non-linear, therefore the resulting distribution of shape parameters is not guaranteed to be normally distributed. Nevertheless, the distribution is approximated as Gaussian to simplify further calculations and the unscented transform is used to propagate the uncertainty in image coordinates to shape parameters.

Once approximations of $\mathbf{s}_1$ and $\mathbf{s}_2$ have been calculated, a measure of shape similarity is determined. The means of $\mathbf{s}_1$ and $\mathbf{s}_2$ can be viewed as two measurements of some underlying shape in $\mathbb{R}^3$ where the covariances, $\boldsymbol{\Sigma}_{\mathbf{s}_1}$ and $\boldsymbol{\Sigma}_{\mathbf{s}_2}$ describe the uncertainty of the respective measurements. This is analogous to the problem formulation of Section 5.2.2. A measure of shape similarity, in the form of a probabilistic distance, is therefore given by,

$$D_S = (\boldsymbol{\mu}_{\mathbf{s}_{k-1}} - \boldsymbol{\mu}_{\mathbf{s}_k})^\top (\boldsymbol{\Sigma}_{\mathbf{s}_{k-1}} + \boldsymbol{\Sigma}_{\mathbf{s}_k})^{-1} (\boldsymbol{\mu}_{\mathbf{s}_{k-1}} - \boldsymbol{\mu}_{\mathbf{s}_k}) + \ln\left(|\boldsymbol{\Sigma}_{\mathbf{s}_{k-1}} + \boldsymbol{\Sigma}_{\mathbf{s}_k}|\right). \tag{6.34}$$

This probabilistic measure, D_S, can now be compared to some arbitrary threshold, Δ_S, as a shape similarity test.

6.6 Comparison of RANSAC and PORUS Execution Times

In this section, a theoretical comparison of probabilistic RANSAC and PORUS execution times are performed. A worst-case sampling strategy for PORUS is assumed – that is, high contamination levels where only linear sampling is performed.

There are two factors which significantly influence the execution time of the RANSAC algorithm. First, according to Equation 4.1, the number of RANSAC iterations, k, is highly dependent on the inlier ratio, ϵ. As ϵ decreases, k grows dramatically, which leads to significantly increased execution times. Secondly, the time spent evaluating each hypothesised model is also proportional to the number of data points, N, as each data point has to be verified. The total cost of a standard RANSAC-based approach is given by

$$c_R = k(c_M + Nc_V), \tag{6.35}$$

where c_M is the cost of instantiating a model and c_V is the cost associated with verifying a hypothesised model against a single data point.

The average total cost of the PORUS, $\bar{c}_P$, is given by,

$$\bar{c}_P = (\bar{k}_P + (N - m))c_S, \tag{6.36}$$

where c_S is the cost of evaluating the shape similarity (or some other inlier test) across viewpoints, $\bar{k}_P$ is calculated from Equation 6.9 and m describes the dimensionality of the data. It takes on average $\bar{k}_P$ sampling instances to identify the first set of inliers and $(N - m)$ instances for linear sampling to identify the remaining points. For standard values of ϵ and m, $\bar{k}_P$ is significantly less than the number of points in a data-set. Since, $\bar{k}_P \ll N$ and it is assumed that $m \ll N$, Equation 6.36 can be approximated as

$$\bar{c}_P \approx Nc_S. \tag{6.37}$$

Equation 6.37 highlights a distinct advantage of PORUS; the complexity is effectively independent of the fraction of inliers as opposed to the RANSAC framework. It is therefore expected that the proposed algorithm will significantly outperform probabilistic RANSAC at low inlier ratios. Furthermore, if it is assumed that $c_V \approx c_S$, then PORUS should on average outperform RANSAC by a factor related to k in terms of efficiency.

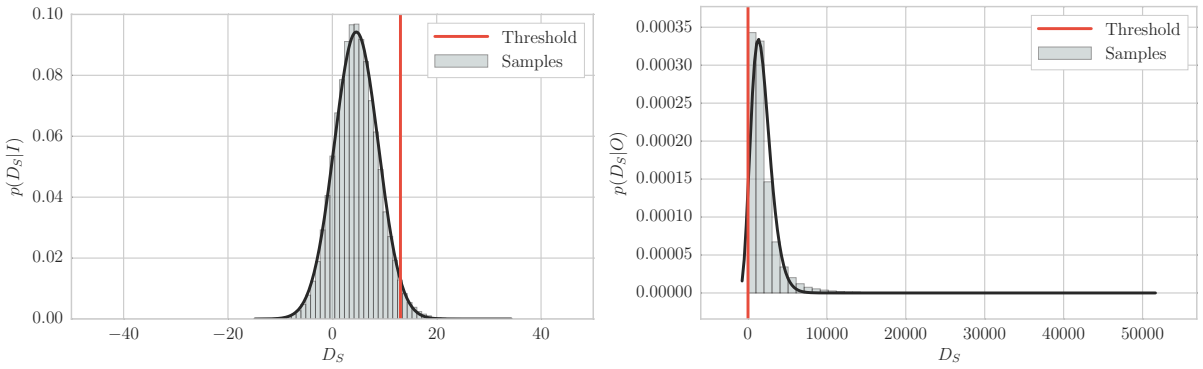
6.7 Experimental Results

Two experiments were performed to verify concepts introduced during the development of the PORUS algorithm in this chapter. Experiments performed to compare PORUS and RANSAC are discussed in

the next chapter. An important assumption was made in Section 6.3, namely, that the probability of a contaminated set passing a shape similarity test, β, is zero. The first experiment aims to support this assumption by investigating the effect of threshold choices under different conditions. The second experiment verifies the adaptive sampling approach, proposed in Section 6.4.3, by comparing costs for linear and adaptive sampling at different inlier ratios. These two experiments are now be discussed in more detail.

6.7.1 Choice of Shape Similarity Threshold

In the first experiment, the effect of the shape similarity threshold, Δ_S, on α and β, defined in Section 6.3, is investigated. A synthetic visual odometry dataset, consisting of two viewpoints and 1000 3D features, was generated using the work of Section 5.4 such that two sequences of matched features, $\mathcal{F}_{k-1}$ and $\mathcal{F}_k$, are generated. The 3D features are uniformly distributed over a cubic volume ($20\,\text{m} \times 20\,\text{m} \times 20\,\text{m}$) centred at the origin of the world coordinate frame. Image features in each sequence are perturbed with noise sampled from a zero-mean normal distribution with $\sigma = 1.0\,\text{px}$, without contamination by outliers ($\epsilon = 1.0$). The feature sequences are then divided into sub-sequences of three matches, so that sequences of matched shape parameters, $\mathcal{S}_{k-1}$ and $\mathcal{S}_k$, are determined from Equation 6.33. For each shape parameter pair, $(\mathcal{S}_{k-1}^{[i]}, \mathcal{S}_k^{[i]})$, a measure of shape similarity, D_S, is determined from Equation 6.34. A histogram of the values obtained for D_S across 1000 independent simulations is shown in Figure 6.10.



(a) **(b)**

Figure 6.10: Verification of shape similarity threshold choice for a noise level of $\sigma = 1.0\,\text{px}$. Threshold, Δ_S, corresponding to $\alpha = P(D_S|I) = 0.95$ is shown in red and distribution fits are shown in black. (a) Conditional distribution, $p(D_S|I)$, obtained from synthetic feature matches. (b) Conditional distribution, $p(D_S|O)$, obtained from synthetic feature matches.

From Figure 6.10a, the samples obtained for D_S appear to be normally distributed. It was therefore decided to model the conditional distribution of D_S, given that all of the observations are inliers, as Gaussian such that,

$$p(D_S|I) = \mathcal{N}(\bar{D}_S, \sigma_{D_S}), \tag{6.38}$$

as indicated by the black line in Figure 6.10a. A threshold, Δ_S, is chosen as

$$\Delta_S = \bar{D}_S + 2\sigma_{D_S}, \tag{6.39}$$

which corresponds to $\alpha = 0.95$ and consequently, a false negative rate of 5%. The experimentally determined α is shown in Table 6.1.

The procedure above is repeated, except outliers are injected as discussed in Section 5.4, where each sub-sequence is randomly contaminated with one, two or three outliers. Once again, a measure of shape similarity is determined for each shape parameter pair. The resulting distribution, $p(D_S|O)$, is shown in Figure 6.10b alongside Δ_S. A chi-squared distribution [89, p. 439] is fit for illustrative

purposes. The majority of the contaminated samples result in $D_S > \Delta_S$, indicating poor shape similarity for measurements containing outliers. A range limited view of this histogram is given in Figure 6.11. Quite clearly, there is a low probability of contaminated samples passing the shape similarity test. This

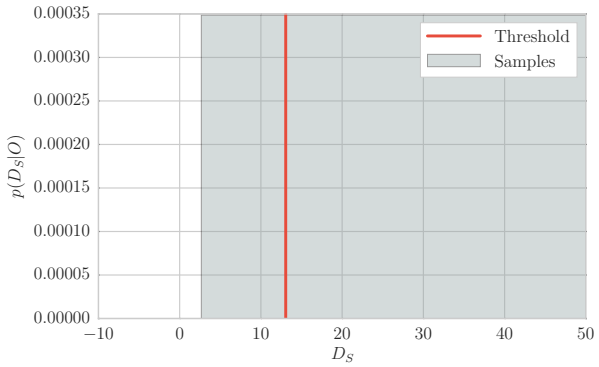
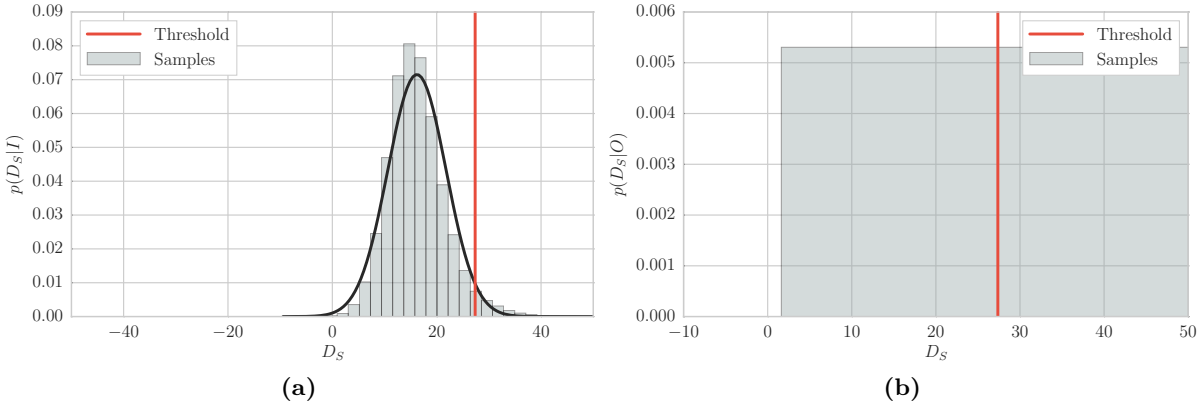


Figure 6.11: Range limited view of Figure 6.10b ($-10 \leq D_S \leq 50$).

corresponds to a β which is very small, as shown in Table 6.1, and supports the assumption that $\beta \approx 0$ made in Section 6.3.

This experiment was repeated for the case of measured image feature matches with a noise level of $\sigma = 5.0\,\text{px}$, resulting in the histograms shown in Figure 6.12 and the values in Table 6.1. Two effects of



(a) **(b)**

Figure 6.12: Verification of shape similarity threshold choice for a noise level of $\sigma = 5.0\,\text{px}$. (a) Conditional frequency distribution, $h(D_S|I)$, obtained from synthetic feature matches and Gaussian fit (black). (b) Range limited view of $h(D_S|O)$ ($-10 \leq D_S \leq 50$)

the higher noise level should be noted. First, the threshold required to maintain $\alpha = 0.95$ is significantly higher due to the nature of the increased measurement noise. Secondly, the probability of a contaminated sample passing the shape similarity test has increased as shown by Figure 6.12b. This is due to the higher threshold required to maintain α. Consequently, for high noise levels, a stricter threshold should be imposed to prevent false positives; however, this stricter threshold reduces the number of inliers that can be used for further calculations and also increases the cost of the algorithm according to Equation 6.36. The results of a low-noise ($\sigma = 0.1\,\text{px}$) experiment are also given in Table 6.1 that show shape similarity is highly effective at rejecting outliers when measurement noise is low.

Two conclusions can be drawn from the above results. First, shape similarity appears to be a viable method of distinguishing between inliers and outliers and secondly, a threshold can be chosen so that the probability of a contaminated sample passing a shape similarity test (false positive) is very low. An important caveat here is that outliers are only introduced in the form of mismatches, and the environment

Table 6.1: Experimentally determined thresholds, Δ_S and corresponding true/false positive rates, α and β.

σ (px)	Δ_S	α	β
0.1	−1.1	0.972	0
1.0	13.15	0.972	0.0002
5.0	27.23	0.967	0.11

is assumed to be static – that is, outliers consistent with secondary motions from dynamic objects are not generated. The value of β may therefore be larger in highly dynamic environments.

6.7.2 Adaptive Sampling Verification

The second experiment verifies the adaptive sampling strategy proposed in Section 6.4.3. The cost of each strategy, c_P, related to the number of inlier tests performed, is determined experimentally for a range of inlier ratios, ϵ, for both linear sampling and adaptive sampling.

Two sequences are defined such that, $\mathcal{W}_1 = \langle w_1^{[i]} \rangle_{i=1}^N$ and $\mathcal{W}_2 = \langle w_2^{[i]} \rangle_{i=1}^N$ where $w_1^{[i]} = w_2^{[i]} = 1$, representing inlier matches between two viewpoints. Elements in $\mathcal{W}_2$ are then randomly replaced with zeros such that $w_2^{[i]} = 0$, where zero elements are representative of outliers, so that ϵ is satisfied. The linear and adaptive sampling algorithms, developed in Section 6.4, are implemented, where m corresponding matches from $\mathcal{W}_1$ and $\mathcal{W}_2$ are sampled and an inlier test is considered passed if all m samples are inliers ($w_1^{[i]} = w_2^{[i]} = 1$). This formulation is representative of a perfect inlier test – that is, $\alpha = 1$ and $\beta = 0$. Samples were drawn until all matches were identified as inliers or outliers.

Experiments were performed across a range of inlier ratios $\epsilon \in \{0.3, 0.4, \ldots, 1.0\}$ for both $m = 3$ and $m = 5$ (representative of a higher dimensional problem) on sequences consisting of $N = 120$ matches. In these experiments it was assumed that a true value of ϵ was available for each experiment. The results of 1000 independent experiments for each inlier ratio are shown in Figure 6.13 alongside the theoretical cost of linear sampling from Equation 6.36. Errors bars indicate the standard deviation of the simulation results.

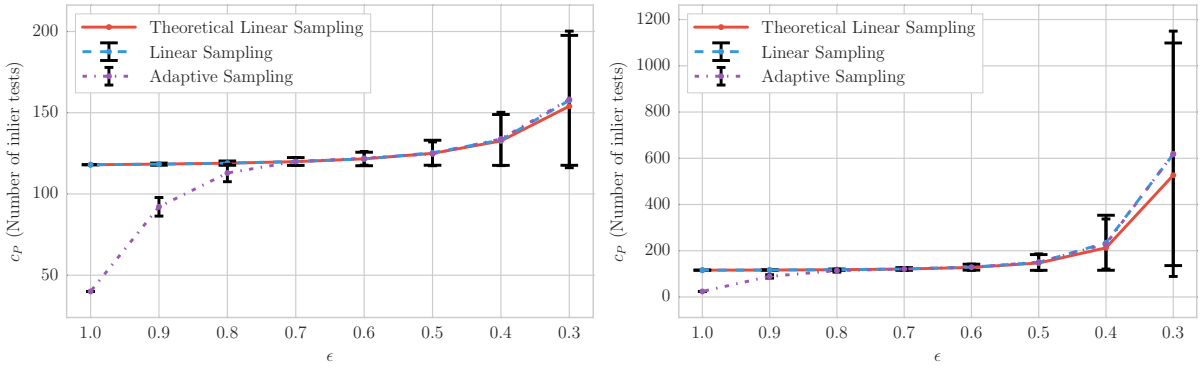


Figure 6.13: Comparison of sampling strategy costs as a function of inlier ratio, ϵ, for $n = 120$. The theoretical cost of linear sampling is shown in red. Experiments for two model complexities are shown, (a) $m = 3$ and (b) $m = 5$.

From Figure 6.13, it can be seen that the theoretical cost of linear sampling matches well with the experimental results. The practical and theoretical cost of linear sampling are equivalent for high inlier ratios in both cases; however, Equation 6.36 appears to underestimate the cost of linear sampling for low inlier ratios in Figure 6.13a, and even more so in Figure 6.13b. This is explained by looking at the derivation of Equation 6.9. Fischler and Bolles [68] model the sampling process as a geometric distribution

– that is, samples are drawn independently with replacement. However, points are not replaced when drawing samples during the search for an initial inlier set, as multiple points are sampled simultaneously, and the value determined from Equation 6.9 is therefore slightly optimistic.

Figure 6.13 also indicates that the use of adaptive sampling increases the efficiency of the PORUS algorithm. At high inlier ratios, adaptive sampling has significantly lower costs than linear sampling in both Figure 6.13a and Figure 6.13b. This improvement can be attributed to greedy sampling being performed at high inlier ratios. As the inlier ratio is decreased, the information gain of greedy sampling becomes smaller and the adaptive sampling approach eventually defaults to linear sampling as seen in Figure 6.13a and Figure 6.13b. As a result, the performance of adaptive sampling and linear sampling are equivalent for low inlier ratios.

The above experiment setup was repeated for $m = 3$, but instead of using the true value of ϵ for each experiment, only an estimate, $\hat{\epsilon}$, was made available. The result obtained using $\hat{\epsilon} = 0.8$, is shown in Figure 6.14. In the region $\epsilon > \hat{\epsilon}$, the adaptive sampling approach still results in lower costs than linear

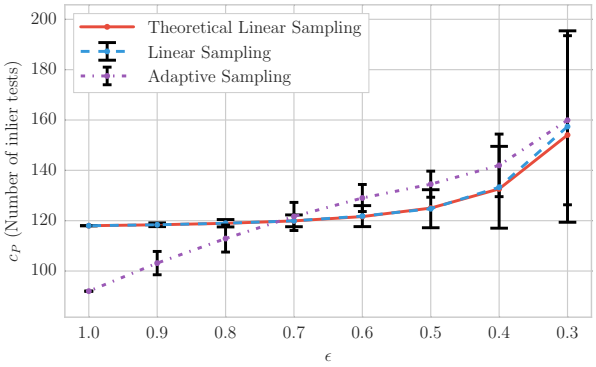


Figure 6.14: Comparison of sampling strategy costs as a function of ϵ where the true value of ϵ is not observed and an initial estimate of the inlier ratio, $\hat{\epsilon} = 0.8$, is used.

sampling but to a lesser extent than that seen in Figure 6.13b. This is attributed to the fact that an underestimated inlier ratio will cause the adaptive sampling approach to default to linear sampling sooner than the ideal case. More importantly, in the region $\epsilon < \hat{\epsilon}$, it can be seen that the adaptive sampling has slightly higher costs than linear sampling. However, this increase in cost is not significant and is a direct result of the overestimated inlier ratio. The adaptive sampling approach follows a greedy sampling strategy for several iterations before the inlier ratio is re-estimated correctly and a linear sampling strategy is followed.

From these results, it can be concluded that adaptive sampling is better suited for PORUS than linear sampling. The reduction in cost at high inlier ratios, seen in Figures 6.13a and 6.13b is significant in comparison to the slight increase in cost incurred for a poorly estimated inlier ratio seen in Figure 6.14. Adaptive sampling therefore performs better than linear sampling in certain cases, and equivalently at worst.

6.8 Chapter Summary

A novel outlier removal technique, given the name PORUS (Probabilistic Outlier Removal Using Shapes) and forming the main contribution of this thesis, was developed and analysed in this chapter. PORUS makes use of inherent shape information to distinguish between inliers and outliers. Akin to RANSAC, PORUS makes use of randomly sampled sets of observations; however, it differs in the sense that a model is not fit to the observations and then verified. Instead, points are evaluated directly in terms of how likely they are to be inliers based on spatial constraints.

The PORUS algorithm was presented conceptually for the well-known line fitting problem alongside RANSAC. The application of PORUS was then extended to the generalised problem of identifying and removing outliers when determining relative transformations in m-dimensional space. It was shown analytically, that the probability of a set of randomly sampled points containing an outlier is related to a measure of shape similarity between viewpoints and moreover, that a measure of shape similarity can be used to distinguish between inliers and outliers with a low probability of false positives.

It was then shown that the mechanism of the PORUS algorithm allows for several sampling strategies, and that the choice of sampling strategy can be formulated as a decision theory problem. The concepts of entropy and information gain were introduced, after which the information gain of two sampling strategies, namely, a linear approach and a greedy approach, were derived. Thereafter, an adaptive sampling strategy was proposed, which maximises the information gain at each sampling instant, to increase the efficiency of the PORUS algorithm.

The concept of shape similarity, required by PORUS, in the context of visual odometry was then investigated. A parametrisation of shape information in $\mathbb{R}^3$ was developed, as well as a probabilistic measure of shape similarity that is based on the work of Section 5.2.2. Thereafter, a theoretical comparison of execution times for RANSAC and PORUS was performed. It was shown that the expected cost of the PORUS algorithm is not dependent on the inlier ratio, which suggests that significant performance improvements are expected at high contamination levels when compared against RANSAC in the next chapter.

The chapter concluded with a set of experimental results verifying several components of the PORUS algorithm. The first experiment focused on the effect of the threshold choice on the probability of rejecting inliers and incorrectly accepting outliers. It was shown that a threshold could be chosen to have a false negative rate of 5% while rejecting almost all outliers, thereby supporting the assumption that the probability of false positives is approximately zero. A second experiment compared the costs of linear sampling against the proposed adaptive sampling approach. The adaptive sampling approach was shown to perform significantly better for high inlier ratios and equivalently for low inlier ratios.

Now that the PORUS algorithm has been formulated, the next chapter focuses on the evaluation of PORUS and RANSAC in terms of robustness and efficiency. Furthermore, PORUS is incorporated into a visual odometry pipeline, and compared against the robust visual odometry framework, implemented in Chapter 5, with both synthetic and practical datasets.

Chapter 7

Experimental Results

In the previous chapter, a novel outlier removal technique, named PORUS, was proposed as an alternative to RANSAC and other hypothesise-and-verify approaches. In this chapter, experiments are performed on both synthetic and practical datasets to compare the performance of PORUS against the probabilistic RANSAC framework developed in Chapter 5.

Several limitations of the standard RANSAC algorithm were discussed in Chapter 4. These included concerns with regards to efficiency at low inlier ratios as well as robustness to noise. Several improvements to the RANSAC algorithm were also discussed and one such improvement, proposed by Brink et al. [82], was implemented in a robust visual odometry framework in Chapter 5. However, the limitations of hypothesise-and-verify approaches in general are still a concern. PORUS was therefore developed in Chapter 6 to directly address the limitations of RANSAC. The probabilistic RANSAC approach acts as benchmark in this chapter, against which PORUS is compared.

The chapter begins with a set of experiments that evaluate the efficiency and robustness of RANSAC and PORUS. Thereafter, the performance of the two algorithms are compared when used for robust motion estimation with synthetic visual odometry datasets. The chapter concludes with experimental results obtained from practical datasets.

7.1 PORUS vs RANSAC

In the first set of experiments, the performance of PORUS and a probabilistic RANSAC framework, developed in Chapter 5, are compared in terms of efficiency and robustness. An overview of the experimental procedure is now provided after which the results are discussed.

7.1.1 Experimental Procedure

The experimental setup of Section 6.7.1 is repeated where two viewpoints of a synthetic visual odometry dataset are generated such that the relative transformation between viewpoints is described by $\mathbf{R}_{\text{truth}}$ and $\mathbf{t}_{\text{truth}}$. Measurements of image features are formed by contaminating sequences of matched features, $\mathcal{F}_{k-1}$ and $\mathcal{F}_k$, with outliers and noise sampled from a zero-mean normal distribution. Thereafter, sequences of reconstructed 3D points, $\mathcal{X}_{k-1}$ and $\mathcal{X}_k$, are determined from triangulation of measured image features. Outlier removal is performed on the two sequences of matched 3D features using the probabilistic RANSAC technique, described in Algorithm 5.1, and the PORUS method, described in Algorithm 6.2. Thereafter, the remaining inliers points are used to estimate relative transformation parameters, $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$, from the motion estimation refinement technique of Section 5.3.

The experimental procedure consists of varying several parameters in relation to a nominal test case defined by the parameters in Table 7.1. The nominal test case is defined as $N = 1000$ matched features with an inlier ratio of $\epsilon = 0.5$ – that is, half of the matches are inconsistent with $\mathbf{R}_{\text{truth}}$ and $\mathbf{t}_{\text{truth}}$, and a noise level of $\sigma = 1.0\,\text{px}$ in image coordinates. The probability of success is set at $\eta = 0.95$, which

Table 7.1: Parameters of nominal test case.

N	ϵ	σ (px)	η	Δ_T	Δ_S
1000	0.5	1.0	0.95	100.0	10.0

is used to determine the number of RANSAC iterations, and the threshold used by the RANSAC[1] in Algorithm 5.1, is chosen empirically as $\Delta_T = 100.0$. The choice of shape similarity threshold, $\Delta_S = 10.0$, is based on the experimental results of Section 6.7.1. Importantly, it should be noted that Δ_T has been chosen in such a way that both RANSAC and PORUS are equally likely to reject outliers. The parameters η, Δ_T and Δ_S are kept constant throughout, while N, ϵ and σ are varied individually across independent experiments.

Several performance measures are used to compare PORUS and RANSAC as N, ϵ and σ are varied in the experiments. A brief description of these measures are given as follows:

- α – The probability of a true positive (an inlier correctly identified as an inlier).
- β – The probability of a false positive (an outlier incorrectly identified as an inlier).
- t_{outlier} – Time spent performing outlier removal in seconds.
- $e_t = \|\mathbf{t}_{\text{truth}} - \hat{\mathbf{t}}\|$ – The norm of the difference between the true and estimated translation vectors.
- $e_R = \|\mathbf{q}_{\text{truth}} - \hat{\mathbf{q}}\|$ – The norm of the difference between the true and estimated rotations in quaternion form.

The true and false positive rates are determined by maintaining a list of injected outlier indices when generating the synthetic datasets and comparing the sets of identified inliers and outliers with true inliers and outliers. Furthermore, both outlier removal techniques were implemented in C++ for realistic timing results.

A total of 100 independent simulations were performed for each experiment. Error bars indicate the standard deviations of the simulation results and shading represents the 95% confidence interval. The various experiments and obtained results will now be discussed in more detail.

7.1.2 Inlier Ratio

In the first experiment, the effect of varying the inlier ratio on performance is investigated for both PORUS and RANSAC. Inlier ratios in the range 0.3–0.9 were tested with the number of matches and noise level kept at their nominal values. The results obtained are shown in Figures 7.1 to 7.3.

As seen in Figure 7.1, the thresholds have been chosen such that RANSAC and PORUS perform equivalently in terms of false positives – that is, the probabilities of identifying an outlier as an inlier, β_{PORUS} and β_{RANSAC}, are the same and more importantly, approximately zero. Equating the performance of these two algorithms in this regard is necessary for a fair comparison, as both PORUS and RANSAC are now equally likely to reject outliers. As the inlier ratio is decreased, the probabilities of true positives, α_{PORUS} and α_{RANSAC}, remain relatively constant for both algorithms. However, α_{PORUS} is consistently higher across all inlier ratios. This is advantageous as a larger set of inliers should result in better motion estimates. Furthermore, it should also be noted that the standard deviation of α_{RANSAC} is significantly larger than α_{PORUS}, and is explained by the fact that RANSAC is not guaranteed to select an uncontaminated set of matches, which results in poorly supported models and small consensus sets. It is therefore expected that estimated transformation parameters using RANSAC will be more inconsistent.

The time taken to perform PORUS and RANSAC, for several inlier ratios, is shown in Figure 7.2a. As predicted by Equation 6.37, the execution time of PORUS is effectively independent of the inlier ratio.

[1] From here on, the probabilistic RANSAC framework proposed by Brink et al. [82] is simply referred to as RANSAC.

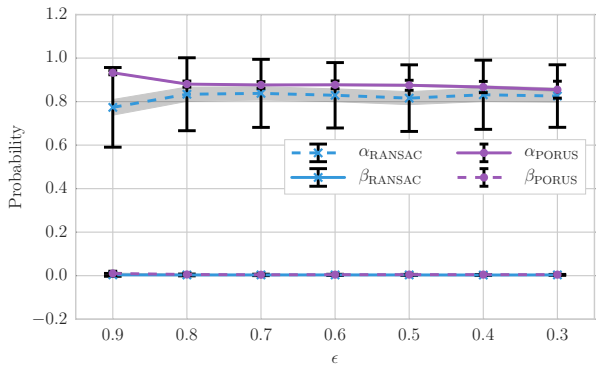
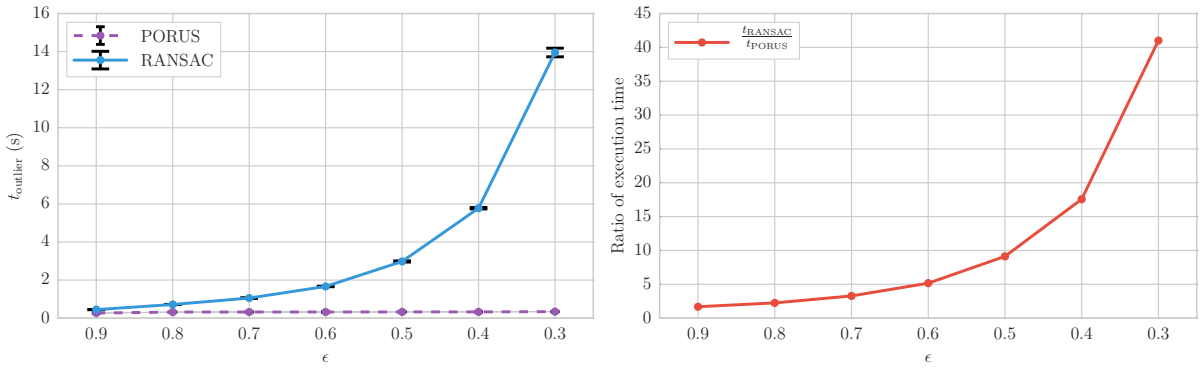


Figure 7.1: Probability of true and false positives, α and β, as a function of inlier ratio for $N = 1000$ points and a noise level of $\sigma = 1.0\,\text{px}$.



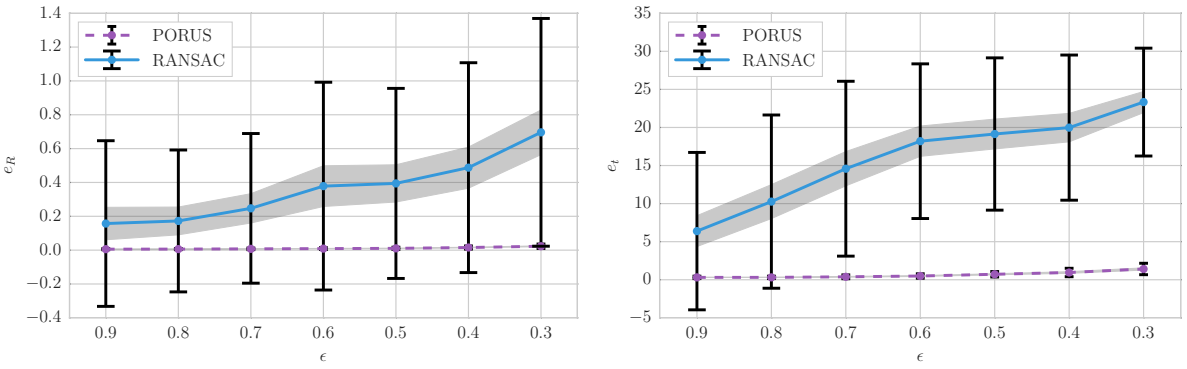
(a) **(b)**

Figure 7.2: Execution time as a function of inlier ratio with $N = 1000$ points and a noise level of $\sigma = 1.0\,\text{px}$. (a) Time taken to perform outlier removal using RANSAC and PORUS. (b) Ratio of execution times, $\frac{t_{\text{RANSAC}}}{t_{\text{PORUS}}}$, shown in a.

The execution time of RANSAC, on the other hand, grows dramatically as the inlier ratio decreases. PORUS therefore significantly outperforms RANSAC for high contamination levels. Figure 7.2b highlights this improvement in performance with an illustration of the relationship, $\frac{t_{\text{RANSAC}}}{t_{\text{PORUS}}}$. At the nominal inlier ratio, $\epsilon = 0.5$, RANSAC is slower by a factor of 10 and 40 times slower at a low inlier ratio of $\epsilon = 0.3$. This is consistent with the prediction made in Section 6.6 that PORUS should on average outperform RANSAC by a factor related to k as calculated in Equation 4.1.

The accuracy of estimated transformation parameters, obtained after performing outlier removal using PORUS and RANSAC, are shown in Figure 7.3 – that is, parameters are estimated using only the inliers identified by each algorithm. As suggested in Figure 7.1, the estimated transformation parameters obtained from RANSAC are significantly more inconsistent than that of PORUS. This is supported by the large standard deviations of e_R^{RANSAC} and e_t^{RANSAC} seen in Figures 7.3a and 7.3b. Once again, these large standard deviations are explained by the fact that RANSAC is only guaranteed to sample a set of uncontaminated points with a probability of η. When no uncontaminated sets are sampled, erroneous parameters are estimated, which leads to the large rotation and translation errors. On the other hand, the mean values of e_R^{PORUS} and e_t^{PORUS} are small, and accompanied by small standard deviations. This indicates that during PORUS very few outliers are misidentified as inliers and as a result, parameter estimation is performed on inliers only. It should be noted that the mean errors of each technique should not be compared directly, as a single incorrect model estimate by RANSAC will result in a large mean error even though the majority of correct model estimates will have relatively small errors.

From this set of results, it can be concluded that PORUS is more adept at handling different inlier ratios than RANSAC. It was shown that PORUS has a higher true positive rate than RANSAC across all



(a) (b)

Figure 7.3: Accuracy of pose estimation as a function of inlier ratio for $N = 1000$ points and a noise level of $\sigma = 1.0\,\text{px}$. (a) Rotation error as function of inlier ratio. (b) Translation error as function of inlier ratio.

inlier ratios, and significantly, an execution time that is not dependent on the inlier ratio. Furthermore, it was shown that PORUS results in more consistent transformation parameter estimates. The focus now shifts to the performance of the two algorithms under different noise conditions.

7.1.3 Noise Level

In the second experiment, the level of noise added to image features is varied while the inlier ratio and number of matches are kept at their nominal values. The results are shown in Figures 7.4 to 7.6.

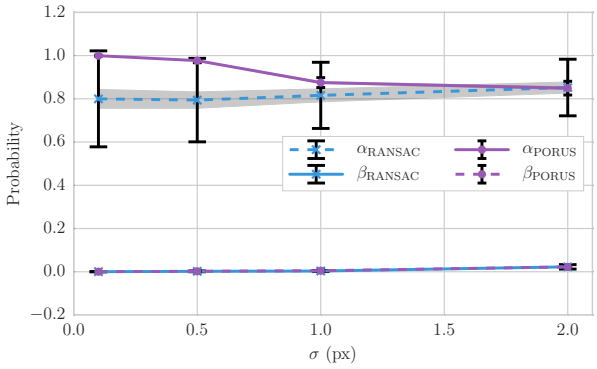


Figure 7.4: Probability of true and false positives, α and β, as a function of noise level for $N = 1000$ points and an inlier ratio of $\epsilon = 0.5$.

In Figure 7.4, it can once again be seen that the thresholds, Δ_T and Δ_S, have been chosen such that both algorithms are equally likely to reject outliers. However, as the noise level in image coordinates is increased, two effects are seen. First, both probabilities of incorrectly identifying an outlier match as an inlier, β_{PORUS} and β_{RANSAC}, increase. This should be intuitive as it becomes difficult to distinguish outliers from very noisy inliers. Secondly, the probability of correctly identifying an inlier match as an inlier, α_{PORUS}, decreases as the noise level is increased. This is explained by the fact that the threshold, Δ_S, was kept constant across all experiments and is based on the result of Figure 6.10, performed at a noise level of $\sigma = 1.0\,\text{px}$. This indicates that the shape similarity threshold is quite sensitive to the level of image noise and needs to be adjusted for high noise levels. However, even with this "strict" threshold at $\sigma = 2.0\,\text{px}$, PORUS still performs equivalently to RANSAC in terms of the true positive rate.

The effect of varying the noise level on the execution time of both algorithms is shown in Figure 7.5. As should be expected, the execution times of both algorithms are independent of the level of image noise.

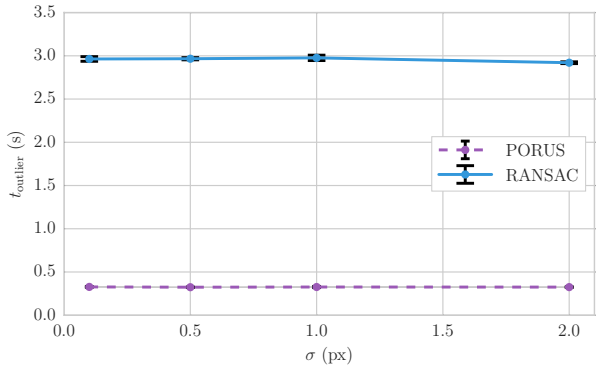
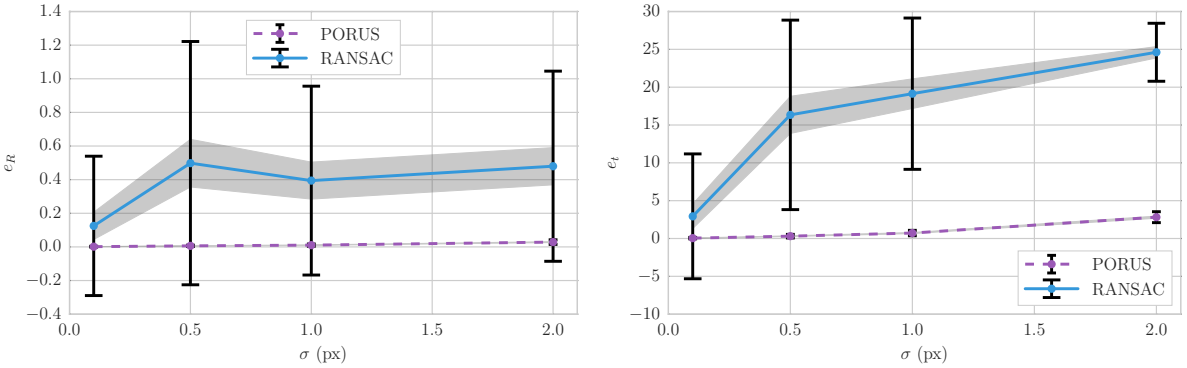


Figure 7.5: Execution time as a function of noise level for $N = 1000$ points and an inlier ratio of $\epsilon = 0.5$.

However, Figure 7.5 does verify the result of Figure 7.3b, which suggests that PORUS is faster by a factor of 10 for an inlier ratio of $\epsilon = 0.5$.

The effect of different noise levels on the accuracy of estimated transformation parameters, obtained after performing outlier removal using PORUS and RANSAC, is shown in Figure 7.6. Similar to the



(a) **(b)**

Figure 7.6: Accuracy of pose estimation as a function of noise level for $N = 1000$ points and an inlier ratio of $\epsilon = 0.5$. (a) Rotation error as a function of noise level. (b) Translation error as a function of noise level.

results of Figure 7.3, it can be seen that the standard deviation of e_R^{RANSAC} and e_t^{RANSAC} is much larger. This can once again be attributed to the fact that RANSAC only samples an uncontaminated set of points with a probability of η. Furthermore, the small errors seen for PORUS indicate that very few outliers are misidentified as inliers. The slight increase in e_t^{PORUS} seen in Figure 7.6b, as the noise level is increased, is a consequence of the larger β_{PORUS} seen in Figure 7.4.

From this set of results, it can be concluded that PORUS is sensitive to the choice of shape similarity threshold and that an increase in noise level has two effects on PORUS. First, the probability of false positives increases as it becomes more difficult to distinguish between outliers and noisy inliers and secondly, the probability of rejecting inliers increases when a strict threshold is used at high noise levels. However, PORUS still outperforms RANSAC under these conditions. The next section investigates the effect of varying the number of matches evaluated.

7.1.4 Number of Matches

The third set of experiments evaluates the performance of PORUS and RANSAC when the number of available matches, N, is varied, while the inlier ratio and level of noise in image coordinates are kept at their nominal values. The results obtained are shown in Figures 7.7 to 7.9.

The probabilities of true and false positives for both techniques are independent of the number of matches as shown in Figure 7.7. It should be noted, however, that once again α_{PORUS} is consistently larger

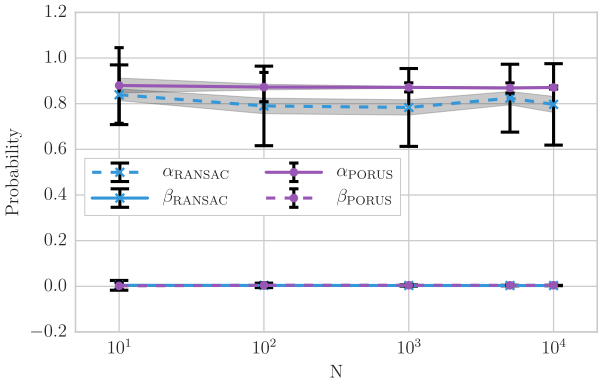


Figure 7.7: Probability of true and false positives, α and β, as a function of N for an inlier ratio of $\epsilon = 0.5$ and noise level of $\sigma = 1.0\,\text{px}$.

than α_{RANSAC} across all values of N and the probabilities of false positives are equivalent. Furthermore, the standard deviations of α_{PORUS} are relatively large for small values of N. It should therefore be expected that the corresponding estimated transformation parameters will also have larger standard deviations.

The effect of varying the number of matches on the execution time of both algorithms is shown in Figure 7.8. As should be expected, the execution times of both algorithms grow as the number of matches

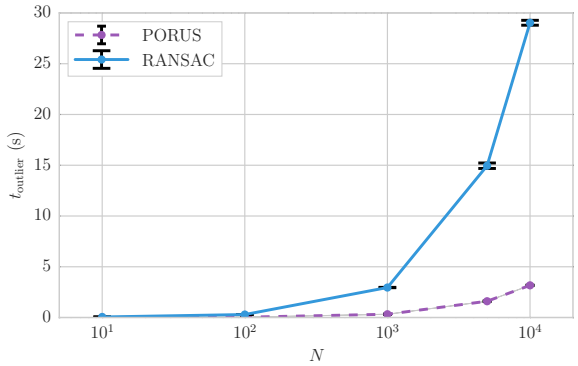
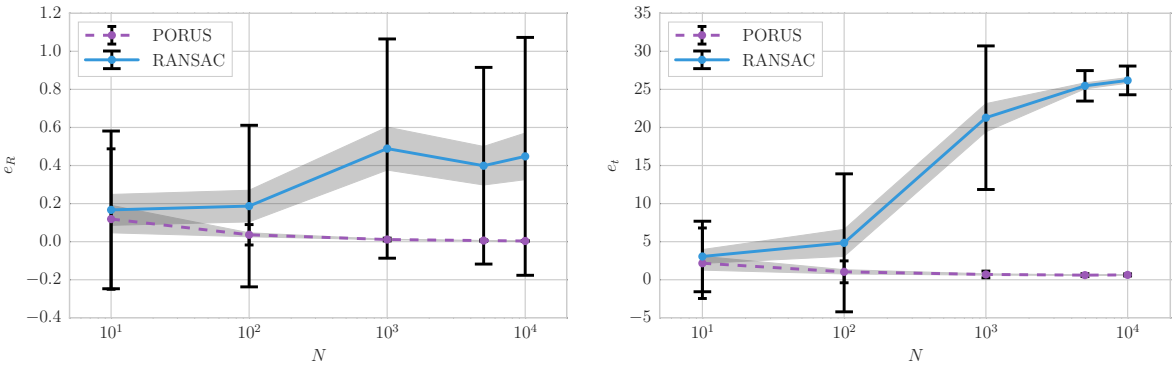


Figure 7.8: Execution time as a function of N for an inlier ratio of $\epsilon = 0.5$ and noise level of $\sigma = 1.0\,\text{px}$.

increases. Importantly, it can be seen that the execution time of PORUS is significantly shorter (by a factor of 10 as predicted by Figure 7.2b) across all values of N.

Two effects of varying the number of matches on the accuracy of estimated transformation parameters should be noted. First, from Figure 7.9 it should be clear that the standard deviations of e_R^{PORUS} and e_t^{PORUS} are large for small values of N. This is a direct result of Figure 7.7 where it was shown that α_{PORUS} had high uncertainty. Importantly, as N is increased the mean value of e_R^{PORUS} and e_t^{PORUS} both tend to zero.

From the results in this section, it can be concluded that the only major effect of varying the number of matches is related to the execution time of the two algorithms. Furthermore, it is shown that PORUS outperforms RANSAC across all values of N. This set of experiments concludes the direct comparison of RANSAC and PORUS. The focus now shifts to the use of PORUS and RANSAC in a robust visual odometry framework.



(a) (b)

Figure 7.9: Accuracy of pose estimation as a function of N for an inlier ratio of $\epsilon = 0.5$ and noise level of $\sigma = 1.0$ px. (a) Rotation error as a function of N. (b) Translation error as a function of N.

7.2 Simulated Visual Odometry

PORUS and RANSAC are now implemented in separate robust visual odometry frameworks, defined in Section 5.1, and used to estimate trajectories from synthetic datasets. In particular, PORUS and RANSAC are compared in terms of their execution time and accuracy of reconstructed trajectories.

Random robot trajectories consisting of 50 robot poses, are generated using the motion model described in Section 5.4. A total of 2000, 3D landmarks are randomly generated and projected to 2D stereo image coordinates across the 50 poses of the generated ground truth trajectory. It should be noted that only realisable measurements are considered - that is, landmarks that fall outside the field of view of the simulated cameras are disregarded. Furthermore, measurements occurring further than 30 m from the robot are ignored. Feature measurements are formed by contaminating projected 2D features with additive measurement noise corresponding to a given noise level, σ. Furthermore, outliers are injected into the set of 2D image measurements according to a given inlier ratio, ϵ. Outlier removal is then performed on the synthetic dataset and identified inliers are used to estimate the relative transformation parameters between viewpoints. The estimated transformations are concatenated to reconstruct the trajectory of the robot.

Synthetic datasets are generated for several combinations of inlier ratio and noise level. Four measures of comparison are used for this set of experiments, where α, β are as defined in Section 7.1.1 and t_{pose} is the time taken to determine the relative pose of the robot at each time step. An additional measure, M_{good}, represents the number of accurate relative transformation estimates obtained per trajectory. The concept of an *accurate* estimate was introduced in Section 4.1.2 and is based on the work Tordoff and Murray [107], where a "good model estimate" is defined as a model which is consistent with at least 75% of the inlier observations. The parameters η, Δ_T and Δ_S are kept at their nominal values defined in Section 7.1.1. Each experiment consists of 100 independent simulations and the mean results are shown in Table 7.2. Two examples of reconstructed trajectories obtained after performing PORUS and RANSAC are shown in Figure 7.10 alongside the results of a visual odometry pipeline that does not perform outlier removal.

Several conclusions can be drawn from these results. First, it can be seen from Table 7.2 that PORUS results in reduced execution times across all experiments and a significant improvement in execution time (factor of five) is seen when the inlier ratio is equal to $\epsilon = 0.5$, as predicted by the results of the previous section. Furthermore, it can be seen that $\bar{\alpha}_{\text{PORUS}}$ is greater than $\bar{\alpha}_{\text{RANSAC}}$ across all experiments, while $\bar{\beta}_{\text{PORUS}}$ and $\bar{\beta}_{\text{RANSAC}}$ are equivalent. Once again, this is consistent with the results of Section 7.1. Interestingly, when no measurement noise is present, $\bar{\alpha}_{\text{RANSAC}}$ is effectively equal to the probability of success, η, and then drops significantly as the level of noise is increased. This is a direct consequence of the fact that a hypothesised model instantiated on a minimal set will not be consistent with all inliers if

Table 7.2: Synthetic visual odometry results.

		RANSAC				PORUS			
σ (px)	ϵ	$\bar{\alpha}$	$\bar{\beta}$	$\bar{M}_{\text{good}}$	$\bar{t}_{\text{pose}}$ (s)	$\bar{\alpha}$	$\bar{\beta}$	$\bar{M}_{\text{good}}$	$\bar{t}_{\text{pose}}$ (s)
0.0	0.5	0.96	0.00	46.72	0.103	1.00	0.00	49.00	0.019
0.0	0.8	0.93	0.00	45.95	0.021	1.00	0.00	49.00	0.016
0.1	0.5	0.76	0.002	31.86	0.089	0.98	0.00	48.98	0.016
0.1	0.8	0.74	0.001	29.01	0.022	0.99	0.00	48.99	0.016
0.5	0.5	0.75	0.005	30.98	0.095	0.92	0.006	48.74	0.017
0.5	0.8	0.76	0.005	29.39	0.021	0.91	0.003	48.93	0.015
1.0	0.5	0.79	0.011	33.04	0.098	0.89	0.009	48.16	0.018
1.0	0.8	0.77	0.005	30.58	0.020	0.87	0.005	48.88	0.015
2.0	0.5	0.82	0.023	35.79	0.086	0.83	0.025	46.29	0.017
2.0	0.8	0.81	0.017	32.41	0.026	0.82	0.013	48.82	0.019

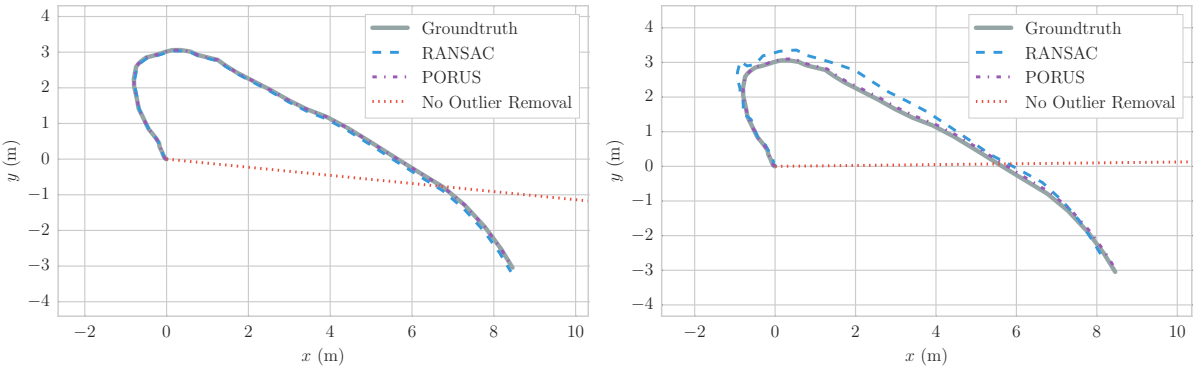


Figure 7.10: Synthetic visual odometry results. Example reconstructed robot trajectories after PORUS, RANSAC and no outlier removal are performed on a synthetic dataset. (a) No measurement noise and an inlier ratio of $\epsilon = 0.8$. (b) Measurement noise of $\sigma = 0.5$ px and an inlier ratio of $\epsilon = 0.8$.

measurement noise is present. A second significant result is that the estimated transformation parameters obtained by PORUS are more accurate. Comparing the values of $\bar{M}_{\text{good}}^{\text{PORUS}}$ and $\bar{M}_{\text{good}}^{\text{RANSAC}}$, it can be seen that a significantly larger number of pose estimates obtained are consistent with more than 75% of the inliers in the dataset when PORUS is performed. This results in more accurate trajectory estimates as shown in Figure 7.10b

It has been shown that the use of PORUS in a visual odometry framework leads to significantly improved execution times and more accurate trajectory estimates for synthetic datasets. The use of PORUS will now be verified on practical datasets.

7.3 Practical Visual Odometry

In order to verify the results obtained in the previous section, both PORUS and RANSAC are incorporated into a practical visual odometry framework and evaluated using image sequences from a well-known autonomous navigation dataset. A brief overview of the dataset and the visual odometry library used is now provided.

7.3.1 KITTI Image Sequences

Practical visual odometry experiments were performed on a set of image sequences available in the *KITTI Vision Benchmark Suite*[2] [122]. The KITTI benchmark consists of several datasets captured by Geiger et al. [122] while driving a moving platform, shown in Figure 7.11, through urban and rural

regions of Karlsruhe, Germany. Stereo sequences (Figure 7.12) consisting of images with a resolution of

Figure 7.11: AnnieWay Volkswagen Passat used as a recording platform for KITTI. The recording platform is equipped with two sets of stereo cameras (colour and greyscale), a 3D Velodyne laser scanner and an inertial navigation system. Image adapted from *Vision meets Robotics: The KITTI Dataset* [123]

1344×391 px are provided at 10 fps. It should be noted that the stereo sequences consist of dynamic scenes. Additionally, accurate ground truth trajectories, obtained from an inertial navigation system, are provided for each dataset in the benchmark, which allows for easy evaluation of autonomous navigation and computer vision techniques. Technical details regarding the sensor layout are provided by Geiger et al. [123].

7.3.2 LIBVISO2

LIBVISO2 is an open-source visual odometry library developed and maintained by Geiger et al. [44]. The reconstruction pipeline present in LIBVISO2 is homogeneous to the generic visual odometry framework discussed in Section 1.2.1 – feature detection, feature matching and motion estimation – and is capable of performing real-time, 6DoF ego-motion estimation using images captured with a calibrated camera

[2]Datasets are available for download at `www.cvlibs.net/datasets/kitti`.

(a)

(b)

Figure 7.12: Examples images from KITTI dataset. (a) Left frame. (b) Right frame.

system. Two implementations of the framework are available and are designed for stereo and monocular visual odometry applications respectively.

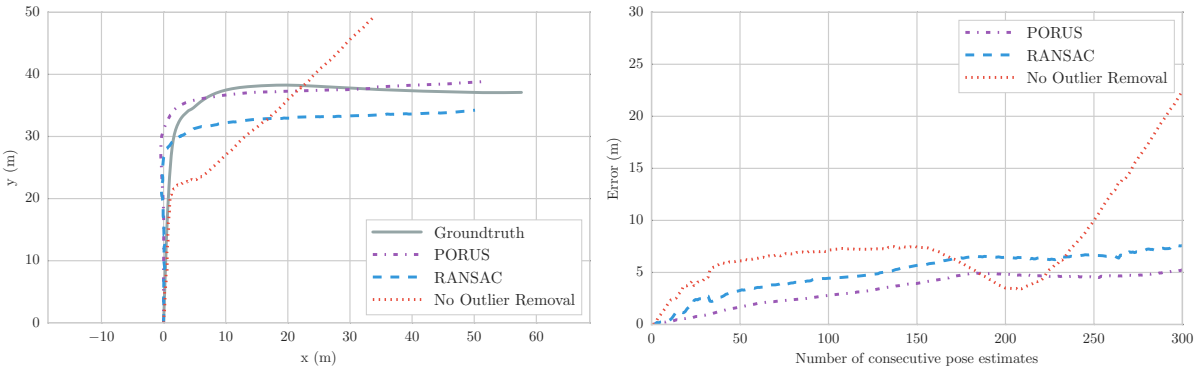
LIBVISO2 is a feature-based approach that makes use of both corner and blob detectors to identify salient points of interest in images. Horizontal and vertical Sobel filters are applied to image frames to identify regions of interest, followed by non-maximum and non-minimum suppression [124] to generate a set of feature candidates for each image. Matched features are determined by searching for correspondences in a circular pattern: starting with the current left image, progressing to the previous left image, previous right image and current right image respectively, before returning to the current left image. Matches are retained if the loop is closed and the first and last feature are the same. The number of features used for motion estimation is reduced by bucketing [125]. Ego-motion estimation is performed by minimising the re-projection error using Gauss-Newton optimisation [126]. It should be noted that LIBVISO2 implements two methods for handling outliers; first, sporadic outliers are removed during the feature matching stage using flow differences [44] and thereafter, their ego-motion stage is performed in a RANSAC scheme – both of these methods were removed to allow for a direct comparison of PORUS and RANSAC.

7.3.3 Experimental Results

Three adapted LIBVISO2 frameworks are now used to perform visual odometry on three image sequences from the *KITTI Vision Benchmark Suite*. The first implementation applies PORUS and a second implementation uses the probabilistic RANSAC approach of Brink et al. [82] to remove outliers before motion estimation is performed. A third implementation implements no outlier removal method and performs motion estimation on all points. The number of RANSAC iterations is kept at $k = 50$, as chosen by Geiger et al. [44], and the parameters, $\sigma = 0.2\,\text{px}$, $\Delta_T = 50.0$ and $\Delta_S = 1.0$ are chosen empirically. A conservative estimate of the inlier ratio, $\epsilon = 0.5$, was chosen. The measures of comparison are the final translation error, e_t^{final}, and the average time taken per pose estimate, $\bar{t}_{\text{pose}}$. The results are summarised in Table 7.3 and Figures 7.13 to 7.15.

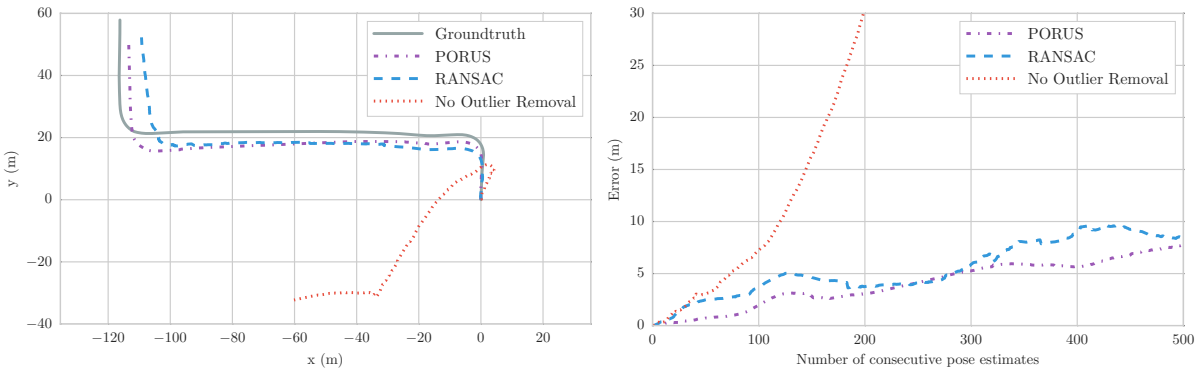
Table 7.3: Practical visual odometry results.

	No Outlier Removal		RANSAC		PORUS	
Kitti Dataset	e_t^{final} (m)	$\bar{t}_{\text{pose}}$ (s)	e_t^{final} (m)	$\bar{t}_{\text{pose}}$ (s)	e_t^{final} (m)	$\bar{t}_{\text{pose}}$ (s)
2009_09_08_drive_0010	22.52	0.025	7.585	0.093	5.266	0.035
2009_09_08_drive_0016	98.11	0.024	8.616	0.094	7.718	0.036
2009_09_08_drive_0021	45.11	0.031	25.03	0.106	10.652	0.042



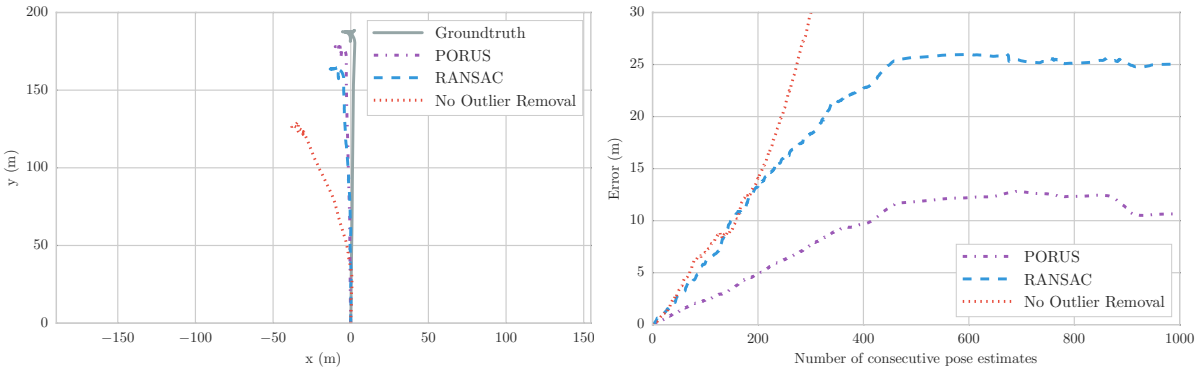
(a) **(b)**

Figure 7.13: KITTI 2009_09_08_drive_0010 (a) Reconstructed trajectories. (b) Euclidean distance error of estimated trajectories as a function of the number of pose estimates.



(a) (b)

Figure 7.14: KITTI 2009_09_08_drive_0016 (a) Reconstructed trajectories. (b) Euclidean distance error of estimated trajectories as a function of the number of pose estimates.



(a) (b)

Figure 7.15: KITTI 2009_09_08_drive_0021 (a) Reconstructed trajectories. (b) Euclidean distance error of estimated trajectories as a function of the number of pose estimates.

From these results it can be seen that PORUS significantly outperforms RANSAC when used for outlier removal in a practical visual odometry framework. For the three datasets tested, the use of PORUS resulted in more accurate trajectory reconstructions. Although drift still occurred, the errors incurred are notably less than the errors from the RANSAC implementation of LIBVISO2. Furthermore, the time required to perform pose estimation using the PORUS implementation is significantly less than the RANSAC implementation as seen in Table 7.3. The timing results shown here indicate that PORUS would allow visual odometry to be performed at 30 fps, which is a significant advantage over the 10 fps obtained by the RANSAC implementation.

7.4 Chapter Summary

In this chapter, PORUS and RANSAC were evaluated across a range of experiments. This includes experiments designed to compare the efficiency and robustness of the two algorithms as well as application experiments where PORUS and RANSAC are incorporated into robust visual odometry frameworks, which are then tested on synthetic and practical datasets.

In the first set of experiments, it was shown that PORUS is more adept at handling different inlier ratios than RANSAC. A significant result was obtained where the execution time of PORUS was shown to be independent of the inlier ratio. Consequently, PORUS was able to perform outlier removal significantly faster for highly contaminated synthetic datasets. Furthermore, it was also shown that PORUS has a higher true positive rate than RANSAC across all inlier ratios. Additional results indicated that PORUS is sensitive to the choice of shape similarity threshold and that an increase in noise level affects both

the probability of false positives as well as the probability of rejecting inliers when performing PORUS. However, PORUS still consistently outperformed RANSAC under noisy conditions. Lastly, it was shown that an increase in the number of data points leads to increased execution times of both algorithms. Importantly, even with very large datasets, PORUS consistently outperformed RANSAC in terms of accuracy and efficiency.

The results of the simulated visual odometry experiments indicated that PORUS is significantly more efficient than RANSAC, resulting in pose estimation being performed five times faster. Furthermore, it was shown that the estimated transformation parameters obtained, after performing PORUS, are more accurate. These results were verified with a practical visual odometry framework where significant improvements in execution time and accuracy were seen when using PORUS over RANSAC.

Chapter 8

Conclusion

This final chapter aims to provide a high level overview of the work performed in this thesis. The chapter begins with a summary of the methods contained in the various chapters as well as brief comments on significant results. Thereafter, an overview of the contributions of this research is provided. The thesis concludes with a discussion of possible future work.

8.1 Summary

This thesis details the development of a novel outlier removal approach, PORUS (Probabilistic Outlier Removal Using Shapes), aimed at vision-based autonomous navigation applications. Specifically, the problem of robust motion estimation in the presence of highly uncertain measurements and large numbers of outliers is investigated. PORUS, proposed as an alternative to the commonly used approach of RANSAC, is implemented in conjunction with a robust visual odometry framework. Generic 6DoF motion is assumed and measurement noise in image coordinates is modelled by Gaussian distributions. A brief summary of the methods and results presented in this thesis is provided below.

Chapter 1 The opening chapter of this thesis introduces the basics of autonomous navigation where a generalised framework is discussed. Accurate localisation is emphasised, and the importance of removing outliers for vision-based motion estimation is made clear. Furthermore, from an investigation of existing outlier removal methods, it is evident that difficulties exist with current techniques.

Part I: The first part of this thesis focuses on the modelling of environmental sensors – in particular, that of a stereo camera pair. Chapter 2 introduces single and multiple view geometry based on a pinhole camera model, and it is shown how the location of 3D features is determined from sets of matched stereo features. In Chapter 3, the uncertainty of stereo measurements is considered where measured image features and processed measurements are modelled as normally distributed random variables. Two techniques used for uncertainty propagation, linearisation and the unscented transform, are investigated and evaluated for approximating the distributions of reconstructed 3D features. The results of several experiments performed on synthetic datasets are given in Chapter 3 – these results show that the unscented transform significantly outperforms linearisation when measurements are highly uncertain.

Part II: In the second part of this thesis, the focus shifts to outlier removal in the context of visual odometry. In Chapter 4, two classes of outlier removal are identified, namely, hypothesise-and-verify based approaches such as RANSAC, and deterministic techniques based on geometric constraints. Several limitations of RANSAC are discussed before improved versions of the algorithm are detailed. A probabilistic RANSAC method is adapted and combined with a robust visual odometry in Chapter 5. Results performed on synthetic visual odometry datasets indicate that the transformed distributions obtained from linearisation may contain significant approximation errors, leading to unreliable outlier removal and incorrect motion estimates. This further supports the results of Chapter 3.

A novel outlier removal approach, PORUS, is developed in Chapter 6 that attempts to address the limitations of the standard RANSAC algorithm. PORUS makes use of inherent shape information to distinguish between inliers and outliers. In a similar fashion to RANSAC, PORUS makes use of randomly sampled sets of observations; however, it differs in the sense that models are not fit to sampled points and verified. Instead, the probability of points being inliers are determined directly based on spatial constraints. In addition, a probabilistic measure of shape similarity is developed for visual odometry applications, as well as an adaptive sampling technique based on decision theory concepts. Results of experiments performed in Chapter 6 show that PORUS is a viable approach for identifying contaminated sets of points, and that the proposed adaptive sampling approach leads to increased efficiency at high inlier ratios. In Chapter 7, PORUS is compared against the probabilistic RANSAC method of Chapter 5. Synthetic datasets are used to evaluate both techniques in terms of efficiency, as well as robustness to noise and high contamination levels. Results indicate that PORUS is significantly more efficient at low inlier ratios and consistently identifies more inliers. Furthermore, estimated transformation parameters obtained from RANSAC are shown to be more inconsistent than those obtained from PORUS. The viability of PORUS was further verified by performing robust motion estimation on synthetic and practical visual odometry datasets. PORUS is shown to obtain more consistent trajectory reconstructions, with significantly reduced execution times, for both simulated and practical experiments.

8.2 Contributions

A summary of the main contributions presented in this thesis is given below.

1. The primary contribution of this thesis is the design and analysis of PORUS, a novel outlier removal technique for vision-based autonomous navigation systems. The advantages of PORUS over RANSAC are twofold. First, the mechanism of PORUS directly addresses one of the limitations of RANSAC, namely, the fact that a large portion of execution time is spent verifying erroneous models. In PORUS, hypothesised models are not generated, and instead the probability of points being inliers is evaluated directly. This leads to significantly lower execution times when datasets are highly contaminated with outliers. Secondly, PORUS is shown to be more robust to measurement noise by identifying inliers more consistently than RANSAC. Furthermore, PORUS has been developed for generalised 6DoF motion. This makes PORUS viable for a wide range of vision-based autonomous navigation applications.

2. A secondary contribution of this thesis was a comparison of the unscented transform and linearisation for uncertainty propagation. Several studies from literature have compared the unscented transform and linearisation without agreement with regards to which technique is universally better. These studies generally compare linearisation and the unscented transform in terms of their performance in an EKF/UKF framework. However, to the best of the author's knowledge, no conclusive work has been done on the accuracy of these approximation techniques when applied to sensor uncertainty propagation in computer vision applications. These findings were presented at the 2016 PRASA-RobMech International Conference and have been accepted for publication in the conference proceedings [127].

3. The similarity measure of Brink et al. [82] was simplified and extended to that of a probabilistic distance. This form is advantageous for several reasons. First, the new form requires fewer matrix calculations making it cheaper to compute and secondly, a negative log likelihood form is more numerically stable for small probabilities.

This thesis therefore constitutes a significant and novel contribution to the field of robotics.

8.3 Future Work

Several avenues for future work are now briefly discussed.

1. A particle approximation was not investigated for uncertainty propagation, which is likely to outperform both linearisation and the unscented transform in terms of accuracy. The computational complexity of such an approach would need to be investigated.

2. There are several possible improvements which could be investigated for the proposed adaptive sampling approach. First, a Monte Carlo sampling approach, similar to that of PROSAC [72], could be used instead of uniform sampling. This requires that the individual probabilities of each point be maintained across sampling instances as done by Tordoff and Murray [107]. Secondly, only a linear sampling strategy and a greedy sampling strategy were investigated. There are other strategies, such as sampling a single uncertain point and two inliers for the 3D case, which may result in a more efficient adaptive sampling approach.

3. The effect of dynamic objects in the environment was not investigated. This would result in multiple rigid body transformations with different transformation parameters for each object. It might be possible to identify multiple objects by using the measure of shape similarity.

4. A different parametrisation of triangle shapes could also be investigated.

Appendix A

A.1 Confidence Intervals

The parameters of a model are often estimated from observed samples. It is therefore useful to introduce the concept of confidence intervals – that is, a range of values which is likely to contain the true value of an unobservable parameter with some probability (known as the confidence level).

Consider the problem of estimating the mean of an unknown Gaussian distribution,

$$X \sim \mathcal{N}(\mu, \sigma), \tag{A.1}$$

from N sampled observations, $(x_1, \ldots, x_N)^\top$. The true mean, μ, is not observed and is approximated using the sample mean,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{A.2}$$

with an estimated standard deviation given by

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}. \tag{A.3}$$

Since the samples are not drawn deterministically, the sample mean itself is not in fact deterministic and can be modelled as a normal distribution,

$$\bar{x} \sim \mathcal{N}(\mu, \sigma_{\bar{x}}), \tag{A.4}$$

where $\sigma_{\bar{x}} = \frac{s}{\sqrt{N}}$ is the standard error. The confidence interval is then simply given by

$$\left(\bar{x} - z^* \sigma_{\bar{x}}, \bar{x} + z^* \sigma_{\bar{x}}\right), \tag{A.5}$$

where z^* is the z-value corresponding to the confidence level (CL). Commonly used CLs and their corresponding z-values are shown in Table A.1. For $z^* = 1.96$, it can be said that the true value of μ lies in the interval $\bar{x} \pm 1.96\sigma_{\bar{x}}$ with a probability of 95%.

Table A.1: Commonly used confidence levels and corresponding z-values.

CL	z^*
0.90	1.645
0.95	1.960
0.99	2.576

Bibliography

[1] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. von Hundelshausen, O. Pink, C. Frese, and C. Stiller, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[2] "Google Self-Driving Car Project," Google Inc., Tech. Rep. August, 2016.

[3] O. F. Vynakov, E. V. Savolova, and A. I. Skrynnyk, "Modern electric cars of Tesla Motors Company," *Automation Technological and Business-processes*, vol. 8, no. 2, pp. 9–18, 2016.

[4] C. E. van Daalen, "Conflict detection and resolution for autonomous vehicles," Ph.D. Thesis, Stellenbosch University, 2010.

[5] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE, 2004, pp. 652–659.

[6] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2008, pp. 3946–3952.

[7] J. Kelly, S. Saripalli, and G. S. Sukhatme, "Combined visual and inertial navigation for an unmanned aerial vehicle," *Springer Tracts in Advanced Robotics*, vol. 42, pp. 255–264, 2008.

[8] J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez, and M. Olivares, "Visual 3-D SLAM from UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 55, no. 4-5, pp. 299–321, 2009.

[9] C. Kerl, "Odometry from RGB-D Cameras for autonomous quadrocopters," Master's Thesis, Tecnhnical University of Mumich, 2012.

[10] D. Wettergreen, C. Gaskett, and A. Zelinsky, "Development of a visually-guided autonomous underwater vehicle," in *IEEE Oceanic Engineering Society Conference Proceedings*, vol. 2. IEEE, 1998, pp. 1200–1204.

[11] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, 2012.

[12] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer Handbook of Robotics.* Springer Berlin Heidelberg, 2008, vol. 23, no. 7-8, pp. 871–889.

[13] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[14] A. Petrovskaya, M. Perrollaz, L. Oliveira, L. Spinello, R. Triebel, A. Makris, J.-D. Yoder, C. Laugier, U. Nunes, and P. Bessiere, "Awareness of road scene participants for autonomous driving," in *Handbook of Intelligent Vehicles*. Springer London, 2012, pp. 1383–1432.

[15] L. Montesano, "Detecting and tracking moving objects from a mobile platform using a laser range scanner," Ph.D. Thesis, Universidad de Zaragoza, 2006.

[16] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, vol. 53, pp. 263–296, 2008.

[17] F. Chenavier and J. Crowley, "Position estimation for a mobile robot using vision and odometry," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*. IEEE Computing Society Press, 1992, pp. 2588–2593.

[18] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 328–342, 1995.

[19] S. Clark and H. Durrant-Whyte, "Autonomous land vehicle navigation using millimeter wave radar," *Proceedings 1998 IEEE International Conference on Robotics and Automation*, vol. 4, no. May, pp. 3697–3702, 1998.

[20] J. Guivant, E. Nebot, and S. Baiker, "Autonomous navigation and map building using laser range sensors in outdoor applications," *Journal of Robotics Systems*, vol. 17, no. 10, pp. 565–583, 2000.

[21] S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 572–578, 1999.

[22] W. Brink, C. E. Van Daalen, and W. Brink, "Stereo vision as a sensor for EKF SLAM," in *22nd Annual Symposium of the Pattern Recognition Association of South Africa*, 2011, pp. 19–24.

[23] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. Munich, "The vSLAM algorithm for robust localization and mapping," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, vol. 2005. IEEE, 2005, pp. 24–29.

[24] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[25] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 29, no. 6, pp. 1052–1067, 2007.

[26] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[27] H. Lategahn, A. Geiger, and B. Kitt, "Visual SLAM for autonomous ground vehicles," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1732–1737.

[28] A. A. S. Souza, R. Maia, and L. M. G. Gonalves, "3D Probabilistic occupancy grid to robotic mapping with stereo vision," in *Current Advancements in Stereo Vision*. InTech, 2012, pp. 181–198.

[29] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proceedings of 8th National Conference on Artificial Intelligent/14th Conference on Innovative Applications of Artificial Intelligent*, vol. 68, no. 2, pp. 593–598, 2002.

[30] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2007, pp. 1–10.

[31] K. Celik, Soon-Jo Chung, and A. Somani, "Mono-vision corner SLAM for indoor navigation," in *2008 IEEE International Conference on Electro/Information Technology*. IEEE, 2008, pp. 343–348.

[32] L. Shao, J. Han, D. Xu, and J. Shotton, "Computer vision for RGB-D sensors: Kinect and its applications," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1314–1317, 2013.

[33] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 127–136.

[34] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D Mapping: Using depth cameras for dense 3D modeling of indoor environments," in *Springer Tracts in Advanced Robotics*, 2014, vol. 79, pp. 477–491.

[35] H. Dong, N. Figueroa, and A. El Saddik, "Towards consistent reconstructions of indoor spaces based on 6D RGB-D odometry and KinectFusion," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1796–1803.

[36] D. Scaramuzza and F. Fraundorfer, "Visual odometry," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[37] L. Clemente, A. Davison, I. Reid, J. Neira, and D. J. Tardős, "Mapping large loops with a single hand-held camera," in *Proceedings Robotics: Science and Systems Conference*, 2007, pp. 297–304.

[38] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A system for large-scale mapping in constant-time using stereo," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, 2011.

[39] C. Taylor and D. Kriegman, "Structure and motion from line segments in multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 17, no. 11, pp. 1021–1032, 1995.

[40] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover." Carnegie-Mellon University, Tech. Rep., 1980.

[41] L. H. Matthies, "Dynamic stereo vision," Ph.D. Thesis, Carnegie Mellon University, 1989.

[42] A. Milella and R. Siegwart, "Stereo-based ego-motion estimation using pixel tracking and iterative closest point," in *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*. IEEE, 2006, pp. 21–21.

[43] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 486–492.

[44] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 963–968.

[45] Yang Cheng, M. Maimone, and L. Matthies, "Visual Odometry on the Mars Exploration Rovers," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1. IEEE, 2005, pp. 903–910.

[46] D. Helmick, Yang Cheng, and S. Roumeliotis, "Path following using visual odometry for a Mars rover in high-slip environments," in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, vol. 2. IEEE, 2009, pp. 772–789.

[47] V. Guizilini and F. Ramos, "Visual odometry learning for unmanned aerial vehicles," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 6213–6220.

[48] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 15–22.

[49] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2007.

[50] Z. Zheng, H. Wang, and E. Khwang Teoh, "Analysis of gray level corner detection," *Pattern Recognition Letters*, vol. 20, no. 2, pp. 149–162, 1999.

[51] W. Brink, "Stereo vision for simultaneous localization and mapping," MSc Thesis, Stellenbosch University, 2012.

[52] D. Scaramuzza and F. Fraundorfer, "Visual odometry : Part II," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[53] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*. Alvey Vision Club, 1988, pp. 147–151.

[54] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Lecture Notes in Computer Science*, 2006, vol. 3951 LNCS, pp. 430–443.

[55] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, vol. 178, no. December. IEEE Comput. Society Press, 1994, pp. 593–600.

[56] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, no. 8. IEEE, 1999, pp. 1150–1157 vol.2.

[57] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[58] N. Govender, "Evaluation of feature detection algorithms for structure from motion," in *3rd Robotics and Mechatronics Symposium (ROBMECH 2009)*, 2009, p. 4.

[59] A. Schmidt, M. Kraft, and A. Kasiński, "An evaluation of image feature detectors and descriptors for robot navigation," in *Computer Vision and Graphics*, 2010, vol. 32, no. February, pp. 251–259.

[60] C. Tomasi, "Detection and tracking of point features," Carnegie Mellon University, Tech. Rep. April, 1991.

[61] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1403–1410 vol.2.

[62] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[63] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.

[64] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 26, no. 6, pp. 756–770, 2004.

[65] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[66] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, p. 629, 1987.

[67] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 13, no. 4, pp. 376–380, 1991.

[68] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[69] D. M. Hawkins, *Identification of Outliers*. Dordrecht: Springer Netherlands, 1980.

[70] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.

[71] O. Chum, "Two-view geometry estimation by random sample and consensus," Ph.D. Thesis, Czech Technical University, 2005.

[72] O. Chum and J. Matas, "Matching with PROSAC — Progressive sample consensus," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 220–226.

[73] C. V. Stewart, "Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, no. 3, pp. 513–537, 1999.

[74] M. Lhuillier, "Automatic scene structure and camera motion using a catadioptric system," *Computer Vision and Image Understanding*, vol. 109, no. 2, pp. 186–203, 2008.

[75] D. Scaramuzza, "Performance evaluation of 1-point-RANSAC visual odometry," *Journal of Field Robotics*, vol. 28, no. 5, pp. 792–811, 2011.

[76] O. Naroditsky, X. S. Zhou, J. Gallier, S. I. Roumeliotis, and K. Daniilidis, "Two efficient solutions for visual odometry using directional correspondence," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 34, no. 4, pp. 818–824, 2012.

[77] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the Mars Exploration Rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[78] F. Bellavia, M. Fanfani, F. Pazzaglia, and C. Colombo, "Robust selective stereo SLAM without loop closure and bundle adjustment," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligent and Lecture Notes in Bioinformatics)*, 2013, vol. 8156 LNCS, no. PART 1, pp. 462–471.

[79] R. Raguram, J. M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *European Conference on Computer Vision*. Berlin, Heidelberg: Springer, 2008, vol. 5303 LNCS, no. PART 2, pp. 500–513.

[80] H. Badino and T. Kanade, "A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion," in *12th IAPR Conference on Machine Vision Applications*, 2011, pp. 185–189.

[81] H. Hirschmuller, P. Innocent, and J. Garibaldi, "Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics," *7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002.*, vol. 2, no. December, pp. 1099–1104, 2002.

[82] W. Brink, C. E. van Daalen, and W. Brink, "Probabilistic outlier removal for robust landmark identification in stereo vision based SLAM," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2822–2827.

[83] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 22, no. 11, pp. 1330–1334, 2000.

[84] G. Bradiski and A. Kaehler, *Learning OpenCV – Computer Vision with the OpenCV Library*, First, Ed. O'Reilly Media, Inc., 2008.

[85] D. Brown, "Decentering distortion of lenses," *Photometric Engineering*, vol. 32, no. 3, pp. 444–462, 1966.

[86] W. Brink, D. Joubert, and F. Singels, "Dense stereo correspondence for uncalibrated images in multiple view reconstruction," in *21st Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, 2010, pp. 39–44.

[87] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960.

[88] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *AeroSense: 11th International Symp. Aerosp./Defence Sens., Simulation and Controls*, 1997, pp. 182–193.

[89] P. Z. Peebles Jr., *Probability, random variable, and random signal principles (4th ed.)*, ser. Communications and signal processing. McGraw-Hill, 2001.

[90] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2nd ed. MIT Press, 2005.

[91] B. Ochoa and S. Belongie, "Covariance propagation for guided matching," in *Workshop Statist. Methods Multi-Image and Video Process.*, 2006, pp. 1–12.

[92] J. Nieto, T. Bailey, and E. , "Scan-SLAM: Combining EKF-SLAM and scan correlation," in *Field and Service Robotics*. Berlin/Heidelberg: Springer-Verlag, 2006, vol. 25, pp. 167–178.

[93] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.

[94] E. Ziegel, W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1987.

[95] R. Raguram, J.-M. Frahm, and M. Pollefeys, "Exploiting uncertainty in random sample consensus," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 2074–2081.

[96] T. Thormählen, H. Broszio, and A. Weissenfeld, "Keyframe selection for camera motion and structure estimation from multiple views," in *Eur. Conference Comput. Vision*, 2004, pp. 523–535.

[97] D. G. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 113–122, 1992.

[98] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[99] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[100] R. van der Merwe and E. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 6. IEEE, 2001, pp. 3461–3464.

[101] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Ann. of Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[102] Q. Wang, S. Kulkarni, and S. Verdu, "A nearest-neighbor approach to estimating divergence between continuous random vectors," in *2006 IEEE International Symposium on Information Theory*, no. 5. IEEE, 2006, pp. 242–246.

[103] F. Pérez-Cruz, "Kullback-Leibler divergence estimation of continuous distributions," in *IEEE International Symp. Inf. Theory*, 2008, pp. 1666–1670.

[104] N. Leonenko, L. Pronzato, and V. Savani, "A class of Rényi information estimators for multidimensional densities," *The Annals of Statistics*, vol. 36, no. 5, pp. 2153–2182, 2008.

[105] J. Duchi, "Derivations for linear algebra and optimization," Stanford, pp. 1–13, 2014.

[106] A. Gelman, "Method of moments using Monte Carlo simulation," *J. Comput. and Graph. Statist.*, vol. 4, no. 1, pp. 36–54, 1995.

[107] B. Tordoff and D. W. Murray, "Guided sampling and consensus for motion estimation," in *Computer Vision ECCV 2002*, 2002, no. May, pp. 82–96.

[108] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[109] A. A. Dempster, N. N. Laird, and D. D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society Series B Methodological*, vol. 39, no. 1, pp. 1–38, 1977.

[110] G. Dubbelman, W. Mark, and F. Groen, "Accurate and robust ego-motion estimation using expectation maximization," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2. IEEE, 2008, pp. 3914–3920.

[111] J. Matas and O. Chum, "Randomized RANSAC with T_{dd} test," in *Brit. Mach. Vision Conference*, 2002, pp. 448–457.

[112] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 30, no. 8, pp. 1472–1482, 2008.

[113] A. Wald, *Sequential analysis*. Courier Corporation, 1973.

[114] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *Proceedings of the DAGM*, 2003, vol. 2781, pp. 236–243.

[115] D. Nistér, "Preemptive RANSAC for live structure and motion estimation," *Mach. Vision and Appl.*, vol. 16, no. 5, pp. 321–329, 2003.

[116] K. B. Pedersen and M. S. Petersen, "The Matrix Cookbook," p. 72, 2012.

[117] B. Herbst, J. du Preez, and S. Kroon, "Machine Learning," p. 142, 2013.

[118] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *The Indian Journal of Statistics*, vol. 7, no. 4, pp. 401–406, 1946.

[119] O. Bottema and B. Roth, *Theoretical Kinematics*. Dover Publications Inc, 1979.

[120] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2013.

[121] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge: Cambridge University Press, 2011.

[122] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[123] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[124] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 850–855.

[125] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligent*, vol. 78, no. 1-2, pp. 87–119, 1995.

[126] W. Gander, M. J. Gander, and F. Kwok, *Scientific Computing - An Introduction using Maple and MATLAB*, ser. Texts in Computational Science and Engineering. Cham: Springer International Publishing, 2014, vol. 11.

[127] A. Chiu, T. Jones, and C. E. van Daalen, "A comparison of linearisation and the unscented transform for computer vision applications," in *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*. IEEE, 2016, pp. 60–66.