# AN ANALYSIS OF 3D SURFACE CURVATURE FEATURES FOR 3D SLAM USING KINECT DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖMER FARUK ADİL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2014

Approval of the thesis:

**AN ANALYSIS OF 3D SURFACE CURVATURE FEATURES FOR 3D SLAM USING KINECT DATA**

submitted by **ÖMER FARUK ADİL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan ________________
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. İlkay Ulusoy Parnas ________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

**Examining Committee Members:**

Prof. Dr. Kemal Leblebicioğlu ______________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. İlkay Ulusoy Parnas ______________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gözde Bozdağı Akar ______________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Afşar Saranlı ______________________
Electrical and Electronics Engineering Dept., METU

Alper Erdener, M.Sc. ______________________
Unmanned Systems Programs Director, ASELSAN

**Date:** ______________________

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name : Ömer Faruk ADİL

Signature :

# ABSTRACT

## AN ANALYSIS OF 3D SURFACE CURVATURE FEATURES FOR 3D SLAM USING KINECT DATA

Adil, Ömer Faruk
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. İlkay Ulusoy Parnas
December 2014, 125 pages

The introduction of affordable and sufficiently accurate range sensors such as TOF cameras, laser scanners and RGBD cameras has significantly contributed to the robotic applications like SLAM. Although range sensors are used extensively and successfully in SLAM applications, there is still room for improvement in the sensor data utilization process. In this thesis a novel approach for the feature extraction part of 3D SLAM is introduced. The use of compact surface curvature features in the SLAM algorithm is proposed which will appear for the first time in the literature. The proposed method uses mean and Gaussian curvature calculations to extract curvedness features from RGBD output of Microsoft Kinect sensor. The extracted features are then fed to the SLAM algorithm to be used in the global data association process. The results are compared to the state-of-the art feature extraction techniques for SLAM, namely plane features, SURF features and corner features on real Kinect dataset sequences which are conventionally used as benchmark for SLAM algorithms.

Keywords: SLAM, Kinect Sensor, RGBD Data, Salient Feature Extraction, Surface Curvature Features, Compact Surface Features

# ÖZ

## KINECT VERİSİ İLE 3-BOYUTLU SLAM UYGULAMASINDA 3-BOYUTLU YUZEY KAVİS ÖZELLİKLERİ KULLANIMI ANALİZİ

Adil, Ömer Faruk
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. İlkay Ulusoy Parnas
Aralık 2014, 125 sayfa

RGBD kameralar, TOF kameralar, lazer tarayıcılar gibi düşük maliyetli ve yeterli derecede hassas ölçüm sunabilen sensörlerin ortaya çıkması, SLAM gibi robotik uygulamalar için önemli katkılar sağlamıştır. Bu mesafe sensörleri, mevcut SLAM uygulamalarında yoğun olarak başarıyla kullanılsa da sensor verilerinin kullanımının iyileştirilmesi ve geliştirilmesi hala mümkündür. Bu tezde, 3-boyutlu SLAM uygulamasında kullanılmak üzere sensor verilerinden özellik çıkarımı için yeni bir method sunulmaktadır. SLAM uygulamarında kompakt yüzey kavis özelliklerinin kullanımı literatürde ilk kez bu çalışmada incelenmektedir. Önerilen metotta ortalama kavis ve Gaussian kavis hesaplamaları, Microsoft Kinect sensöründen alınan RGBD verisinden kavis özellikleri çıkarmada kullanılmaktadır. Çıkarılan kavis özellikleri, SLAM algoritmasının veri eşleştirme adımlarında kullanılmaktadır. SLAM uygulamaları için karşılaştırmalarda sıkça kullanılan gerçek Kinect veri kayıtları kullanılarak yapılan uygulamalar sonucunda bulunan sonuçlar, literatürdeki güncel ve kabul gören SURF özellikleri, düzlemsel özellikler ve köşe özellikleri gibi özellik çıkarma teknikleri ile karşılaştırılmaktadır.

Anahtar Kelimeler: SLAM, Kinect Sensörü, RGBD Verisi, Özellik Çıkarımı, Yüzey Kavis Özellikleri, Kompakt Yüzey Özellikler,

*To my wife…*

# ACKNOWLEDGEMENTS

I am grateful for my Advisor Assoc. Prof. Dr. İlkay Ulusoy for her valuable support, vision, knowledge and the will for innovation.

I would like to thank all of my previous instructors for guiding me towards the unending curiosity and search for the new.

I am thankful to Aselsan for the support and cooperation during my graduate education.

I am indebted to my family for the love, respect and selflessness to this day.

I can never be thankful enough to God for bestowing me such a perfect wife, Zehra Adil, whom I owe infinitely much and love eternally.

# TABLE OF CONTENTS

# LIST OF TABLES

**TABLES**

# LIST OF FIGURES

**FIGURES**

# LIST OF ABBREVIATIONS

| | |
|---|---|
| SLAM | Simultaneous Localization and Mapping |
| RGBD | RGB + Depth Data |
| EKF | Extended Kalman Filter |
| FastSLAM | A Factored Solution to SLAM Problem |
| TOF | Time-of-Flight |
| ICP | Iterative Closest Points |
| DoF | Degree of Freedom |
| SIFT | Scale-Invariant Feature Transform |
| SURF | Speeded-up Robust Features |
| RANSAC | Random Sample Consensus |

# CHAPTER 1

# INTRODUCTION

## 1.1 Problem Definition and Motivation

Autonomous agents have long been taking a vital role in many fields in today's world. Modern industry is a product of autonomous agents' hard work. It is natural to expect that the daily life of people would get its fair share of the help of autonomous agents. Driverless cars [1] have been running down the streets, they have been racing against other driverless cars on a rally championship [2]. A robot welcomes and guides guests in a technology museum. Future life is likely to give bigger roles to those "machines with minds" [3].

In order to assign heavier duties to autonomous agents, it is imperative that they should have already gained fundamental abilities. One of the most basic abilities for any autonomous operation is to navigate through unknown environments for the sake of real world compatibility. Although this sounds trivial for an intelligent-born human, it is a rather tedious task for a man-made agent. Aware of this situation, a vast amount of work is devoted for the problem of knowing the whereabouts for an autonomous robot. This problem consists of knowing the neighborhood information and its own location with respect to them. Although the problems of a robot localizing itself and mapping its neighborhood are relatively easier tasks, achieving both of them at the same time is a serious challenge. This is sometimes expressed as the proverbial “chicken and egg” problem [4]. However, this problem is addressed quite well so far under the name of Simultaneous Localization and Mapping (SLAM).

Although the SLAM can be considered as a solved problem [5], there is plenty room for SLAM methods' maturity, efficiency and success. Thanks to the SLAM techniques being quite complicated, there are a number of sub-sections still to be progressed for the better. The mathematics behind the SLAM techniques can be said to be well settled and proven to be practical. On the other hand, sensor data management and processing techniques are still a work in progress. This is partly due to the evolution of sensor technologies on a regular basis. As the state of the art technologies offer more to autonomous systems, the focus of data processing algorithms could be shifted towards new ideas.


**Figure 1: A Mobile Robot with a Kinect Sensor [6]**

In this thesis, a new idea is introduced in the domain of feature based SLAM. In the literature, SLAM techniques generally adopt point, line and plane features as landmark measures [7] [5]. Those approaches are designed to be well suited for indoor environments and are proven to be thusly so far. There are a number of advantages of such techniques making them attractive for SLAM researchers, however, the techniques proposed in this thesis are never explored before and they could help SLAM algorithms in many ways. In 3D SLAM algorithms, surface features have been used mainly as planar features. We investigate the use of surface curvature features of the 3D environment data. Although this would seem intuitionally straightforward after the plane approaches, the literature has never seen a study a SLAM algorithm with surface curvature metric. Like the existence of solid SLAM algorithms, there is a good amount of powerful algorithms in the surface curvature feature extraction literature. Thus, this study aims to unite those confident techniques and document the results for the first time.

### 1.2 State of the Art

SLAM literature could be categorized based on either the main probabilistic approach adopted or on the sensor data utilization techniques used. We will start with the main SLAM approaches as the literature on those methods could be said to be quite well settled. There are certain *de-facto* approaches which have proven their maturity over time.

The first approach to emerge as a comprehensive solution is the Extended Kalman Filter (EKF) method. EKF provided an analytical solution and yielded efficient algorithms. The roots of this technique go back to the work by [8]. The success of this technique then attracted works such as [9], [10] and [11] to follow. In time, extensions for EKF methods are studied. The first consideration was the quadratic complexity of the algorithm. In [12] this issue is addressed and with a more efficient implementation of EKF, the algorithm was able to operate in real time. Another defect of the EKF algorithm is observed to be its vulnerability to data matching errors. This problem arises due to the Gaussian error assumption during the linearization. Gaussian assumption requires single peak for the probability distribution, however, this cannot be expected to strictly hold. Thus, as the real world data error deviates from the Gaussian assumption, failure to match the observation to the map becomes more and more likely which, in turn, could drive the algorithm towards the cliff of divergence. There are extensions such as [13] to overcome this problem by a multi-hypothesis approach. This approach was able to limit the local maxima problem to some extent. Another inefficiency caused by the linearization error in EKF is addressed by Unscented Kalman Filter (UKF) [14] which introduces the idea of having sample models other than Gaussian making it possible to choose the best model for the corresponding problem. Presently, after years of contributions, EKF stands as a reliable and proven method and a nearly finished article.

Grid-based methods are also early approaches for the SLAM problem the examples being [15], [16], [17], [18]. Mainly, the idea is to have the probability likelihood over the space of all possible states for the robot where the state space is discretized by the use of grids. The existence of such a discretization alone introduces an additional source of error which causes reluctance even from the start. The addition of the fact that this approach requires a much bigger amount of memory on top of the discretization error, those approaches did not receive much attention over time.

Particle filter approaches, however, emerged as a very powerful opponent to EKF algorithms. [19], [20] introduced belief distribution of the robot state over the particles, unlike the distribution over grids as in grid-based methods. Thus, this multi-hypothesis approach was able to eliminate the grid requirement as in grid-based methods which also eliminated the additional discretization errors. Just like grids, the number of particles could be used as a parameter to balance the trade-off between performance and memory. This can be viewed as having the multi-hypothesis advantage of the grid-based methods without having the discretization problems. FastSLAM method was introduced shortly afterwards by [21].The biggest impact of this method was that the algorithm complexity became linear with respect to the number of features used in SLAM algorithm which was quadratic in the case of EKF SLAM. Although this algorithm also eliminates the problem of divergence caused by the instability of singe-hypothesis approach of EKF, FastSLAM has its own nemesis: the problem of diminishing good particles. However, this problem is comparatively less probable as the multi-hypothesis structure is able to recover from getting stuck around false peaks.

The final notable family of SLAM methods is based on scan-matching techniques. The main principle is, in fact, the tracking methodology. This includes the alignment of consecutive sensor measurements and calculation of the state transition based on the minimization of relative transformation. The early example for 2D data is studied by [22]. For 3D data, [23] introduced much renowned Iterative Closest Point (ICP) algorithm. Although not a global solution due to the tracking based approach, SLAM literature witnessed successful implementations of ICP in many 3D SLAM algorithms as in [24].

In the scope of this thesis, we will be more focused on the sensor data representation and utilization part of SLAM. We can examine the SLAM solutions in three main groups in terms of data representations as raw data, grid-based and feature-based representations. Amongst them, as mentioned in [25], feature-based are mostly preferred. There are decisive reasons why feature-based methods are beneficial. Firstly, in raw data and grid based approaches, the amount of data processed causes a problematic computational complexity and memory requirements. Feature-based methods are able to represent the data in a more compact mathematical form which reduces both the time complexity and memory demand. Also, grid-based methods introduce additional error to the system due to discretization. Another advantage of feature-based methods is that the mathematically compact data representation comes in handy for the mapping process. Thus the literature is inclined towards these techniques. This thesis study also adopts a feature-based approach and aims to contribute in the feature utilization part of the SLAM problem. Hence, the following sections of the literature survey are allocated to the feature extraction methods used in the SLAM algorithms.

Based on the sensor data output of the robot, feature extraction methods used in SLAM systems can be investigated under two main categories: 2D features for vision sensors and 3D features for range sensors. Our thesis study uses a feature extraction method that lies beneath the family of 3D feature extraction methods. Thus, 3D feature extraction methods will be reviewed as relevant work in the literature.

[26] is one of the most cited works in the 3D SLAM domain. In this study, RGB and depth data is utilized using planar features of the environment making use of RANSAC (Random Sample Consensus) and ICP (Iterative Closest Points) algorithms. Basically, planar patches in the observed scene are detected by using RANSAC on the extracted point features and then the detected consecutive patches are matched through ICP algorithm. In [27], a similar solution is proposed where SURF is used to extract point features from RGB images and then aligned with other frames by use of RANSAC algorithm. In [25], where a SLAM technique based on planar segment approach, again the scene is decomposed into planar segments via RANSAC algorithm and then the smaller segments are grown into bigger segments by a breadth-first region growing algorithm. It is possible but not necessary to mention many other studies based on planar feature extraction for SLAM. However, the literature does provide an example of surface feature utilization other than plane features in SLAM algorithms. There are 3D surface feature extraction studies like [28] stating that the suggested methods "*would*" apply quite well on SLAM solutions,

nevertheless, no implementation or results could be found in the existing studies. This is the gap that our study aims to fill as a contribution.

### 1.3 Thesis Contribution

This thesis study aims to contribute to the existing feature-based SLAM techniques by introducing the Surface Curvature Feature based SLAM. The use of surface curvature features in SLAM domain is a completely novel approach. As reviewed in the state-of-the-art section, the existing 3D SLAM techniques are mainly interested in planar surfaces and their features. Planar features are useful for they can represent indoor environments and some man-made objects quite efficiently. Also, planar features are computationally efficient to extract which primarily make them a rational choice for robotic applications like SLAM.

However, SLAM algorithms could get further valuable information from the sensor data. The proposed use of surface curvature features provides SLAM algorithm the following benefits:

- Curved surfaces yield more discriminative information compared to planar surfaces. It is more likely to mismatch two different planar segments. The normal vectors are expected to be salient mostly, however, other features like their size could be easily measured different at each appearance in the scene which could cause matching errors. In surface curvature features, it becomes quite hard to have two surfaces having both matching shapes and sizes.
- Planar features are sensitive to occlusion. The size of a planar region is directly affected by the size of any possible occluded region. On the other hand, when an analytical surface is fitted to a surface, only a partial section of the curved surface is sufficient to deduce the shape of the complete surface and even the size if the surface is a closed symmetrical surface. The missing data on the occluded regions are automatically interpolated during the surface fitting process.
- Surface curvature features are additional sources of descriptive information for the environment. This could improve the performance of SLAM algorithm for the cases when there is a lack of planar landmarks in some specific environment. Given sufficient computational power, surface

curvature features could improve SLAM at any given case as compared to the sole use of planar features.

## 1.4 Outline of the Thesis

Brief information about the SLAM problem definition and feature extraction part of the problem is described in first part of the introduction. The state of art is mentioned in the second section of the introduction part. The contributions of the thesis work are explained after pointing out the open field in the literature.

In Chapter 2, after a literature review on SLAM studies, the literature on 3D feature utilization for SLAM problems is detailed as this thesis work aims to improve the feature utilization side of the SLAM application.

In Chapter 3, a theoretical background is given on SLAM concepts and two major SLAM algorithms, namely EKF SLAM and FastSLAM.

In Chapter 4, the theoretical concepts behind the surface curvature feature extraction used in the SLAM algorithm is given.

In Chapter 5, the solution implementation design is presented. The integration of the surface curvature feature extraction method into the FastSLAM algorithm is explained.

In Chapter 6, the repeatability of the proposed surface curvature feature technique is experimented on a segment of real Kinect data and analyzed for data association purposes.

In Chapter 7, the results of the experiments and analysis based on the results are stated. The performance of the proposed novel approach is discussed.

In Chapter 8, the thesis study is concluded with an overview of what are gained and what to be followed next.

# CHAPTER 2

# LITERATURE REVIEW ON 3D SURFACE FEATURE UTILIZATION FOR SLAM

The spreading use of range sensors in robotic applications [29] has naturally pushed the methods from 2D to 3D universe. Range sensors provide valuable spatial information in especially indoor environments for purposes of obstacle avoidance, path planning, object manipulation, remote operation and autonomous navigation where a global positioning system (GPS) is not available [30]. In order to achieve such intelligent abilities, the utilization of range sensor data is vital as in image processing methods for vision sensors (cameras). The success of the representation of the environment determines how realistic the belief of the robot about its surroundings could be. In SLAM applications and other applications which, at some stage, require the association of range sensor data pairs have to rely on 3D data processing performance. As the most of such methods are feature based approaches, the 3D feature detection and description techniques are of utmost importance.

On the grounds that this thesis work primarily aims to improve SLAM performance by means of a new approach in data representation for the 3D data obtained from the range sensor, it is inevitable to investigate the already rich literature on 3D feature detection and description techniques. This review will provide the fundamental insight into various well-known state-of-the-art approaches and a panoramic perspective over most of the 3D feature utilization methods. Upon this overview, we will delve further into the subsets of 3D feature extraction literature which are considered to be relevant to the approaches followed, namely surface feature and surface curvature feature approaches.

## 2.1 3D Feature Detection and Description Methods for Data Association in SLAM Problem

The domain of 3D feature detection and description techniques is a vastly broad field. The attraction towards 3D features is an outcome of the complexity of the methods to infer depth information from 2D vision data [31]. Thus, the computational cost devoted to the derivation of range information is discarded by sensor technologies capable of a directly supplying the depth information for the scene. This is, beyond a reasonable doubt, a tremendous benefit for especially real time applications as in the robotics field. Still, to fully exploit such a saving on data processing, the feature detection and description techniques have to keep up with their 2D counterparts. The perpetual problems of uneven illumination and textures of the surfaces in the possibly unknown and uncontrolled environments are other important issues that are conveniently avoided in range sensor based approaches. Recent studies demonstrate that 3D feature literature is not too far away from satisfactory results. In the following sections, recent work on 3D feature utilization approaches and techniques are given within the scope of range sensor information processing for data association in SLAM domain and possibly other closely related applications.

In [31], the authors compare 2D and 3D SLAM performances on their benchmark framework by running them on the same environment in real time. They use Kinect as their vision and range sensor and Hokuyo laser scanner as the benchmarking sensor. The 3D SLAM method is the RGBD SLAM method presented in [32]. In this method, although the features are detected and matched from the RGB images of the scene via SURF feature detection and description method, the mapping process involves the depth values of the interest points found in the former part. Thus, although the feature extraction is performed in 2D, consecutive two frames are aligned via the RANSAC process based on 3D point locations of the feature points which is why this method could be somehow investigated in 3D feature approaches category. The benchmarking 2D SLAM method is chosen as GMapping [33].

The work in [34] is rather relevant to this thesis work not only in the 3D feature utilization side but also some other aspects such as object-based approach and curvature feature-based data association methods. We will put the other aspect aside to be later discussed in their respective sections in the remainder of this literature survey part. As far as the 3D feature usage for data association is concerned, the adopted method accepts the Kinect-style 2D organized 3D sensor input and detects 3D interest points by making use of local surface variations, namely local curvatures. For the curvature-based key point detection, shape index (SI), mean-Gaussian curvatures (HK), shape index-curvedness (SC) and factor quality (FQ) by [35] are implemented, analyzed and compared for performance parameters such as repeatability. Data association for object recognition application is completed by the matching process in which the descriptors extracted around the detected key points are compared between the model patches and the test patches defined as the neighborhood. The descriptors are chosen as the shape index values and the surface normal difference of the key point with respect to the average of its neighboring points. Thus, the feature vector basically consists of a shape index value in the range [0, 1] and a cosine value measured between the aforementioned surface normals. The study also includes the comparative analysis of the curvature-based detector methods; however, this discussion is left for the more related curvature-based methods literature. The results of the study yields a repeatability of as high as 80% and a recognition rate of as high as 96.4% in the best configuration of investigated methods. From the perspective of the work in thesis, the object recognition is not tested in a SLAM application and it stands as the biggest difference as compared to the proposed system. Also, the objects are analyzed point-wise and expanded as connected neighborhood patches whereas in our work, an initial spatial segmentation and a generalized quadratic surface representation is adopted for object detection. For the description, the same curvature-based features are used, however, not in the point level but in the surface level.

Edge detection is also popular in 3D feature utilization in SLAM applications. A recent example of edge detection-based localization is studied in [36]. In this system, color image output of the Kinect sensor is used to detect lines in the scene. The RGB image choice is due to the use of color and texture around edge regions for matching process. Once the lines are detected, they are tested for their curvature being low enough to meet the linearity tolerances. To gain robustness against noise, Gaussian filtering and dilation steps are applied. Then the detected lines are converted to 3D lines through the depth map information and checked whether the line still satisfies the linearity conditions when the depth information is also considered. After the detection of the lines are finalized, they are compared between two consecutive frames for matching. The intercepts and inclinations of the compared lines are considered as descriptors in the matching process. The system is tested in EKF SLAM application with real Kinect data recorded in a laboratory environment and the results are found to be satisfactory by the authors.

In [37] 3D features are extracted and used for object classification. The method of choice for detection is the application of the absolute value of the determinant of Hessian matrix formed by box filters of second order Gaussian derivatives. This process is applied to the gridded 3D data into smaller voxels (volumetric pixels). After the detection of interest points, the description is achieved by a 3D adaptation of 2D SURF method.

A framework for RGB-D mapping using Kinect-style depth images is presented in [26]. In this work, both the RGB and depth images are utilized in the mapping application in which TORO [38] is used as a graph-based SLAM optimization method. For data association, a two stage operation is performed the first stage of which is processed on the RGB image through the use of SIFT feature detection. In the second stage, the consecutive frames are aligned by RANSAC application on the 3D points around detected interest point locations. Thus, it is safe to summarize this method as a combination of 2D feature detection and 3D matching. The motivation of the study is to globally reconstruct the map of indoor environments.

Another study that exploits the dual output of RGB-D cameras for data association problem is given in [38]. The authors have benefitted from RGB images via Canny edge feature detection in 2D. As emphasized in their explanations, it is straightforward to refer to the same point from RGB image to depth image and vice versa. Thus, the 3D counterpart of the detected feature points in 2D image is conveniently referred to and the corresponding 3D edges are found and used as 3D feature descriptions. The scope of the study is determined up to the registration of the consecutive images by means of ICP algorithm. Hence, this process could be the data association part of a SLAM application; however, as the motivation in this work is to improve the image registration part, they did not opt to evaluate the performance of the registration inside a SLAM loop.

The combined approach of 2D detection and matching 3D point-wise association is also noted in [27]. In this work, the RGB images are used to detect visual features and match consecutive frames using SIFT, SURF and ORB (Oriented FAST and Rotated BRIEF, where FAST stands for *Features from Accelerated Segment Test* and BRIEF is the name for *Binary Robust Independent Elementary Features*) descriptors. The matched features locations are then used to obtain the 3D point pairs for frame pairs. The optimal transformation is then obtained by means of a RANSAC application to conclude the data registration process. The performance is evaluated on a publicly available RGB-D SLAM dataset that the authors themselves have recorded. A quite similar system is also proposed in [39] where SIFT and SURF methods are applied and analyzed as feature extraction techniques for SLAM.

A very recent study utilizing the fusion of RGB and depth image output of the Kinect sensor to achieve a more accurate 3D tracking technique is presented in [40]. Although the study does not aim at global feature matching purposes, there are useful results and analysis for the improvement of 3D feature matching in data association between consecutive frames. Not surprisingly, the 3D features are in the point level instead of higher level features such as surfaces or objects as the tracking processes do not require loop closing. This constitutes the reason for the review of this study to be located under the headline of 3D features. The author suggests a system which receives the intensity and depth images from the sensor. Then, simultaneously, detects interest points on both of the images based on the "*cornerness*" measures within local rectangular grid patches of a pre-determined size. For the intensity image, the mean error performance of the Harris and Shi-Tomasi corner detection methods are compared and Harris features are chosen to be settled with. On the other hand, a similar operation is followed on the depth image by making use of the curvature properties of the local surface. The peaks of the shape index (SI) measurement calculated based on the principal curvature values are detected as the 3D interest points. The study shows that the mean tracking error performance in the case of combined detectors of intensity and depth images yield better results as compared to the cases in which only one of them are in use. Then the fusion of the detected points commences in a manner that is inversely related to the deviation of the *cornerness* of the regions around the neighborhood of the detected points. This way, more confident feature points are allowed to proceed to the EKF stage where the Kalman Gain is modified to alter the weights of 3D and 2D interest points based on their deviation. Matching performance of the system is compared against the ICP and other tracking methods and it is shown that the system outperforms the others in mean projection error; mean and absolute pose error values. As far as the SLAM problem is concerned, the discriminative power of the surface curvature property is noted in the success of the tracking performance results although only the shape index is used as curvature feature instead of possible combinations of mean and Gaussian curvatures as in this thesis work. The given study, of course, cannot be fully referred to as it deals with local tracking problem; nevertheless the success of 3D curvature feature matching encourages the feature extraction approach adopted in this thesis study which also relies on the surface curvature properties.

In [41] the problem of place recognition is studied in the mobile robotics domain on the grounds that the autonomous mobile robotics is the main field where the place recognition is investigated intensively. In this work, the laser range data input from the sensor is pre-processed to obtain a depth image similar to that of commercially available RGB-D sensors like Kinect. The reason for this transformation lies in the aim of processing the depth information in a 2D image-processing sort of manner. Depth image representation yields a structural organization for data points which allows somewhat easier gridding operation. The proposed system firstly detects the interest points by the application of Laplacian of Gaussian (LoG) operator on the depth image. The resulting interest points basically result in the points which have distinctively different depth values within their neighborhood. Nevertheless, in awareness of the false positive interest points due to occlusions, they feel in need for the elimination of some initially detected interest points. For this purpose, they filter out the very high local gradient points by the assumption that they belong to an occluding object edge. Although not mentioned, however, this could potentially cause to miss any object edges as any object is an occluding body for any background region. We assume though, the background effect is handled accordingly against such a case. The authors also find the interest point forming a line undesirable as they prefer corner-like regions rather than edge-like regions and also prefer to keep the interest point number low. After the interest points are finalized, a fixed-size patch around each point is chosen for the production of feature descriptors around the interest points. Within the patch neighborhood, the local gradients along x and y directions are calculated and normalized into the range of $[0, 1]$ where the values near zero are planar regions and values near one are sharply curved regions. Finally, the extracted features are compared against the previously recorded features via a scoring system based on location and feature description matching. The results are evaluated on a real world range data using the SLAM system, TORO [42].

In [30], a SLAM system is proposed which is designed to be used in assistive robots for indoor environments. The system utilizes Kinect sensor RGB and depth data by means of the 3D feature extraction method, namely ORB [43]. The study suggests a system that detects and describes feature points using ORB feature representation which is reported to be chosen for its robustness, invariance to rotation, faster execution and lack of licensing costs as opposed to the patented SIFT and SURF techniques.

## 2.2 Surface Feature Detection and Description Methods for Data Association in SLAM Problem

In the 3D feature detection and description methods section, the approaches mostly utilize low level features focusing on points, curves and lines. Although these low-level feature-based techniques have proven to be successful and stable in many SLAM implementations some of which are mentioned in the previous section, the benefits of using higher level features urge SLAM applications towards higher level features. The semantic approaches could exploit the rich information obtained from higher level features such as surface features [44]. This is may not be vital for solely navigational purposes in plain environments; however, it is deadly important in the cases of object interaction and object manipulation. The fully extracted surface of a coffee mug would be a very helpful input to a service robot which is expected to bring that mug to the disabled user at home. But when a massive group of extracted corner points of the same mug would require vast additional processing and reasoning before any sort of interaction, like grabbing. Another advantage of higher-level utilization is the compact representation of the environment [44]. To represent a planar wall region with a single feature vector containing normal, size and location information as compared to having thousands of feature points with low level features is a fatal difference as all other processes build on this fundamental representation. The compactness of the representation provides a cumulative computational saving for the scan matching, loop closing and mapping process in the SLAM chain. When a feature-wise dense environment is considered, the execution time required for data association will differ dramatically for the cases of comparing a few vectors instead of one-to-one checking of several thousands of point features representing the same amount of environment data. Computational cost reduction is not the only benefit of compact representations; rather the robustness gained by the inclusion of bigger amount of data is also important in the sense that possibly erroneous data sections could be compensated by the dominance of other proper data sections belonging to the same object.

The 3D surface feature techniques in the SLAM problem are dominated by the plane representation-based proposals. Apart from the compactness benefits, the motivation for the tendency to plane representations could lie in the simplicity of detection and description on top of the fact that most of the indoor environments consist of planar regions. The outcome of the studies dealing with plane features in SLAM applications verify that the plane feature representations deserve the attention they drive. Including the plane feature-based methods, there are quite a number of studies using surface features as high level features for data association purposes in SLAM scenarios or other relevant problems. It is aimed to mention some of the most recent and milestone examples of those studies in the remainder of this section.

The study presented in [45] is directly motivated by the benefits of using higher level features in SLAM loop instead of low level features such as 3D point or edge features. The method actually suggests a hierarchical feature extraction process in which 3D points are first detected and described for scan matching. In the next phase of the process, the selected 3D points are tested for whether they form acceptable planar surfaces. For this detection purpose, RANSAC approach is adopted and the descriptor vector for the decided plane constitutes of 9 elements which are the plane origin location and 2 orthonormal vectors for the plane normal. The main idea of this work is to augment this 9-element plane feature vector in the SLAM state vector. The study also suggests the same approach for line representations, however, as the surface features are under consideration in this section, only the plane part is mentioned. The solution system is tested on real data and it is concluded that the compactness gained from the reduction of the feature vectors in the SLAM state vector pays off in reduction of the execution time. This is one of the main motivations behind using surface features instead of 3D point features. However, the initial 3D feature extraction process is still performed, only the data association process effectively works with reduced amount of feature vectors. Thus, the computational cost is relaxed only on the matching side not the feature extraction side.

Another work proposed for underlining the positive effect of higher level features is given in [44]. This work considers the environmental features in the object level and utilizes the object level features for the global matching process, also known as loop closing, rather than for solely local scan matching purposes. The authors' proposed system starts processing the sensor data by first segmenting the scene into planar and non-planar clusters. Both of the groups are used as landmarks, however, kept separately. The detected planar regions are registered using surface normals and least squares fitting. The criteria for plane categorization include the minimum number of inliers and low curvature indicated by the low variance of surface normals along the planar region. The non-planar regions are further segmented with either a connected component approach or a graph-based approach. The process also features a semantic noise filter that eliminates extremely big or small clusters which pose a high risk of erroneous data. The thresholds are chosen based on the average size of the common indoor objects. The segmented clusters are then treated as objects and then matched based on their centroid location, CSHOT descriptor [46] and its bounding box properties. The matching mechanism works in a safety-first manner in which the nearest descriptor match between the observed and known landmarks is performed in both directions, only when the both matches agree then the association is confirmed. Additionally, the spatial constraints are also applied to make sure the objects are consistent not only feature-wise but also location-wise. The system detects a matched new observation in terms of whether the loop closure occurred by matching the same object after a period of no detection or a regular match occurred either by direct matching or partial matching and merging. The performance of the system is evaluated on a real Kinect sensor data record that the authors themselves had collected. The object recognition rate, the amount of reduction up to 30% in the false positive rate by the help of object refinement and the success in the loop closure are emphasized in the results and conclusions sections. This work adopts a similar approach to the proposed system in this thesis work in the spatial scene segmentation parts and the overall object based approach, however, the feature extraction preferences differ and the quadratic surface generalization is not assumed in this work. Although the mentioned method seems to be making use of planar features as an additional utilization, however, the method proposed system in this thesis actually includes the planar regions silently under the quadratic surface generalization which also accommodates planar surfaces.

The surface features of the objects are utilized in [47] with the limitation of surfaces lying horizontally with respect to the sensor frame. The reason for such an assumption for the surfaces is due to the underlying motivation for the design of assistive service robots. In household environments, the objects are usually kept in horizontal surfaces such as tables, desks, counters, shelves and so forth, thus the service robot is assumed to be interacting with such surfaces most of the times. The SLAM system gets the 3D input and deals with the 3D point cloud starting with an iterative application of RANSAC in order to find planes in the scene. This method is applied on the whole scene and at each iteration; the largest plane that is roughly horizontal and contains sufficient amount of inliers. If those conditions are met, the selected data constituting the plane is registered in to the map and if not the said data is removed from the point cloud under consideration without an update into the map. The matching of the detected planes are performed by the inspection of overlapping between the pair of planes as projected in the global map. Hence, there is no use of feature extraction in the process, instead only location and pose matching are considered. Actually, the method uses the planar surfaces only for the mapping purposes. The localization problem is left to the line features extracted from the indoor wall regions. The method is tested in a regular home environment with a mobile robot driven manually and range data obtained by stopping the robot and making a sensor scan. Thus the algorithm is run offline on the recorded data. As the design criteria prefers having not detecting planes instead of false positives, some planar regions such as smaller tables in the distance and shelves too close to one another are missed by the mapping system. However, the results in general serve the purpose in the sense that the localization and mapping of the most significant horizontal planar surfaces is achieved. The extension of this study into the SLAM method in which the planar surface features are used as landmarks in the loop closing including other planar surfaces than horizontal planes is presented in [48].

## 2.3 Curvature Feature Detection and Description Methods for Data Association in SLAM Problem

Curvature features of object surfaces in the robotic application environments produce valuable information as they are able to provide affine invariant descriptors [28]. For applications like SLAM where the data association is a vital problem, the performance of feature detectors and descriptors is a game-changing factor. The appearance of range sensors in such applications did not only help to avoid the sensitivity of 2D images to conditions such as illumination and texture, but also offered the chance to extract additional complex information from surfaces. The extraction of surface topology improves the descriptor performance and thus results in better data matching in many applications fields with face recognition being an important example where the pits and peaks of human face provides significantly robust and repeatable description features [35]. Curvature information of the surface is also persistent in the partial occlusion scenarios in which although some properties such as size, edge or corner information might get lost. Nevertheless, if the said surface is continuous and symmetrical to some degree, then the curvature feature is valid for any observable section of the surface. The description power of curvature features is also another source of attraction in the sense that these features provide robust, affine invariant and repeatable performance. Hence, although not yet quite common, the utilization of curvature features in robotic applications as in SLAM problem is also promising. As will be noted, it is hard to find direct use of curvature features in SLAM systems, which is partly the source of the motivation for this thesis work, though the literature on data association and recognition contains valuable work with substantial results. Thus, in the remaining parts of this section the use of surface curvature features in data association process of SLAM problem and more often other related problems.

A quite recent and relevant study is reported in [34] which presents a local surface curvature feature based object recognition technique. The aim of the study is to obtain salient and repeatable key points under view variation which is a typical scenario for robotic applications. The system traverses the input depth image exploiting the structured distribution of data points and calculates curvature values from differential geometry within a local neighborhood patch. The curvature calculations include Gaussian curvature (K), mean curvature (H), shape index (SI) and proposed factor quality measures. Then maxima and minima are found based on selected combinations of these curvature measures which are later evaluated in the results section of the study. The key points are detected thusly as the extreme points of the curvature values throughout the surface. The performances of the mentioned descriptors are evaluated for stability under noise and viewpoint variation. Figure 2 is found in the authors' article to depict the results of the repeatability analysis performed on a publicly available 3D object dataset. The objects are isolated frames in the dataset and the methodology adopted is to define a repeatability measure based on the percentage of stable key points detected in one view of the same object with respect to another translated and rotated view. If the factor quality descriptor is left aside, the results suggest that all other curvature-based features lead to a solid repeatability in the levels of %80 to %90. Although the theoretical approach is quite relevant to the basis of this thesis work, there are differences to note. For one, SLAM performance is not evaluated in the given work which would reflect a more realistic problem environment. In our thesis study, a compact surface based curvature feature is used instead of the point-wise feature representation in the referred study. However, in the scope of feature detection and description from the range sensor data, this study stands as an enlightening preview for our work to reach improved SLAM performance by means of better sensor data utilization for data association.

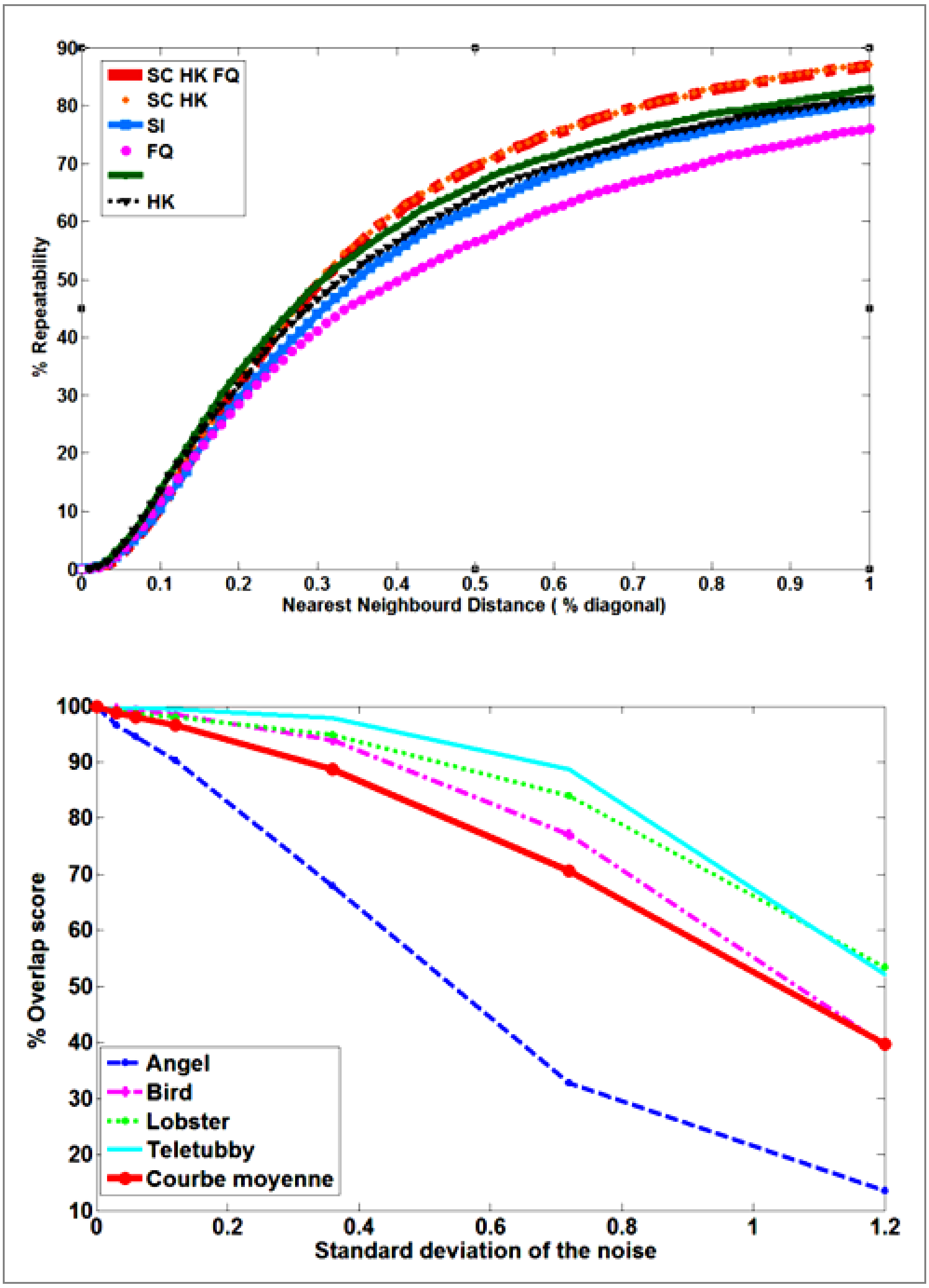


**Figure 2: The repeatability analysis of the curvature feature descriptors under viewpoint change and noise conditions in [34]**

### 2.4 Object-based Techniques in SLAM Applications

The work in this thesis aims to utilize a compact representation for 3D surfaces. For that reason, it is necessary to view the literature in the perspective of Object-based

SLAM. Basically, we can refer to Object-based approaches as the compact representations of the environment where the fundamental elements in the environment are object-level regions. Although we try to demonstrate the benefits of a method that falls under the object-based approaches in the SLAM domain, in the literature object based approaches appear mostly on applications of SLAM in dynamic environments. This is natural in the sense that object-based approach is an option in SLAM in static environments; however, it is quite unavoidable for SLAM in dynamic environments as in real world, not points or lines but "*objects*" move. If everything is stationary, then it is straightforward to break the environments into any convenient sub-sections such as planes, lines or points. Nevertheless, when there are moving objects in the environment, there has to be an additional consideration on whether the regions of interest are moving or stationary. Thus, the intuitive representation would naturally be based on object level regions. The literature, therefore, contains most of the object-based techniques in studies devoted to SLAM in dynamic environments. In this section, we will be mainly reviewing object-based SLAM approaches which are vastly on SLAM with moving objects.

The pioneering work on SLAM in dynamic environments is presented in [49]. In this work, a mathematical framework is proposed under the name of Simultaneous Localization, Mapping and Moving Object Tracking (SLAMMOT) which combines SLAM in dynamic environments and detection and tracking of the respective moving objects. The illustration for the overall system is given in Figure 3.

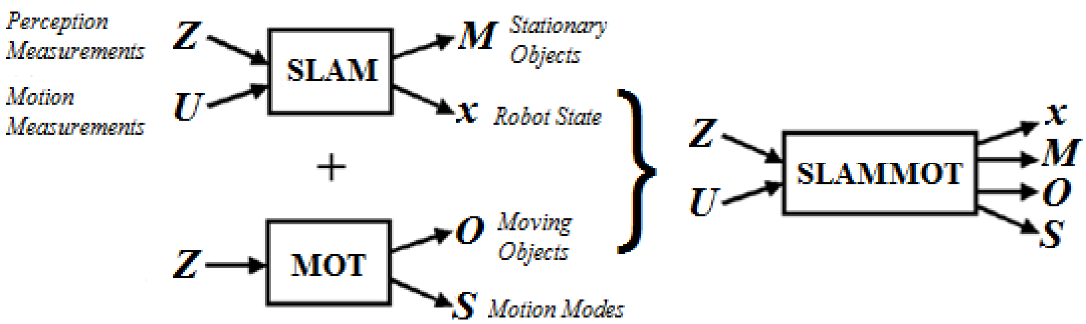


**Figure 3: SLAM Process and Moving Object Tracking Process (MOT) combining into Simultaneous Localization, Mapping and Moving Object Tracking (SLAMMOT) Process [49]**

The motivation for this work is described as to build a framework for autonomous safe driving in the urban traffic environments. The detection and tracking of moving objects requirement of the system therefore arises from the dynamic nature of the traffic environment. When the elements of a traffic environment are considered, it is seen that a large variety of objects could be found in such an environment. The objects could be buildings, walls, sign posts, traffic lights, humans, cats, vehicles, bicycles, trees, shops, sidewalks and so forth. Due to the diversity of the object types, it is not feasible to have a finite model database or certain object characteristics. For that reason, the authors proposed a "*free-form*" object representation which assumes no a priori constraint on the objects to be detected and tracked. In this free-form object representation, the scan points from the range sensors are segmented based on solely the distance criterion. Thus, the only assumption of the method about the object candidates is the connectedness of the scan points forming the objects. To give a brief insight into the segmentation process, it could be noted that considering the urban traffic environments, they have determined the constraint that two different segments cannot have points which are closer than 1 meter in distance. This is the initial coarse segmentation applied on the scene data, however, more precise segmentation of the objects rely on the performance of the SLAMMOT algorithm over time frames.

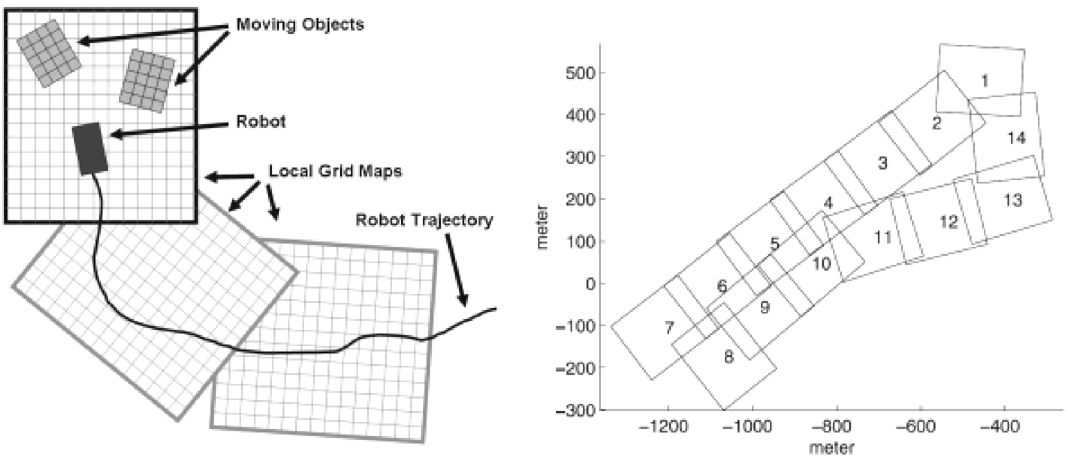


**Figure 4: Free-form Object Representation in Partially Overlapping Grid Maps [49]**

In SLAMMOT method, the local localization is achieved by ICP (Iterative Closest Points) algorithm and the sensor measurements are stored into a local grid map as stationary or moving objects. The objects are obtained through the segmentation of the scene into object clusters based on only the distance criterion. The only features of the segmented objects are the locations of the centroids of the data points belonging to the objects. This is the result of the generalization to free-form objects. The grid map size is determined as 160m and 200m for the width and length, respectively with a grid resolution of 0.2m. The overlapping regions are the 40m margins of the grid borders. The aforementioned sizes are determined experimentally for the application in [49], however, it is noted that the sizes can be adjusted on-line in practice. Figure 4 depicts the local grid maps and the overlapped combinations of those maps over the trajectory. Each local grid map is formed by detected objects which are localized by ICP matching. Those locally formed grid maps are then utilized as *3 DoF features* for the feature based EKF SLAM algorithm used for the global SLAM loop. Thus, the global SLAM is achieved by matching of the local grid maps via a 3 DoF feature vectors which represent the location and orientation of each local grid map. As seen in Figure 4, there are 14 local grid maps, in other words, 14 global features for EKF SLAM algorithm. Thus, as far as the feature extraction and description part of the SLAM is concerned, a truly original method is used which is quite different from the approach used in the proposed method for this thesis work. The object features used in the local localization part, however, could be compared to the object representation used in the proposed method. In [49], only the location of the segmented objects are used as features, dropping even the orientation of the objects as with the distance criteria for their free-form object segmentation a reliable orientation detection is not possible. This approach again differs from the proposed method due to the use of 3D surface curvature features in the object feature vector in addition to the location information. The absence of orientation, however, is common in both approaches as orientation provides misleading information for quadratic surfaces when the partial visibility of surfaces is considered.

The preceding work [49] on object-based SLAM is given in relatively more detail due to the fact that the most of studies in this field deals with the similar SLAM in dynamic environments problem and the referenced work is the pioneering work among others. In [50], a moving object detection algorithm is proposed along with visual SLAM on a small-size humanoid robot. The system has a monocular vision camera as the only sensor, from which the distance information is obtained using consecutive image frames. The moving object detection is achieved concurrently with SLAM; however, the objects are registered in a start-up procedure before the SLAM operation. Thus, the categorization of moving and stationary objects is performed offline. The features used in EKF algorithm are selected as SURF features. Hence, this method actually uses a point feature, namely SURF, and then recognizes objects by matching the strongest features belonging to the objects. Objects are not represented directly by object features but through the point features on them. Although the authors themselves refer their work to be based on object recognition and object-based SLAM, the implementation is somewhere between object feature representation and point feature representation. The approach is thus quite different from the proposed 3D surface feature representation based method in which strict object segmentation precedes the quadratic surface fitting and feature extraction from the fitted surface.

Another work featuring an object-based approach is presented in [51]. The motivation of the work is basically about the indoor use of mobile robot help for humans. The data representation is accordingly based on typical household objects and doors. Their proposed system uses a laser range finder to detect lines in 2D space which are then used for door detection. A stereo camera is used to recognize the detected doors by means of SIFT feature matching against the previously detected doors. This method considers only the doors for SLAM and uses line detection and SIFT features for door object utilization. This way, the authors choose to deal with a limited amount of objects among the objects in a scene which is quite different from the proposed approach in which a more generalized family of objects which have quadratic surfaces is considered.

Object mapping is handled by the help of SLAM method in [52]. In this work, the map of the environment is first mapped by means of stereo SLAM application with Canny edge detector for data association. The mapped scenes are then segmented into objects which are modeled in a database beforehand. For this segmentation, first the objects are detected as composed of edge points found in SLAM application and then compared to the objects in the database in terms of SIFT descriptors of the objects in the database and detected objects from the scene.

In [53], a different object-based SLAM method is introduced in which no feature extraction is involved. The study features a mobile system that navigates and collects environment data with a 2D laser range scanner which is driven by a servo motor in order to obtain 3D range data. The 3D range data is then segmented into two main groups where the vertical point clusters which are connected to the ground level are considered as robust landmarks and the "*overhang*" point clusters which have a certain gap below them from the ground level are considered as obstacles and not landmarks. The latter group is segmented as such due to their distortion effect on the 2D projection process of the vertical type landmarks. For example, a tree body is considered as a landmark and the when the system tries to project the tree body onto the 2D xy-plane, the leaves and branches on top of the body distort the projection and injects undesired points into the landmark map. In summary, the method uses raw segmented vertical point clusters as the SLAM landmarks and although it is object-based, it does not describe the objects with their features. The method only makes use of a subset of raw data clusters.

An interesting application of SLAM is studied in [54] where the system is designed to help people interact with the objects previously learnt by the agent. The human user wears the system equipments which consist of camera, odometry sensor and a hand-held screen. As the human traverses the environment, the system recognizes the objects that are stored in its database and displays helpful instructions on the screen to the human user. The use of SLAM basically serves for the localization of the objects of interest and the consistent mapping of the environment to the user via the hand-held screen. Apart from its interesting motivation, the method uses classical mono-camera SLAM and benefits from SIFT features for data association between the known objects and the live scene objects.

Another example of object-based SLAM implementation is given in [55]. The study aims to make use of natural scene objects, especially trees. In the experimental work, the robot registers and recognizes tree objects primarily by the help of SIFT features and some other segmentation and image processing techniques.

In [56] and [57], SLAM with moving object tracking (SLAMMOT) problem is handled with the motivation of computational saving. The latter work is based on the first work except that it tries to accomplish the task with a monocular vision instead of stereo vision. In both studies, the SLAMMOT problem is approached in the way that tries to estimate the robot state, the static map and the moving object trajectories at the same time by separating the classical SLAM problem and the moving object tracking problem. Separate filters for each moving object are defined which renders the landmarks and robot state independent similar to the *d-separation* concept in FastSLAM [21]. As far as the feature detection and description parts are concerned, in [56], a simple Harris corner detector is used and the intelligence to manage features is left to higher level mechanisms that process and reason the interest points. In general, the images are divided into uniform but variable size grids and the Harris corner features found within those grids are analyzed in terms of the categorization into groups of moving objects, stationary landmarks or objects that are neither. In [57], however, the feature detection and description are left out of discussion by performing a manual data association and feature injection.

Object detection is studied for home environment service robot SLAM scenarios in [58] and [59]. The former work is based on an off-line known landmark database construction. Thus, the process of adding new landmarks to the map, also referred to as the database, is omitted from the thesis discussion. The authors basically aim to analyze the performance of the data association performance of their proposed method which relies on multi-scale Harris corner detection and SIFT descriptors of those detected corners. The novelty of their work lies on the concept of points displaying group transformations which, in turn, lead to object detection and tracking. As the main focus is the home environment, their data association process is optimized for mostly planar geometrical objects commonly encountered in such environments. The latter work similarly targets the home environments and is implemented with the help of an off-line training of objects of interest. The object matching performance of histogram based techniques is analyzed within EKF SLAM framework in terms of color, gradient and Laplacian histograms. Although both two studies deal with object based SLAM, their implementations do not fully correspond to a mobile robotic application as they heavily include off-line computation.

Another object SLAM example is given in [60] in which, multi-body moving objects are also considered in SLAM. The moving objects are not reconstructed as they are assumed to be usually smaller foreground objects which are not quite feasible to map. They also state that for some applications, it is sufficient to notice the moving object rather than fully describe. The method aims to segment, detect and track multiple moving objects through the temporal process. In the tracking process, groups of FAST (Features from Accelerated Segment Test) corner features are detected. Although all of the feature points are tracked, only a selected subset of them is allowed to be the part of the 3D map. Matching for the tracking is performed on the patches generated around the feature points based on the vicinity, affine similarity and occlusion criteria. The tracked features are then fed to the motion segmentation process in which the independently moving feature groups are detected considering the camera motions and epipolar geometry. Then, the convex hull of the segmented moving feature points is described as the moving object.

An important work on object-based SLAM is presented in [61]. Apart from the other work mentioned among the object-based SLAM that deal with moving objects in dynamic environments; this study focuses on the object representation part of the SLAM. From that aspect, the motivation of this work is similar to that of this thesis study. The computational benefit of using higher level features such as object features instead of lower level features such as points, planes or patches is emphasized. With the compactness of the object level representation, a comparable performance is reported to be achieved at a lower computational and map storage cost. This philosophy is quite parallel to our approach to the problem of data representation in SLAM. However, there is a fundamental difference in the object detection mechanism. The authors' method requires a pre-run object database model generation in which the common objects in the operation environment are modeled using marching cubes technique where each object is observed and modeled in isolation. This is not consistent with the complete autonomous operation as the object detection process requires a type of preliminary training. In addition to this important difference, the adopted object recognition technique is based on a generalized Hough Transform approach [62] as compared to the quadratic surface representation approach in this thesis work.

## 2.5 Review of the Surface Feature and Curvature Feature Detection and Description Techniques for Use in SLAM Applications with Range Sensor Data

This thesis study aims to achieve improvement on the performance of the state of the art 3D SLAM techniques, namely FastSLAM [21], by means of a compact 3D feature extraction technique that is desired to respond to the challenges in the sensor data utilization for mobile robotics applications. Thus, the feature detection and description techniques are expected to provide some fundamental properties in order to be feasible in robotic applications. The extracted features must be robust as the mobile robots navigate through unknown environments with possibly different conditions. Viewpoint invariance is necessary due to the fact that the robot may have different views of the same part of the scene and needs to recognize and associate these views. Another important demand is for the repeatability which is vital for the loop closure scenarios in which previously seen objects might be encountered several times and the features derived from that object must be repetitively consistent for successful association. From this perspective, the methods investigated in this literature survey chapter will be reviewed in this section. The point feature-based techniques such as the ones making use of SIFT, SURF, ORB [43], Canny, Harris and other features are herein disregarded as our thesis study is built upon the compact surface representation approach.

In [34], a rather relevant study in terms of 3D curvature feature utilization is reported. The suggested method makes use of surface curvature measurements, namely, shape index, mean and Gaussian curvature, shape index and curvedness in order to detect key points at the maximal points within surface patches. After the detection of key points, descriptors around them are defined as shape index which is within the range of [0, 1] and the surface normal difference margin with respect to its neighborhood which is represented by the cosine value of the said normal vectors. Although this method is not tested under the SLAM problem, it is straightforward to apply the same method that is able to prove itself in the object recognition problem in the data association process of the SLAM applications. Thus the results of this study stand as a positive reference for our curvature feature utilization proposal except that our approach handles the curvature measurements in the surface level in a compact manner whereas in the aforementioned work curvature features are treated as key points. The analysis and results of the referenced study is given in further detail in the chapter about the proposed system details.

A semi-3D feature detection and description method is proposed in [40] as a solution to accurate tracking harnessing both of the dual output image types which are of intensity and depth information. The most fundamental difference of this work from the other Kinect fusion methods given in [26], [27] and [38] is that the key point detection is not strictly performed on the 2D intensity image. This is a fatal difference in the sense that intensity images are much more sensitive to the environmental conditions such as illumination, texture and color diversion and shadow regions which are important limitations for mobile robotics systems expected to work in unknown and uncontrolled environments. In the method, however, although the system benefits from the readily available intensity image, the detection process has another choice which is the 3D feature key point detection based on surface curvature properties. The system detects key points simultaneously on both intensity and depth image grids and proceeds with the less deviating detection which automatically eliminates the possible problems with intensity images and vice versa although not as often as the former case. For the 3D feature case, the method chooses to utilize the maximal shape index features for description and prefers corner features for the 2D case. Thus, the surface curvature feature approach is similar to our thesis work in the sense that the principle curvatures are exploited; however, the point-wise feature representation makes the difference. Also, the method is purely targeted for tracking and not for loop closing globally which explains the point feature approach as it is the common choice in tracking methods.

A study that is directly aimed at the demonstration of compact higher level feature utilization in SLAM problem is proposed in [45]. In their proposed system, the authors perform feature extraction in a hierarchical manner which starts with 3D interest point detection and followed by detection of the possible planes formed by the respective 3D interest points. The detected planes are then represented by 9-element feature descriptor vectors consisting of plane origin and two orthonormal vectors with respect to the plane surface normal. Then those planar surface feature descriptors are augmented into the SLAM state vector which, in turn, means that the planar feature vectors are used not only for local scan matching, but also for the global loop closing phenomenon of the SLAM problem. This concept is the main connection between this work and our thesis study. Albeit, our approach aspires to take this idea further into a feature description that is able to represent more general surfaces, namely quadratic surfaces, rather than only planar surfaces. But still, the mentioned study gives a good estimate of what the compact surface feature representations could bring for an improved SLAM performance. Mainly, it is shown by the authors that, planar surface features provide comparable performance to point level features at a significantly lower computational cost thanks to the compactness of the surface features. This is quite parallel to the idea that our thesis work is built upon. The results that are verified on real Kinect data will be elaborated further in the proposed system discussion.

A similar work exploiting the compact surface feature representation, namely planar features, is presented in [44]. The notable difference is the use of non-planar regions as object candidates in addition to the planar surface feature utilization. Thus, on the grounds that this work combines compact surface feature approach with object level feature approach, the solution gets one step closer to our proposed system. However, our system does not perform a categorization of planar or non-planar surfaces; instead, planar surfaces are implicitly covered as they are a subset of quadratic surfaces. The authors' suggested method first segments the scene in a connected component manner as in our method and proceeds further by classifying the segments into the groups of planar and non-planar surfaces. Then treats both of the detections as landmarks and uses them in the loop closing feature matching process. In the object features, the authors check location-based and bounding box properties for matching. The performance of the system is investigated on real Kinect record and promising results are achieved the details of which will be mentioned in the system proposal chapter.

In [63] a comprehensive evaluation of 3D key point detectors is presented. The benchmark methods are chosen among the most popular techniques in the 3D feature literature. The methods are tested on a publicly available RGB-D Kinect object dataset [64]. The main interest of the evaluation is the "*repeatability*" of the detectors against the factors such as noise, view point change and occlusion. The metric is based on the method in [65] which is defined as the difference of extracted key points before and after the applied transformations of rotation, translation and scale change. In the results of this work which will be given in more detail in proposed system chapter, it is seen that the curvature features have a good balance in computational complexity, rotational invariance, translational invariance and scale invariance. In general, curvature features result in the best computational performance with a %30 difference to the nearest method and similar performance to other methods which validates our choice as far as the accuracy and computational complexity criteria are considered.

Another informative evaluation is conducted in [66] where presents a comparison of local feature descriptors including a notably large variety of descriptors is presented. The curvature description methods including mean, Gaussian curvature, shape and curvature index result in one of the best performances in the evaluation that considers the deviation of the feature values and the distinctiveness of features as a measure of how likely a point from one set is undesirably matched to another set. The results of this work stands as another supportive factor for the use of curvature features for our data association purposes.

As the related literature does not contain a directly comparable method employing both the surface curvature feature and the compact surface approaches, it is found fair to investigate them separately and to try to infer the results as a combination. In light of the point curvature feature and compact planar surface feature methods proposed in the literature, our method intends to incorporate the benefits of compact surface features and the robustness the discriminative power of surface curvature features. This is the foundation that this thesis is built on and the motivation for the use of compact quadratic surface representation and surface curvature features.

## 2.6 Multi-scale Approach for 3D Feature Extraction in SLAM Applications

In 2D feature matching applications, the benefits of multi-scale approaches are observed in the *de facto* methods in the literature, namely SIFT and SURF. Both of these methods are based on texture variation which is a scale-dependent property of the surfaces. Thus, the need for such multi-scale approaches is obviously legitimate. However, in 3D universe, with the additional depth information, different types of features could be extracted from the surfaces one of which is the surface curvature variation. As explained in [67], surface variation is invariant to re-scaling. Thus, for the surface curvature features, multi-scaling might not be too critical. Moreover, the scale space representations are mostly considered for point-feature matching cases where the features drawn from the same surfaces at different scales vary significantly especially in the quantity. However, in the case of this thesis work, the surface matching is not processed point-wise but compact quadratic surfaces are matched based on their surface curvature properties for the whole segmented surface. Thus, it could be anticipated that the multi-scale implementation of the surface curvature feature extraction is probably a computational cost that does not return equivalent benefit for the matching performance. This is indeed the case for the results of our experimentations. As no significant effect of multi-scale surface curvature feature extraction is observed, the multi-scale approach is discontinued in order to reduce the computational cost that does not pay off.

# CHAPTER 3

# THEORETICAL BACKGROUND FOR SLAM

There seems to be many types of SLAM algorithms, however, there are two main approaches which constitute the basis for others. These two approaches are EKF-based and particle filter-based techniques. In fact, EKF is usually a part of particle filter-based techniques in the sense that it is used within each particle in the particle filter methods. Thus, it is more explanatory to investigate EKF SLAM first and then FastSLAM [21] which is the state of the art technique for particle filter-based SLAM.

## 3.1 3D 6DOF EKF SLAM

In this thesis work, FastSLAM technique is used as the probabilistic filter for the SLAM problem. Although the main focus of this work is the feature extraction part, it is vital that the de facto techniques should be used for implementation and testing.
As FastSLAM also contains EKF in the particle level, the theory behind EKF SLAM algorithm is presented.
In the formulizations, the notation in [68] is used. The expressions are stated such that they apply to any type of landmark captured by any type of 3D sensor. This is to isolate the thesis work which is about the feature extraction for landmark detection park from the state of the art SLAM techniques which are used for testing.

### 3.1.1 Definitions

In Figure 5, a symbolic representation of a SLAM environment is given. In this environment, we will denote the related variables as given in the following definitions.

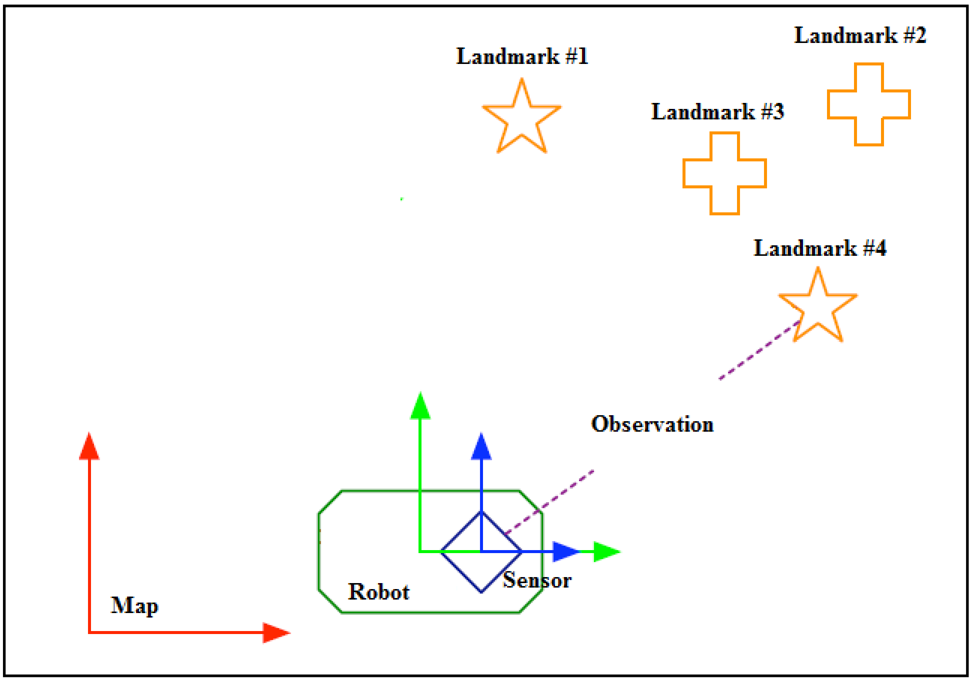


**Figure 5: An Example of SLAM Environment [69]**

The robot state consists of 3D Cartesian coordinates and the bearing angles, namely yaw, pitch, roll, where the state variable is named as $x_r$ given in (3.1).

$$x_r = [x\ y\ z\ \phi\ \chi\ \psi]^T \tag{3.1}$$
$$\phi: yaw, \qquad \chi: pitch, \qquad \psi: roll$$

As illustrated in Figure 5, at any given time frame $k$, there expected to be a number $L$ of landmarks created from the extracted features of the sensor measurements. Those landmarks are referenced by global coordinates $y_i$ given in (3.2).

$$y_i = [x_i\ y_i\ z_i]^T \tag{3.2}$$

The complete state vector is defined as $x$ and is made of the augmentation of the robot state vector and all $L$ landmark position vectors up to the time frame $k$ at that instance as given in (3.3).

$$x = \begin{bmatrix} x_r \\ y_1 \\ \vdots \\ y_L \end{bmatrix} \tag{3.3}$$

As the SLAM problem is handled in probabilistic approach, there will be a probabilistic estimation for the state variables. This is modeled as a multivariate Gaussian at each time frame $k$ with mean $\hat{x}_{kk}$ and covariance $P_{kk}$. The size of the covariance matrix is $(6 + 3L) * (6 + 3L)$. The state is an augmented matrix of robot state and landmark locations. This matrix is the heart of EKF and it contains the complete state of the environment and the robot at each time frame as given in (3.4).

$$P_{kk} = \begin{bmatrix} P_{xx|6x6} & P_{xy1|6x3} & \cdots & P_{xyL|6x3} \\ P_{y1x|3x6} & P_{y1y1|3x3} & \cdots & P_{y1yL|3x3} \\ \cdots & \cdots & \cdots & \cdots \\ P_{yLx|3x6} & P_{yLy1|3x3} & \cdots & P_{yLyL|3x3} \end{bmatrix} \tag{3.4}$$

Practically, this covariance matrix is initially set to zeros. This means that the robot location is placed to the center of the global map. This convention is mainly due to the ease of calculations.

EKF iterates through this covariance matrix updating the belief values of the state variables based on the prediction with motion and update with sensor measurements. The prediction and update (correction) cycle of EKF is depicted in Figure 6.

The equations for the prediction and update steps are given in (3.5) to (3.11).

$$Prediction:$$

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \tag{3.5}$$

$$P_{k|k-1} = \frac{\partial f}{\partial x} P_{k-1|k-1} \frac{\partial f}{\partial x}^T + Q_k \tag{3.6}$$

$$Update\ (Correction):$$

$$\hat{y}_k = z_k - h(\hat{x}_{k|k-1}) \tag{3.7}$$

$$S_k = \frac{\partial h}{\partial x} P_{k|k-1} \frac{\partial h}{\partial x}^T + R_k \quad (3.8)$$

$$K_k = P_{k|k-1} \frac{\partial h}{\partial x}^T S_k^{-1} \quad (3.9)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k = \quad (3.10)$$

$$P_{k|k} = (1 - K_k \frac{\partial h}{\partial x}) P_{k|k-1} \quad (3.11)$$

Where $f(.)$ represents the motion model and $h(.)$ stands for the observation model. The above two cycles of calculations lie at the core of the EKF process. At each iteration of the EKF cycle, the estimations could have less and less errors if the landmarks provide sufficient information.

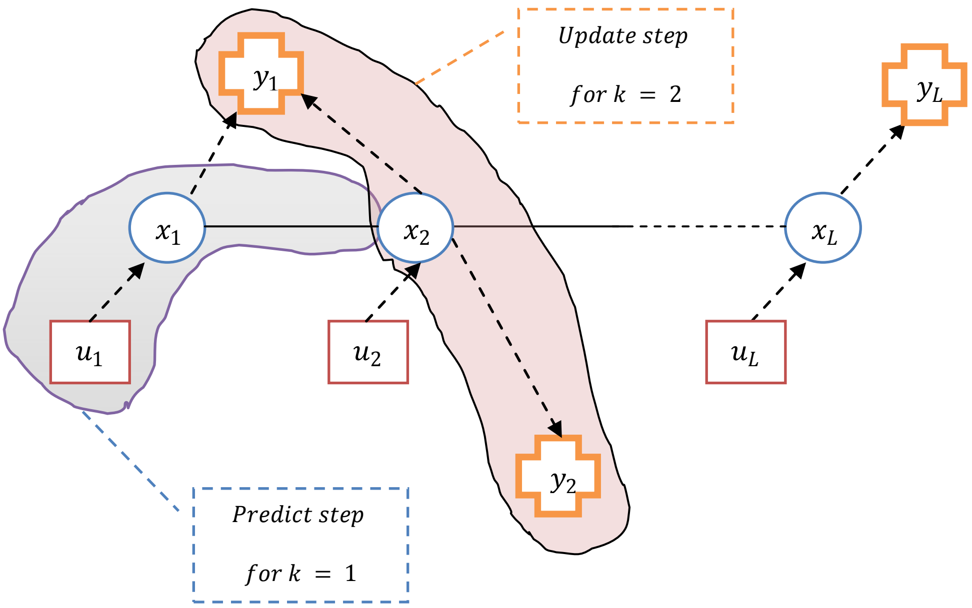


**Figure 6: Prediction and Update Steps**

### 3.1.2 Prediction Step Calculations

Prediction step is defined as the next pose estimation of the robot based on the motion control data, namely $u_k$. This vector represents the "*noisy*" belief of the robot about its location and pose based on measurements from sensors such as odometer, inertial sensor and others. Thus, $u_k$ vector is assumed to be of 6D as it contains both the location and the pose change between the consecutive states at time frames $k-1$ and $k$.

The odometry information is supposed to be "*extra*" information that improves the error rate of the estimations. Actually, EKF works well without any motion information from the sensors. The decisive information basically comes from the range sensors. Hence, we can virtually discard odometry information from the cycle by setting the uncertainty to a large value. Nevertheless, any useful information could improve the estimations in this probabilistic filtering algorithm hence the odometry data is also considered in EKF.

### 3.1.3 The Motion Model

The robot is assumed to be the only moving object on the map. At each transition between time steps, robot state changes as given in (3.12) and (3.13).

$$x_{r\{k\}} = f_r\left(x_{r\{k-1\}}, u_k\right) \tag{3.12}$$

$$u = [x_u\ y_u\ z_u\ \phi_u\ \chi_u\ \psi_u] \tag{3.13}$$

$f_r$ then becomes by making use of the homogeneous coordinates as in (3.14);

$$\begin{cases} x_k = x_{k-1} + R_{11}x_u + R_{12}y_u + R_{13}z_u \\ y_k = y_{k-1} + R_{21}x_u + R_{22}y_u + R_{23}z_u \\ z_k = z_{k-1} + R_{31}x_u + R_{32}y_u + R_{33}z_u \\ \phi_k = \phi_{k-1} + \phi_u \\ \chi_k = \chi_{k-1} + \chi_u \\ \psi_k = \psi_{k-1} + \psi_u \end{cases} \tag{3.14}$$

where, $R_{ij}$ corresponds to the transition matrix obtained by augmenting the rotation matrix and the translation vector.

### 3.1.4 State Vector Update

In the prediction step with the motion data, only the robots pose changes. Hence, only the robot pose variables are updated based on the formulations given in (3.15). The rest of the state vectors occupied by the landmark positions remain unchanged.

$$\hat{x}_{r\{k|k-1\}} = f_r\left(\hat{x}_{r\{k-1|k-1\}}, u\right) \tag{3.15}$$

### 3.1.5 Covariance Matrix Update

The new prediction based on the motion model and the control data is now reflected on the covariance matrix for the complete state vector. In order to find the updated covariance matrix, the first step is to compute the Jacobian of the transition matrix with respect to the state vector.

$$\left.\frac{\partial f}{\partial x}\right|_{(6+3L)*(6+3L)} = \begin{pmatrix} \left.\frac{\partial f_r}{\partial x_r}\right|_{6x6} & 0|_{6x3} & 0|_{6x3} & \dots & 0|_{6x3} \\ 0|_{3x3} & I|_{3x3} & 0|_{3x3} & \dots & 0|_{3x3} \\ 0|_{3x3} & 0|_{3x3} & I|_{3x3} & \dots & 0|_{3x3} \\ \dots & \dots & \dots & \dots & \dots \\ 0|_{3x3} & 0|_{3x3} & 0|_{3x3} & \dots & I|_{3x3} \end{pmatrix} \tag{3.16}$$

The special structure of the Jacobian in (3.16) allows for the manipulation in (3.17).

$$x_r = \begin{pmatrix} \frac{\partial f_r}{\partial x_r} P_{xx} \frac{\partial f_r}{\partial x_r}^T & \frac{\partial f_r}{\partial x_r} P_{xy1} & \dots & \frac{\partial f_r}{\partial x_r} P_{xyL} \\ P_{y1x} \frac{\partial f_r}{\partial x_r}^T & & & \\ \dots & & & \\ P_{yLx} \frac{\partial f_r}{\partial x_r}^T & & & \end{pmatrix} + Q_k \tag{3.17}$$

Only the first row and the first column is updated and all other entries in the covariance matrix remain unchanged. The $Q_k$ matrix represents the noise induced from the motion step as the odometry data contains some error with respect to the actual pose change. Also, since landmarks are static, it is sufficient to compute only the sub matrix for the robot state variables.

$$Q_{kv}|_{6x6}6x6 = \left.\frac{\partial f_r}{\partial u}\right|_{6x6} U|_{6x6} \frac{\partial f_r}{\partial u}^T \tag{3.18}$$

The sub matrix calculation given in (3.18) is to be later added upper-left sub matrix which represents the robot pose covariance.

### 3.1.6 The Update Step

The update step is where the range sensor measurement is propagated to the belief distribution of the robot. As in the motion model, a sensor model is used to represent the noise distribution of the sensors used for landmark detection.

### 3.1.7 The Sensor Model

The sensor model for the range-bearing sensor is given in (3.19).

$$z_i = h_i(x_r, y_i) \tag{3.19}$$

where $x_r$ is the robot pose and $y_i = (x_i \; y_i \; z_i)^T$ is the vector for the landmark locations.

Any observation vector $z_k$ is defined as in (3.20).

$$z_k = \begin{pmatrix} r \\ \alpha \\ \beta \end{pmatrix}, where \begin{cases} r: Distance\ to\ the\ landmark\ (Range) \\ \alpha: Azimuth\ to\ the\ landmark\ (Yaw) \\ \beta: Elevation\ to\ the\ landmark\ (Pitch) \end{cases} \tag{3.20}$$

Those parameters are related to the Cartesian coordinates of the landmarks with respect to the robot as in (3.21) to (3.23).

$$r = \sqrt{\bar{x}_i^2 + \bar{y}_i^2 + \bar{z}_i^2} \tag{3.21}$$

$$\alpha = \tan^{-1} \frac{\bar{y}_i}{\bar{x}_i} \tag{3.22}$$

$$\beta = -\tan^{-1} \frac{\bar{z}_i}{\sqrt{\bar{x}_i^2 + \bar{y}_i^2}} \tag{3.23}$$

Then, using the homogeneous coordinates, the equation in (3.24) is obtained:

$$\begin{pmatrix} & & & \bar{x}_i \\ & & & \bar{y}_i \\ & & & \bar{z}_i \\ 0 & 0 & 0 & 1 \end{pmatrix} = R(x_r)^{-1} \begin{pmatrix} & & & x_i \\ & & & y_i \\ & & & z_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.24}$$

This time, only the three coordinates are sufficient to calculate the function $h_i$.

### 3.1.8 The Observation Noise

The matrix R used in the preceding equations is the error induced by the sensor measurements which also represents the uncertainty for the respective sensor given in (3.25).

$$R = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_r^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{pmatrix} \tag{3.25}$$

The variance values in the matrix are characteristic values for various types of sensors. The diagonal nature of the matrix reflects the assumption of the independence between the noises for each coordinate component [68].

### 3.1.9 The Kalman Correction

This step is for the calculation of the $S$ matrix used in equations (3.8) and (3.9). With this calculation given in (3.26), all the equations featured in the prediction-correction cycle is are addressed.

$$S_k = \frac{\partial h}{\partial x} P_{k|k-1} \frac{\partial h}{\partial x}^T + R_k \tag{3.26}$$

### 3.1.10 Landmark Introduction to the Map

If at some time frame $k$, a new landmark is introduced into the existing map with $L-1$ landmarks after the feature extraction and matching steps, then the system is expanded according to $L$ number of landmarks as in (3.27) and (3.28):

$$\hat{x} := (\hat{x}\ \hat{y}_L) \tag{3.27}$$

$$\hat{y}_L = y_L(\hat{x}_r, h_L) \tag{3.28}$$

In (3.27) and (3.28), the state vector is updated by the insertion of the $L^{th}$ landmark $\hat{y}_L$ which is obtained by inverse sensor model calculation. The complete covariance matrix is then expanded in a straightforward manner in which the number of both the columns and the rows are increased by one. The inverse sensor model is applied as given in (3.29).

$$y_L(x_r, h_L) = \begin{pmatrix} x_L \\ y_L \\ z_L \end{pmatrix} = x_r \begin{pmatrix} h_r \cos h_\alpha \cos h_\beta \\ h_r \sin h_\alpha \cos h_\beta \\ -h_r \sin h_\beta \end{pmatrix} \tag{3.29}$$

The range (distance) and bearing angles yaw and pitch are converted to the global map coordinates thusly.

In summary, the EKF steps are processed as given in Table 1:

**Table 1: The Overview of EKF SLAM Steps**

| Step | Definition |
|---|---|
| 1 | The next robot pose is estimated with (3.5). |
| 2 | The complete covariance matrix is updated with (3.16). |
| 3 | Observation prediction with (3.20). |
| 4 | Computation of matrix S (3.26). |
| 5 | Computation of Kalman gain through the inverse of matrix S (3.9). |
| 6 | The state vector update with (3.10). |
| 7 | The complete covarience matrix updates (3.11). |
| 8 | If necessary, the introduction of a new landmark. |

### 3.2 FastSLAM

FastSLAM [21] is a novel approach to the SLAM problem which is proposed based on a fundamental assumption in the probabilistic relations of the states. FastSLAM has overcome the problems of EKF caused by single hypothesis approach. It is considered in FastSLAM that if the trajectory of the robot is known, then the measurements probabilities would be independent from each other [70]. Thus, FastSLAM makes this assumption in order to be able to avoid the complete covariance matrix which is of size $(6 + L) * (6 + L)$. There are two major benefits of this:

- As an ever growing covariance matrix is avoided, the number of landmarks is no longer causing the algorithm to have quadratic complexity. Also in EKF, as the covariance matrix grows with the addition of landmarks, complex matrix operations suffer gradually, however, for FastSLAM, as there is no covariance matrix, it handles large number of landmarks seamlessly.
- FastSLAM addresses another problem of EKF SLAM which is the sensitivity of the algorithm to data association problems which could go so far up to divergence as the algorithm relies on a single belief hypothesis. FastSLAM adopts particle filter approach and it can be said that it has $M$ number of different EKF SLAM algorithms running in parallel and after each time step, particles are regenerated such that the better beliefs survive more. So, when there are data association problems and some EKF SLAM particles are losing the track, if the number of particles is sufficient, there are some other particles having better belief probabilities. After the end of time steps, FastSLAM diminishes the worse particles and replaces with the resamplings of the better particles. This approach drastically reduces the chances of divergence due to poor landmark detection and matching.

In the following sections, the principles of FastSLAM are given in more detail. However, as EKF SLAM part is given with full formulations and mathematics, in FastSLAM sections, that much finer detail will not be needed. Instead, more of a conceptual explanation will be given.

### 3.2.1 Theoretical Concepts

In FastSLAM, the SLAM problem is elaborated as a Dynamic Bayesian Network (DBN). The network structure is as seen in Figure 7.

In DBNs (Dynamic Bayesian Networks), the probability is propagated through other probabilistic nodes. If there exists a node in the trail which is "*observed*", in other words, known, then the nodes connected by the observed node cannot affect each other and thus become independent. This is also called "*d-separation*". Further detail can be found in [70], however, in Figure 8 a brief example is given.

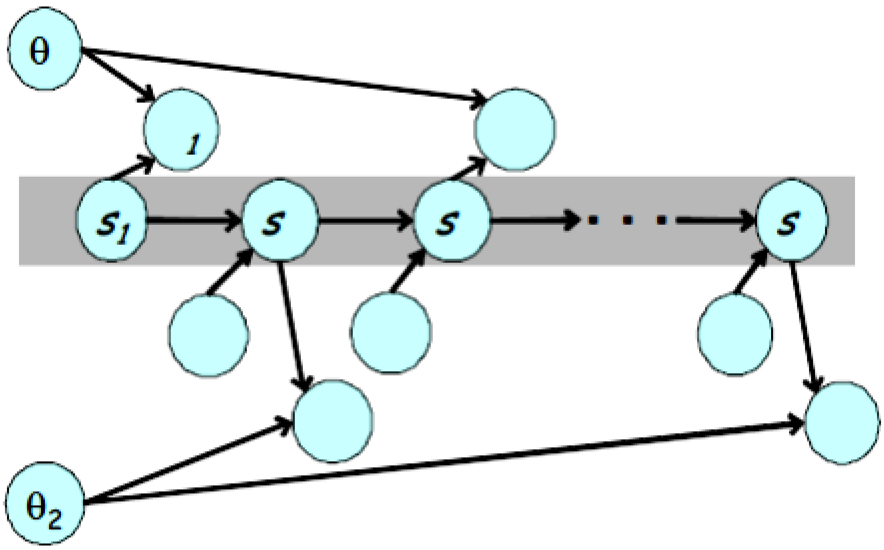


**Figure 7: SLAM Problem as a DBN [21]**

Thus, the idea in FastSLAM is to assume that the trajectory of the robot, which refers to the robot pose. This assumption renders the control data and the sensor data independent.

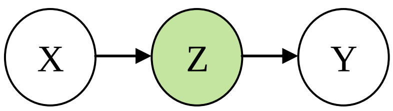


**Figure 8: D-separation in DBNs [70], if Z is observed; X and Y are no longer dependent**

Before the explanation of the formulations, the definitions of the variables are given in (3.30).

In Figure 9, the SLAM problem is shown as a DBN and the gray-shaded nodes are the robot pose stated assumed to be known, in other words, observed. This assumption breaks the propagation trail between $u_2$ and $z_2$, $u_3$ and $z_3$, $u_n$ and $z_n$. The broken links are shown in red.

$$\begin{aligned}
&s_t\text{: The robot pose at time frame } t\\
&u_t\text{: The control data (like odometry)}\\
&z_t\text{: The observation at time } t\\
&\Theta_{n_t}\text{: The landmark that the observation is made}\\
&n_t\text{: The data association of observations and map}
\end{aligned} \tag{3.30}$$

In general, the assumption of known trajectory leaves all observations and control data independent within the relative time frame. As such an assumption is sacrificed, the SLAM calculation can be simplified from (3.31) to (3.32).

$$\begin{aligned}
&p(s_t, \Theta | z^t, n^t, u^t)\\
&= \eta p(z^t | s_t, \Theta, n^t)\\
&\cdot \int p(s^t | s_{t-1}, u^t) p(s_{t-1}, \Theta | z^{t-1}, n^{t-1}, u^{t-1}) ds_{t-1}
\end{aligned} \tag{3.31}$$

$$\begin{aligned}
&p(s^t, \Theta | z^t, u^t, n^t)\\
&\qquad = p(s^t | z^t, u^t, n^t) \prod_{n=1}^{N} p(\theta_n | s^t, z^t, u^t, n^t)
\end{aligned} \tag{3.32}$$

(3.32) states that the iterative calculation of SLAM state is now decomposed into $N + 1$ estimator products. The first component in (3.32) is the path estimation and the other $N$ products are the estimations for each landmark location.

In FastSLAM, the independence provided by the known robot trajectory assumption is utilized by particle filter approach, namely Rao-Blackwellized particle filter [71].With the particle filter use, multi-hypothesis property is gained. Each particle is an instance of (3.32) where there is one estimations per landmark summing up to a total of $N$ estimations which is the number of landmarks. Each landmark estimation is implemented as EKFs. Thus, if there are $M$ particles in the particle filter, $N \cdot M$ Kalman filters are used.

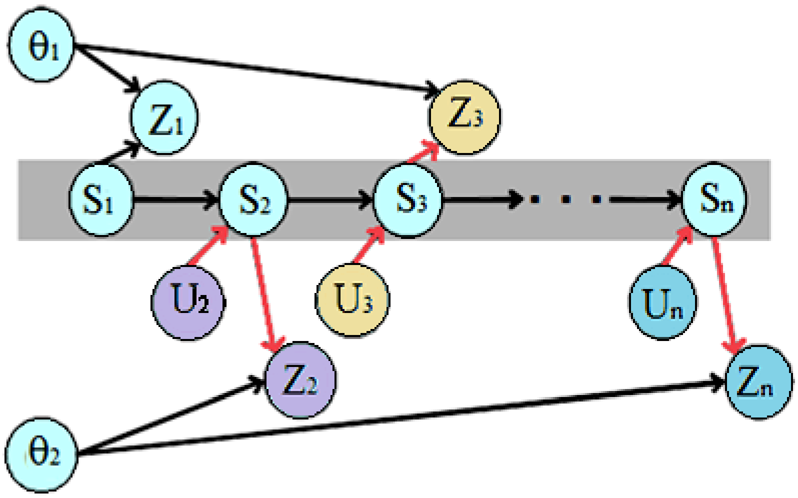


**Figure 9: D-separation in SLAM Network [21]**

The particle filter and EKF being the basis, FastSLAM is implemented in four recursive stages. Figure 10 shows the basic steps of FastSLAM algortihm. In first step, classical EKF prediction step is performed for each particle. The next robot pose is estimated based on the previous pose and the control data. In the second step, standard EKF correction step is performed for each particle where the robot pose estimation is corrected based on the sensor observations. Up to that point, everything is equivalent to EKF steps except for the fact that not only one EKF is implemented but $M$ EKFs are implemented simultaneously. This gives the FastSLAM algorithm the "*multi-hypothesis*" feature. Because, at that step, the algorithm possesses $M$ alternative beliefs about the landmark locations to choose from. That is what the algorithm actually does in the third step, it selects the particles with less uncertainty as the better samples by giving weights to each particle based on how certain they are with respect to the map. Then in the last step, the algorithm regenerates the particle population such that there are more of the better fit particles and less of the worse fit particles in the next generation of particle population.

```
1.   Sample a new robot path given the new control
2.   Update landmark filters corresponding to the new observation
3.   Assign a weight to each of the particles
4.   Resample the particles according to their weights
```

**Figure 10: Basic FastSLAM Algorithm [21]**

As in Monte-Carlo approaches and genetic approaches, the number of particles provides the control on how close the estimation will asymptotically get to the truth. As expected, number of particles determines the tradeoff between computational cost and error performance. Figure 11 displays an overview of the FastSLAM data structure.

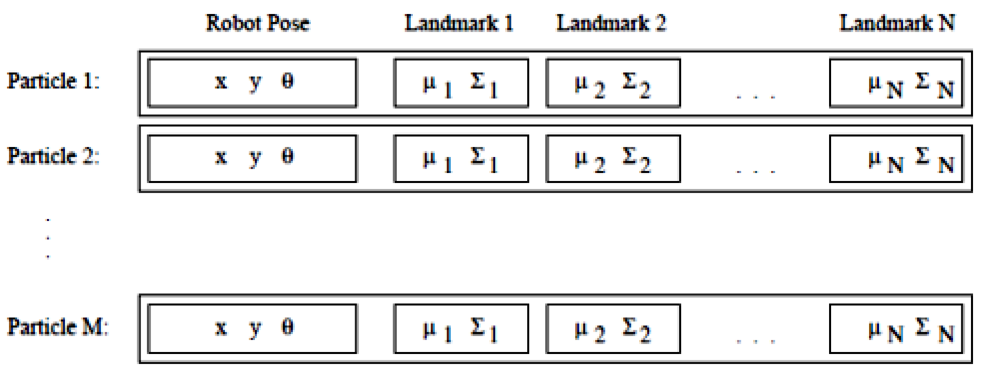


**Figure 11: FastSLAM Structure [21]**

# CHAPTER 4

# 3D SURFACE CURVATURE FEATURE EXTRACTION FROM RANGE DATA

Before the introduction of reliable and affordable 3D range sensors such as TOF cameras, laser scanners, Kinect-style RGB plus depth sensors, SLAM studies were mostly based on 2D vision data which is 2D projection of the 3D world onto the image plane. Although there are quite an amount of successful SLAM work with 2D data there are still limitations as compared to the range data like sensitivity to occlusion and lighting conditions. Thus, the studies began to incline to be based o range data which represents 3D world significantly better. The first appearance of Kinect data in a SLAM implementation is seen in [26]. Then the use of Kinect and other range sensors constantly increased. As put in the state-of-the-art section of the Introduction chapter, there exist many proven work in the literature that uses range data in SLAM.

The most extensively used methods are usually based on planar feature extraction as in [6] [25] [24]. However, in SLAM literature, there is no implementation example of surface curvature features other than planar features. This is actually the motivation for this thesis study.

Related work on 3D surface features includes appearance based techniques [72] [73] [74] [75], object silhouette based methods [76] [77], 3D correlation methods [78] [79], exhaustive search techniques [80] [81] and the most notable family of methods which are the local surface descriptor based methods [82] [83] [84] [85] [86] [87] [88] [89]. In SLAM and other robotic applications, expectations from features are basically robustness, saliency and to be automatic especially if the robot is expected to operate in unknown environments. Thus, in this thesis, a multi-scale 3D surface feature extraction method is adopted which is a local surface descriptor based technique. In this technique, local surface curvature is considered as the descriptor.

## 4.1 Review and Analysis of the Surface Feature and Curvature Feature Detection and Description Techniques for Use in SLAM Applications with Range Sensors

In this section, the foundations of the proposed feature detection and description technique will be explained. The results of the relevant work will be analyzed to demonstrate the feasibility of the adopted methods in our problem of data association improvement in SLAM problem. To the best of our knowledge, the most relevant studies are referenced in this section; however, there is not a direct application of surface level curvature feature utilization evaluated in SLAM applications.

To have an overview of the related work from the literature on the use of curvature features and compact surface features in the data association process in general or within the SLAM algorithm, we will briefly mention the methods that are presented in more detail in the literature chapter. These studies are the considered as reference for the determination of the data utilization process in our thesis work.

In [34], 3D curvature features are explored as 3D key points in the object recognition problem. In this method, mean curvature, Gaussian curvature, shape index and curvedness values of the surfaces are calculated and the maximal points within the fixed patch regions are detected as key points. Around the detected key points, feature descriptor vector is defined as the shape index which is a value in the range [0, 1] and the surface normal vector difference between the key point and its neighborhood which is a cosine value. This feature detection and description method based on surface curvature properties is tested on a real 3D object dataset under the rotation, translation and noise conditions. The key performance factor is defined as repeatability which is decided as the stability of the percentage of the feature points of the same object under varying conditions. Figure 12 summarizes the results from which it could be drawn that surface curvature features displayed a repeatability rate at the levels of %80 to %90. This result is accepted as a reference for the repeatability of our feature utilization system for data association problem in SLAM.

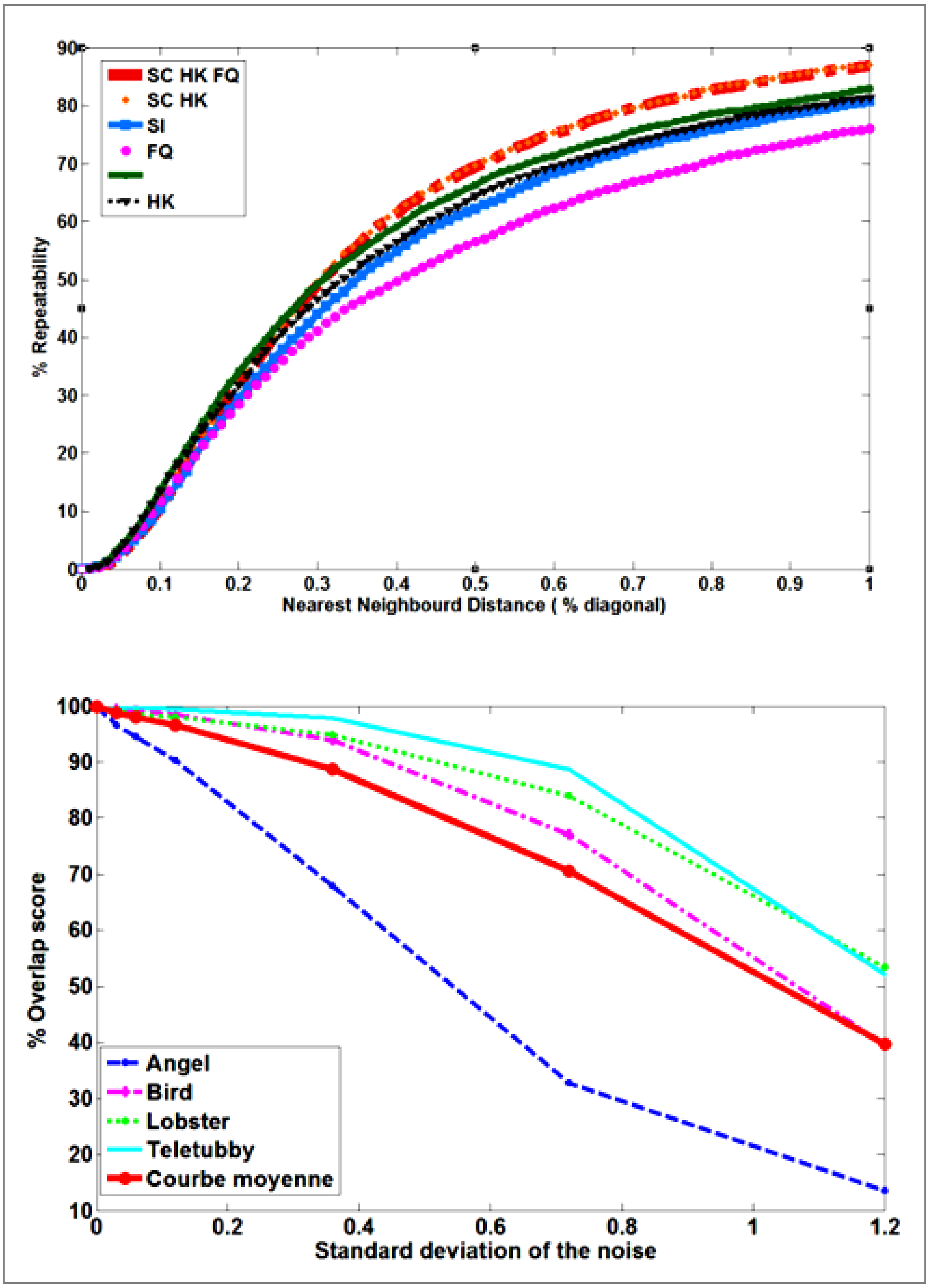


**Figure 12: The repeatability analysis of the curvature feature descriptors under viewpoint change and noise conditions in [34]**

An evaluation of local feature description techniques including the surface curvature features is presented in [66]. In this study, the feature descriptors are compared based on two criteria one of which is the deviation of the feature values and the other is the distinctiveness which is defined as the ability to maintain the matches consistent before and after the change of viewpoint and error conditions. Figure 13 shows the respective results. The blue bars and green bars represent the spread of feature values and the rate of false matches under view point changes where the smaller the bars the better the descriptor performance. Thus mean curvature, shape index and curvedness index descriptors are observed near the top performances which again supports the surface curvature approach in the proposed system.

As our aim is to exploit the robustness and the invariance of surface curvature features in a compact manner, we will now review the compact surface features as used for data association problems as in SLAM. The closest approach in the SLAM literature is the planar feature approach as compact surface representation. On the grounds that planes are the less complex form of curved surfaces, it is safe to investigate the related work on the planar surface feature utilization for SLAM as far as the compact surface concept is concerned. A quite recent work uses planar surface features is reported in [44]. In this method, the scene is segmented into planar and non-planar regions where both of the groups are registered as landmarks; however, the latter group is treated as object candidates. Plane feature descriptions and the location and bounding box properties of the objects are augmented into the SLAM state and used for global matching during the loop closing stage. The objects mentioned here are the outcomes of the filtering from the candidates by semantic criteria such as reasonable size and spatial variance in order to distinguish from plain noise. When we consider the fact that what our study aims to achieve is the more generalized form of this method that is additionally capable of detecting and describing more complex curved surfaces such as quadratic surfaces. As the authors emphasizes in Figure 14, the most important conclusion to draw for our proposal is the success of the compact surface feature landmarks in the loop closing stage which yields the global matching for SLAM.

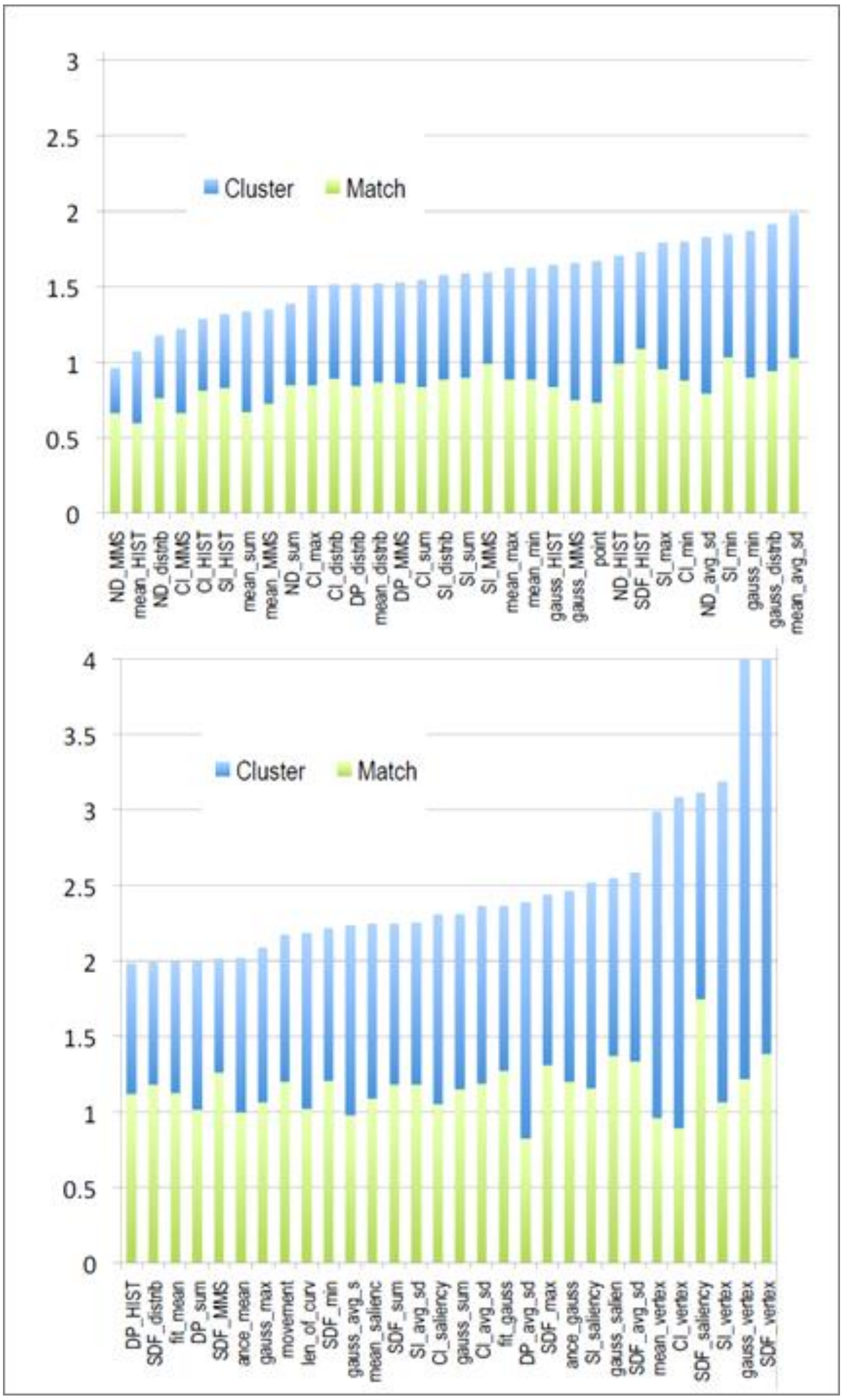


**Figure 13: The deviation and distinctiveness analysis results, green bars represent the deviation of feature values from the mean (smaller is better) and blue bars show the tendency of mismatches (smaller is better) [66]**

Another similar approach is documented in [45] which directly focuses on the benefits of compact surface representation in SLAM applications. The planar surface features are described by a 9-element feature vector that consists of three position elements and two orthonormal vectors to the plane normal of both three elements. These features are used for loop closing process of the SLAM. The authors emphasize that by utilizing compact surface features in SLAM, comparable error performance could be achieved as compared to computationally more complex point feature based techniques. This result is also a source of confidence for the compact quadratic surface feature approach adopted in our work.

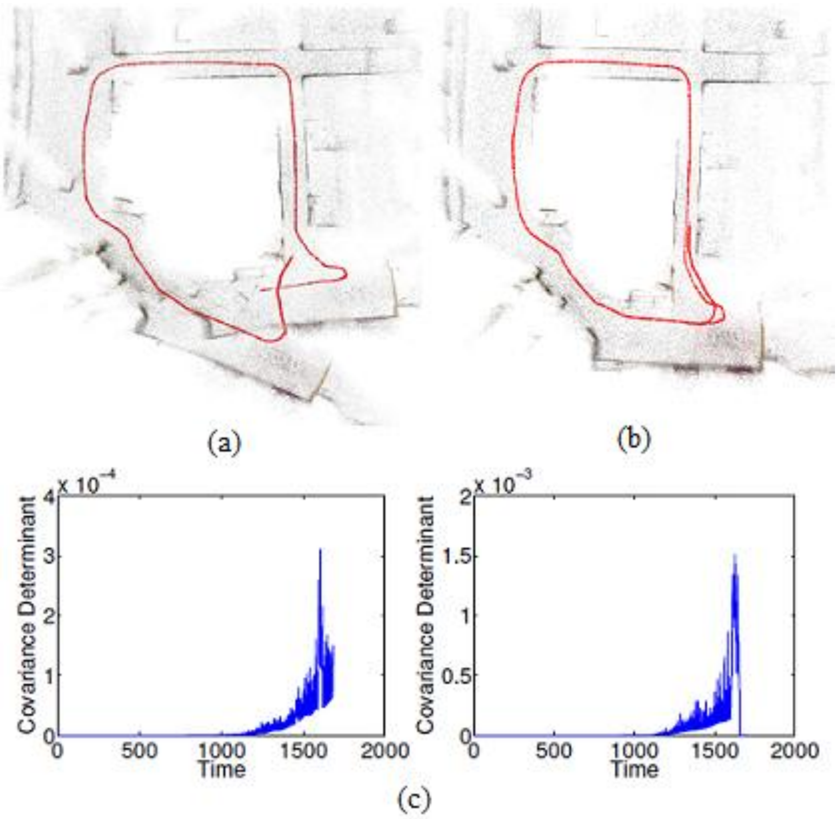


**Figure 14: (a): Robot trajectory without loop closing with landmarks (b): Robot trajectory with landmarks for loop closing (c): The respective covariance determinant values as an indication of errors [44]**

Based on the analysis and results in the literature, our system design relies on the surface curvature feature description techniques for a discriminative and stable feature matching and prefers to utilize the features in the higher level surface approach to achieve compactness which serves to save from computational cost for the ultimate real time prospects that any robotics application should consider. The repeatability analysis for the feature detection and description processes of our proposed system will be investigated thoroughly in Chapter 6 which is dedicated to this analysis.

Thus, the performance analysis is left aside for the relevant chapter and in the following sections the details of the feature detection and description techniques adopted are explained.

## 4.2 Local Surface Curvature

Local surface curvature is determined by the amount change in the surface normal from one point to another. In other words, local surface curvature indicates how the surface bends around each point on the surface. We can define a "*shape operator*" that yields the amount of change in the surface curvature change between two points on the surface. If we let $M \in R^3$ be a surface and $n$ be the normal vector to the surface at the surface point $p \in M$, then a shape operator can be defined using directional derivative $D_v$ as given in (4.1).

$$s(v_p) = -D_v n \tag{4.1}$$

$D_v n$ is the directional derivative of $n$ in the direction of $v_n$ around point $p$ and is defined as in (4.2)

$$D_v n = \lim_{h \to 0} \frac{n(p + hv_p) - n(p)}{h} \tag{4.2}$$

It is obvious that such a shape operator would yield zero if the surface is ideally planar. Using the shape operator defined, the sharpest changes in the surface normal can be found which is named as the principle curvature at a point on the surface.

## 4.3 Principal Curvatures

Principal curvatures are the extremal changes in the surface normal at a point. This determines the dominant curvature directions for a point on the surface. Let $u_p$ be the tangent vector for the point $p$ on the surface $M$. Assume further that $\|u_p\| = 1$ then the normal curvature along $u_p$ can be defined as in (4.3).

$$k(u_p) = s(u_p) \cdot u_p \tag{4.3}$$

The critical points for $k(u_p)$ are found as $k_1$ and $k_2$ corresponding the maximum and minimum values. These points are defined as the "*principal curvatures*" representing the most and least curvature values. Unit vectors $e_1$ and $e_2$ yielding the maximum and minimum values of the normal curvature are the "*principal directions*" leading to the definitions mean curvature $H$ and Gaussian curvature $K$.

$$k_1 = H + \sqrt{H^2 - K} \tag{4.4}$$

$$k_2 = H - \sqrt{H^2 - K} \tag{4.5}$$

Geometrically shown in Figure 5, $k_1$ and $k_2$ can be viewed as the vectors showing the directions at which the surface is bending the most positively and negatively.

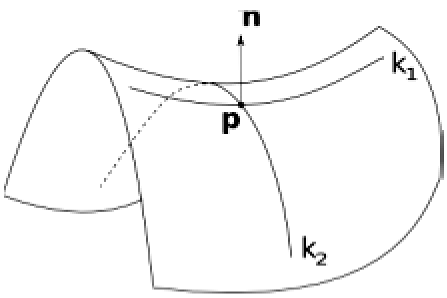


**Figure 15: Principle curvatures at point p [28]**

### 4.4 Shape Index and Curvedness

Additional high-level measurements can be drawn from the principle curvatures such as shape index and curvedness. Shape index yields a measure of how convex/concave the surface is and is given in (4.6).

$$S_I(p) = \frac{2}{\pi}\tan^{-1}\frac{k_1 + k_2}{k_2 - k_1} \qquad (4.6)$$

From (4.6) it is seen that the shape index can take values within the range $[-1,1]$, however, if the surface is planar then shape index becomes undefined as both of the principle curvatures are zero. Shape index is an indicator for the surface curvature that contains both mean and Gaussian curvature.

Similarly, another measure of surface curvature that reflects the combination of two principle curvatures is the "*curvedness*" feature. Curvedness represents the bending energy for each point on the surface and is given as in (4.7).

$$c_p = \sqrt{\frac{k_1^2 + k_2^2}{2}} \qquad (4.7)$$

The curvedness yields the strength or sharpness of the surface curvature. Curvedness is a more informative measure then the Gaussian curvature which vanishes on parabolic edges. Curvedness value is lost only on planar regions. As far as the feature extraction is concerned, shape index and curvedness features are affine invariant [40].

### 4.5 Calculation Local Differential Properties

There are numerous techniques to approximate the local curvature features like mean and Gaussian curvature and curvedness [90]. Among them, the method in [91] is widely accepted and has a very close estimation for local curvature feature values. In this method, a "*jet*" which is a truncated Taylor expansion is fitted to the surface patch. This enables the surface curvature features to be analytically available. The approximation is determined by the performance of the fitting.

In [92], it is shown that any continuous curve or surface can be represented as a "*height function*" graph as given in (4.8) for the point $p = (x, y, z)$ in 3D space.

$$f(x) = J_{B,n}(x) + O(x^{n+1})$$
$$x = (x, y), \qquad z = f(x), \qquad O(x^{n+1}) = HOT \tag{4.8}$$

$J_{B,n}(x)$ is the $n^{\text{th}}$ order Taylor expansion of the height function given in (4.9).

$$J_{B,n}(x, y) = \sum_{k=1}^{n} \sum_{j=0}^{k} B_{k-j,j}\, x^{k-j} y^{j} \tag{4.9}$$

where,

$$B_{k-j,j} = \frac{1}{(k-j)!\, j!} \frac{\partial^k f(0,0)}{\partial x^{k-j} \partial y^{j}} \tag{4.10}$$

The given Taylor expansion represents the tangent plane with the first order features. Second order features correspond to the principal curvatures. The third order features could be an estimate to the directional derivatives of the principal curvature lines. The equation for the jet has a canonical form and is given in (4.11).

$$J_{B,3}(x, y) = \frac{1}{2}(k_1 x^2 + k_2 y^2) + \frac{1}{6}(b_0 x^3 + 3b_1 x^2 y + 3b_2 x y^2 + b_3 y^3) \tag{4.11}$$

Where, $k_2, k_2$ are the defined principle curvatures, $u = (b_0, b_1), v = (b_3, b_2)$ are the directional derivatives of the principle curvatures along the curvature lines. This fitting method is applicable to not only the mesh surfaces, but also to the point clouds.

### 4.6 Curvedness Feature Extraction

Given a 3D surface in the form of either a mesh or a point cloud as in this thesis study, we can extract salient features based on the curvedness measures of the 3D

surface. The idea is to choose the points with extreme values of curvedness. This way, the sharpest and the most robust feature points are obtained.
In order the feature extraction to be multi-scale, the size of the patch that constitutes the neighborhood of the point is considered. The size of the patch for the curvature measurement is chosen in multiple values by selecting the amount of neighboring depth image pixels in our case. Each surrounding array of pixels is named as “rings”. If we set ring, $r = 1$, that means the curvature around the point is calculated within the 3x3 patch from the depth image. Similarly, if $r = 2$, 5x5 depth image patch will be used as shown in

Figure 16.

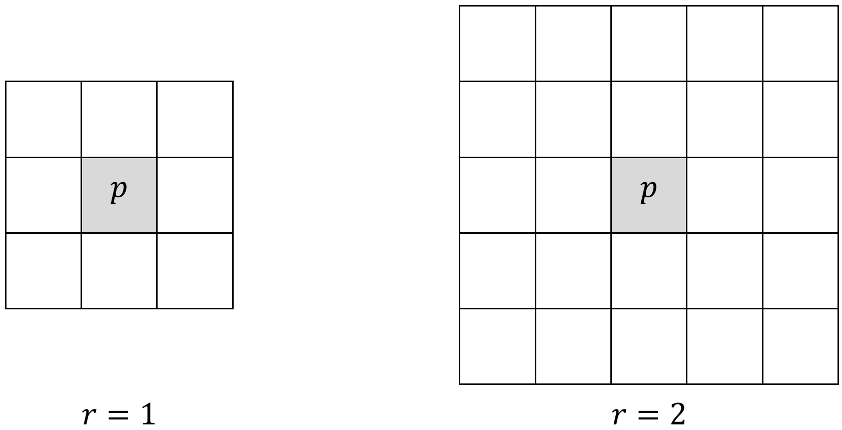


**Figure 16: Example of scale selections with r=1 and r=2**

In general, the curvedness-based feature extraction algorithm is given in [28].
The multi-scaling process is similar to many other techniques in the main principle. In Hessian matrix-based feature extraction methods such as SIFT and SURF, the scale spaces are constructed by changing the filter window size which is analogous to the changing of the neighborhood rings in the curvature feature extraction technique. The multi-scale approach was first implemented for the surface curvature extraction method, however, is later discarded as the benefits of it to the system is outweighed by its additional computational cost. This could be an expected result in the sense that, unlike methods like SIFT and SURF, our method does not rely on any texture

detail on the images; but is purely based on surface curvature, i.e. the relative positions of data points with respect to each other within each segmented quadratic surface. Thus, this relative distribution of data points is, to some extent, stable whether the surface is observed in the near vicinity or further away. Hence, with this intuition and the results of experiment, the multi-scale approach was discontinued for the feature extraction process.

---

**Algorithm** Multi-scale Curvedness-based Feature Extraction Algorithm

---

**Data:**

$\mathbf{P} = \{\mathbf{p}_i \in \mathbf{R}^3\}$: set of 3D points sampled from the surface.

$\mathbf{R} = \{r_k\}$: a set of scales.

**Algorithm:**

**for** $r \in \{r_k\}$ **do**
  **for** $\mathbf{p} \in \{\mathbf{p}_i\}$ **do**
    Find the neighbourhood $N_r$ at scale $r$
    Fit a jet to $N_r$
    Compute principal curvatures $k_1$ and $k_2$
    Compute the curvedness $c_{\mathbf{p}}$

$$c_{\mathbf{p}} = \sqrt{(k_1^2 + k_2^2)/2}$$

  **end for**
  Features are positions $\mathbf{p}$ having extremum values $c_{\mathbf{p}}$ both in the neighbourhood of radius $r_k$ as well as over the above and below scales $(r_{k-1}, r_{k+1})$.
**end for**

**Figure 17: Multi-scale Curvedness-based Feature Extraction Algorithm [28]**

# CHAPTER 5

# PROPOSED SYSTEM FOR 3D SURFACE CURVATURE FEATURE BASED SLAM

In this thesis, the aim is to implement 3D SLAM algorithm with 3D surface curvature features about which there is no existing study in the literature yet. Thus, the main focus is not the SLAM algorithm itself, but the use of 3D surface curvature in SLAM. Thus, the SLAM part of the implementation is considered as a benchmarking tool and is based on the existing state-of-the methods [93] [94].

In the implementation, a real world dataset is used which is TUM (University of Technology, Munich) SLAM Benchmark Dataset [6]. This dataset is recorded with a Kinect sensor around indoor environments. The choice was motivated by the thought that both planar and curved surfaces should be present in the dataset as the surface curvature feature extraction method is compared against other methods including ones with planar features. Also, in order to observe the loop closing behavior, both of the sequences complete a loop path. The availability of the ground truth information for the robot path is another consideration which important for the sake of reliable error measurements. It is important to note that the supplied ground truth is a proper information obtained from external camera measurements which is significantly more reliable than some other ground truth assumptions such as the output of state of the art estimation techniques.

**Figure 18: TUM RGBD SLAM Dataset Scene Samples [6]**

## 5.1 Overview

The proposed system could be considered as a generalization for planar feature based SLAM systems. Our solution is designed to use a more general surface representation which is not only able to utilize planar surfaces but also curved surfaces.

The introduction of more types of surfaces into the SLAM algorithm is actually quite intuitive. From the perspective of humans, who are naturally intelligent mobile agents, to learn the whereabouts and remember previously visited places is mainly based on matching some distinctive features from the environment. Planes are distinctive in this sense, as humans, we tend to remember notable planes in our vicinity. However, if we consider a large ball in the scene, that is also quite memorable for humans, maybe even more distinct from the planes due to the fact that planes resemble each other whereas more complicated surfaces such as a spherical surfaces are less likely to have that level of similarity.

The complexity of the surfaces of interest is also an important issue to consider. If we once more consider as a human, it is more difficult to remember a terrain surface with many complex and unorganized surface patches, however, it is quite easy to match a previously observed bowl on a table.

Thus, in this SLAM system, the extracted surfaces from the environment are determined as quadratic surfaces. This provides a suitable tradeoff between surface distinctiveness and complexity.

The flow of the system begins with the introduction of motion data from odometry. The next state prediction is done based on the odometry data and the motion model. FastSLAM initializes the particles with the first prediction belief values. In the next step, the sensor data is processed. First, the raw data from the scene is clustered via hierarchical clustering method. The clustering method has to be unsupervised in order to have a fully automated system which is able to work in unknown environments. The surface patches obtained by hierarchical clustering are then fit to an algebraic quadratic surface. The fit is either accepted or declined based on the goodness of the surface fit. If the fit is sufficiently accurate, then the surface patch is approved as a quadratic surface. At that stage, the analytical parameters of the surface which represents the extracted patch are available. That means any mathematical feature can be drawn from the surface. In this work, Mean and Gaussian curvature based features are considered and calculated by making use of differential geometry calculations. The extracted surface features and the location of the surface constitute the landmark feature vector.

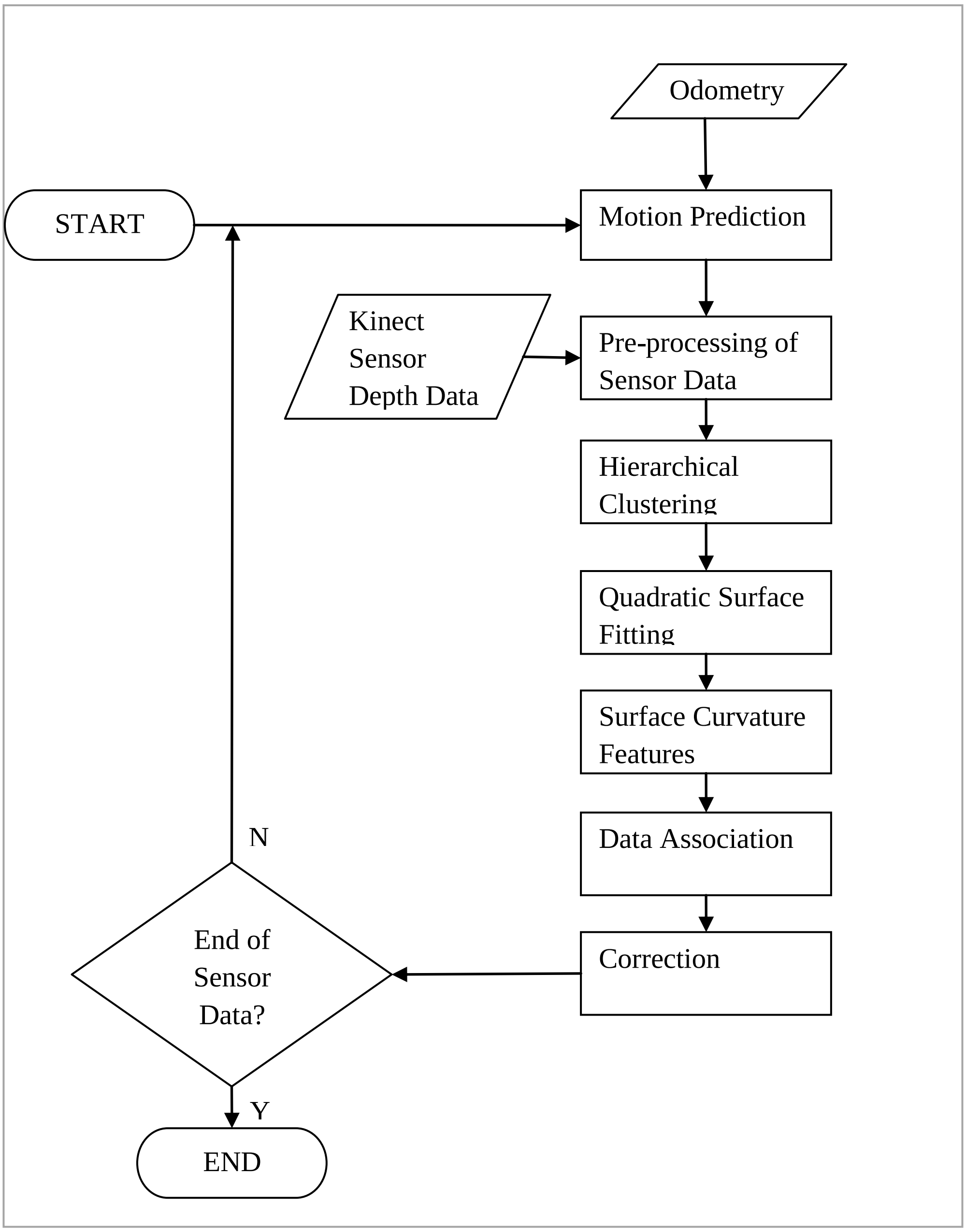


**Figure 19: Overview of the Proposed System**

In the update step of the FastSLAM algorithm, the feature vectors representing the landmarks are compared against those of newly observed surfaces in search of possible matches. This is the data association step of the SLAM algorithm. After that step, the prediction and update steps are repeated till the end of sensor data, updating and lowering the errors iteratively. The overall process is depicted in Figure 19.

In remainder of this chapter, the details of implementation will be given for each section of the algorithm.

## 5.2 The Environment

The proposed system is implemented in MATLAB on real Kinect data from TUM SLAM Benchmark Dataset [6]. FastSLAM implementation is based on [93] and [94]. which is considered as the state of the art for our surface curvature feature SLAM assessment.

## 5.3 The Motion Update

The Kinect dataset does not provide sufficient odometry data which leaves no choice but to induce artificial errors in order to have a motion prediction. Thus, in this implementation, the odometry is obtained by noise injection.

## 5.4 Pre-processing of Sensor Data

Kinect output is considerably noisy, however, as can be seen in the amount of work dealing with Kinect data, the noise levels are in a manageable range. Also, our method contains surface fitting procedure which implicitly has smoothing effects. Thus, there is not much need for a complicated pre-processing. Only a series of median filtering and mean filtering is applied with a local window of $5x5$ neighborhood.

## 5.5 Hierarchical Clustering

A sample Kinect sensor output is given in Figure 20. The scene is relatively sparse; however, the need for clustering the image into meaningful patches is apparent.

**Figure 20: RGB and Depth Output of Kinect Sensor [27]**

Hierarchical clustering method was chosen due to the assumption that there is no *a priori* information available about the environment. Hence, we can only make use of connectivity property of man-made objects in the scene. The connectivity information can be captured if the distance metric of the clustering is chosen accordingly, which is usually the Euclidean distance for point sampled data. For the depth data given in Figure 20, the corresponding dendogram and the labeled data points are given in Figure 21 and Figure 22, respectively.

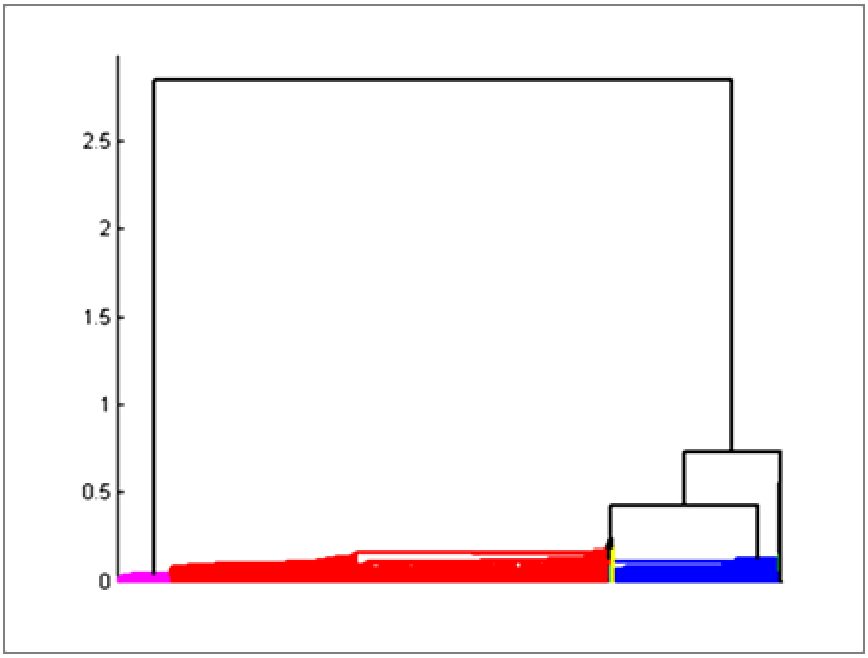


**Figure 21: Dendogram for the Hierarchical Clustering**

Dendograms display in a tree-like structure the discrepancy of the labeled points in the clustered data. There are in fact 9 clusters although not visible in Figure 21. This is due to some clusters being too small and lay within small regions like the yellow region in the dendogram. Such small regions, along with other regions with certain properties are eliminated from the data after the initial clustering. The properties of the regions to be removed are as follows:

- Regions with less number of point below the threshold,
- Regions that have high depth variance,
- Regions that are further away from the Kinect sensor reliable range,
- Regions that are not fully observed within the frame.

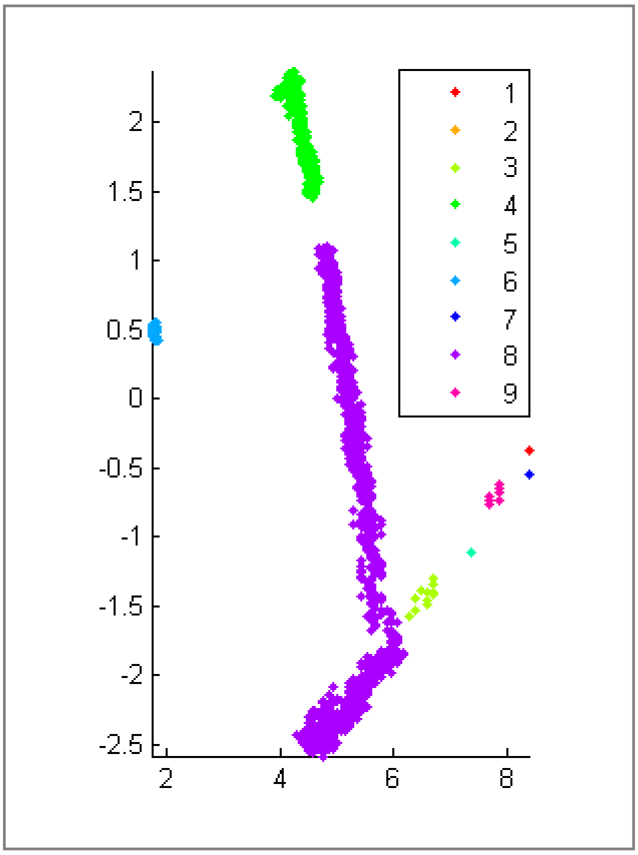


**Figure 22: Labeled Point Clusters**

Among the clusters seen in Figure 22, only the blue and green labeled clusters remain due to the aforementioned constraints. The implementation of hierarchical clustering algorithm is included within the MATLAB software; however, the design of the clustering still must be done through the parameters such as distance metric, cut-off threshold and neighborhood linkage. The distance metric selection was already done as Euclidean distance due to the nature of the Kinect data. For cut-off threshold, the best way is to observe the experimental results as the calculations for the ideal cases would be distorted by the noisy measurements. The cut-off threshold is thusly determined as 0.2 meters.

### 5.6 Compact Feature Representation Approach

In the proposed system, compact higher level feature representation for the sensor data is the core of the design criteria. Some of the benefits of using higher level features for data association in SLAM applications are studied and explained in [45] and [44] as well as in the sections of this thesis which are about the literature survey and results of the experimentation. Briefly, the use of such features provides more robustness due to the inclusion of a large amount of data points and their correlation, better computational efficiency thanks to the more compact feature vector to represent large regions, more distinct features for better matching performance and a semantically more meaningful representation that helps both the mapping processes and human understanding of the scene. The success of the compact planar feature representations as in [25], [44], [47] and [48] has inspired this thesis study to advance further into the realms of compact feature representation by the generalization from planar surface features towards quadratic surface features which allow to exploit the differential geometry of the surfaces with no loss of the ability to represent planar surfaces. The motivation is to extend the compact higher level features by making use of more discriminative and more complex surface features, namely the curvature properties. This extension is expected to improve the data association performance in two main ways; by the inclusion of curved surfaces into the SLAM features that were previously left out due to the planar surface limitations and by utilizing the distinctiveness of the curved surface features as compared to the planar surfaces which are more likely to lead to possible false positive matches as the planar surfaces usually resemble each other.

## 5.7 Quadratic Surface Fitting

Quadratic surface fitting is used in the state of the art techniques for surface curvature calculations as in [95] [96] [97] and many others. The main idea is to first fit a quadratic surface to the clustered data points, and then calculate the surface curvature features analytically via the fitted algebraic surface.

Quadratic surface fitting lies at a good point between memorization and generalization of the surfaces. One degree less means to have only planes and one degree more results in overly complex surface that are not suitable for matching purposes. For general purpose indoor applications, the surfaces are expected to have regular surfaces like planes, ellipsoids, cones, cylinders, spheres, hyperbolas and so forth. Such surfaces can be perfectly represented by quadratic surfaces which are algebraically polynomials of second degree. It is also preferable in terms of computational cost. Polynomial fitting of second degree is quite computationally effective, so that even the MATLAB implementation with thousands of points executes instantly. The second degree polynomial fitting for the quadratic surfaces is pretty much the finished article in the literature. Almost all methods adopt least squares minimization approach. In our system, this efficient method is used.

To have a better representation from the very beginning of the SLAM algorithm prevents from the further introduction of error. Figure 23 is a very clear example of the difference between the fitting performances. In the first image, blue cloud is the surface points and the grey thin line is the edge of the fitted plane. The second image displays the same surface data points with green dots and the fitted quadric surface as the dark sheet. It is worth noting that even with near planar surfaces, quadratic fit provides an apparent improvement. Although the data points have a low level of curvature, quadratic fitting is apparently less erroneous. If this is a very large group of points, even that much of a difference could make up to meter level error margins. Also, if the curvature of the surface is higher and higher, the planar fitting will be less and less accurate and in turn it will be unable to fit to a plane.

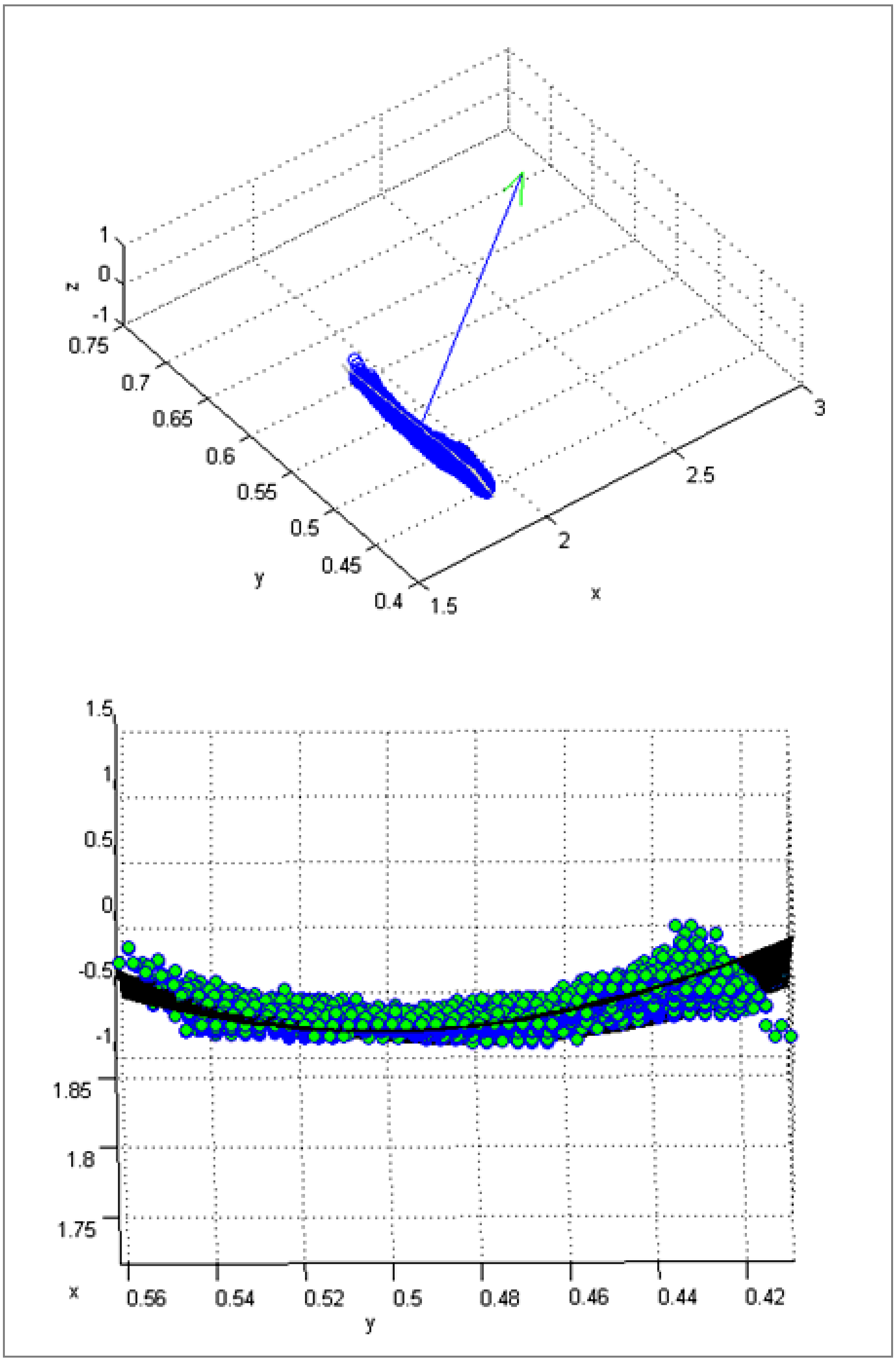


**Figure 23: Planar and Quadratic Surface Fitting results for the same data (X, Y and Z-axis values are coordinates in meters)**

Our method addresses this issue by generalizing the surface representation from planar only to quadratic surfaces. Although it is possible to fit deeper with cubic representations and higher, it is not feasible to try and match such overly complex surfaces. In fact, that much of accuracy could be deadly in the sense that it could fit surfaces to combination of several objects which stay in close distance. This is not a failure mode in the quadratic fitting as the goodness of the fitting performance is a constraint of feature extraction. So, if the surface is not fit properly, in other words, the underlying geometry of the surface points cannot be represented by second degree polynomials, and then the SLAM algorithm will simply ignore this measurement. This is an *admissible* approach in the sense that we try to minimize the possible additional computational noise.

The mathematics of the least squares minimization for the fitting is given in Appendix C and could be further reviewed from [95], [96], [97].

The result of quadratic surface fitting is a 6 parameter vector that algebraically describes the surface as given in (5.1).

$$z(x,y) = p_1x^2 + p_2xy + p_3y^2 + p_4x + p_5y + p_6 \tag{5.1}$$

The vector returned from the fitting algorithm is given in (5.2).

$$p = [p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5 \quad p_6] \tag{5.2}$$

The availability of an algebraic representation makes any mathematical manipulation possible for the surface. The curvature calculations are analytically calculated based on those parameters.

## 5.8 Surface Curvature Features

The surface curvature features are implicitly defined by the derivatives of the surface. The exact calculations of Mean Curvature and Gaussian Curvature are given in (5.4) and (5.5), however, approximated calculations yield sufficiently close values. Hence, the calculation of Mean and Gaussian curvature features is given in (5.3) to (5.5).

$$d_u = p_4$$

$$d_v = p_5$$

$$d_{uu} = 2p_1 \tag{5.3}$$

$$d_{vv} = 2p_3$$

$$d_{uv} = 2p_2$$

$$H = (d_{uu} + d_{vv} + d_{uu} * d_v * d_v + d_{vv} * d_u * d_u - 2 * d_u * d_v * d_{uv})/(2 * ((1 + d_u * d_u + d_v * d_v)^{1.5})) \tag{5.4}$$

$$K = (d_{uu} * d_{vv} - d_{uv} + d_{uv})/((1 + d_u * d_u + d_v * d_v)^2) \tag{5.5}$$

From Mean and Gaussian curvatures $(H, K)$ it is possible to define a curvedness value which is affine invariant [28] as given in (5.6).

$$C = \sqrt{(2 * H^2 - K)} \tag{5.6}$$

Thus, the feature vector for the clustered surface patch is now complete. The vector content is given in (5.7).

$$f = [x \quad y \quad z \quad H \quad K \quad C] \tag{5.7}$$

The use of Mean and Gaussian curvature values for surface matching is a preferred method in the literature for 3D surface matching. Nevertheless, the use of these features in SLAM algorithm is the novelty of this study. Our aim is to observe the results for this elegant surface matching technique in the SLAM domain in which the range sensor data is vital and to have an improvement on sensor data utilization could prove quite useful.

## 5.9 Data Association

This work is based on unknown correspondence case of SLAM which means that it is not known which sensor measurement is associated to which previously observed landmark. This relation is found during the algorithm through the state covariance matrices. Although data association problem could be seen as simply comparing the measurement feature vector against all the known landmarks, the difficulty lies in the quality of the comparison. It is not straightforward to decide whether an object is equal to another as in the case of numbers. The key factor was the surface curvature feature extraction section. The more descriptive and discriminative the feature vector, the better the data association will be.

Data association concept is less of a problem with FastSLAM which adopts multi-hypothesis belief propagation. This leaves a safety margin for data association errors. However, with less number of particles or more sparsely sampled scans could compensate for the advantage of multi-hypothesis approach. Thus, although FastSLAM could recover from data association errors unlike EKF SLAM, the error performance is still dependent on the associating the data correctly.

## 5.10 Measurement Correction

The mathematics of Kalman update is given in detail in Chapter 3, hence will not be given again here. As far as this thesis study is concerned, the measurement update step is a straightforward implementation of FastSLAM. The main focus was to improve the features obtained from sensor measurements. The rest of the implementation solely tries to keep up with the state of the art. The novelty of this work is to use the surface curvature features inside SLAM algorithm.

## 5.11 Real Time Application Discussion

In a recent work [98], Mean and Gaussian curvature from a surface range data is estimated using FPGA technology. The benefits of low-level and parallel processing in this application, made it possible to compute the same operation in dramatically shorter execution time as compared to the high-level programming environments such as MATLAB. In this work, the range data of size 128x128 was processed and curvature values were calculated within duration of as short as 471μs which is a quite promising speed for real time considerations. If the rest of the SLAM algorithm was implemented on such a hardware then real time operation would seem practical and feasible.

# CHAPTER 6

# ANALYSIS ON THE SURFACE CURVATURE FEATURE MATCHING STABILITY

In this chapter, the repeatability and the distinctiveness of the chosen surface curvature features are analyzed based on the experimentation with the real Kinect data sequences. This analysis is carried on the grounds that although the success of the surface curvature feature matching process could be implicitly observed from the SLAM performance, it would be informative to investigate the feature matching alone as some other factors could be affecting the SLAM performance other than the feature matching process. Thus, for such an analysis, in order to isolate all other parameters, the Kinect scans are read and clustered by hand, then the quadratic surface fitting and surface curvature features are extracted. For this procedure, scans are chosen in a larger period in order to observe varying scene conditions. The details of the analysis are presented in the following sections.

## 6.1 Methodology

Basically, for feature detectors and descriptors, the measure of performance is repeatability and distinctiveness. For the first measure, the surface curvature feature elements of the feature vector, namely H, K and C elements, are tracked across the Kinect scans at different time intervals which tests the repeatability, scale and view-point invariance of the surface curvature features. For the latter measure, the distinctiveness of the surface curvature features are compared across the measurements taken from the surfaces of a bowl, a smaller glass and a planar region in order to see whether the features are able to discriminate different objects surfaces from one another. For all feature extraction processes, the segmentation of the relevant object surface is done manually by choosing the region of interest within a drawn polygon in order to eliminate all other factors other than the feature extraction process itself. In the analysis discussion, a through experimentation including many object types is not aimed as it is beyond the scope of this work and also the results of the feature matching performance are implicitly observed within the overall performance of the SLAM algorithm.

### 6.2 Repeatability Analysis

In order to investigate the repeatability of the surface curvature features, a sequence of Kinect data is chosen such that throughout the sequence scale and view-point changes occur so that different data samples collected from the same object surface.

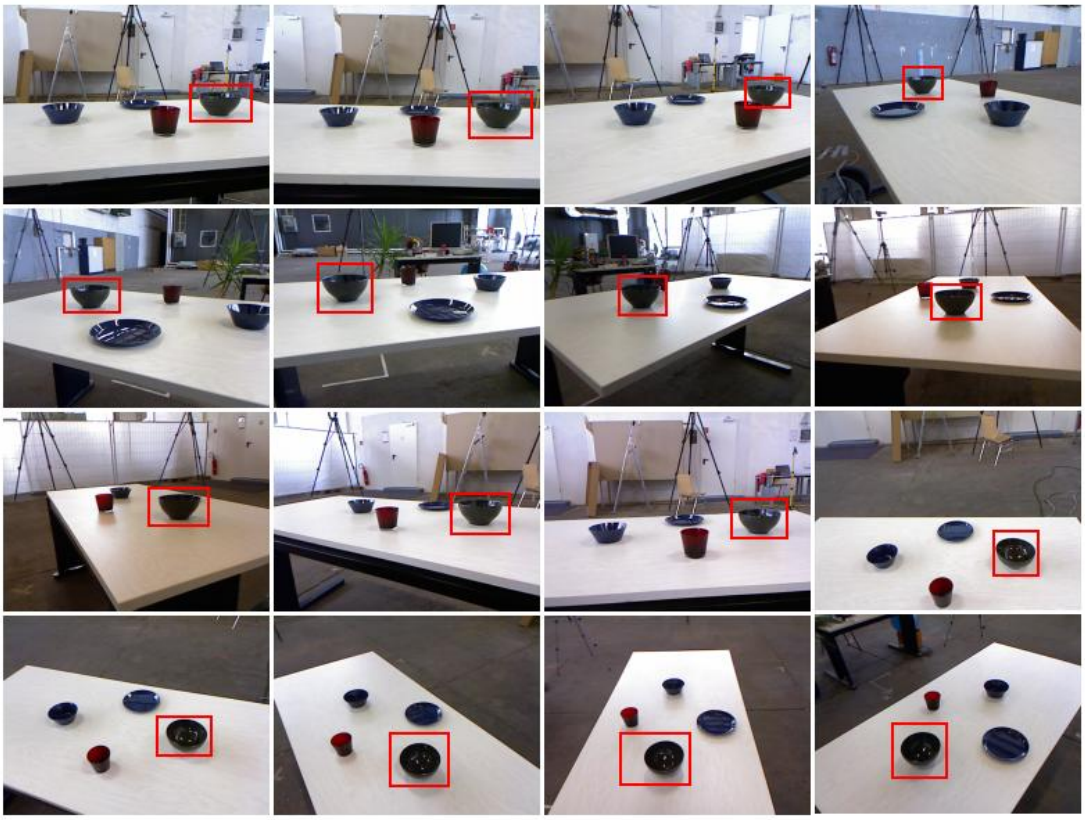

**Figure 24: Scale and view-point variation during the scan sequence [27]**

Figure 24 shows the variation of the scenes across sixteen scans at different time intervals. The tracked object surface is shown as the bowl near one edge of a table in the scene. The scale change is visible especially between the scenes that the sensor locations are at the opposite sides of the table. The viewpoint change is extensive between the scenes that the sensor displays the bowl almost from the top, sides and oblique angles. Thus, the experimentations are carried out under significant amount of scale and viewpoint changes which is able to represent the real application conditions.

The variation of the H, K and C values corresponding to the surface curvature feature elements of the feature vector is given Figure 25. To begin with, deviations in all of the feature values are quite acceptable in the sense of repeatability when the significant scale and view-point changes are taken into account. As the Gaussian curvature value, namely K, is constantly very low, the curvedness value is seen to closely follow the mean curvature value.

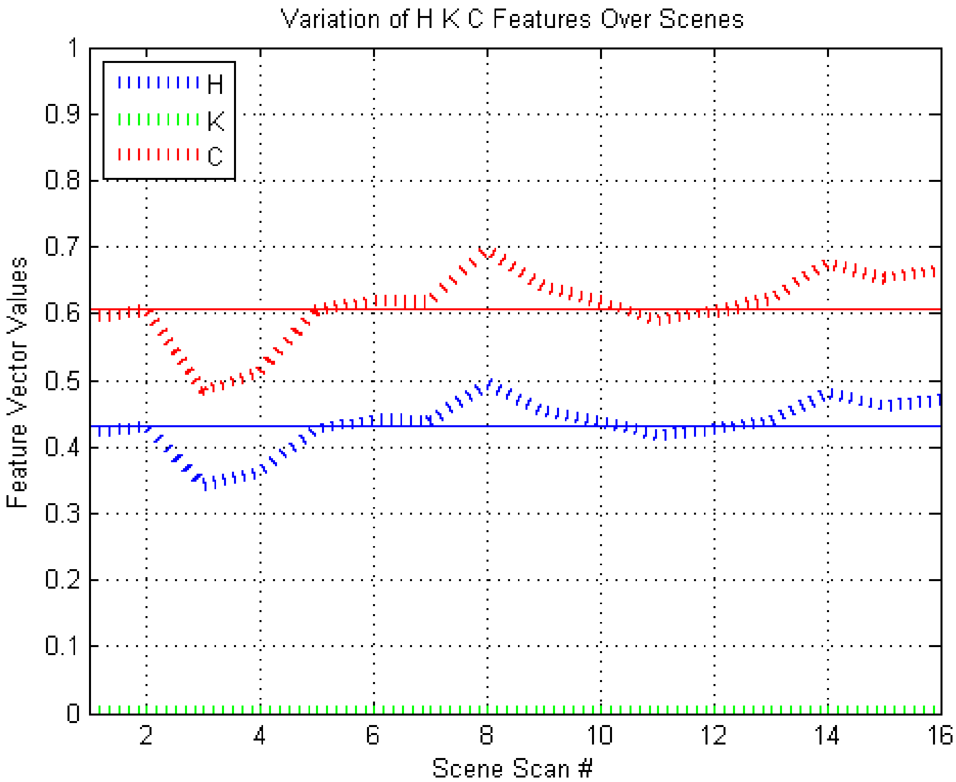


**Figure 25: The variation of H, K and C values over different scene scans**

The mean and variance for the H, K and C values measured across different scale and view-point conditions are given in Table 2.

The consistency of the values is also observed in mean and variance values of the feature values as given in Table 2. The variations of the values across the scene scans at different time instances are also seen to be stable. This stability is vital for any mobile robotics application in the sense that the mobile robot could observe the same object surface at rather different angles and distances. Still, the mobile robot has to be able to recognize that the measurements are taken from the same object surface, otherwise such data association problems accumulate and might render the estimations useless. However, in this experimentation it is observed that the feature values belonging to the same object surfaces are quite robust against significant changes in the scale and viewpoint variations. Thus, the analysis of repeatability and invariance to scale and view-point changes properties of the proposed surface curvature feature method are concluded.

**Table 2: Mean and variance of the surface curvature feature elements**

| | Mean Curvature (H) | Gaussian Curvature (K) | Curvedness (C) |
|---|---|---|---|
| **Mean Value** | 0.4306 | 0.0019 | 0.6077 |
| **Variance** | 0.0076 | 0.0000 | 0.0149 |

### 6.3 Distinctiveness Analysis

In this section, analysis on the distinctiveness of the surface curvature features is presented. For this purpose, the experimentation for the previous section containing the bowl object is compared against the same measurements taken from the table surface. The surface fitting process actually makes the difference felt even before the feature extraction as seen in Figure 26. The difference in the mean values of the surface curvature feature values also reflects a similar result as seen in Table 3.

The measurements for the smaller cup are quite noisy and it is difficult to obtain stable data for long sequences. Thus, only a rough mean value is provided in Table 3. It is noted that the table surface curvature values are near the zero level as it is a planar surface. Thus, the difference from a curved surface is quite decisive. The distinctiveness between the bowl and the smaller cup is also notable in the sense that they are similarly curved surfaces; however, the smaller cup is more curved as it has a smaller radius on the base and top regions. To conclude the distinctiveness analysis, it is observed in the experimentations that the surface curvature features are able to discriminate non-curved and differently curved surfaces seen in various scale and view-point conditions.

It should be noted that, the surface fitting operation provides a complete surface representation even in the case of missing data points by the help of the surface symmetry. Thus, if the shape of the surface is notably asymmetrical, then due to the quadratic surface assumption, the representation may have a higher fitting error. Nevertheless, common indoor objects, or in a broader sense, man-made objects usually have a certain degree of symmetry which allows the quadratic surface assumption to represent the object surfaces reliably with low fitting error values.

**Table 3: Mean and variance of the surface curvature features from bowl and table surfaces**

| | **Mean Curvature (H)** | **Gaussian Curvature (K)** | **Curvedness (C)** |
|---|---|---|---|
| **Bowl Surface (Mean)** | 0.4306 | 0.0019 | 0.6077 |
| **Bowl Surface (Variance)** | 0.0076 | 0.0000 | 0.0149 |
| **Table Surface (Mean)** | -0.0740 | -0.0006 | 0.0830 |
| **Table Surface (Variance)** | 0.0400 | 0.0003 | 0.0478 |
| **Smaller Cup (Mean)** | 1,6627 | -0,2091 | 2,3955 |

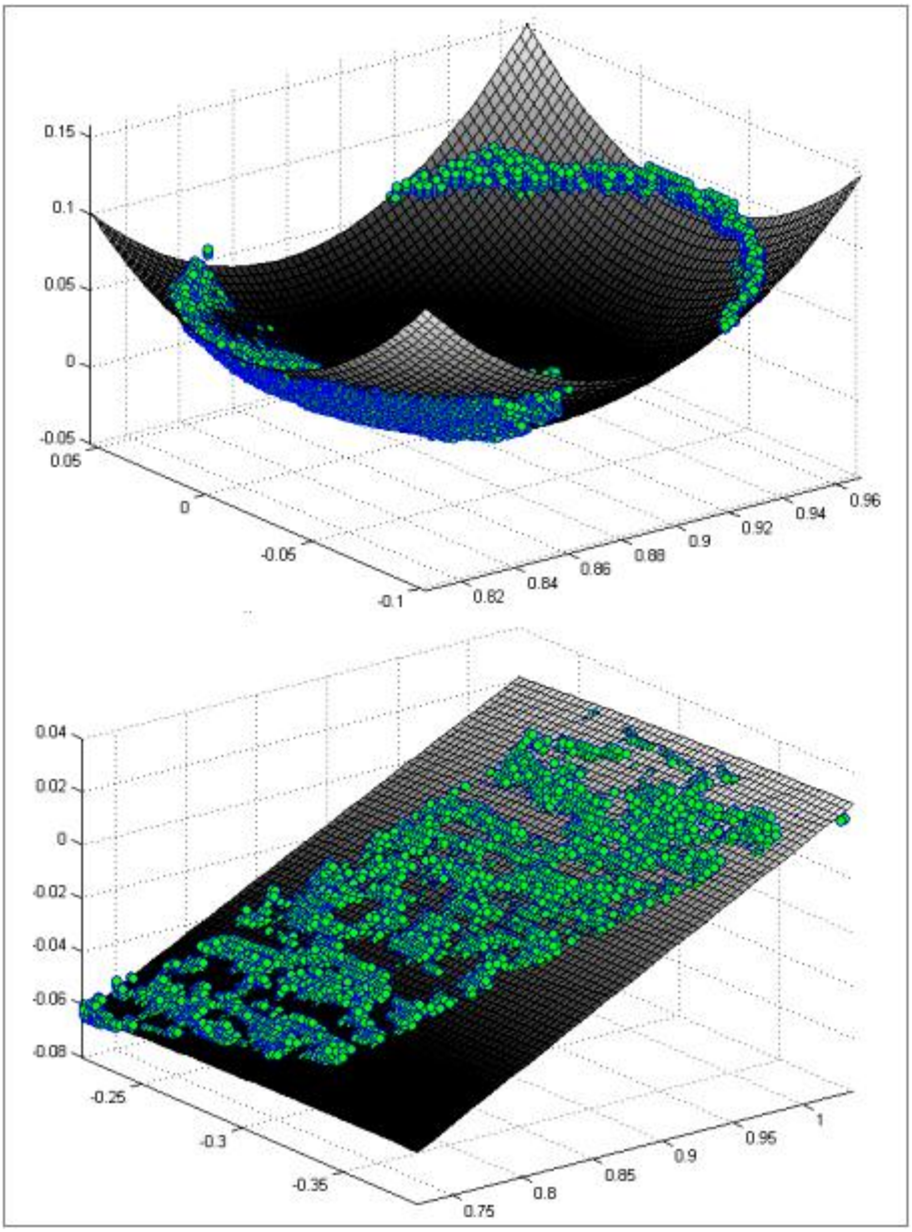


**Figure 26: Quadratic surfaces fitted to the surface of the bowl and the table (X, Y and Z-axis values are coordinates in meters)**

# CHAPTER 7

# RESULTS AND ANALYSIS

The proposed method introduces a novel approach for the feature representation in SLAM algorithm. Thus, the only way to experiment and observe the results is to compare against existing methods. The first method to compare would naturally be the plane feature based SLAM. In principle, our work could seem to be a more generalize form of a surface representation as compared to planar approaches. However, it is not as straightforward as it would sound to utilize curved surfaces in 3D space. Many algorithms working on planar surfaces would not work on curved surfaces as the constancy of planar surfaces allow numerous approximations and assumptions. That completely changes the solution approach. Although it is not possible to investigate and compare the results in every aspect of the problem due to lack of 3D Kinect dataset diversity among the publicly available resources, there is still a good chance to observe fundamental properties of the compared techniques.

In this chapter, basically the error performances and the computational cost analysis will be given. The final path and landmark estimations will be also compared against planar feature based, SURF based and corner feature based SLAM methods.

## 7.1 Benchmark Environment

In order to evaluate the performance of the proposed method, a set of SLAM feature techniques and Kinect dataset records are employed. The selected SLAM feature techniques for comparison are plane feature-based SLAM, SURF feature-based SLAM and corner feature-based SLAM methods. The planar surface based technique is a natural competitor due to the fact that the planar features are also surface features as in the case of the proposed surface curvature feature based SLAM. The other two techniques are included in the benchmarking domain in order to have not only a similar approach but also fundamentally different approaches. Point feature approach in SLAM represents the idea of having a dense landmark map which prefers to have a better error performance in expense for more computational power demand. Among such techniques, SURF and corner features are selected for both their popularity in SLAM applications and their conceptual diversity.

The experimentations are carried out on MATLAB version 2011a 64-bit running on a PC with Windows 8 operating system installed. The computer hardware features Intel Core i7-3632QM processor at 2.20GHz, 6GB RAM and 2GB Nvidia GeForce GT-635M graphics processor. The processed depth images count up to 100 frames each of which contain 640 by 480 pixels.

### 7.1.1 Dataset Used in the Experiments

Although there is a good number of publicly available Kinect dataset packages on the web, there are only a few packages which provide sufficient ground truth data, if any, a suitable path featuring loop closure and object variety in the scenes. The SLAM benchmarking dataset provided by [6] in Technical University of Munich. The given name of the dataset is actually self-explanatory: "*RGB-D SLAM Dataset*". Thus, the main purpose of the record is to provide a medium for SLAM implementation, evaluation and benchmarking. This dataset proves quite useful as it supplies reliable ground truth data constructed using external calibrated camera system which is valuable information in the sense that the ground truth is obtained independently without being perturbed by the system itself. Other equivalent dataset options either lack a proper ground truth or ground truth is assumed to be the output of most optimal algorithm the authors decide. In that sense, the most sensible dataset of choice is left as the one chosen in this thesis work. There are some sequences suitable for object recognition, camera calibration, dynamic object tracking and so forth, however, the main focus is the SLAM studies and the records are mainly used thusly. A recent study which performs the experiments on this Kinect dataset is [27]. The authors evaluate the state of the art SLAM techniques on the publicly available records belonging to the aforementioned dataset. The evaluated SLAM methods are based on the OpenCV [99] implementations [100] of the feature extraction and matching techniques SIFTGPU [101], SURF [102] and ORB [43]. Other than the mentioned work, although being published quite recently in 2012, the dataset is cited by 130 studies [103] some of which are on SLAM [104] [105] [106], RGB-D mapping [107], localization [108], tracking [109] [110], ICP [111] and scene reconstruction [112].

From many Kinect data sequences, two records were chosen based on the surface types observed in the scenes. One of the records mainly contains planar or nearly planar object surfaces which provides a challenge between planar and surface curvature methods in the comfort zone of planar feature based method. The result of the execution on this sequence will be informant on how the surface curvature method performs on planar surfaces which are one of the less complex quadratic surfaces. In this scenario, planar and surface curvature features are expected to have comparable results because although quadratic surface representation is able to describe more types of surfaces with less error values, the absence of such complex surfaces in the sequence should prevent such difference. Conversely, the other sequence contains curved surfaces which are expected to differentiate the complex surface description power of the proposed system from the planar feature based system. The point feature based methods, namely SURF features and corner features, are not considered in the discussion of surface types as they work with a large number of landmarks and there will be points of interest for them in almost any scene. The selected dataset sequences are listed in Table 4 with their basic properties.

In summary, the first dataset in Table 4 is chosen for observing the performances of the SLAM techniques with mostly planar and near-planar surfaces. Also, this sequence is recorded on an actual mobile robot, which provides more direct analysis considering the real mobile application of SLAM; however, as the ground surface is mostly flat, no significant change in some dimensions is observable. The second dataset is chosen in order to analyze the performances of the benchmarking SLAM methods on planar and curved surfaces combined. Additionally, as the sequence is recorded with a hand-held sensor, it is possible to obtain results with significant variation in all six dimensions. Thus, through the experimentations, it is expected to observe the SLAM performance in different environments with various objects surface types and with different motion characteristics such as a mobile robot platform motion or the hand-held motion.

**Table 4: Properties of the Experimenting Dataset Sequences**

| Sequences | Duration (s) | Length (m) | Avg. Trans. Velocity (m/s) | Avg. Rot. Velocity (deg/s) |
|---|---|---|---|---|
| **freiburg2_pioneer_360** | 72.75 | 16.118 | 0.225 | 12.053 |
| **freiburg2_dishes** | 100.55 | 15.009 | 0.151 | 9.666 |

Figure 27 shows one scene from the second benchmarking Kinect sensor record. The curved surfaces in the scene make it challenging for the planar surface feature based SLAM to properly represent the object surfaces. Figure 20 is from the first record in which the foreground objects have planar or nearly planar regions that are mostly suitable for plane fitting.

Also, the first recorded data is taken from the measurements of Kinect sensor mounted on a moving Pioneer robot. As the ground of the environment is flat, there is not much variation in elevation, roll and pitch directions. In this case, the results do not give much information about the error performance on these axes. The second record, however, is taken from hand-held Kinect movements, which not only provides variation on all axes, but also features more abrupt moves as compared to the robot movement. Hence, the second Kinect data is a more challenging benchmark in the sense that the sensor moves freely in the air.

**Figure 27: Kinect Data record with curved surfaces in the scenes [27]**

## 7.2 Benchmark Methods

The benchmarking methods are chosen from state-of-the-art techniques. The implementations for these techniques and the main FastSLAM algorithm are mainly based on the work in the literature [94] [93]. Thus, those methods are presented only briefly.

### 7.2.1 Planar Feature Based SLAM

The SLAM technique used as a benchmark is based on planar segments with normal vectors. In this technique, the planar surfaces are detected by making use of a modified version of RANSAC in which a seed planar region is selected and a large amount of points are included in the plane fitting. For the point clusters that satisfy plane constraints are then described with its central location, area and normal vector to the plane surface. Thus, the landmarks are represented as planar regions with their location, area and normal vector. The feature vector contains 3 elements for central location, 3 elements for normal vector representation and 1 element for the area information as given in (7.1).

$$X = \begin{bmatrix} x_c \\ y_c \\ z_c \\ n_x \\ n_y \\ n_z \\ a \end{bmatrix} \tag{7.1}$$

The initial clustering of the scene is the same as the proposed SLAM technique. Thus, the same object candidates arrive at feature extraction parts of the algorithms. The planar feature based system tries to its best to represent those surfaces as planes.

### 7.2.2 SURF Feature Based SLAM

SURF features provide point features at sharp regions on the object surfaces. There is not much need for an initial clustering; however, such a clustering will save computational power by avoiding less stable background measurements which are out of Kinect sensor comfort zone. Thus, the object clustering is performed as in other benchmarking techniques. The features are represented by their location, scale and Laplacian as given in (7.2).

$$X = \begin{bmatrix} x_c \\ y_c \\ z_c \\ s \\ L \end{bmatrix} \tag{7.2}$$

SURF is a Hessian based feature detector like SIFT. The scale entry in the feature vector provides the scale invariance of the features and the Laplacian parameter provides robustness as it represents the trace of the Hessian matrix. For more detail on the SURF feature extraction method, Appendix A could be visited.

### 7.2.3 Corner Feature Based SLAM

Corner features are seen as the features with only point location data. As the benchmark method, Shi & Tomasi's minimum eigenvalue method is preferred. This algorithm is integrated in the standard library of MATLAB. In this method, basically, the corners are detected by analyzing the Taylor-approximated auto-correlation of the image and its shifted version which yields the image gradient. Then the minimum eigenvalue points of the gradient matrix which corresponds to the fastest change. The feature vector resulting from the corner detection is then simply the location of the detected corner as given in (7.3).

$$X = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{7.3}$$

The fundamentals of Shi & Tomasi corner feature extraction method is given here only briefly, Appendix B provides further information.

## 7.3 Performance Analysis

The performance of the proposed system as compared to the state of the art benchmark methods is measured via the experimentation on real Kinect data simulations. For the sake of diversity in the experimentations, two Kinect records that are different in terms of surface types and motion characteristics. Mainly, the first sequence is recorded on a Pioneer platform and the environment contains mostly planar surfaces and the second sequence is recorded from a hand-held Kinect sensor and the environment contains both planar and curved surfaces.

### 7.3.1. Execution with Kinect Data from Moving Robot

In this part, the plane feature based SLAM, SURF feature based SLAM and corner feature based SLAM are compared against the proposed method in terms of estimation errors. The implementations of the benchmark algorithms are based on [93] and [94].

Figure 28 and Figure 29 depict the total translational and rotational error results, respectively for the FastSLAM algorithm running with four different feature matching techniques. These two graphs reflect the overall performance of the benchmark methods. In terms of error performance, it is observed that SURF feature method yields the best results. Surface curvature feature and corner feature methods follow with close results; however, surface curvature feature method error results are slightly better. Planar feature method stands at the last spot; nevertheless the results are still comparable to the other three methods. These scores are quite conformant with the expectations. SURF reaches the best error performance by using more complex and quantitatively more features at the expense of a higher computational complexity. Corner feature matching is less complex as compared to SURF and this is visible in the inferior results in spite of the similarly high amount of features used. Two compact surface feature methods, namely plane and surface curvature methods are observed to catch up with other two point feature methods. In fact, surface curvature method is seen to yield better results than not only plane feature method but also the corner feature method. It is parallel to the expectation that the quadratic surface curvature representation would have more expressive power than the planar features. It should be noted that planar feature and surface curvature feature methods are higher level compact representations using very few number of features, however, these methods, especially surface curvature feature method, are able to yield comparable results to the more complex point feature methods.

The loop closing behaviors of algorithms differ noticeably. Corner and plane methods seem to have smaller loop closures whereas SURF and surface curvature methods seem to have dramatic loop closure error reductions. However, the loop closing occurs at similar time instances (around the step 90) which correspond to the moment when features are re-encountered after a period. For the other SLAM phase behaviors, it is firstly noticed that there is no "*no feature observation*" as the dataset sequence starts with object surfaces within the field of view. In the "*features observed, before loop closure*" phase, for all of the four methods, there is an increasing trend in the error values, where local differences are observed due to the structural differences of the method approaches.

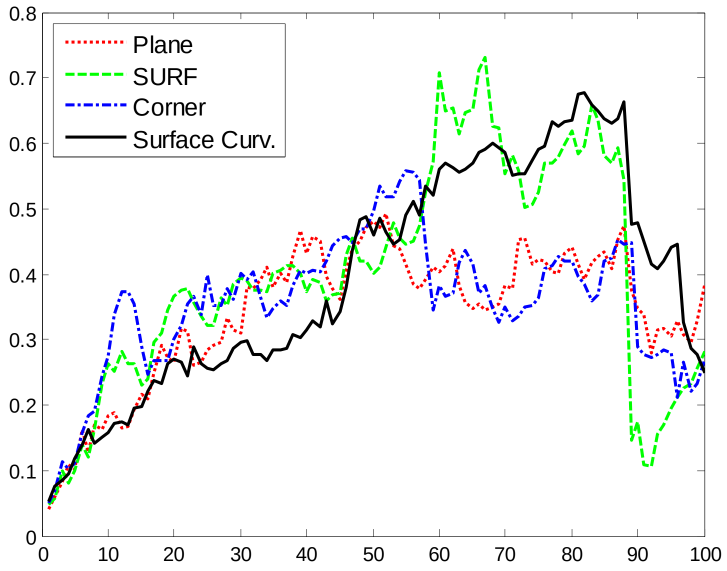


**Figure 28: Total Translational Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

If the translational error results in x, y and z dimensions are considered separately as given in Figure 30, Figure 31 and Figure 32, it is noted that the best performances differ between the three dimensions. SURF, corner and surface curvature feature methods yielded best results for the errors in x, y and z dimensions, respectively. However, the overall trend is consistent with the total translational error comparison.

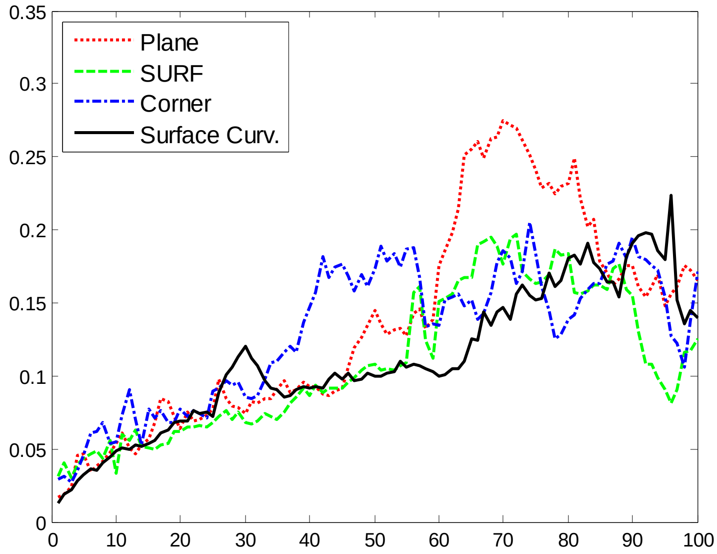


**Figure 29: Total Rotational Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

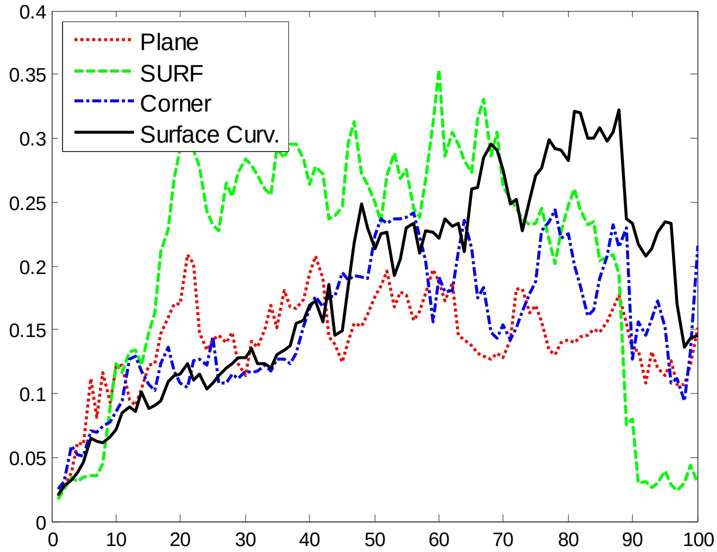


**Figure 30: X-axis Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

The errors in the rotational axes display different characteristics as compared to the translational axes errors. The errors in yaw axis are naturally higher as the robot moves on a flat surface where roll and pitch values do not vary significantly. Thus the most informant results are for the yaw axes which indicate that the SURF feature method performs the best, followed by surface curvature feature method. Plane and corner feature methods have similar results and follow the other two methods.

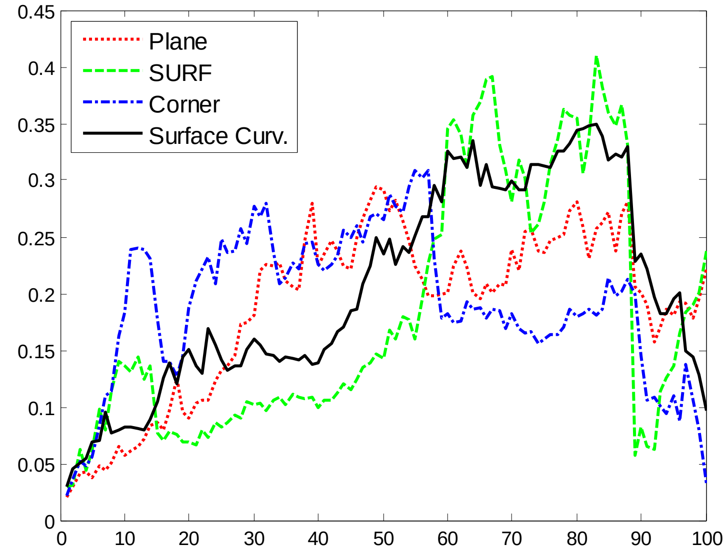


**Figure 31: Y-axis Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

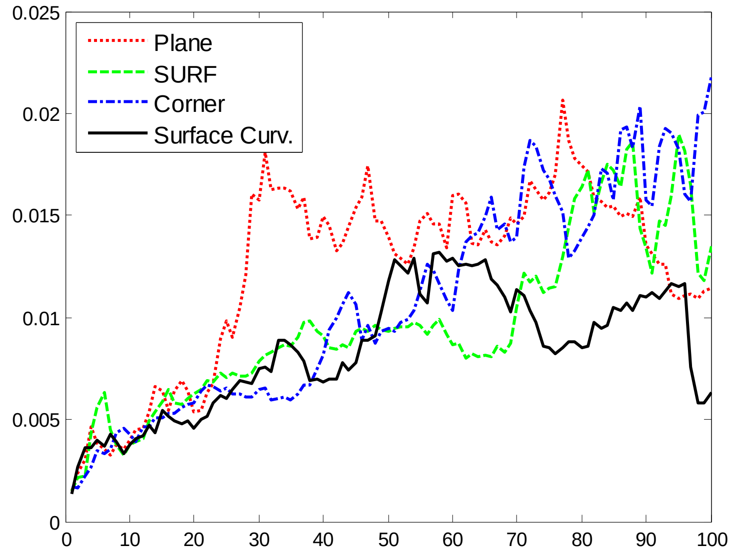


**Figure 32: Z-axis Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

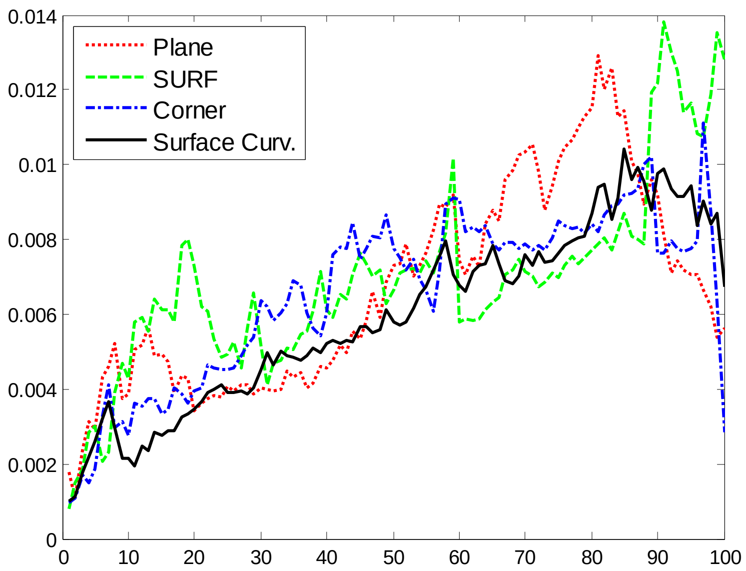


**Figure 33: Roll-axis Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

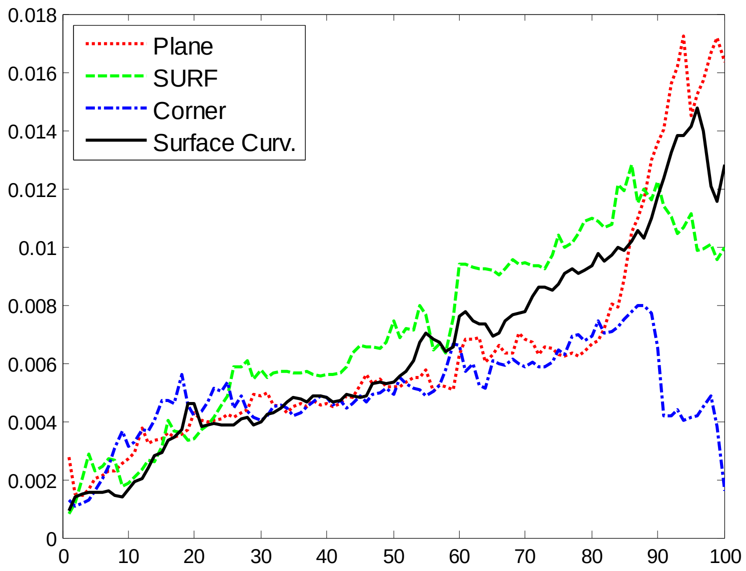


**Figure 34: Pitch-axis Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

The error performances in rotational axes, namely roll, pitch and yaw, are given separately in Figure 33, Figure 34 and Figure 35, respectively.

Number of features is almost like the trade-off parameter between performance and computational cost. However, still the compactness advantage lies beside the plane and surface curvature methods. The computational cost comparison is implicitly involved in the landmark number discussion.

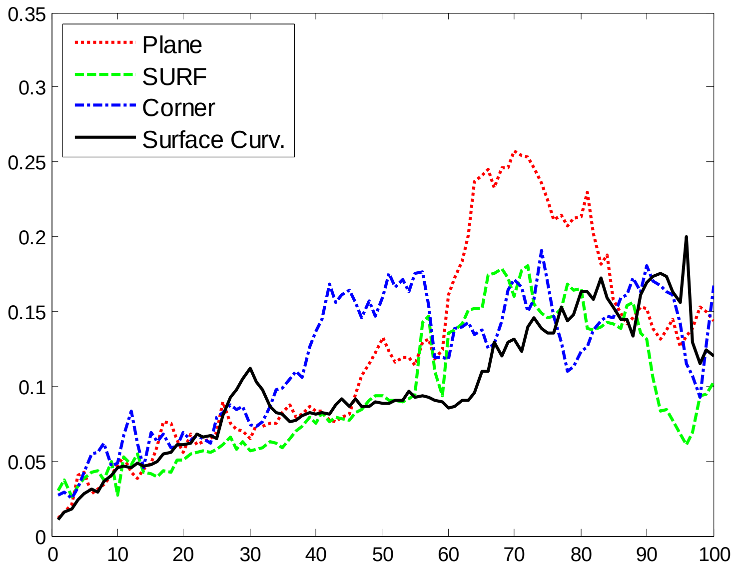


**Figure 35: Yaw-axis Error Performance Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

In the final estimation of path and landmark locations, the results can be said to be as expected. The most computationally complex and thus, the most robust SURF technique seem to have made nearly perfect data associations whereas other methods had a small number of data association faults. The path estimation seems equivalent in all of the methods that is partly because of the success of the measurement update cycles and partly due to a fairly simpler path followed. The final path and landmark location estimation results are given in Figure 36.

Table 5 shows the execution times of the four algorithms. In accordance to the previous discussion, the performance costs computational power. The performance of the proposed method is satisfying in the sense that although the results with SIFT features are better, a close performance is achieved by our compact surface curvature features at a lower computational cost.

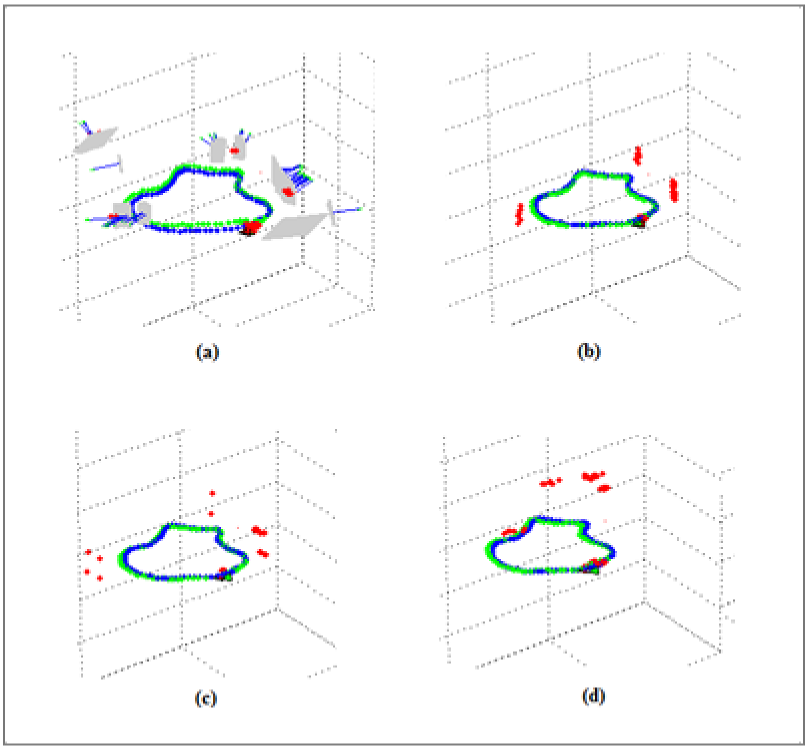


**Figure 36: Final Path and Landmark Locations Comparisons for (a) Plane features, (b) SURF features, (c) Corner features, (d) Surface Curvature Features (proposed)**

**Table 5: Execution Times (Based on the conditions given in the section *7.1 Benchmark Environment*)**

| | Plane Features | SURF Features | Corner Features | Surface Curvature Features |
|---|---|---|---|---|
| **Time (sec)** | 322.6 | 1266.0 | 365.2 | 352.4 |

### 7.3.2. Execution with Kinect Data from Hand-held Movement

As the object surfaces and scenes are more complex and there are more foreground objects in this record, the execution times are higher as compared to the first record. The point features are excessively affected from the complexity of the scenes in terms of execution times. This makes the use of point features less feasible. Also, the performance of the point features does not vary much with the scene complexity or the movement characteristics of the sensors. However, although not feasible, for the sake of a complete observation, the point feature methods, namely SURF and corner features, are included in the experimentation.

Figure 37 shows an example of an analytical surface fitted to a surface patch from the Kinect data. The first figure shows the scatter of points that emerge from the initial clustering of the scene. Those points actually belong to one of the bowls seen in Figure 27. The second figure shows the surface fitted to represent that data It could be noted that the fitted surface was able to capture the original and proves a fine representation for a *bowl*. Planar features either could not fit a plane to this data which is the optimistic scenario, or they could fit a plane which would be almost random for this data. In our benchmarking algorithm, fortunately, such a surface is rejected as a plane and not introduced as an additional source of error into the SLAM algorithm.

Figure 38 and Figure 39 display the total translational and rotational error performance results of the four methods. The overall behavior is consistent with the surface representation performance. In this sequence of Kinect data which contains more complex surfaces, the quadratic surface representation yields better fitting results as seen in Figure 37. In general; the error rates are very low due to the path of the Kinect sensor which keeps the dishes on the table within the field of view most of the time. Thus, the error rates stay low as the algorithm is able to observe the most of the features continuously. The results express that SURF and surface curvature features have the better error performance as compared to the plane and corner feature methods. It is important to note that the performance difference between the surface curvature feature and the plane feature methods become more distinct as compared to the first Kinect sequence due to the increased amount of curved surfaces. Another observation to be made is that the increased complexity of the scene caused corner features to be less stable on the grounds that the corner locations begin to overlap with the increase in the feature density.

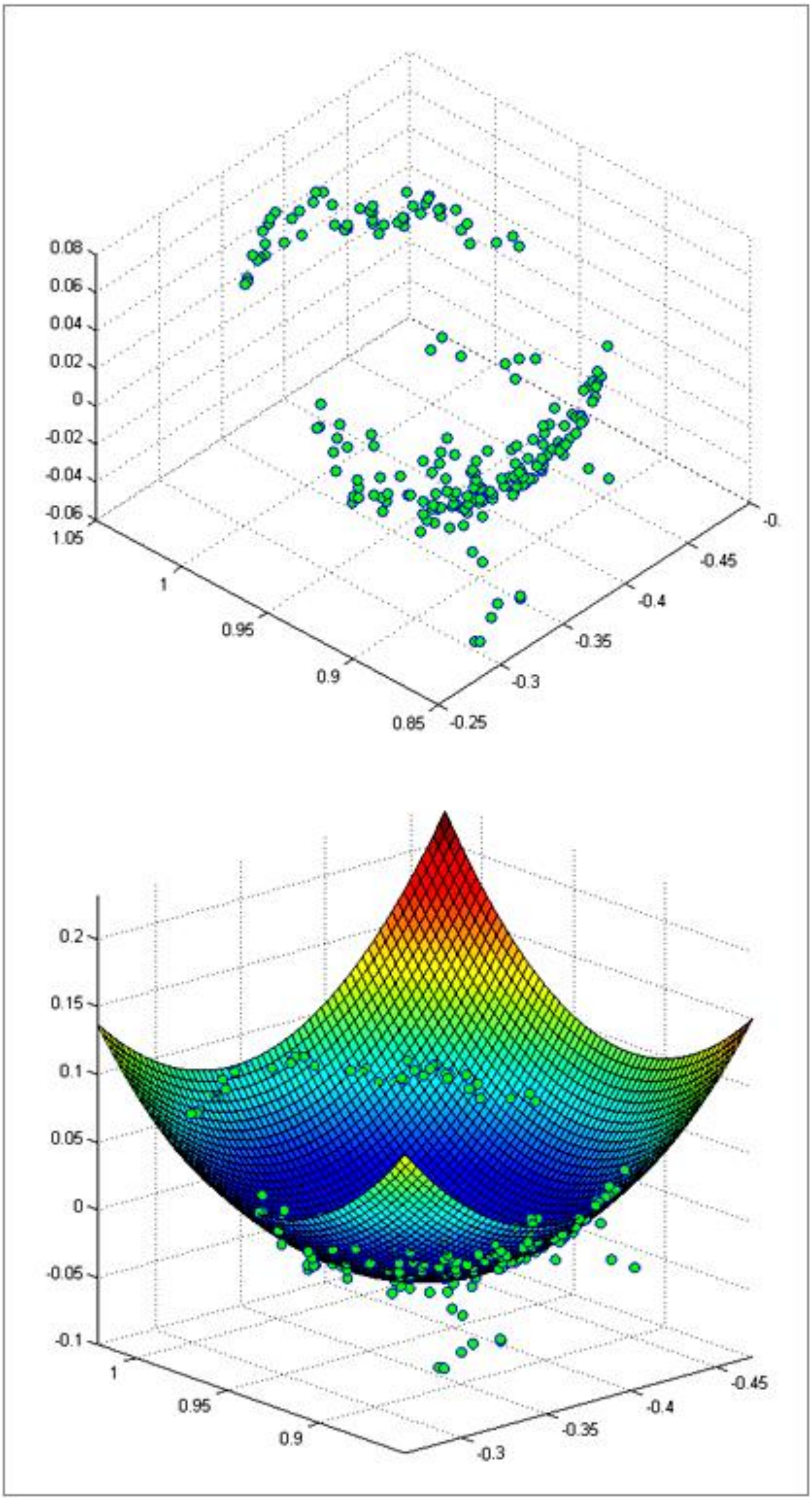


**Figure 37: A Sample Surface Fit from the Second Kinect Record (X, Y and Z-axis values are coordinates in meters)**

The better results in surface representation are then propagated to the overall error results of the SLAM implementation. From Figure 38 and Figure 39, we note that although the error rates are low, there is a significant amount of error performance difference final and average error values of plane feature and surface curvature feature methods. Although this is consistent with expectations, considering the dominance of curved surfaces in the scene. Thus, the expressive power of the quadratic surface representation displays its added value upon the planar surface representation.

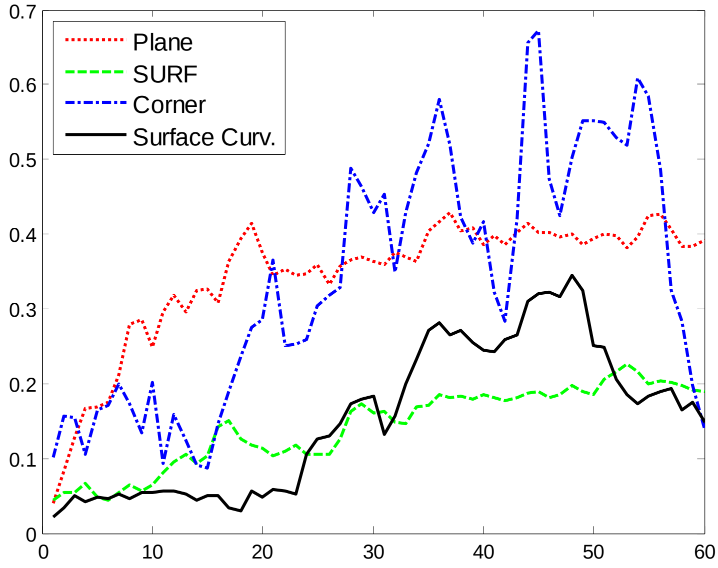


**Figure 38: Total Translational Error Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

When each of the translational and rotational error components are considered separately, the results are as given in Figure 40. Unlike the results in the first Kinect sequence, the error results of each dimension are given together as the values are low and thus it is more useful to observe them together. It is seen that in the most active dimensions, namely x,y and yaw, surface curvature feature method yields the superior performance. This meets the expectation that the impact of the more surface curvature feature utilization would be more distinct as such complex surfaces cannot be easily represented by planar features. The point feature approaches, namely SURF and corner features, are not affected significantly as they deal with low level point features which are almost always utilized to some extent. Nevertheless, corner feature method are observed to have failed to adapt to a dense feature environment as its feature stability is not as powerful as SURF method.

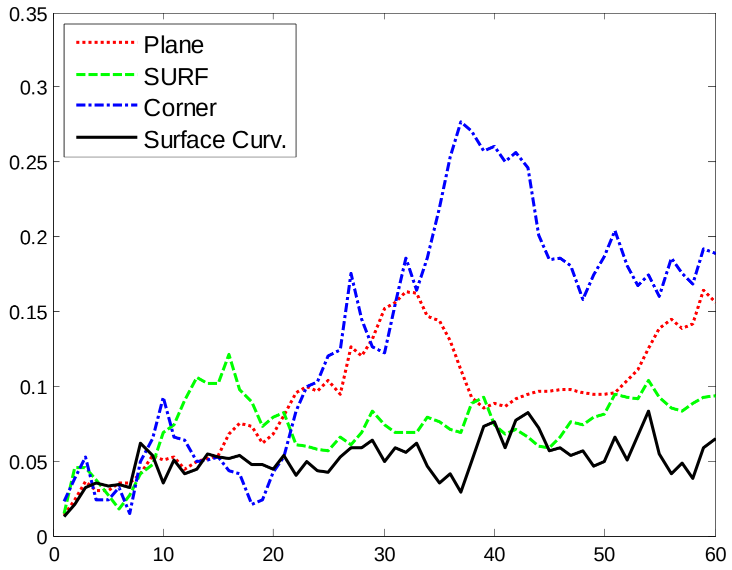


**Figure 39: Total Rotational Error Comparison (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

The computational cost performance is in accordance with the results in the first Kinect record when compared to each other. However, in absolute values, the execution times are significantly longer due to the increase in the amount of data to be processed. In the first record, an important part of the scene images are from background regions whereas in the second record, the scenes are taken in closer range and also the angle of the sensor does not allow much of the background to enter into the field of view. Thus, the execution times are measured as high as the values given in Table 6. In summary, it could be drawn from the results that with surface curvature features, it is possible to obtain superior error performance at a comparably less computational cost which shows that the aim of using compact surface features has been achieved.

**Table 6: Execution Times (Based on the conditions given in the section *7.1 Benchmark Environment*)**

| | Plane Features | SURF Features | Corner Features | Surface Curvature Features |
|---|---|---|---|---|
| **Time (sec)** | 6322 | 10166 | 6465 | 8482 |

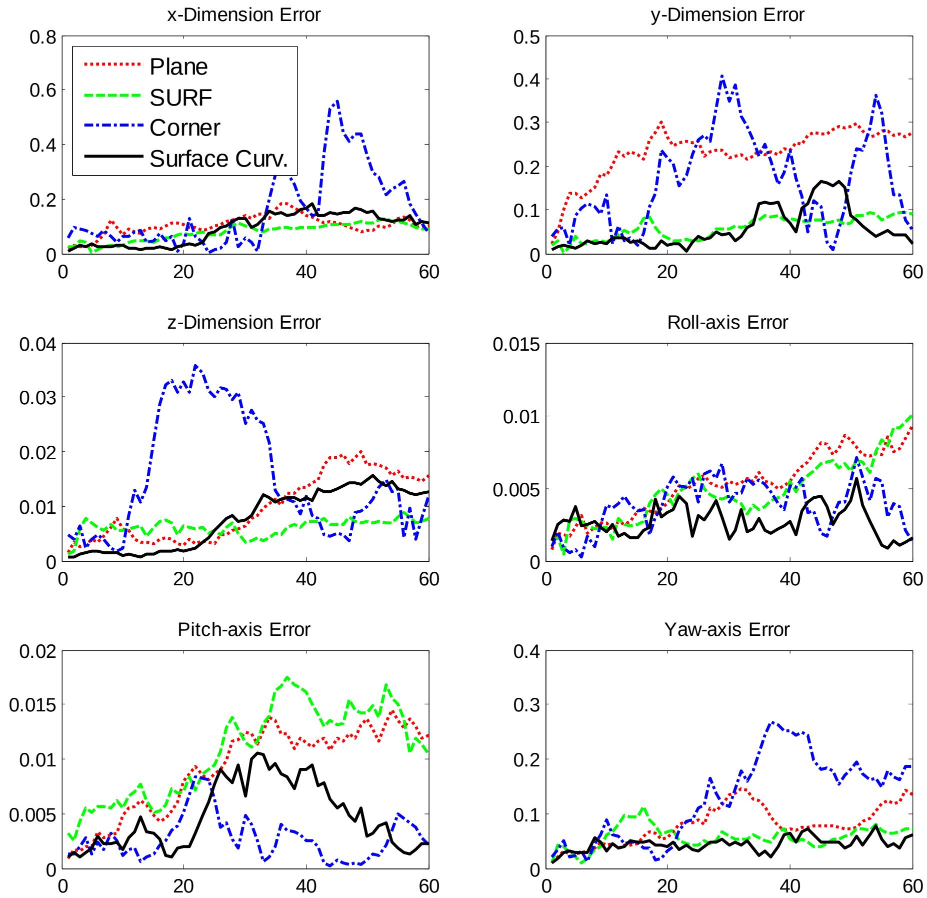


**Figure 40: Error performance in x, y, z, roll, pitch and yaw dimensions. (X-axis: number of depth image frames, Y-axis: per-particle errors in meters)**

### 7.3.3. Analysis of Higher-level Compact Feature Effects in SLAM

The benchmarking feature representations could be categorized into three main groups in terms of the complexity and compactness. Corner features are considered as low-level point features, SURF features stand as higher-level point features as the feature points are chosen based on the local surface geometry. The planar features and surface curvature features, on the other hand, are higher-level compact features that are able to describe the same region of scene data with a 6 or 7 element feature vector instead of possibly hundreds of feature points as in the case of corner features and especially SURF features which extracts a substantial amount of feature points from a given scene section. Figure 28 and Figure 29 show the results of the four feature types used in the data association of the same SLAM algorithm running on the same Kinect sensor dataset. It is observed that SURF features result in the best error performance at the expense of more than 50 feature points to be tracked. Although the corner features yielded about 20 features, the error performance was significantly worse than that of the SURF features. The use of plane and surface curvature features, however, result in a comparably close performance to that of SURF features at the cost of a dramatically lower number of features which is in the range of 5 to 10. The execution times given in Table 5 are an immediate indication of what this compactness could grant for SLAM applications. In a broader sense, the ability to represent the same set of scene data more compactly with fewer features provides savings in several stages of the SLAM algorithm which yields a cumulative benefit. The feature extraction stage might be still complex however, right after the compact surface representation is obtained; the remaining path is a peaceful downhill. For the FastSLAM method as in this thesis work, the number of features lead to a logarithmic complexity whereas in the case of EKF-SLAM the complexity grows quadratically with the number of features as the cross correlations between the landmarks are kept throughout the operation. This reflects the global matching side of the SLAM problem which is visible in the growth of the SLAM state variables. The data association phase composed of consecutive pair-wise comparison within the monotonically growing database of feature landmarks kept inside the SLAM state. In addition to that, the local scan matching process is also affected by the compactness of the features for the complexity considerations. For the real time applicability of

SLAM algorithms, the use of compact features for data association is quite promising. Another practicality of the compact representations emerges during the visual mapping processes. If the SLAM environment is to be reconstructed, then the higher level surface representations such as planar or quadratic surfaces would prove to be a better input and requires less further processing than point feature representations.

# CHAPTER 8

# CONCLUSIONS AND FUTURE WORK

The implementation of surface curvature features inside SLAM is introduced. This is a novel approach in the sense that although the surface curvature features have been used in surface matching methods, there has been no attempt to use that technique within SLAM domain. This makes sense due to the fact that in especially indoors robotic environments, such curved surfaces are encountered. Instead of reducing them down to planar level or even completely ignoring, this method tries to utilize these surfaces as a means of features for landmark representations. Another advantage is that the method implicitly contains planar features when the fitted surface is a planar patch. Thus, without loss of generality, the planar feature based SLAM could be incrementally improved.

The experimental results support the theoretical expectations. The following results are drawn from this thesis work:

- A new approach is introduced for feature extraction in SLAM domain which makes use of surface curvature features extracted locally from clustered range data. It is shown that quadratic surfaces can be utilized for SLAM algorithms.
- The repeatability and distinctiveness analysis of the proposed surface curvature feature extraction method is verified with the experiment conducted on selected object surfaces that are observed under various view point changes. It is shown that the proposed surface curvature feature extraction method produces successful feature representations for the data association problem in the SLAM application.
- The performance of the proposed method yielded satisfactory results in the sense that it is able to utilize curved surfaces and provides improvement as compared to the sole use of planar surfaces.

- After planar approaches, quadratic surface approach verified that compact representations are useful for SLAM applications as they provide more robust representations and fewer amounts of features which, in turn, lead to a significant computational cost reduction.
- The real time concept is discussed and an existing application of Mean and Gaussian curvature estimation is mentioned which is able to execute in microsecond levels. This loosely proves the real time applicability of the proposed feature extraction method.

The results indicate that selected surface curvature features are promising in the data association problems in mobile robotics applications such as SLAM. The surface fitting approach was quite successful, however, it can be said to be almost ideal. Some higher level features could be driven from the algebraic parameters of the fitted surface patches.

Another extension would naturally be the real time application. The existence of some methods encourages such a long term project which involves hardware programming.

The use of surface curvatures may lead the way towards a better terrain representations and better outdoors SLAM. The feasibility of this approach on terrain surfaces will be investigated.

Maybe, the most practical extension of this SLAM work is to extend the operation to be applicable in dynamic environments instead of static. In real life, it is rather hard to safely assume that everything in the environment except the mobile robot itself is static. Thus, if the SLAM is ever going to be blend in daily life activities in the full and literal sense; SLAM with Moving Object Tracking (SLAMMOT) ability must be at hand. Therefore the first priority for further research on SLAM is aimed towards the SLAM in dynamic environments which have considerable common aspects with this thesis work such as compact object representation, no *a priori* assumption for object surfaces.

# REFERENCES

[1] S.Thrun, Google X Driverless Car. Wikipedia. [Online] http://en.wikipedia.org/wiki/Google_driverless_car, (Accessed: 2014, Dec.)

[2] Stanley: The Robot that Won the Darpa Grand Challenge. S. Thrun, s.l. : Wiley Interscience, 2006, Journal of Field Robotics.

[3] J. Haugeland, Artificial Intelligence: The Very Idea. London : MIT Press, 1985.

[4] Vision-Based SLAM: Stereo and Monocular Approaches. T. Lemaire, 3, Toulouse : Springer Science, 2007, International Journal of Computer Vision, Vol. 74.

[5] Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. H. Durrant-Whyte.

[6] A Benchmark for the Evaluation of RGB-D SLAM Systems. Sturm, J. s.l. : Proc. of the International Conference on Intelligent Robot Systems (IROS), 2012.

[7] Simultaneous Localization and Mapping (SLAM): PartII. H. Durrant-Whyte, s.l. : IEEE Robotics & Automation Magazine, 2006.

[8] Estimating uncertain spatial relationships in robotics. R. Smith, P. Cheeseman. s.l. : Autonomous Robot Vehicles, 1990.

[9] Mobile robot localization by tracking geometric beacons. H. F. Durrant-Whyte, J. J. Leonard. s.l. : IEEE Transactions on Robotcs and Automation, 1991.

[10] The SPMap: A probabilistic framework for simultaneous localization and map building. J. A. Castellanos, M. M. Montiel, J. Neira, and J. D. Tardós. s.l. : IEEE Transactions on, 1999.

[11] Feature-based multi-hypothesis localization and tracking using geometric constraints. K. O. Arras, J. Castellanos, M. Schilt, and R. Siegwart. s.l. : Robotics and Autonomous Systems, 2003.

[12] Simultaneous localization and map building using natural features and absolute information. J. Guivant, F. Masson, and E. Nebot. s.l. : Robotics and Autonomous Systems, 2002.

[13] Active global localization for a mobile robot using multiple hypothesis tracking. Kristensen, P. Jensfelt and S. s.l. : IEEE Transactions on Robotics and Automation, 2001.

[14] A new extension of the kalman filter to nonlinear systems. Uhlmann, S. J. Julier and J. K. s.l. : AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation, 1997.

[15] Sensor fusion in certainty grids for mobile robots. H. Moravec, 1988.

[16] A probabilistic framework for robot perception and navigation. Elfes, A. s.l. : Carnegie Mellon University, PhD thesis, 1989.

[17] Active markov localization for mobile robots. D. Fox, W. Burgard, and S. Thrun. s.l. : Journal of Artificial Intelligence Research, 1999.

[18] Experiences with an interactive museum tour-guide robot. W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner,S. Thrun. s.l. : Artificial Intelligence, 1999.

[19] Novel approach to nonlinear/non-gaussian bayesian state estimation. N.J. Gordon, D.J. Salmond, and A.F.M. Smith. s.l. : IEEE Proceedings of Radar and Signal Processing, 1993, Vol. 140.

[20] Particle filters for mobile robot localization. D. Fox, S. Thrun, W. Burgard, and F. Dellaert. 2001.

[21] Fastslam: a factored solution to the simultaneous localization and mapping problem. M. Montemerlo, S. Thrun, D. Koller, Wegbreit B. s.l. Proceedings of the AAAI National Conference on Artificial Intelligence, 2002.

[22] Robot pose estimation in unknown environments by matching 2d range scans. Milios, F. Lu and E. s.l. : Computer Society Conference on Computer Vision and Pattern Recognition, 1994.

[23] A method for registration of 3-d shapes. McKay, Paul J. Besl and Neil D. s.l. : IEEE Transactions, 1992.

[24] A Robust RGBD SLAM System for 3D Environment with Planar Surfaces. Su, Po-Chang. s.l. : Univercity of Kentucky, 2013.

[25] Feature-based 3D SLAM. J. Weingarten, 2006.

[26] RGB-D Mapping: Using depth cameras for dense 3D modeling of indoor environments. P. Henry, M. Krainin, E. Herbst, X. Ren , D. Fox. 2010.

[27] An Evaluation of the RGB-D SLAM System. F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard. s.l. : IEEE Int. Conf. on Robotics and Automation (ICRA), 2012.

[28] 3D Surface Matching from Range Images using Multiscale Local Features. Ho, Huy Tho. s.l. : The University of Adelaide, 2009.

[29] Infrared Range Sensor Array for 3D Sensing in Robotic Applications. Y. Do, J. Kim. s.l. : International Journal of Advanced Robotic Systems, 2013.

[30] 3D Visual SLAM for an Assistive Robot in Indoor Environments Using RGB-D Cameras. L. Meng, C. W. de Silva, J. Zang. s.l. : International Conference on Computer Science & Education, 2014.

[31] Using the Kinect as a Navigation Sensor for Mobile Robotics. A. Oliver, S. Kang. s.l. : Proceedings of the 27th Conference on Image and Vision Computing New Zealand, 2012.

[32] Real-time 3D visual SLAM with a hand-held RGB-D camera. N. Engelhard, F. Endres, J. Hess, J. Sturm, W. Burgard. s.l. : Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, 2012.

[33] GMapping. OpenSLAM. [Online] http://openslam.org/gmapping.html, (Accessed: 2014, Dec.)

[34] 3D Keypoints Detection for Objects Recognition. A. Shaiek, F. Moutarde. s.l. : 16th International Conference on Image Processing, Computer Vision, and Pattern Recognition, 2012.

[35] Scale-space approach for the comparison of HK and SC curvature descriptions as applied to object recognition. E. Akaggunduz, O. Eskizara, I. Ulusoy. s.l. : ICIP, 2009.

[36] Self Localization with Edge Detection in 3D Space. V. Ailani, D. Prakash, K. S. Venkatesh. s.l. : Journal of Image and Graphics, 2013.

[37] Hough Transform and 3D SURF for robust three dimensional classification. Jan K., Mukta P., Geert W., Radu T., Luc van G. s.l. : Proceedings of the European Conference on Computer Vision, 2010.

[38] Incremental Registration of RGB-D Images. I. Dryanovski, C. Jaramillo, J. Xiao. s.l. : ICRA, 2012.

[39] Building a 3D Map from RGB-D Sensors. V. Högman, s.l. : KTH, School of Computer Science and Communication, 2012.

[40] Accurate 3d Tracking Using Visual And Depth Data. O. S. Gedik, s.l. : Ph.D. Thesis, Middle East Technical University, 2014.

[41] Robust Place Recognition for 3D Range Data based on Point Features. B. Steder, G. Grisetti, W. Burgard. 2010.

[42] Non-linear constraint network optimization for efficient map learning. G. Grisetti, C. Stachniss, and W. Burgard. 2009.

[43] ORB: An efficient alternative to SIFT and SURF. E. Rublee, V. Rabaud, K. Konolige, G. Bradski. s.l. : IEEE International Conference on Compuper Vision, 2011.

[44] SLAM with Object Discovery, Modeling and Mapping. S. Choudhary, A. J. B. Trevor, H. I. Christensen, F. Dellaert. s.l. : IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014.

[45] Discovering Higher Level Structure in Visual SLAM. A. P. Gee, D. Chekhlov, A. Calway, W. Mayol-Cuevas. s.l. : IEEE Transactions On Robotics, 2008.

[46] A combined texture-shape descriptor for enhanced 3D feature matching. F. Tombari, S. Salti, L. Di Stefano. s.l. : IEEE International Conference on Image Processing (ICIP), 2011.

[47] Tables, Counters, and Shelves: Semantic Mapping of Surfaces in 3D. A. J. B. Trevor, J. G. Rogers III, C. Nieto-Granda, H. I. Christensen. s.l. : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010.

[48] Planar Surface SLAM with 3D and 2D Sensors. A. J. B. Trevor, J. G. Rogers, H. I. Christensen. s.l. : IEEE International Conference on Robotics and Automation (ICRA), 2012.

[49] Simultaneous Localization, Mapping and Moving Object Tracking. C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, H. Durrant-Whyte. 9, s.l. : The International Journal of Robotics Research, 2007, Vol. 26.

[50] Visual SLAM and Moving-object Detection for a Small-size Humanoid Robot. Y. T. Wang, M. C. Lin, R. C. Ju. 2, s.l. : International Journal of Advanced Robotic Systems, 2010, Vol. 7.

[51] Cognitive maps for mobile robots—an object based approach. S. Vasudevan, S. Gachter, V. Nguyen, R. Siegwart. Zurich : Robotics and Autonomous Systems, 2007, Vol. 55.

[52] 3D Object Mapping by Integrating Stereo SLAM and Object Segmentation Using Edge Points. Tomono, M. s.l. : Springer, 2009.

[53] Using 3D Laser Range Data for SLAM in Outdoor Environments. C. Brenneke, O. Wulf, B. Wagner. s.l. : Intelligent Robots and Systems, 2003.

[54] Combining monoSLAM with Object Recognition for Scene Augmentation using a Wearable Camera. Castle, R. O. Oxford : Image and Vision Computing, 2010.

[55] Vision-Inertial SLAM using Natural Features in Outdoor Environments. Asmar, D. C. s.l. : University of Waterloo, 2006.

[56] Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach. Sola, J. s.l. : Ph.D. dissertation, Institut National Politechnique de Toulouse, 2007.

[57] Use a Single Camera for Simultaneous Localization And Mapping with Mobile Object Tracking in dynamic environments. D. Migliore, R. Rigamonti, D. Marzorati, M. Matteucci, D. G. Sorrenti. s.l. : IEEE Inernational Conference on Robotics and Automation (ICRA), 2009.

[58] Data Association Using Visual Object Recognition for EKF-SLAM in Home Environment. S. Ahn, M. Choi, J. Choi, W. K. Chung. s.l. : International Conference on Intelligent Robots and Systems, 2006.

[59] Integrating SLAM and Object Detection for Service Robot Tasks. P. Jensfelt, S. Ekvall, D. Kragic, D. Aarno. s.l. : IEEE International Conference on Intelligent Robots and Systems, 2007.

[60] Realtime Motion Segmentation based Multibody Visual SLAM. A. Kundu, K. M. Krishna, C. V. Jawahar. s.l. : ICVGIP, 2010.

[61] SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, A. J. Davison.

s.l. : IEEE International Conference on Computer Vision and Pattern Recognition, 2013.

[62] Model globally, match locally: Efficient and robust 3D object recognition. B. Drost, M. Ulrich, N. Navab, S. Ilic. s.l. : Proceedings of the IEEE Conference on Computer Vision, 2010.

[63] A Comparative Evaluation of 3D Keypoint Detectors in a RGB-D Object Dataset. S. Filipe, L. A. Alexandre. s.l. : International Conference on Computer Vision Theory and Applications, 2014.

[64] A large-scale hierarchical multi-view RGB-D object dataset. K. Lai, L. Bo, X. Ren, D. Fox. s.l. : International Conference on Robotics and Automation, 2011.

[65] A Performance Evaluation of 3D Keypoint Detectors. S. Salti, F. Tombari, L. D. Stefano. s.l. : International Conference on 3D Imaging, Modeling, Processing and Transmission, 2011.

[66] Local shape descriptors, a survey and evaluation. P. Heider, A. Pierre-Pierre, R. Li, C. Grimm. s.l. : Eurographics Workshop on 3D Object Retrieval, 2011.

[67] Multi-scale Feature Extraction on Point-Sampled Surfaces. M. Pauly, R. Keiser, M. Gross. s.l. : Eurographics, 2003.

[68] MonoSLAM: Real-Time Single Camera SLAM. A.J. Davison, I. Reid, N. Molton, O. Stasse. s.l. : IEEE Transactions on Pattern Analysis and Machine, 2007.

[69] Impact of landmark parametrization on monocular EKF-SLAM with points and lines. J. Sola, 2011.

[70] D. Koller, N. Friedman. Probabilistic Graphical Models Principles and Techniques. 2009.

[71] K. Murphy, S. Russell. Rao-blackwellized particle filtering for dynamic bayeisan networks. s.l. : Sequential Monte Carlo Methods in Practice, 2001.

[72] Eigenshapes for 3D Object Recognition in Range Data. R. Campbell, P. Flynn. s.l. : IEEE Int'l Conf. Computer Vision and Pattern Recognition, 1999, Vol. 2.

[73] Hebert, A. Johnson and M. s.l. : EEE Trans. Pattern Analysis and Machine, 1999.

[74] Visual Learning and Recognition of 3-D Objects from Appearance. H. Nayar, S. K. Murase, s.l. : International Journal of Computer Vision, 1995.

[75] Face Recognition using Eigenfaces. M. Pentland, A. Turk, s.l. : Proc. IEEE, 1991.

[76] Silhouette-Based Isolated Object Recognition through Curvature Scale Space. F. Mokhtarian, s.l. : IEEE Trans. Pattern Analysis and Machine Intelligence, 1995.

[77] On Recognizing and Positioning Curved 3D Objects from Image Contours. Kriegman, J. Ponce and D. s.l. : Proc. IEEE Workshop Interpretation 3D Scenes, 1989.

[78] Object Recognition using Three-dimensional Correlation of Range Images. J. Guerrero-Bermdez, J. Meneses, and O. Gauldrn. s.l. : Optical Engineering, 2000.

[79] Three-Dimensional Joint Transform Correlator. J. Rosen, s.l. : Applied Optics, 1998.

[80] RANSAC-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. C. Chen, Y. Hung, and J. Cheng. s.l. : IEEE Trans. Pattern Analysis and Machine Intelligence, 1991.

[81] A Graph Matching Approach to 3-D Point Correspondences. Don, J. Cheng and H. s.l. : Int'l J. Pattern Recognition and Artificial Intelligence, 1991.

[82] Point Signatures: A New Representation for 3D Object Recognition. Jarvis, C. Chua and R. s.l. : International Journal of Computer Vision, 1996.

[83] COSMOS: A Representation Scheme for 3D Free-Form Objects. Jain, C. Dorai and A. s.l. : IEEE Trans. Pattern Analysis and Machine Intelligence, 1997.

[84] Recognizing Objects in Range Data using Regional POint Descriptors. A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. s.l. : Proc. European Conf, 2004.

[85] 3D Object Recognition from Range Images Using Local Feature Histograms. G. Hetzel, B. Leibe, P. Levi, and B. Schiele. s.l. : IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2001, Vol. 2.

[86] Hebert, A. Johnson and M. s.l. : IEEE Trans. Pattern Analysis and Machine Intelligence, 1999.

[87] 3D Object Recognition from Range Images using Pyramid Matching. Guskov, X. Li and I. s.l. : Proc. IEEE Int'l Conf. Computer Vision, 2007.

[88] A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images. A. Mian, M. Bennamoun, and R. Owens. s.l. : International Journal of Computer Vision, 2006.

[89] On Recognizing and Positioning Curved 3D Objects from Image Contours. Kriegman, J. Ponce and D. s.l. : Proc. IEEE Workshop Interpretation 3D Scenes, 1989.

[90] A Survey of Methods for Recovering Quadrics in Triangle Meshes. Petitijean, S. s.l. : ACM Computing Surveys, 2001.

[91] Estimating Differential Quantities using Polynomial Fitting of Osculating Jets. Pouget, F. Cazals and M. s.l. : Computer Aided Geometric Design, 2005.

[92] A Comprehensive Introduction to Differential Geometry. Spivak, M. s.l. : Publish or Perish, 1999.

[93] Bailey, Tim. http://www-personal.acfr.usyd.edu.au/tbailey/software/matlab_-utilities.htm, (Accessed: 2014, Dec.)

[94] M. Yörük, Performance Comparison of Point and Plane Features For SLAM. M.Sc. Thesis, METU, 2014

[95] 3D free-form object recognition in range images using local surface patches. H. Chen, B. Bhanu. s.l. : Pattern Recognition Letters 28 , 2007.

[96] Faithful Recovering of Quadric Surfaces from 3D Range Data. Werghi, N.

[97] Surface Parameterization in Volumetric Images for Curvature-Based Feature Classification. Quek, F. K. H. 2003.

[98] Real Time 3D Surface Feature Extraction On FPGA. Tellioğlu, Z. H. Ankara : METU Library, 2010.

[99] OpenCV. http://opencv.org/. [Online], (Accessed: 2014, Dec.)

[100] Learning OpenCV: Computer Vision with the OpenCV Library. G. Bradski, A. Kaehler. s.l. : O'Reilly Media, 2008.

[101] SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). Wu, C. 2007.

[102] Speeded-Up Robust Features (SURF). H. Bay, A. Ess, T. Tuytelaars. s.l. : Computer Vision and Image Understanding, 2008.

[103] Google Scholar. [Online] scholar.google.com. (Accessed: 2014, Dec.)

[104] Deformation-based Loop Closure for Large Scale Dense RGB-D SLAM. T. Whelan, M.Kaess, J. J. Leonard, J. McDonald. s.l. : International Conference on Intelligent Robots and Systems (IROS), 2013.

[105] Dense Visual SLAM for RGB-D Cameras. C. Kerl, J. Sturm,D. Cremers. s.l. : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013.

[106] A Robust Fusion Method For RGB-D SLAM. T. Liu, X. Zhang, Z. Wei, Z. Yuan. s.l. : Chinese Automation Congress (CAC), 2013.

[107] Robust Real-Time Visual Odometry for Dense RGB-D Mapping. T. Whelan, H. Johansson, M.Kaess, J. J. Leonard, J. McDonald. s.l. : IEEE International Conference on Robotics and Automation (ICRA), 2013.

[108] Robust Localization Using RGB-D Images. Y. Oh, S. Oh. s.l. : International Conference on Control, Automation and Systems (ICCAS), 2014.

[109] Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions. E. Bylow, J. Sturm, C. Kerl, F. Kahl, D. Cremers. s.l. : Robotics: Science and Systems (RSS) Conference, 2013.

[110] Tracking an RGB-D Camera Using Points and Planes. E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, T. Garaas. s.l. : IEEE International Conference on Computer Vision Workshops, 2013.

[111] Adaptive Iterative Closest Keypoint. J. Ekekrantz, A. Pronobis, J. Folkesson, P. Jensfelt. s.l. : European Conference on Mobile Robots (ECMR), 2013.

[112] Towards Consistent Reconstructions of Indoor Spaces Based on 6D RGB-D Odometry and KinectFusion. H. Dong, N. Figueroa, A. El Saddik. s.l. : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014.

[113] Good Features to Track. J. Shi, C. Tomasi. s.l. : IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[114] OpenCV. Shi-Tomasi Corner Detector & Good Features to Track. [Online], (Accessed: 2014, Dec.)

# APPENDIX A

# METHOD OF SPEEDED-UP ROBUST FEATURES (SURF)

SURF (Speeded-Up Robust Features) [102] is introduced as a scale- and rotation-invariant feature detection and description technique. The method uses a Hessian matrix-based measure for the detection and a distribution-based approach for the description with the aim of achieving superior repeatability, distinctiveness and robustness.

At the core of the fast computation capability of the method, lies the integral image formation which speeds up the box type convolution filter executions. An entry $I_{\Sigma}(x)$ at the location $x = (x, y)^T$ in the integral image is defined as the sum of all pixel values in the image $I$ within the rectangular region formed by the corners at the origin and the point under consideration, $x$ as given in (A.1).

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \tag{A.1}$$

After the integral image is formed, then the calculations at any region of the image with any size is calculated simply by three addition operations as depicted in Figure 41.

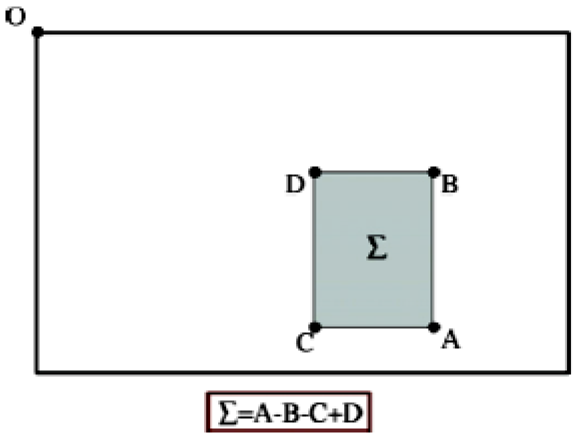


**Figure 41: Only three additions to calculate the sum of intensities inside a rectangular region of any size [102]**

Blob-like structures are detected at the locations where the determinant of the Hessian matrix is maximum. The scale selection is also carried out in the Hessian matrix as given in (A.2) where $x = (x, y)$ is a point in an image I, the Hessian matrix is found for location x and scale σ.

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad \text{(A.2)}$$

$L_{xx}(x,\sigma)$, $L_{xy}(x,\sigma)$ and $L_{yy}(x,\sigma)$ terms stand for the convolution of the Gaussian second order derivative with the image at pointx. Gaussian second order partial derivatives are approximated as given in Figure 42. In this figure, first two images show the images obtained by the calculations of $L_{xx}$, $L_{xy}$ and $L_{yy}$ whereas; last two images are the approximations to them, namely $D_{xx}$, $D_{xy}$ and $D_{yy}$. As visible in Figure 42, the weights are kept simple at $1$, $-1$ and $-2$ instead of more complex values for computational efficiency. After the approximations, the determinant of the Hessian matrix is found as given in (A.3) where w is a constant used to balance the expression for the Hessian determinant and found as given in (A.4) for the scale of 1.2 which is the lowest scale for the detection process.

$$det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{A.3}$$

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912 \ldots \cong 0.9 \tag{A.4}$$

In (A.4), $|.|_F$ is the Frobenius norm. Although this constant is obviously different for each scale, based on the experimental results, there was no significant effect of the change, the value is kept constant. The result of determinant approximation of the Hessian matrix is therefore, yields the blob response in the image at the respective location.

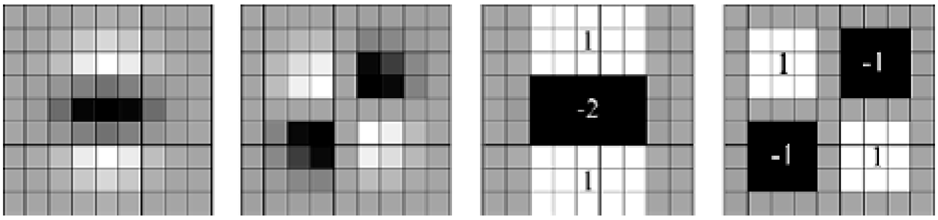


**Figure 42: Left to right; discretised Gaussian second order derivative in y and xy directions and their approximations, respectively [102]**

The authors show that the approximation for the second order Gaussian derivatives result in comparable performance as compared to the original calculations with a repeatability rate ranging from %65 to %95 against the rotation of the images from 0° to 180°.

The scale invariance is maintained by exploiting the integral image once more. Contrary to the other scaling approaches such as reducing the image size, the filter box size is increased which does not increase the computational cost at upper scales thanks to the integral image conversion where the computations at any size are for exactly the same cost. The scale space is divided into octaves which are the series of consecutive scales. The minimum scale corresponds to a 9x9 box filter. Then the next scale filter size is chosen by increasing the size by one third of the initial size at each side. Thus, the second scale is 15x15 by adding $3 = 9(1/3)$ to each lobe of the box. In this manner, the octave has another two up-scaled filter boxes and then a new overlapping octave starts at 15x15 size. This could be seen as a smooth transmission from one octave to the other. With the addition of a new octave, the coverage of the scale increases as seen in Figure 43.

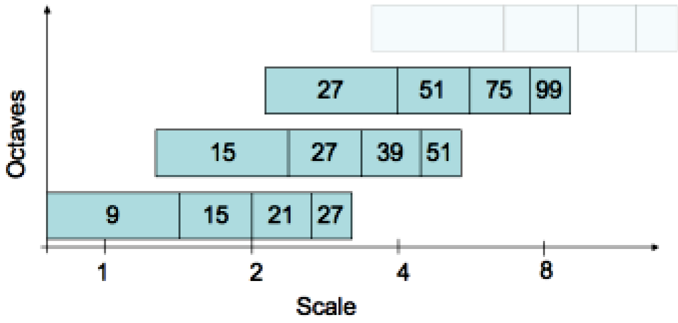


**Figure 43: Three octaves, the horizontal axis yields logarithmic scales [102]**

The interest points are then localized by application of non-maximum suppression within a $3x3x3$ neighborhood followed by the determination of the maxima of the Hessian determinant interpolated in scale and image space.

After the interest points are detected, the descriptor vector is formed with 64 elements. First, for rotation invariance the orientations of the interest points are determined by finding the Haar wavelet response in $x$ and $y$ directions near the image and scale space neighborhood. Then a square region around the interest point is placed the size of which is determined by the scale at which it is detected as shown in Figure 44.

**Figure 44: The sizes of descriptor windows at different scales [102]**

Then the region is divided into $4x4$ sub regions for each of which Haar wavelet responses, namely $d_x$ and $d_y$, are calculated at $5x5$ sample points in horizontal and vertical directions. For each square, wavelet responses are calculated and the 2x2 sub regions of each square yields the actual descriptor values. These are the summations $d_x$, $|d_x|$, $d_y$ and $|d_y|$ as computed relatively to the grid orientation. The descriptor building process is depicted in Figure 45.

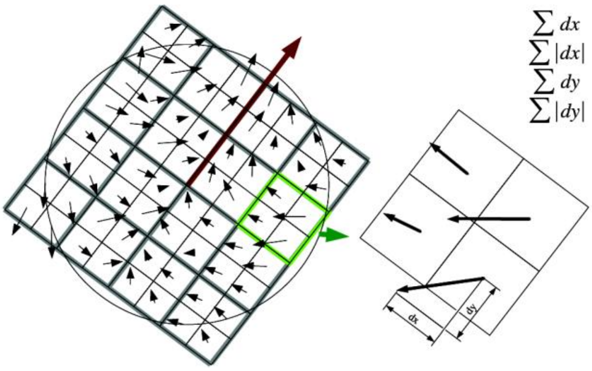


**Figure 45: The descriptor building process [102]**

The summations in the absolute values are used in order to keep the polarity information of intensity changes. Thus, each sub region is left with a 4-element feature vector which, in turn, makes a total of 64 elements for the resultant descriptor of each interest point.

# APPENDIX B

# SHI & TOMASI CORNER DETECTION METHOD

This is a well-known corner detection method also known as Minimum Eigenvalue Method which is introduced as a part of the work in [113] which deals with the visual tracking problem. In this method, the gradients of the input image, $I$, is found in the x and y directions, namely $I_x$ and $I_y$, respectively. Then the matrix $M$ is formed by the squared gradients as given in (B.1).

$$M = \begin{bmatrix} (I_x)^2 & (I_x I_y)^2 \\ (I_x I_y)^2 & (I_y)^2 \end{bmatrix} \tag{B.1}$$

After the sum of the squared difference matrix, M, is calculated the eigenvalues of this matrix are found. The smaller of the eigenvalues corresponds to the corner metric matrix. Then it is only a matter of defining a threshold as in (B.2) for the measure of how sharp should the interest be in order to be counted as a corner.

$$\min(\lambda_1, \lambda_2) > \lambda \tag{B.2}$$

This corner feature extraction method could be seen as the modification of Harris corner method in order to achieve better video tracking results. The main difference with respect to the Harris corner detection lies in the final metric calculation given in (B.2). In the case of Harris corner detection, the metric is defined as in (B.3) instead.

$$\lambda_1\lambda_2 - \mathrm{k}(\lambda_1 + \lambda_2)^2 > \lambda \tag{B.3}$$

A sample execution of Shi-Tomasi corner feature is given in Figure 46.

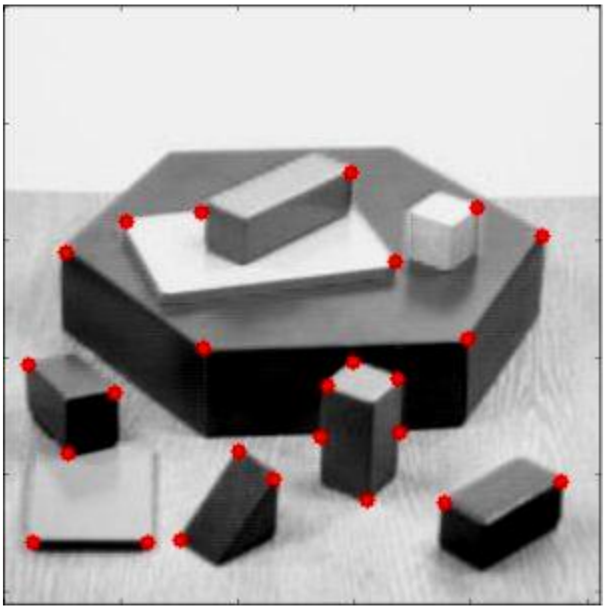

**Figure 46: Corner features detected by Shi-Tomasi method as in [114]**

# APPENDIX C

# NON-LINEAR LEAST SQUARES SURFACE FITTING

For the quadratic surface fitting process used in the proposed system, a non-linear least squares error minimization approach is adopted in which Levenberg-Marquardt method is used for iteratively adjusting the fitting parameters towards a better fit. The implementation of the method is supplied by the MATLAB software. The theoretical details are explained in the remainder of this section.

The problem of fitting model is given in (C.1).

$$y = \varphi(X, \beta) + \varepsilon \tag{C.1}$$

where,

$y: nx1\ vector\ of\ responses$

$\varphi: function\ of\ X\ and\ \beta$

$\beta: mx1\ vector\ of\ coefficients$

$X: nxm\ design\ matrix\ for\ the\ model$

$\varepsilon: nx1\ vector\ of\ errors$

Then the least squares problem can be formulated as given in (C.3).

$$\min_{x \in R^m} f(x) = \sum_{i=1}^{n} (y(x_i) - \varphi(x_i, \beta))^2 \tag{C.3}$$

The residual vector is denoted as given in (C.4).

$$F(x) = \begin{bmatrix} y(x_1) - \varphi(x_1, \beta) \\ y(x_2) - \varphi(x_2, \beta) \\ \cdots \\ y(x_n) - \varphi(x_n, \beta) \end{bmatrix} \tag{C.4}$$

Normally, the least squares function can be minimized via unconstrained optimization, certain characteristics of the problem could be used to improve the iteration efficiency. The said characteristics is observed in the special structure of the gradient and Hessian matrix of least squares matrix. If we denote $J(x)$ as the $mxn$ Jacobian matrix of $F(x)$, $G(x)$ as the gradient vector of $f(x)$, $H(x)$ as the Hessian matrix of $f(x)$ and $H_i(x)$ as the Hessian matrix of each $F_i(x)$ then the equations in (C.5) are obtained.

$$\begin{gathered} G(x) = 2J(x)^T F(x) \\ H(x) = 2J(x)^T J(x) + 2Q(x) \\ Q(x) = \sum_{i=1}^{n} F_i(x) \cdot H_i(x) \end{gathered} \tag{C.5}$$

The matrix $Q(x)$ tends to zero as the residual $\|F(x)\|$ tends to zero as a result of $x_k$ approaching to the solution. This is an effective way to determine the direction of the iterations towards the optimality point. In Levenberg-Marquardt method, this search direction is chosen as the solution of the linear set of equations given in (C.6).

$$(J(x_k)^T J(x_k) + \lambda_k I) d_k = -J(x_k)^T F(x_k) \tag{C.6}$$

In (C.6), $\lambda_k$ is the control scalar to determine the magnitude and direction of $d_k$. If this scalar is chosen as zero, the method is identical to Gauss-Newton method and if this scalar tends to infinity then the method becomes equivalent to the steepest descent iteration. Thus, Levenberg-Marquardt method is a cross between the steepest descent and Gauss-Newton. In other words, it relies on the trade-off between speed and robustness. In the case of quadratic surface fitting, the least squares minimization problem is set as given in (C.7).

$$\begin{aligned}
&n{:}\, Number\ of\ data\ points\ to\ fit\ the\ surface\ to\\
&y{:}\, The\ z-value\ of\ the\ data\ points\\
&\varphi{:}\, The\ quadratic\ polynomial\ function\ for\ fitting\\
&\beta{:}\, Parameters\ of\ quadratic\ surface
\end{aligned} \tag{C.7}$$

The error vector and the model matrix are formed from the definitions in (C.7). After that point, the least squares minimization process determines a starting parameters set for the quadratic surface, finds the error vector as the difference of the data values and the evaluation of the fitted function with initial quadratic surface parameters. Then calculates the direction for the next iteration and iterates until the error between the actual data point values and the fitted function responses is minimized.