TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department of Electrical Power Engineering and Mechatronics

DEVELOPMENT OF VISUAL-INERTIAL ODOMETRY BASED REAL TIME LOCALIZATION SYSTEM FOR GNSS DENIED ENVIRONMENT

VISUAAL-INTERSIAAL-PÕHISE REAALAJAS ASUKOHA TUVASTAMISE SÜSTEEMI ARENDUS KASUTAMISEKS GNSS VABAS KESKKONNAS

MASTER THESIS

Student:	Nazrul Nazeer
Student code:	223610MAHM
Supervisor:	Mairo Leier, Phd.
Co supervisor:	Uljana Reinsalu, Phd
Co supervisor:	Dhanushka Chamara Liyanage, Phd

Tallinn 2024

AUTHOR’S DECLARATION

Hereby I declare, that I have written this thesis independently.

No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

14 May 2024

Author: ....

Thesis is in accordance with terms and requirements

14 May 2024

Supervisor: …........................

Accepted for defence

“.......”....................20… .

Chairman of theses defence commission: ....

/name /

Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I, Nazrul Nazeer

1. grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Development Of Visual-Inertial Odometry Based Real Time Localization System For GNSS Denied Environment"

supervised by Mairo Leier, Phd
co-supervised by Phd Uljana Reinsalu and Phd Dhanushka Chamara Liyanage

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

14 May 2024 (date)

[1] *The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.*

ABSTRACT

Author: Nazrul Nazeer *Type of the work:* Master Thesis

Title: Development of visual-inertial odometry based real time localization system for GNSS denied environment

Date: 06.05.2024 87 *pages (the number of thesis pages including appendices)*

University: Tallinn University of Technology

School: School of Engineering

Department: Department of Electrical Power Engineering and Mechatronics

Supervisor(s) of the thesis: Mairo Leier, Phd; Dhanushka Chamara Liyanage, Phd; Uljana Reinsalu, Phd

Abstract:

GNSS denied localization is an important topic in today's world. Indoor localization is also a topic that is discussed when GNSS denied localization is discussed as in both cases the absence of GNSS is the root problem that is being tried to address. The complete absence of GNSS signals can prove disastrous in modern connected world. Autonomous system that are capable of motion would find themselves in a dead end without a constant update on the current location. There exist several systems that has been studied in this thesis to address these challenges.

The final implementation by the end of this thesis entails a visual inertial odometry method developed to independently estimate the trajectory for up to 400 meters with less than an absolute trajectory error of 3 meters. The thesis also proposes a hardware for the said method to test the proposed solution. The proposed method is tested also on dataset such as KITTI and the performance of the proposed solution is then compared to other state of the art systems. In addition, the functioning of the proposed solution is tested in simulated random adverse situations. The result from all tests shows that the objectives set for the thesis was met and the results were satisfactory and adhering to the goals set.

Keywords: Visual odometry, stereo odometry, IMU odometry, sensor fusion, Kalman filter, error state extended Kalman filter, GNSS, KITTI, Real-time localization

LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Nazrul Nazeer

Lõputöö liik: Magistritöö

Töö pealkiri: Visuaal-intersiaal-põhise reaalaja asukoha tuvastamise süsteemi arendus kasutamiseks GNSS vabas keskkonnas

Kuupäev: 06.05.2024

87 lk *(lõputöö lehekülgede arv koos lisadega)*

Ülikool: Tallinna Tehnikaülikool

Teaduskond: Inseneriteaduskond

Instituut: Elektroenergeetika ja mehhatroonika instituut

Töö juhendaja(d): Mairo Leier, Phd; Dhanushka Chamara Liyanage, Phd; Uljana Reinsalu, Phd

Sisu kirjeldus:

Ilma GNSS-ita lokaliseerimine on tänapäeva maailmas oluline teema. Lokaliseerimine Siseruumides lokaliseerimie puhul on on GNSS-vaba asukoha tuvastus oluline teema, mida püütakse lahendada. GNSS-signaalide täielik puudumine võib kaasaegses ühendatud maailmas osutuda saatuslikuks. Autonoomsed süsteemid, mis on võimelised liikuma, satuksid ummikusse ilma praeguse asukoha pideva värskendamiseta. Nende probleemide lahendamiseks on selles lõputöös uuritud mitmeid süsteeme.

Lõplik teostus selle lõputöö jooksul hõlmab visuaal-intertsiaalse odomeetria mõõtmise meetodit, mis on välja töötatud trajektoori sõltumatuks hindamiseks kuni 400 meetri ulatuses, kui absoluute positsioneerimise viga on alla 3 meetri. Lõputöö käigus töötatakse välja ka arendatud meetodi testimiseks vajalik riistvaraline lahendus. Väljapakutud meetodit testitakse muuhulgas ka KITTI andmekogul ja seejärel võrreldakse pakutud lahenduse toimivust teiste tuntud metoodikatega. Lisaks testitakse pakutud lahenduse toimimist simuleeritud juhuslikes ebasoodsates olukordades. Kõikide testide tulemused näitavad, et lõputööle seatud eesmärgid said täidetud ning tulemused olid rahuldavad ja vastasid püstitatud eesmärkidele.

Märksõnad: Visuaalne odomeetria, stereo odomeetria, IMU odomeetria, andurite liitmine, Kalmani filter, veaoleku laiendatud Kalmani filter, GNSS, KITTI, reaalajas lokaliseerimine

Department of Electrical Power Engineering and Mechatronics

THESIS TASK

Student: Nazrul Nazeer, 223610MAHM

Study programme: MECHATRONICS, MAHM

Main specialty:

Supervisor: Phd, Mairo Leier

Co-supervisor: Phd, Dhanushka Chamara Liyanage

Co-supervisor: Phd, Uljana Reinsalu

Thesis topic:

Development Of Visual-Inertial Odometry Based Real Time Localization System For GNSS Denied Environment

Visuaal-intersiaal-põhise reaalaja asukoha tuvastamise süsteemi arendus kasutamiseks GNSS vabas keskkonnas

Thesis main objectives:

1. Understand the application constraint and advantage of visual-inertial odometry system in GNSS denied localization.
2. Development of visual-inertial odometry system.
3. Test and analyze the developed system

Thesis tasks and time schedule:

No	Task description	Deadline
1.	State of the art, literature review	23 February 2024
2.	Implementation, collection of data, experiments	23 March 2024
3.	Finishing writing the main text	31 April 2024
4.	Supervisor corrections, recommendations	13 May 2024

Language: English **Deadline for submission of thesis:** 13 May 2024

Student: Mohamed Nazrul Mohamed Nazeer .... 13 May 2024
/signature/

Supervisor: Mairo Leier, PhD .... 13 May 2024
/signature/

Consultant: .... “.......”............2024
/signature/

Head of study programme: Anton Rassõlkin, PhD .... “....” 2024
/signature/

CONTENTS

List of figures

List of tables

PREFACE

The master thesis describes in essence the possible application of visual-inertial odometry system in GNSS denied environment. The thesis is developed on the basis and for an ongoing project. However, the specifics and detail of this project is left out from the topic of the thesis as not relevant parts loosely connected to the specific thesis task.

Without the endless love and encouragement of my mother Kamila Hassim and Father Mohamed Nazeer I would never have been able to complete my graduate studies. To them I dedicate my success and share my happiness with them as they have shared with me all that they have. No amount of what I would do would be even comparable to what they have done for me. For I am who I am because of them. To mom, you are the world to me.

Extended gratitude is extended and expressed to Mairo Leier Phd, who has been not only a supervisor of the thesis but also mentor for the last 2 plus years of my academic and professional career. Uljana Reinsalu Phd deserves a special note for being a strong motivation and backbone of the team providing her invaluable knowledge and support throughout the development of the thesis.

List of abbreviations

2D	Two Dimensional
3D	Three Dimensional
ATE	Absolute Trajectory Error
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
DAQ	Data Acquisition Unit
DOF	Degrees Of Freedom
ES-EKF	Error State Extended Kalman Filter
FPS	Frames Per Second
GNSS	Global Navigation Satellite System
GT	Ground Truth
GPS	Global Positioning System
GPU	Graphical Processing Unit
IMU	Inertial Measurement Unit
KNN	K–Nearest Neighbor
LC	Loop Closure
ML	Machine Learning
MONO	Monocular
OF	Optical Flow
OpenCV	Open-Source Computer Vision
RFID	Radio Frequency Identification
RMSE	Root Mean Squared Error
RPE	Relative Pose Error
RSSI	Received Signal Strength Indicator
SLAM	Simultaneous Localization and Mapping
SIFT	Scale Invariant Feature Transform
UWB	Ultra-Wide Band
VO	Visual Odometry
Wi-Fi	Wireless Fidelity
w.r.t	With Respect To
YOLO	You Only Look Once

1 INTRODUCTION

Ever changing rules and dynamics of the world dictates the requirements of different technologies to solve trivial problems. With systems such as global positioning system (GPS), Galilieo, GLONASS or BeiDou which are a subset of GNSS (Global Navigation Satellite System) the era of localizing oneself begun. Being able to know one’s location in the world has great impact in civil, military and disaster relief situations and applications. However, the use of this technology entailed some restrictions and more recently certain counterproductive systems could restrict the use of this technology in certain areas or make the application of this technology less reliable. These areas could be anywhere from buildings, underground caves, tunnels to war zone.

1.1 Background

The ability to geo-locate, primarily establishes the ground for autonomous capabilities. Modern day self-driving cars for example rely heavily on GNSS systems for navigation and autonomous driving. Autonomous driving vehicle has a market that is expected to reach 557 billion dollars by the year 2026 [1], demonstrating the need for location aware system. Localization systems are not only required by autonomous driving vehicles but are also required for aircraft's, ships and all possible types of vehicles. There has been a recent growing trend in the application of unmanned aerial vehicle and unmanned ground vehicles. Which further only implies the requirement of localization systems as an existential criterion.

1.2 Motivation

It’s imperative to find a solution to autonomize for localization in environment where it’s not possible to use GNSS based localization system. Unfortunately adding more GNSS satellites is not a solution, given the physical properties of how these systems work this is not as straight-forward as one would want. Development of localization solution for GNSS denied areas or where GNSS has been affected is therefore an important topic. Localization usually describes the ability to track the movement or the location of an entity in space. Systems that can perform localization are called positioning systems.

Types of positioning systems include radio based and non-radio-based technologies. Examples of radio-based technologies that are used for indoor positioning are Bluetooth/Wi-Fi RSSI (received signal strength), UWB (ultra-wide band), ToA (Time of Arrival) and ToD (Time of Departure). Radio based system has shown limited potential when the applicational requirement is considered. Radio based system often require infrastructural changes (limits scalability) and require additional hardware to facilitate the working of the system and for the improvement of the accuracy of the system. Radio-based system is also prone to interference, jamming and noise which makes this system vulnerable and not reliable for application in challenging environments. On the other hand, non-radio-based technologies which use inertial measurements, visual data, or LIDAR data to localize are of relevance because of scalability advantages accuracy and cost. Camera and inertial measurement sensors are often not so expensive and affordable. Almost every modern-day vehicle in any shape or form has these sensors already integrated in them.

1.3 Objective

In this thesis we will focus on non-radio-based positioning systems. We will specifically develop a positioning method based on visual data from sensors and inertial measurements to track the location in a GNSS denied area. The requirement that would be expected to be achieved by the proposed method are,

1) The developed method must be capable of estimating its current location based on visual features and inertial measurements as the only input.

2) The proposed method should be able to demonstrate robust tracking for at-least a cumulative distance of 400 meters in mostly outdoor setting with an absolute trajectory error of less than or equal to 3 meters for the entire displacement of 400 meters.

3) Robustness of the proposed method in adverse event such as severe data loss will be tested.

The methods will be developed and proposed solutions will be implemented in the thesis. Answering these questions by both reviewing other state of the art approach and by implementation of a vision-based inertial positioning system would allow us to understand our system better. The key research questions being:

1) Comparison by review of literature the accuracy, scalability, and cost-effectiveness between the vision based indoor localization system and existing radio-based positioning technologies like BLE, UWB and WI-FI?

2) How can the system address challenge such as low texture environment and dynamic scene for robust GNSS denied localization?

3) How accurately and consistently can the implemented positioning system determine the position in an GNSS denied environment?

4) How robust is the proposed method when in adverse events that hinders the ability of the system to compute its trajectory. How would the proposed method cope for this using other sensors?

We will test our deployed solution with real-life benchmark dataset called KITTI. We will test our method also by analyzing the proposed method on custom data collected using custom hardware built. In the next Chapter we will review the background of visual odometry by understanding the different classification methods and investigate some of the existing state of the art methods. We will then discuss in Chapter 3 how the proposed method is developed, and the tools used, along with the description of the testing environments and details. In Chapter 4 we will present the quantitate results of the experiments and analyze and discuss the results in Chapter 5 and 6 respectively.

2 LITERATURE REVIEW

To estimate the trajectory as the agent (a vehicle or an object with the visual-inertial odometry system) travers across an unknown or known map is the primary task of a visual odometry system and vital task for autonomous application [2]. Visual odometry system is so called such a system as it's based on visual data to infer motion of the agent. Visual odometry is also an incremental estimator type. That means that the next estimates are added to the previous estimate. A visual inertial odometry system is a visual odometry system coupled with an inertial odometry system in a fusion approach. Different non GNSS positioning systems such as BLE or UWB based systems are developed for indoor application. Considering an indoor situation, we find that using radio-based positioning system does not yield beneficial result given the cost of deployment and scalability issue of these system. In [3] sub meter level accuracy has been shown based on prior works using UWB, Wi-Fi and BLE. The coverage area and coverage accuracy of such system largely depends on the number of deployed devices [4], partly because radio signals weaken over propagation and line of sight requirements. Visual positioning system is not affected by such shortcomings and there is no requirement of increasing the number of devices across a given infrastructure to achieve a better accuracy. There is also this fundamental difference that visual odometry system are placed on the agent itself and other radio-based positioning system are placed on infrastructure around the required area and a receiver is placed on the agent. Radio based system such as BLE, Wi-Fi, UWB or RFID requires additional hardware for improved performance [4] which might entail the requirement of infrastructural changes. Inertial measurement only based localization system by itself is prone to error. The inertial measurement unit readings of acceleration and angular velocity are mainly used in localization by this method. The readings themselves contain biases and often are noisy in most cases where a mid-end inertial unit is used. Like the visual odometry system, inertial odometry system is also an incremental estimator type system. An error in previous estimate will affect all other future estimates.

For BLE and WI-FI based positioning system both accuracy and cost effectiveness of the systems are low. Accuracy is impacted by and depends on the number of access points, interference, and line of sight conditions, and an increase in the number of access points to increase the accuracy increases the cost of the system. UWB or cellular based positioning system have high accuracy however their cost effectiveness is lower given the requirement of additional hardware and the cost of this hardware. Unlike the BLE and WI-FI systems UWB and cellular systems are scalable [4]. IMU only based positioning system are low in accuracy and cost, high in scalability. Visual odometry systems accuracy depends on the type and approach of the system which we will discuss further. However visual odometry system are highly scalable. Table 2.1 provides a qualitative overview of the comparison in a tabular form. The definition of low, high and medium are comparison to each other.

Table 2.1 Comparison of different positioning systems based on key metrics.

	BLE	**WI-FI**	**UWB**	**IMU**	**Cellular**	**Visual odometry**
Accuracy	Low	Low	High	Low	**High**	Medium
Scalability	Low	Medium	Low	**High**	**High**	**High**
Cost effectiveness	Low	Low	Low	**High**	Low	**High**

A visual odometry system can be categorized based on their sensor makeup. That is for example if the system consists of a single camera, based on which the motion is estimated then the system is called monocular visual odometry system. If the system contains two cameras set up in a stereo configuration, then this system is called stereo visual odometry system. Stereo visual odometry have inherent benefits compared to monocular visual odometry. As the monocular visual odometry lacks the ability to infer the scale of the motion given the setup lacks additional sensors that could output a sense of scale to the visual odometry system [5]. Visual odometry system can be classified also based on the type or the approach taken. These are feature based approach, appearance-based approach, and hybrid approach.

Recent development of certain monocular visual odometry approaches has used deep learning-based algorithms to overcome to an extent the scale ambiguity issue. However stereo visual odometry is superior to monocular visual odometry given it’s set up overcomes the problem of scale without any additional complicated algorithms. The third type of the system consist of an RGB-D setup. Which is a RGB camera combined with depth information from stereo setup or TOF sensor. Both monocular and stereo visual odometry setup can be performed on monochrome image, it’s not compulsory for a RGB camera setup. A stereo setup could be easily made to work in monocular visual odometry method. However, the converse is not possible without certain assumption such as assumption that dictates a constant height to a known surface in the frame. The below Figure 2.1 illustrates details of classification in visual representation.

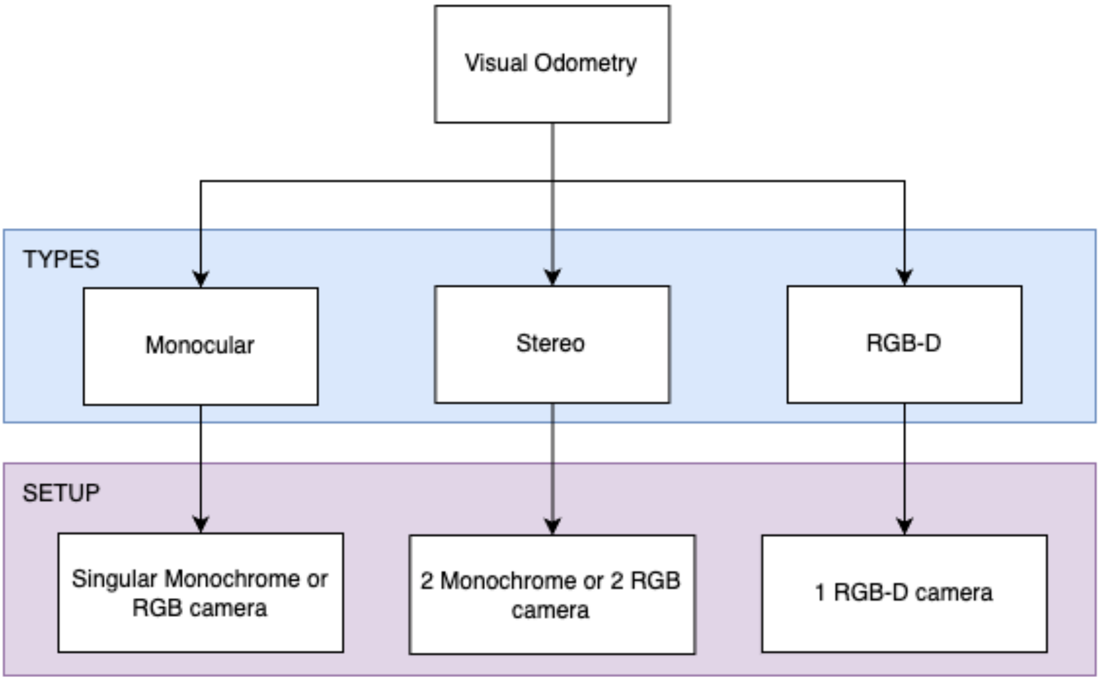
Visual Odometry
TYPES
Monocular
Stereo
RGB-D
SETUP
Singular Monochrome or RGB camera
2 Monochrome or 2 RGB camera
1 RGB-D camera

Figure 2.1 Overview of the types of visual odometry and the associated setup required for the type.

Given over 3 decades of work in the field. We can also classify visual odometry systems by the different approaches such as geometric approach, machine learning approach and hybrid approach, every approach can be categorized into one of three possible classes which are feature based method, appearance-based method, and hybrid methods. It is important that we distinguish SLAM (simultaneous localization and mapping) from Visual odometry at this stage. Visual odometry is a sub system of SLAM or a component of SLAM and is only used in estimation of the ego-motion. However, SLAM system also focuses on map building. SLAM systems are generally more accurate given an event of loop closure. This is an event where the agent onto which the SLAM system is setup, crosses its own path during motion, an event in which a further set of algorithms such as bundle adjustment and graph optimization can be applied to improve the accuracy of the estimates. The trajectory estimates themselves for a SLAM system are from the Visual odometry method (or some other method). The primary focus is on visual odometry method the inertial method will aid the visual odometry method. Therefore, much of the focus on the method proposed is on the visual odometry and so is the state-of-the-art analysis.

2.1 Geometric based method

2.1.1 Feature based approach

Feature based method gained its popularity since the inception of visual odometry. It has been used as the fundamental approach in the early stage of research contributing to visual odometry. Works of Nister et al [6], Howard [7], Cumani [8], Benseddik et al. [9], Naroditsky et al. [10], Jiang et al. [11] and Parra et al. [12] according to [13] all contribute and/or describe this approach. Feature based method was the visual odometry method used on the mars rover. Both rover curiosity and spirit had onboard visual odometry system that computed the 6 DOF pose of the rover [14]. Furthermore in [14] its acknowledged that what initially was meant to be an extra feature (visual odometry) showed remarkable performance such that it ended up being used as critical vehicle system.

In feature-based approach two consecutive images at time t and $t-1$ are used, salient feature in each of the image is detected. These detected features are then matched across both the images. The result is a set of corresponding image points on both the image which can then be used to estimate the rotation and translation of the system. To detect the salient feature a detection algorithm is applied over the image or parts of the image. These algorithms are called feature detection algorithm. They are used not only in visual odometry application but also in image stitching and various other computer vision related applications. Some of the most well-known algorithms are SIFT, SURF, ORB, AKAZE, Good feature to track, BREIF etc. These algorithms are notable for either their robustness to the environment, the computational efficiency, scaling invariance or brightness invariance. Salient features are detected based on their uniqueness and robustness. It’s needed that the best algorithms find the most unique set of points in an image that can be clearly identified in the corresponding image, with consideration for the computational requirements which must be low and invariability to rotation and scale changes of the image and strong adaptability to brightness changes. This is what characterizes an optimal algorithm for the specific application as mentioned in [15]. After the salient points are detected across the two images there needs to be a way to describe these points. In other words, we need to be able to describe these points so it can be identified in the next consecutive image. Descriptors are unique description of a given feature point. They are computed using descriptor algorithms such as SIFT, ORB, SURF etc. The idea of descriptor is it can be used to find the location of the salient feature in the next image. To match the salient feature detected and described matching algorithms such as brute force or FLANN based matcher are employed. These algorithms find the set of image points that are present in both the images and are matched. From these points the motion can then be estimated. Motion is estimated using motion estimation algorithms (2D to 2D, 3D to 3D and 2D to 3D). Tables 2.2, 2.3 and 2.4 describe the working steps of the different motion estimation algorithms. F represent the frame from the camera and t represents the time. F_{lt} and F_{rt} represents the left and right frame at time t from the camera.

Feature based methods are extremely effective in texture rich environment [13][16]. In texture less environment the method struggles or completely fails. If there are not enough correspondence matched per the requirement of the motion estimation algorithm or are no correspondence that can be matched, then the visual odometry system fails. And this is where the appearance-based method is advantageous over the feature-based method.

Table 2.2 2D-2D motion estimation algorithm

```
Get frame $F_{t}$
With $F_{t}$:
    Extract and match features from $F_{t-1}$ and $F_{t}$
    Compute essential matrix
    Compute transformation matrix
    Motion = previous motion * transformation matrix
```

Table 2.3 3D-3D motion estimation algorithm

```
Get frame $F_{lt}$ and $F_{rt}$
With $F_{lt}$ and $F_{rt}$:
    Extract and match feature of $F_{lt-1}$ and $F_{lt}$
    Compute depth map from $F_{lt-1}$ and $F_{rt-1}$ and from $F_{lt}$ and $F_{rt}$
    Triangulate the 3D points for $F_{lt-1}$ and $F_{lt}$
    Compute transformation matrix from 3D points
```

Table 2.4 2D-3D motion estimation algorithm

```
Get frame $F_{lt}$ and $F_{rt}$
With $F_{lt}$ and $F_{rt}$:
    Extract and match feature of $F_{lt-1}$ and $F_{lt}$
    Compute depth map from $F_{lt-1}$ and $F_{rt-1}$
    Triangulate the 3D points for $F_{lt-1}$
    Compute cameras pose (PnP)
```

2.1.2 Appearance based approach

As described in the feature-based method, appearance-based method has an advantage over the later. This is because of the way the appearance-based method works. The estimation of the motion is done using the optimization of the photometric error in this approach [16]. In the feature-based method if the salient feature matched is incorrect than the estimation of the motion is affected, and the rectification of the error could be computationally expensive [16].

However, the appearance-based method utilizes the information form the entire image to deduce a robust estimate [16]. Region-based matching and optical flow-based approaches are the two main categories of appearance-based methods [16]. In template-based approach a patch or region of an image from the current frame is extracted and attempted to be matched onto the next frame. The displacement and rotation of the system is estimated using the matching of this template across the next frame. The algorithm initializes by extracting to consecutive images, after which the template which is a patch or region of intrest of the first image is extracted and matched with the second image at hand through normalized cross correlation. The pixel displacement from the template and the maximum correlation point is extracted. From these pixels information the motion of the system is estimated using the known intrinsic and extrinsic parameters of the camera.

The optical flow-based method works differently, the OF method computes the brightness pattern displacement from one picture frame to the next using the intensity values of the adjacent pixels [13]. Those algorithms which estimates the displacement for every pixel in the image is known as dense optical flow and conversely if the displacement of only selected pixel is computed then it’s referred to as the sparse optical flow. In the former the motion is estimated by aggregating all the computed motion vector of the individual pixel and employing robust algorithms such as RANSAC for the estimation of the motion. In the sparse optical flow however, we track the selected feature to the next frame and then employ one of the algorithms such as 2D-2D or 2D-3D motion estimation algorithms such as in the feature-based approach. One of the drawbacks of appearance-based method is that the motion between the two consecutive frames cannot be significant or large. This is because of the assumption and reliance of this appearance-based approach on pixel intensity and brightness as a matching criterion. And significant motion between consecutive frame can have a large difference in pixel variation which then cannot be tracked correctly or detected.

2.1.3 Hybrid based approach

Having their own advantages and disadvantages for both features based and appearance-based methods. The hybrid approach aims to fuse the best in both aforementioned approaches to estimate the motion robustly. Thus, forming a coalition that is invariant to any environment. There are a few different ways such an approach can be designed. One such design was proposed by [17] [13].

In this work [17] [13] the appearance-based approach was used to estimate the rotation of the system and the feature-based approach was employed to estimate the translation. In this way the hybrid-based approach shows more robustness in low textured environment and consecutively also when there is significant motion between the frames. One other way to design a hybrid-based approach would be to selectively employ each of the algorithm given a metric that evaluates the significance of the motion such as the velocity of the system and the presence of texture. Sensors such as vehicle speed sensor and frame rate of the camera can be also employed as criteria that would decide on the application of the appearance-based methods. When the motion between the consecutive image is small, identified by lower velocity at a high frame rate the appearance-based approach can be employed. Failure in either of the criteria the feature-based approach can be employed. By analyzing the number of corners in an image, a sub metric on the amount of texture in a frame can be estimated. Which can be used to decide the application of the feature-based approach. As mentioned in the beginning of the Chapter Visual odometry is of the type incremental estimator. Therefore, if an estimate becomes erroneous because of incorrect or noisy feature matches or incorrect approximations of the pixel then this error is propagated moving forward.

2.2 Machine learning based method

Modern visual odometry methods are increasingly based on deep learning approach or a combined approach given the availability of more computational resources [16]. Deep learning-based method negates the requirements of some of the pre required knowledge such as for example the intrinsic and extrinsic parameters of the camera or cameras for the visual odometry system. While traditional algorithms such as SIFT, SURF and ORB etc. may have each their advantages, the deep learning approach can form a better alternative which combines the best of all this method [18]. One of the initial works in the application of machine learning based visual odometry method is highlighted by the works of Roberts et al [17]. Later, several works such as [2], [19 - 22] highlighted the applicability of deep learning methods on visual odometry. One of the drawbacks when machine learning based method is compared to geometric based method is the computational resource requirement and subsequent hardware requirements. If we intend to optimize the models for speed, then there is a requirement for additional hardware such as tensor processing unit. However, optimizations have tradeoff in terms of accuracy of the system. Machine learning based method also can be classified like the geometric based approach in to feature based approach, appearance-based approach, and hybrid approaches. The later in this realm could also involve the application of geometric based method in conjunction with machine learning based method.

There are number of different ways the deep learning-based method can be designed. In both the feature-based approach and appearance-based approach which can be grouped together as geometric based approach, the required step to estimate the motion is represented by different algorithms at each stage such as feature detectors, matchers, dense optical flow estimation or motion estimation algorithms etc. Deep learning models can be employed to either to replace any one of these stages in the visual odometry algorithms or multiple different stages or the entire system. For example, [23] proposed a technique to learn good feature which are unique salient feature from an ego motion estimation task. This method could be replacement for the feature detection stage of the geometric based system. Further work such as of Zhaou et al [19] suggested to learn single view depth estimation and visual odometry in an end-to-end fashion. In this case the shortcoming of the monocular system that was described earlier is rectified by replacing the stage with a monocular depth model that could best approximate relative depth. Coupled with an end-to-end system this replaces a traditional visual odometry system or in other words a geometric based approach. A complete replacement of the geometric based method is proposed by Wang et al [20]. Where the input is a sequence of images, and the output is the transformation matrix [2].

2.3 State of the art

Over the last two decades or so several state-of-the-art systems have been developed with competitive and improving results over time. Table 2.5 depicts some of the state of the art from the works of several authors. End-to-End network means that every stage of the visual odometry system is a machine learning based approach including the motion estimation part.

Table 2.5 A few state-of-the-art systems, their types, and approaches

Method	Mode of operation	Type	Approach
VISO2 [24]	Stereo	Geometric	Feature based
ORB-SLAM [25]	Mono/Stereo/RGB-D	Geometric	Feature based
DSO [26]	Mono/Stereo	Geometric	Appearance based
Depth-VO Feat [21]	Mono/Stereo	ML	End-to-end
SC-SfMLearner [22]	Mono/Stereo	ML	End-to-end
DF-VO [2]	Mono/Stereo	ML	Hybrid

VISO2 was initially designed for 3D reconstruction of a scene. It has implementation of Kalman filter for further refinement of the estimation’s VISO2 is based on a feature-based approach of visual odometry. ORB-SLAM 2 is a SLAM system. The system Contains robust techniques such as key frame insertion, loop detection, bundle adjustment etc. It was designed to work in real time. ORB-SLAM system has shown great robustness and efficiency amongst all other visual odometry system. However, it’s noted in [2] that sometimes ORB-SLAM 2 suffers from tracking failures or unsuccessful initialization. ORB-SLAM 2 is a feature-based variant of visual odometry like the VISO 2. DSO sparse – direct method, in which sparse features are extracted and used to estimate the motion. Depth-VO Feat is a machine learning based visual odometry system. The proposed pipeline of the system is shown on Figure 2.2. A combination of CNN based pose estimation with depth CNN allowing for a monocular approach with scale estimation. The system is a multi-stage system with each stage relaying on machine learning models as a replacement to otherwise a geometric solution. DF-VO uses combination of optical flow network, single view depth network and geometric method for motion estimation and hence can be classified into a hybrid approach. Figure 2.3 represents the architecture of the DF-VO system as proposed by the authors. It’s important to understand, for example the state-of-the-art system DF-VO was trained on the same dataset that it was evaluated. Leading to believe that the results of machine learning system can be largely influenced from prior knowledge and unseen or untrained environment can prove a challenge for such system. Which would lead to a requirement for continuous model training over new data frequently for better performance. This could lead to a drawback for the machine learning based approach. The drawback being the in-availability of enough data for training. And subsequently may lead to decrease efficiency and performance in compassion to non-machine learning based methods such as the geometric based method.

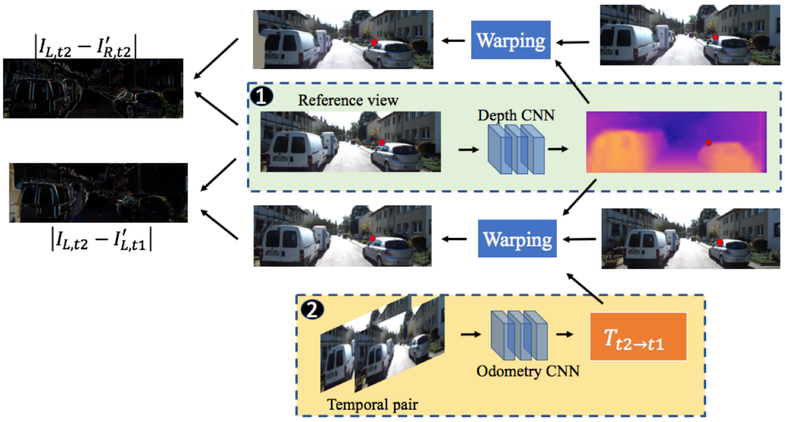

Figure 2.2 Proposed technique for Depth-VO feat visual odometry [21]

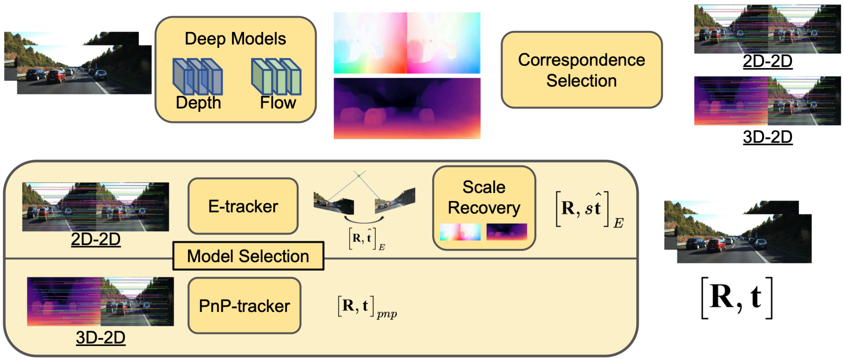
Deep Models
Depth
Flow
Correspondence Selection
2D-2D
3D-2D
2D-2D
E-tracker
$[\mathbf{R}, \hat{\mathbf{t}}]_E$
Scale Recovery
$[\mathbf{R}, s\hat{\mathbf{t}}]_E$
Model Selection
3D-2D
PnP-tracker
$[\mathbf{R}, \mathbf{t}]_{pnp}$
$[\mathbf{R}, \mathbf{t}]$

Figure 2.3 DF-VO system architecture [2]

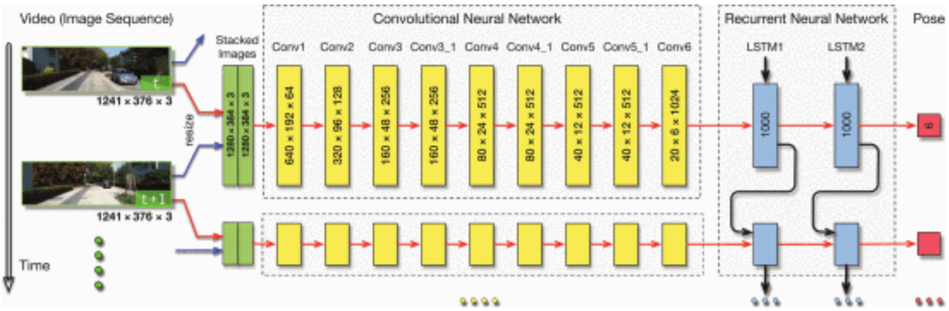

Figure 2.4 End-to-end architecture based on CNN proposed by [20]

In short, the above Figures 2.2 and 2.3 represent the different type of combination of approaches in the case of machine learning based approach for visual odometry can be designed. End-to-End approach can also be considered as an appearance-based approach in geometric terminology as the model uses all pixel of the image to compute the pose estimation. Figure 2.4 depicts an end-to-end machine learning based visual odometry approach as proposed by [20]. Where the input to the system is two images in a temporal sequence and the output is the pose estimation. Figure 2.5 depicts the classification of geometric based and machine learning based state of the art based on the primary approach.

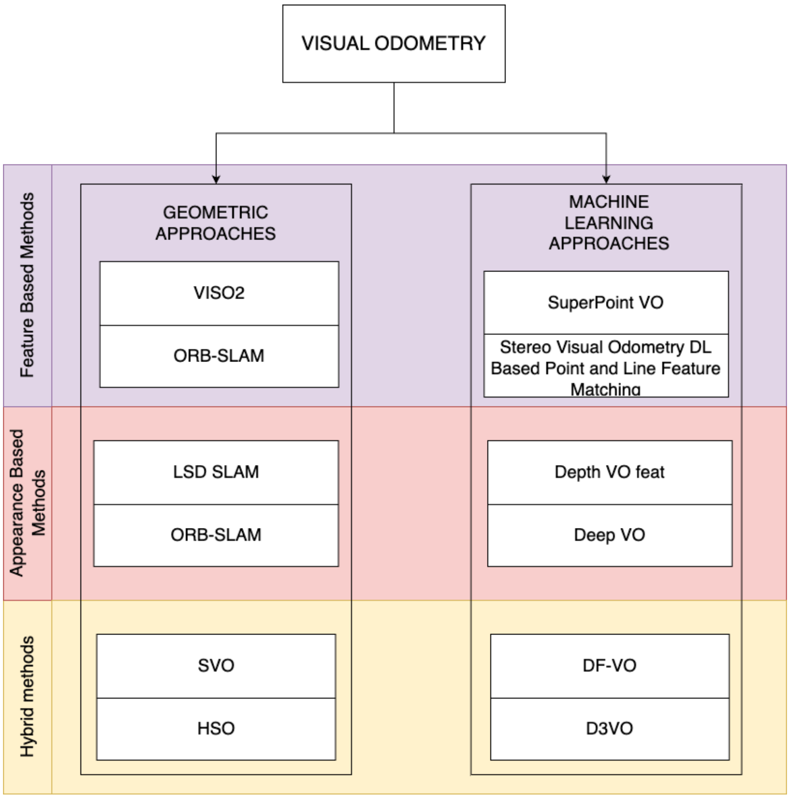

Figure 2.5 Visual odometry systems classified by approaches taken based on existing methods

Few systems that are proprietary and implements a visual inertial system are Apple ARKit, Google ARCore, Intel RealSense T265 and Stereolabs ZED 2. The systems mainly Apple ARKit and Google ARCore require apple operating system and android operating system to run respectively. The Intel real sense T265 is a stand-alone off the shelf component that can be purchased and integrated, the output of this system is 6 DOF position estimates. According to [27] both Apple ARKit and GoogleARCore have demonstrated high accuracy and stability in both indoor and outdoor environments. However, T265 and ZED2 are only recommended for their compatibility with integrating systems.

It’s important to note there exist several other systems all of which have not been considered. Only those visual odometry system that has been used as a benchmark for other works and mainly in [2] has been reviewed. Out of all this system, the ORB-SLAM 2 system and more recently ORB-SLAM 3 system are considered for benchmarking of other visual odometry system given their excellent and efficient performance [2].

Considering the discussion above on state-of-the-art systems and based on the comparative analysis that we would perform in Chapter 4, It’s a valid question as to why when certain systems such as ORB-SLAM 2 or DF-VO is available open source is there a need to develop a system from scratch. Not all methods are suitable for all types of application. There is serval constraint that may restrict the use of a system for a particular application, for example, DF-VO [2] is a machine learning based system that requires extensive computation resources compared to the resources available in single board computer such as raspberry pi or Nvidia jetson nano. At the same time real time system such as ORB SLAM 2 is based on ROS framework. The applicational requirement sometimes do not have ROS framework as a part of the software stack. Therefore, the direct application of this system is limited. Furthermore, to better understand this field of computer vision and to be able to contribute to the field it’s imperative that one learn the fundamental of visual odometry. Also, scratch build of the system allows for easier integration and upgrading of the system overtime which is difficult to be achieved using off the shelf devices such as intel real sense T265 or ZED 2. In some cases, such as ZED 2, have not made their algorithm public therefore constraining the applicability of this solution into custom projects.

2.4 Dataset setup

In order to verify a proposed solution, it is necessary to evaluate the system on known truth. There by enabling us to calculate a quantitative metric to represent the efficiency and applicability of the system. KITTI [28] and Oxford Robotcar [29] are two popular benchmarking datasets [2]. These datasets contain recorded image through outdoor driving situation of various length and complexity. This dataset represents the real-life information that is available to visual odometry or SLAM system. The availability of these data helps us to perform and test our system without the need for hardware setup that would otherwise increase the complexity for the development of visual odometry system. These datasets do not only contain sequence of images, but they contain LIDAR point cloud data, inertial measurement unit readings and high-quality ground truth. The dataset is meant not only for visual odometry or SLAM but are also meant for development in the field of object detection, fusion etc. For our experimentation and testing purposes we will work with the KITTI dataset. The KITTI dataset contains several splits [28]. The KITTI Odometry split is the one that we will consider. This split contains 11 driving sequence with ground truth and camera poses. Each sequence is different in terms of their speed or distance travelled or even in different situation such as highway, urban areas, city etc. Figure 2.6 shows the setup with which the KITTI dataset was constructed. The sensors such as camera, LIDAR, GPS and IMU are mounted onto a vehicle and the information from this device are recorded as the vehicle moves along the road.

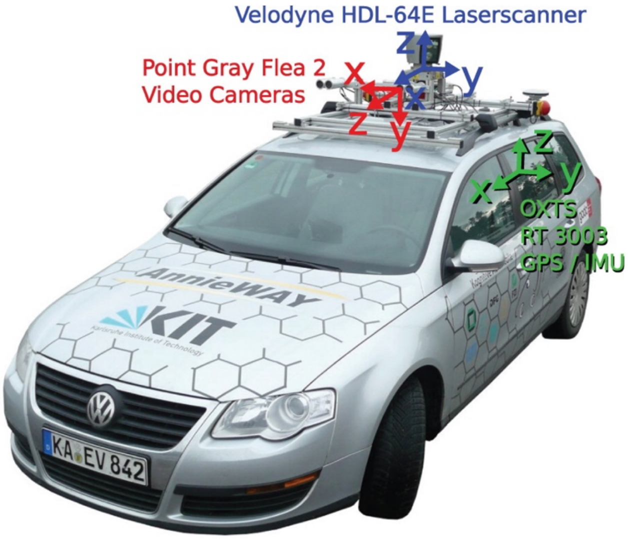

Figure 2.6 The car with the sensors that was used to collect data for the KITTI dataset [28]

In order to evaluate the results of the visual odometry system, it’s important to understand what the output of the system are. The system is supposed to output a 6 DOF estimate which are position estimates in X, Y and Z direction and roll, pitch, and yaw estimates. In order to evaluate the system, we can compare the system estimates to ground truth. Ground truth is the real or true estimate of the system. These are usually obtained from high quality GPS receiver or strictly self-computed. There exist few metrics across which the evaluation of the system must be made in order to be compared against other state of the art system. In [2] the author estimates the following metrics, average translational and rotational error per 100 meters in percentage for up to 800 meter. Absolute trajectory error which is the root mean squared error between the estimation from the visual odometry system and the ground truth. Relative pose error, which matches frame to frame error in both translation and rotation.

Since we would be using the KITTI dataset for our experiments we would analyze the above metrics for the same.

$$ATE = \sqrt{\frac{1}{N}\sum_{k=1}^{N} | p_k^{est} - p_k^{gt} |^2} \qquad (2.1)$$

$$\mathrm{RPE_{trans}} = \frac{1}{N-\Delta}\sum_{t=1}^{N-\Delta} | (Q_t^{-1}Q_{t+\Delta})_{\mathrm{trans}} - (P_t^{-1}P_{t+\Delta})_{\mathrm{trans}} | \qquad (2.2)$$

The equation 2.1 represents the Absolute Trajectory Error as the square root of the average of the squared Euclidean distances between the estimated positions and the ground truth positions over all N timestamps or positions in the trajectory. p_k^{est} represents the estimated trajectory and p_k^{gt} represents the ground truth. Equation 2.2 describes the formula for the computation of the average RPE translational. The formula is for error in translation over a fixed interval represented by Δ. $P_t \; and \; P_{t+\Delta}$ are pose estimates at time t. $Q_t \; and \; Q_{t+\Delta}$ are corresponding poses in the ground truth trajectory. $(P_t \; and \; P_{t+\Delta})_{trans} \; and \; (Q_t \; and \; Q_{t+\Delta})_{trans}$ are the translational component of the estimated and ground truth poses respectively. The same formula as in 2.2 can also be used to calculate the rotational RPE in which case the rotational component is used instead of the translational component.

The average translational error is given by the ATE when N is 100 is evaluated and N at 200 is evaluated so on until N is 800 is evaluated. After which the average of the ATE average over each sequence is added and divided by the number of sequences, and then converted to percent. This gives us the estimate of the average translational error in percentage. RPE indicates the local accuracy of the trajectory of a fixed interval, whereas the ATE represents the global accuracy of the trajectory.

In order to compute the metric, we will use the readily available tool proposed and developed by the author of [2]. This would allow us to easily compare our results with the results from the state-of-the-art approaches on which evaluation was produced based on the above metrics. The uses of dataset such as KITTI would be the best way to represent out system. However, the current system that is being developed is for both outdoor and indoor environments and as such in order to test the systems performance when certain advantages such as large baseline between cameras offered by the KITTI dataset are not available is crucial. Therefore, an indoor dataset based on off the shelf hardware would be collected and system would be evaluated on it.

2.5 Conclusion

So far, we have understood that visual odometry is a process in which the motion of a vehicle is incrementally estimated primarily from vision-based sensor such as cameras. Visual odometry can be classified either based on the hardware or on the approaches. The hardware-based approach classifies the visual odometry into monocular, stereo, and RGB-D systems. The visual odometry method can be also classified into geometric based approach and machine learning based approach. Both categories can be further divided into feature-based approach, appearance-based approach, and hybrid approach. Table 2.6 highlights the fundamental problems that differs for each approach. These issues are consistent only with the geometric based approach. As a well-trained and designed machine learning based approach could learn to be robust against these issues. Which is an advantage to using machine learning based approach.

Table 2.6 Overview of problems in different VO approaches

Problem	**Scale of the problem (Low, Medium, High)**		
	Feature based	**Appearance based**	**Hybrid approach**
Computational requirements	Medium	High	Medium
Lighting changes	Low	High	Low
Scale changes	Low	Medium	Low
Large motion	Low	High	Low
Low texture scenes	High	Low	Medium
Real time	Medium	Medium	Medium
Hardware complexity	Medium	Medium	Medium

Here low represent that the problem in question is easily solvable without any significant changes to hardware or software. An example of this would be for example trying out different feature detector for feature-based approach. Medium represent the requirements of complex algorithms or additional hardware in order to solve the issue. Hardware complexity for all cases is ranked medium as to suggest that by default it’s possible to have monocular visual odometry system for all approaches, but if scale is in question and for a more complete visual odometry system the requirement of a stereo system or RGB-D system increases the complexity of the problem. A high rank suggest that the approach might fail and needs a significant modification or changes in order for the visual odometry system to be robust without failures. Machine learning based VO approaches are computational inefficient then their geometric based counterparts. However, all the other problems for machine learning based approach can be describes as low given that a machine learning based system is able to learn over time to improve the model making it robust to the problem.

Visual odometry system are only as good as their hardware and approaches. And the type of hardware or approach depends on the intended purpose and design. There is no one approach for all; however, this is soon to be realized given the progress in machine learning based approach for visual odometry system. In the next Chapter we will look at the proposed system. Which will be based on feature-based approach combined with machine learning approach.

3 METHODOLGY

As we discussed in the previous section that visual odometry can be classified based on the hardware composition as monocular, stereo or RGB-D odometry. For our implementation we have implemented stereo based visual odometry. To allow us to operate in both monocular and stereo setting individually, necessary adjustment and incorporation of both 2D-to-2D and 2D-to-3D algorithms have also been implemented. However, unless otherwise specified, the method uses stereo visual odometry approach. The visual odometry method that is implemented is a feature-based approach of the geometric variant with machine learning based add-ons to improve our method, making the implemented method a hybrid approach by design. While the state of the art considered in the previous section for the machine learning based approach and hybrid approaches are vitally different from our implementation, our approach can still be classified as a hybrid approach. A feature-based approach was selected for the system as it offered robustness against large scale motion between frame compared to appearance-based approach. Also, the lighting changes would be of less effect to this approach then the appearance-based approach, since we plan to implement our method in different environments this is important for us. However, the susceptibility of this approach to low texture scenarios is a problem which we plan to address using the proposed method. Figure 3.1 below defines the proposed method. The Figure outlines the different nodes of the system and forms together a pipeline. Visual odometry methods proposed by [30], [31] and [32] implement a method where dynamic objects are filtered out with and without using machine learning based methods, our system also use a similar approach. The above system implemented by the authors, however, do not evaluate the performance of the proposed method on the KITTI dataset. In our case we will evaluate the performance of the system on the KITTI dataset and compare it with other state of the art systems.

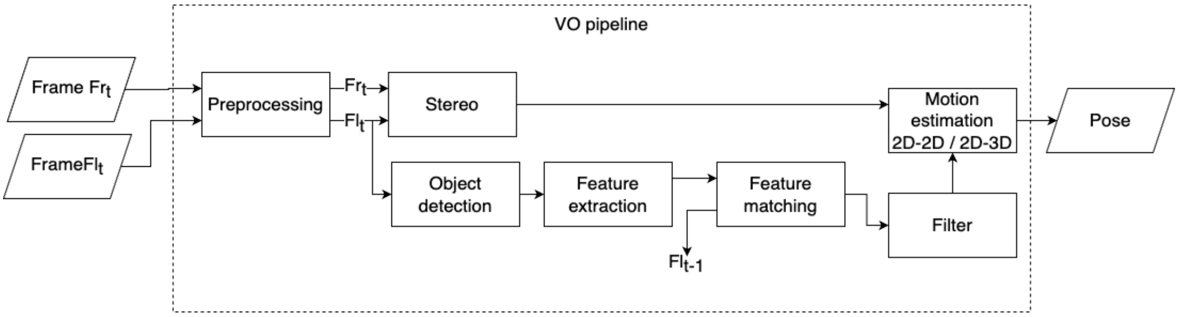

Figure 3.1 Pipeline of the proposed visual odometry method

The implemented visual-inertial odometry method comprises of the visual odometry pipeline and the fusion pipeline. The fusion pipeline with the visual odometry is show on Figure 3.2. Figure 3.2 also represents the entire proposed method. There are two types of coupling for a given filter. Either tightly coupled or loosely coupled. The difference between the coupling is that in a loosely coupled filter, the filter or fusion is applied to the processed positional estimate, which is considered noisy, in the tightly coupled variant however the filter is applied on raw values before preprocessing it to final estimates. In our implementation the implemented Kalman filter for fusion is loosely coupled. Tightly coupled applications have higher accuracy then loosely coupled filter. However, the complexity of the implementation of a loosely coupled filter is less compared to the other approach. Also, our proposed method uses complementary sensors. This is because the error from the sensor in this case would not be correlated. And in the event one of the systems fails the other system could continue working.

The visual-inertial odometry method therefore, in this case is a fusion system where estimates from visual odometry and IMU are used to predict the next state. In the event there is no predictions from visual odometry system the prediction of next state solely relies on estimates from the IMU, and the opposite also remains true. Therefore, it’s important to establish the fact that in this case given the current setup of the fusion system the estimate from the fusion system cannot be better than the visual odometry system. The current application of the fusion system is better described as a compensator and act as to smooth the visual odometry estimates primarily. The best use case scenario for this system is therefore validated by measuring the robustness of the system when there is occasional or large drop in visual odometry estimates.

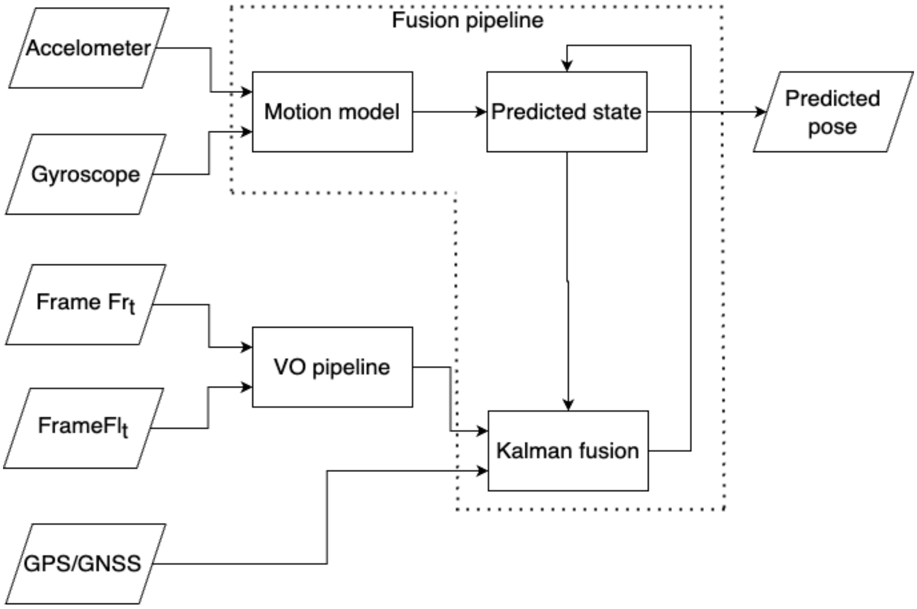

Figure 3.2 Proposed method for the fusion system

3.1 Hardware setup

For a visual odometry method as was discussed in the previous Chapter, the hardware set up determines the type of visual odometry system. In our implementation we set up our system to work in both monocular and stereo mode. For this the setup requires two cameras and a processing unit. Additionally, an inertial measurement unit is also incorporated for the fusion node. Therefore, the implemented visual odometry method requires two monochrome cameras in stereo configuration, a computing unit, and an inertial measurement unit altogether forming a system. We will test our method on both a publicly available, widely used benchmarking dataset KITTI and data collected by a DAQ unit that was developed for this thesis. It is necessary to outline each of the setup as some assumptions and calculation depend on the placement an orientation of the sensors.

3.1.1 KITTI setup

The KITTI odometry setup was done on a car as shown on Figure 2.7. From [28] we get the following Figure 3.3 that outlines the setup.

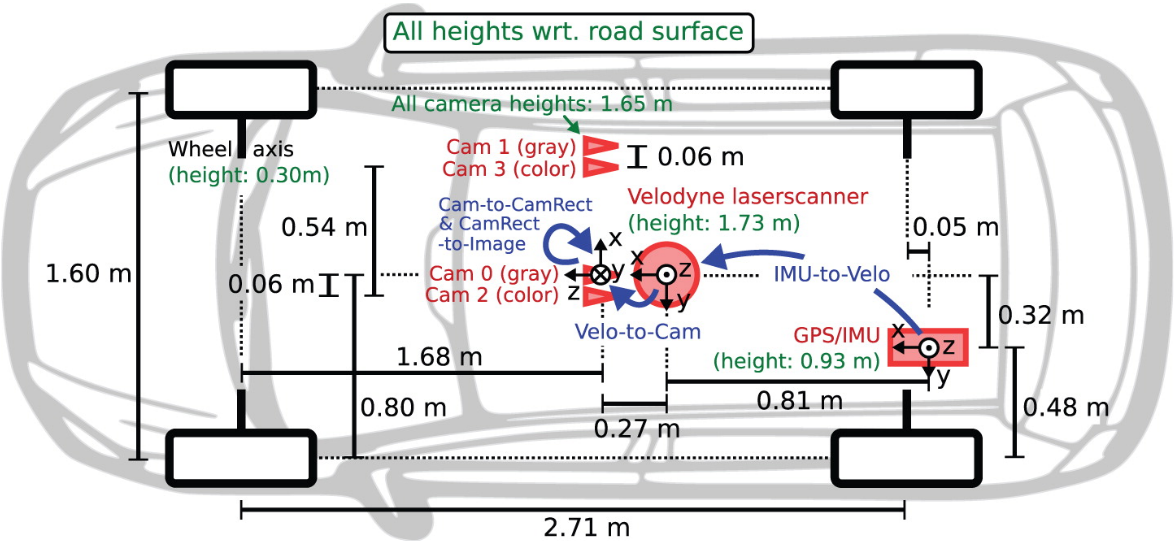

Figure 3.3 Placement of sensors in the vehicle (agent) used to collect the KITTI dataset [28]

While the KITTI setup contains a lot of additional sensors, however for our system we require the data from the following mounted sensors.

1) Cam 1 (gray)

2) Cam 0 (gray)

3) GPS/IMU from the OXTS sensor mounted on the rear end as shown on the image.

Both Cam 1 and Cam 0 are Point Gray Flea2 grayscale cameras (FL2-14S3M-C), 1.4 Megapixels, 1/2” Sony ICX267 CCD, global shutter. The lenses mounted on this camera are of type Edmund Optics lenses, 4 mm, opening angle ~ 90°, vertical opening angle of region of interest (ROI) ~ 35°. The distance between the cameras is approximately 54 cm. An OXTS RT3003 inertial and GPS navigation system was used for GPS and IMU data which provides 6 DOF data at 100 Hz. In order to record all information, they have used the following computing setup with two six-core Intel XEON X5650 processors and a RAID 5 hard disk storage. Additionally, it is also worked out from the information provided that the frame rate at which images are captured and stored from the camera is at 10 Hz [28].

The following coordinate system is noted from the Figure 3 with respect to the car:

1) Cam 0 and Cam 1: X = right, Y = down and Z = Forward

2) GPS/IMU: X = forward, Y = left and Z = up

3.1.2 Our setup

Figure 3.4 outlines the setup in drawing and Figure 3.5 shows the mounting on the vehicle for our hardware setup.

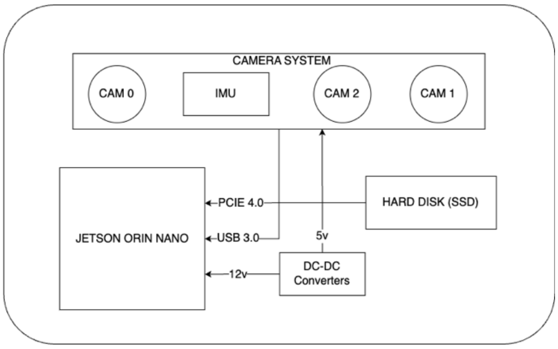

Figure 3.4 Hardware setup overview

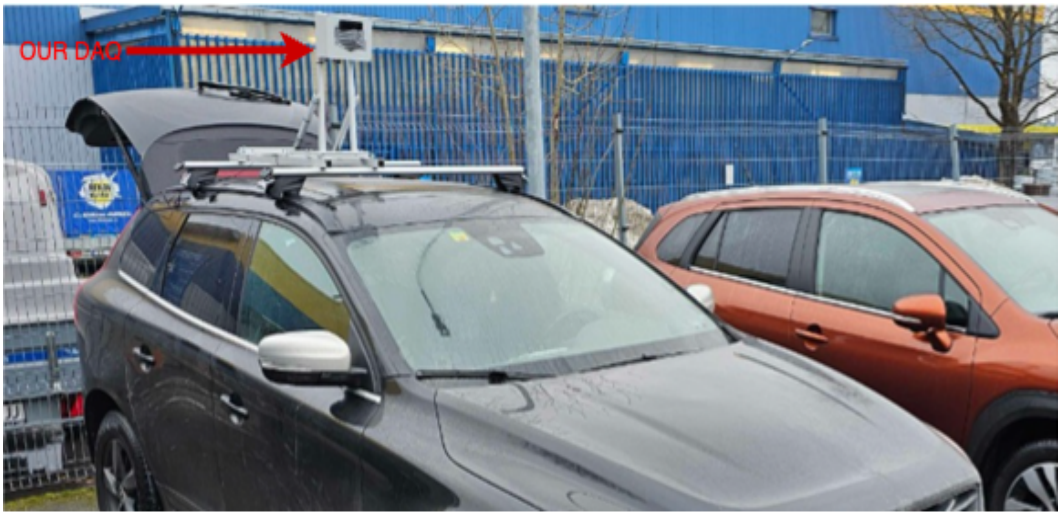

Figure 3.5 Setup mounted on a vehicle

The setup compromises of the following hardware:

1) OAK – D LR

2) Jetson Orin Nano 8GB

3) DC – DC Converter 12v to 5V

Both Cam 1 and Cam 0 are RGB camera AR0234, 2.3MP, global shutter. The distance between the camera is 15cm. The inertial measurement unit is a BNO085 9-axis. The sampling of both the IMU and the cameras are synchronized. The computing unit is a jetson Orin Nano 8GB. An external solid-state drive is attached to the jetson via the PCIe 3.0 slot on the Orin Nano. The computing unit acquires the frame from the camera and the IMU data and stores it at approximately 20Hz. There is occasionally a drop in approximately 2 to 4 frames bringing down the frame rate to 18 - 16 frames per second sometimes.

Coordinate system with respect to the car:

1) Cam 0 and Cam 1: X = right, Y = down and Z = Forward

2) IMU: X = up, Y = left and Z = Forward

3.2 Software setup

The software is written in two parts. The visual odometry pipeline and the fusion pipeline. Each pipeline is to be run parallel. If and when there is a location estimate available from the visual odometry pipeline, then estimate is fused. Else the fusion pipeline predicts the state only based on accelerometer and gyroscope data from the IMU. Parallel computation is also necessary as IMU is a high-rate sensor compared to visual odometry system. Our fusion system can fuse visual odometry estimates, GPS/GNSS estimates and IMU estimates when available. This allows for the proposed method to be invariant to sampling rate of the sensor and allows for the fusion of sensor sampling at different rates. Each system will be described in detail in the below section.

This pipeline as shown on Figure 3.1 compromises of several nodes that must be individually briefed on.

3.2.1 Preprocessing

It is not possible to use raw images from the sensor for the proposed visual odometry method. This is because every image captured by the camera through a lens have barrel distortion. To remove any distortion, we need to post-process the images. In the post processing step the intrinsic and distortion parameters of the camera are used for undistorting the images. Figure 3.6 shows an example of a raw image captured using an imaging sensor on the left and the undistorted image on the right using the parameters of the camera. The camera parameters which contain the intrinsic, extrinsic, distortion matrixes are computed by the calibration of the camera using an image with known shape. Like for example on Figure 3.6, the checkboard in the image was used to calibrate the camera for this example.

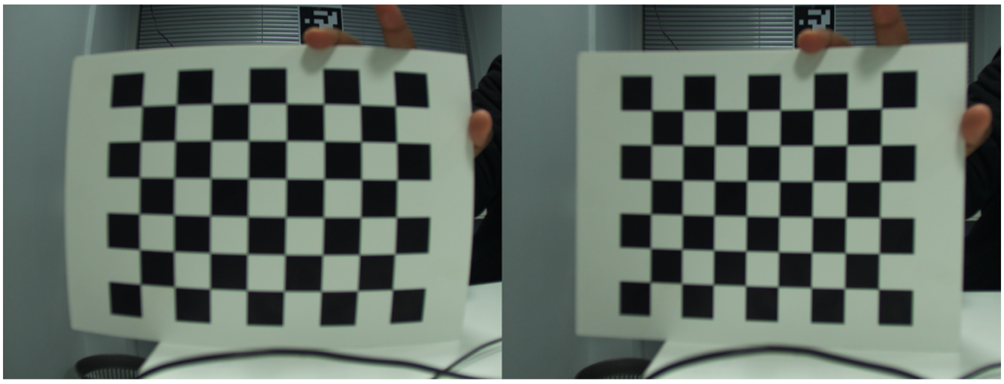

Figure 3.6 On the left is the distorted image and on the right is undistorted image

We can see that without un-distortion the image appears to be curved outwards. This would become a problem for the visual odometry system during any of the feature identification, matching, stereo or motion estimation processes. A preliminary requirement to this node is that the images are rectified. When the images are rectified corresponding features in the image lie along the same horizontal plane. This alignment is crucial for the stereo node. The parameters that are required to rectify images are the intrinsic parameter and extrinsic parameter. Using which a transformation matrix can be computed which along with the intrinsic parameters would align both images along the same horizontal line in a process that can be referred to as stereo rectification. Figure 3.7 and 3.8 shows undistorted images before rectification and after rectification respectively.

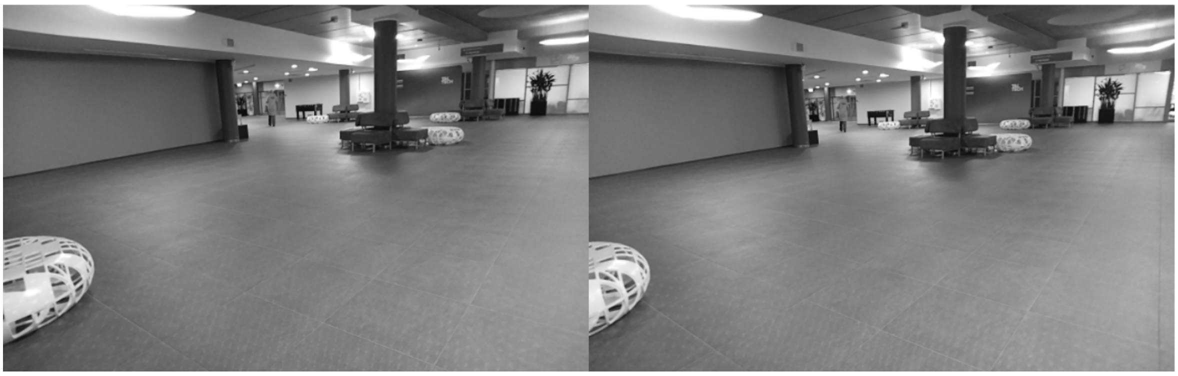

Figure 3.7 Stereo images before rectification

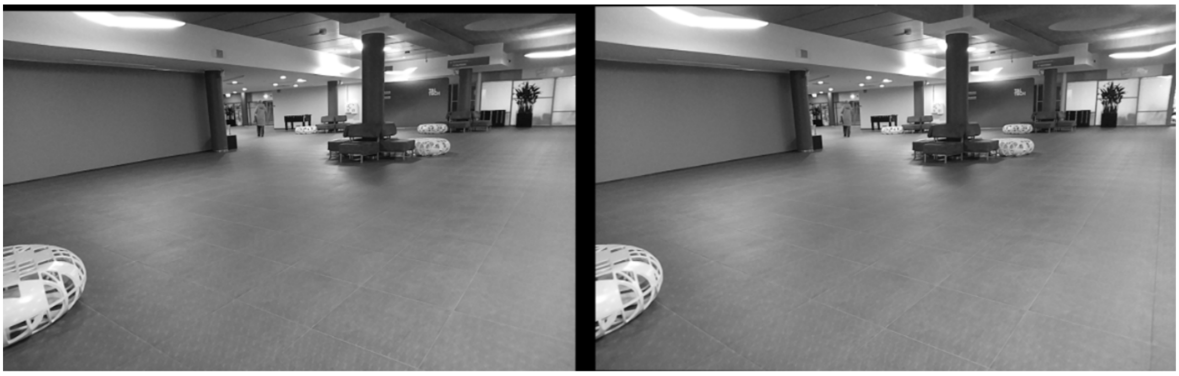

Figure 3.8 Stereo image after rectification

As we can see from the rectification process the size of the image after rectification is different (small of rotated) from before the rectification. In our case the KITTI dataset provides undistorted rectified images, in which case we would bypass the preprocessing node. The OAK camera system used in our own setup also outputs rectified streams.

3.2.2 Feature detection

As discussed in Chapter 2 for the feature-based approach salient features must be extracted from the frame. There are several algorithms for this purpose. Geometric based algorithm such as SIFT, SURF, ORB, AKAZE, FAST, GFTT etc., machine learning based algorithm such as Super point [33] and LIFT are available. For the proposed method, we will use the geometric based algorithm for feature detection. The reason for this choice is as follows:

1) All geometric based approaches are computationally more efficient than their machine learning counter parts given a resource constrained device without any additional hardware such as accelerators or GPU.

2) Same results under same condition or repeatability is an advantage from the use of geometric based detectors. This is because geometric based detector works based on mathematical and statistical principals. This is an advantage that allow us to test our system with different settings and setup while making sure other factor remain the same allowing us to fine tune our system and possibly explain the results.

3) Ease of implementation and integration into the system.

Given the vast number of choices available in geometric based algorithm, we propose to proceed forward with using SIFT as the detector for the visual odometry method.

SIFT is chosen over SURF due to restriction on the usage of SURF for commercial purpose as it is still patented and therefore the unavailability of this detector in an open-source implementation such as OpenCV without any additional work though which is not the recommended approach. SIFT is better than SURF when it comes to scale, however SURF is better in both rotation invariance and blur invariance. Also, it is estimated that SURF is three times faster than SIFT [35]. It’s understood that alternatives such as ORB and FAST demonstrate speed which is unmatched by SIFT. However, from the analysis performed by [36] we find that SIFT is the most accurate algorithm and the author suggest its application in scenarios where there are less points which are features is fundamental for a feature-based approach for visual odometry implementation, as feature-based method suffers in situation with low features (low texture). The detector extracts the features and description of the feature known as descriptors. The feature is extracted only in specified region. A mask with the region to be extracted is computed for every frame based on the object detection node.

Dynamic object in the scene can also cause a correctly tracked feature point to be an erroneous feature point for the visual odometry method. This is because with dynamic objects, when the object is moving slower or faster than the agent, the tracked feature point will have different location in the tracked image which is not a result from the motion of the agent only, but also from the motion of the dynamic object in the scene. Results on the effect of dynamic filtering will be discussed in the next Chapter. Filtering of matches based on dynamic objects in the scene that are considered capable of motion is necessary. As explained earlier it is necessary to remove matches that may have travelled not only with respect to our motion but also because of the motion of the object on which the feature is tracked. A good example would be a car front of the agent and the road on which the agent is travelling. Features from the road that are tracked across the frame display translation and rotation in the next frame purely based on the motion of the agent and most probably not based on any external factors (an exception would be a stone on which a feature was matched but got hurled around by other factors). On the other hand, a feature on a car that was matched travelling in front of the agent, would have a rotational and translational motion in the next frame based on the motion of the agent and the motion of the other car combined. This situation can have a profound affect if not filtered out. A basic filtering technique would be to detect objects that could move significantly in a scene such as car, bus, trolleys, pedestrians, and bicycles and then excluding them, meaning all matches from those objects are not considered. For our system we use a yolov4 tiny [37] model to detect these objects. The reason we use an outdated model while significantly better version of the model such as YOLOv8 or YOLOv9 exist is because of its lightweight requirement and ease of integration. A mask is formed based on the bounding box output from the model on the classes predicted. Which is then used by the detector to perform matching only in unmasked region of the frame.

3.2.3 Feature matching and filtering

The next stage is to establish correspondence between the images from which the features were extracted, through an iterative step of matching and filtering, until enough key points are extracted. Feature matching can be done using algorithms such as BF (brute force) or FLANN based matcher. There also exist machine learning based matcher such as Superglue [38]. For our proposed method we would use the BF based matcher. The Brute force matcher works by comparing every descriptor from image A to descriptors from Image B to find the best match. The match is found by calculating Euclidian distance in this case (Hamming distance is computed instead when binary descriptors are used). This is a computationally expensive task which only further increases as the number of features from the detection increases. Additionally, we employ KNN based matching coupled to BF. This outputs two closes match that was found during matching for each salient feature matched.

Due to repetitive patterns, occlusion, or noise in the image not all matches by the matcher will be correct. These incorrect matches can be considered as outliers and have to be filtered out. Filtering the matches using Lowe's ratio test is the first filtering stage implemented. Lowe's ratio test was proposed by [39]. The idea is to compare the distance between the closest match and the second closet match. The Lowe's ratio test is based on the assumption that the closest match is the best match than the next closest match which could be noise. By taking the ratio of the distance of the first match to the second match and comparing it to a threshold we can filter the match as a good match or an incorrect match. In our system this test is implemented dynamically. Our system starts off from the lowest threshold set and increments the threshold by 0.05 for every iteration the motion estimate fails or there are less matched points than the required. We have found empirically that starting with a ratio of 0.4 has yielded better results.

Estimating the homography from the filtered matches allows us to further filter out the matches by considering only those features which are on the same plane in both images. Homography based filtering is effective from a theoretical point of view in situation where there is only purely rotation motion. However, it's important to note that, implementation of this filtering technique is also conditioned on the ability to estimate the motion. If the motion is unable to be estimated, then the filtering method is neglected. RANSAC is used to compute the homography. The working knowledge of RANSAC is important as it is also used in the motion estimation node of our system.

RANSAC stands for Random Sample Consensus. The assumption on which RANSAC is predicated is that an inlier in an observed set of data can be explained by a model and outliers are those points that does not fit the model. In order to better understand its working, consider this simple example where there is a set of data. Most number of points in this data lie along a line, there are also points in the data that randomly lie around the line which in this case are the outliers. RANSAC must estimate the best line that fits the majority of this points. How RANSAC does this is, RANSAC initially selects a subset of the data, in this case only two points are needed to calculate the line based on the equation of the line. RANSAC then uses this line and calculates how many points fits these lines or are close to this line based on a threshold. The steps are repeated a fixed number of iterations, after which the model which fits the highest number of points in the data is considered the best estimation. We can already understand that the quality of the data is going to play a crucial role in the determination of the model. A greater number of inlier than outlier points will yield a better model than a set containing the opposite. In our application it is crucial that the Lowe's ratio test yield the best filtered matches and additional outliers in this case are removed.

3.2.4 Stereo – depth map generation

There are three processes in this node. Disparity map generation, depth map formation and depth map filtering. We discussed in subsection 3.2.1, during preprocessing we rectify the images. Rectification of the images are necessary in a stereo configuration when the result needed is to estimate the depth of the scene. Estimating the depth of the scene is what gives the stereo visual odometry its advantage over monocular visual odometry. By rectifying the images, we ensure that features in both images lies in the same horizontal plane.

Disparity map is a representation in visual form of the difference in the coordinates of the same feature in two rectified images. In order to calculate the disparity, the algorithms for each pixel in the left image, searches a match in the same horizontal row in the right image within a window. The shift required for this match to occur is known as disparity. All the calculated disparities are compiled into an image called disparity map.

Depth from disparity is calculated using the following equation:

$$depth = f_x * \frac{baseline}{disparity} \tag{3.1}$$

Here the units of the depth and baseline are in millimeters and the unit of the focal length at x and disparity is in pixels. From this equation it can be deduced that the disparity and depth are inversely related. This means changes in the disparity value closer to zero changes the depth by a large margin. And the opposite also holds true. Also, we can deduce that if either of the baseline or the focal length (distance between the camera lens and the image sensor) is increased then it will result in increased depth at same disparity. Which suggest that the depth accuracy is dependent on certain physical properties of the camera and setup. The depth map is then calculated, which essentially is a map that contains an associated depth to each pixel.

No matter the best approach in matching there are some presences of outliers or inconsistency in the matches leading to a noisy depth. In order to rectify this issue as much as possible, our visual odometry system incorporates a median filter (empirical experiments found a kernel size of 3 being the best for the given setup). This filter acts to smoothen out noises and irregularities. Also, it's important to note that in scenarios without enough texture on the surface the depth map computed is not accurate. An example of this would be the sky. Or a large uniform wall.

3.2.5 Motion estimation

Iterative steps of action solve for the estimation of the camera pose. In our visual odometry method our main estimation algorithm is 2D-to-3D based approach. For this approach the following is required:

1) 3D coordinates of the feature extracted and filtered from the previous frame in the system. Since depth for every pixel is computed, by indexing the pixel location of the feature points on the depth map we can extract the Z coordinate from the depth map. The X and Y coordinates in the 3D world is computed using the known intrinsic parameter of the camera. The formula for the computation of X or Y in 3D world coordinate system is:

$$X = \frac{u * c_x}{f_x} * depth \tag{3.2}$$

$$Y = \frac{v * c_y}{f_y} * depth \tag{3.3}$$

   Here u and v are the image coordinate in the image plane. c_x and c_y are the optical center and f_x and f_y are the focal lengths. All four of these parameters are obtained from the camera's intrinsic parameters.

2) Next the 2D coordinate from the current frame is required. This are from the feature matches with the current frame.

3) We filter out the matches further based on two conditions. First being the scale of the change of the motion. By computing the change in location of the pixel in both the frame and comparing it to a threshold, with the condition that if the feature has a matched point on the other image that is displaced more than the threshold then the point is invalidated. This threshold is increased for every

unsuccessful estimation, until there is a successful estimation. Along with this a second additional stage of filtering is done using an implementation of a depth search-based estimation. In this approach initially the motion is estimated using closest points only, failure would cause the depth search window to increase causing more points to be considered for the estimate. The depth at closer range is generally more accurate than at the very far. Closer to the agent the presence of texture on surfaces leads to a better estimate of depth.

4) Using the perspective n point algorithm coupled with RANSAC for robust estimate we compute the rotational and translational matrix that describes the motion of the agent.

The iterative approach in this node and the overall system is expected to have a better and robust behavior towards incorrect matching and in scenarios where there are few matches like for example in highways or indoor environments.

3.2.6 Fusion setup

Our filter will estimate the vehicle state which consist of the position and the velocity. Each of the vehicle state and the velocity state are 3 dimensional. There for our state estimate is represented by a 6-dimensional state vector. The equation are deduced from the sources [40] and [41].

$$x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix} \in R^6 \tag{3.4}$$

Here x_k is the state vector estimated. p_k and v_k are the position and velocity states respectively. The input to the motion model node (refer to Figure 2) of the system is a 3-dimensional vector u_k. f_k represents the linear acceleration estimation from the IMU.

$$u_k = [f_k] \in R^3 \tag{3.5}$$

The predicted state using the motion model is computed using the following equations which represents the motion.

$$p_k = p_{k-1} + \Delta t \,.\, v_{k-1} + \frac{\Delta t^2}{2}\,(f_{k-1} + g) \tag{3.6}$$

$$v_k = v_{k-1} + \Delta t \,.\, (f_{k-1} + g) \tag{3.7}$$

Both motion model equation is modelled based on the general model of motion. Here p_{k-1} and v_{k-1} represent the previous position and velocity state respectively. Δt represent the time between the state estimates. g is the gravitational constant.

The error state is modeled as:

$$\delta x_k = \begin{bmatrix} \delta p_k \\ \delta v_k \end{bmatrix} \in R^6 \tag{3.8}$$

And the error dynamics is computed by:

$$\delta x_k = F_{k-1}\delta x_{k-1} + L_{k-1} n_{k-1} \tag{3.9}$$

Where:

$$F_{k-1} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & -f_{k-1} * \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

$$L_{k-1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.11}$$

In the above equation 1 stands a 3 x 3 identity matrix.

Prediction step is affected by noise:

$$n_k \sim \mathcal{N}(0, Q_k) \tag{3.12}$$

$$Q_k = \Delta t^2 \begin{pmatrix} \sigma^2_{accel} & 0 \\ 0 & 0 \end{pmatrix} \tag{3.13}$$

The observation is affected by noise that is modelled by:

$$o_k \sim \mathcal{N}(0, R_k) \tag{3.14}$$

$$R_k \ = \ \Delta t^2 \begin{pmatrix} \sigma_{cam}^2 & 0 \\ 0 & \sigma_{cam}^2 \end{pmatrix} \tag{3.15}$$

Based on this, a continuous loop-based architecture is given by the following:

1) Update the state with the acceleration estimates from the IMU using the motion model.

2) Propagate uncertainty using:

$$\check{P}_k \ = \ F_{k-1} P_{k-1} F_{k-1}^T \ + \ L_{k-1} Q_{k-1} L_{k-1}^T \tag{3.16}$$

3) If visual odometry or GPS measurement is available

    a. Kalman gain is computed:

$$K_k \ = \ \check{P}_k H_k^T \ (H_k \ \check{P}_k \ H_k^T \ + \ R)^{-1} \tag{3.17}$$

    b. Estimating the error state:

$$\delta x_k = \ K_k \ (y_k \ - \ \check{p}_k), where\ y_k\ is\ the\ observed\ position. \tag{3.18}$$

    c. Correct the predicted state using the error state:

$$\hat{p}_k \ = \ \check{p}_k \ + \ \delta p_k \tag{3.19}$$

$$\hat{v}_k \ = \ \check{v}_k \ + \ \delta v_k \tag{3.20}$$

3.2.7 Sensor transformation

In order to fuse data from various sensor its necessary to ensure that all data from sensor is represented in a single global coordinate frame. Data in a different coordinate frame is different from the same data in another coordinate frame. An example of this is the orientation of the IMU and camera from the car used in KITTI as seen on Figure 3.3. In order to map for example a point from the camera frame to the IMU coordinate frame we need to convert each data point in the camera coordinate frame to the IMU coordinate frame using a rigid body transformation matrix.

A transformation matrix is a 3 by 4 matrix which comprises of a 3 by 3 rotation matrix and a 3 by 1 translation matrix. The rotation matrix describes how a point in a coordinate system changes from a coordinate system A to coordinate system B. The translational matrix represents the change in the location of this point in the coordinate system after the transformation. The transformation matrix can be compared to a mapping matrix. Transformation matrices display the following property, if there are three coordinate systems A, B and C and there are two transformations matric T_{AB} and T_{AC}which describes the change of coordinate system from A to B and B to C respectively. The change in coordinate system from A to C is given by:

$$T_{AC} \; = \; T_{AB} \; x \; T_{AC} \qquad (3.21)$$

Transformation matrix also adhere to the property which displays the following behavior:

$$T_{AC} \; = \; [T_{CA}]^{-1} \qquad (3.22)$$

The inverse of a transformation matrix is valid. And would undo the transformation that was done by the transformation.

For the KITTI dataset we have chosen to convert all data points that are estimated to the IMU coordinate frame. In this case we selected the IMU coordinate frame as the global coordinate frame.

Transformation from IMU coordinate system to the velodyne (LIDAR) coordinate system T_{IV} for the KITTI dataset is:

$$T_{IV} \ = \begin{bmatrix} 1 & 0 & 0 & -0.81 \\ 0 & 1 & 0 & 0.32 \\ 0 & 0 & 1 & -0.79 \end{bmatrix} \tag{3.23}$$

We can see from the above matrix that there is no rotation associated. And this is evident from Figure 3.3, as both the velodyne and IMU coordinate frame are oriented in the same way.

Transformation matrix from the velodyne coordinate system to the camera (CAM 0) coordinate system is given as T_{VC_0}:

$$T_{VC\_0} \ = \begin{bmatrix} 0 & -1 & 0 & -0.007 \\ 0 & 0 & -1 & -0.074 \\ 1 & 0 & 0 & -0.334 \end{bmatrix} \tag{3.24}$$

Since there is a change in the axis orientation between the velodyne and the camera the rotation matrix is affected unlike the equation in 3.23.

So based on the property of the transformation matrix discussed above, the transformation matrix from IMU to CAM 0 is given by T_{IC_0}:

$$T_{IC\_0} \ = \ T_{IV} \ x \ T_{VC\_0} \tag{3.25}$$

And based on the second property of inverse we can obtain the transformation matrix to convert from camera coordinate system to IMU coordinate system by taking the invers of T_{IC_0} which gives us T_{C_0I}:

$$T_{C\_0I} = \begin{bmatrix} 0 & 0 & 1 & 1.14 \\ -1 & 0 & 0 & -0.81 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{3.26}$$

Our system considers CAM 0 as the left camera, features are tracked on this cam. And consecutively the pose is also estimated with respect to this camera coordinate system.

Overall logic of the entire system for both the visual odometry system and fusion system is represented on Figure 3.9 and Figure 3.10 respectively.

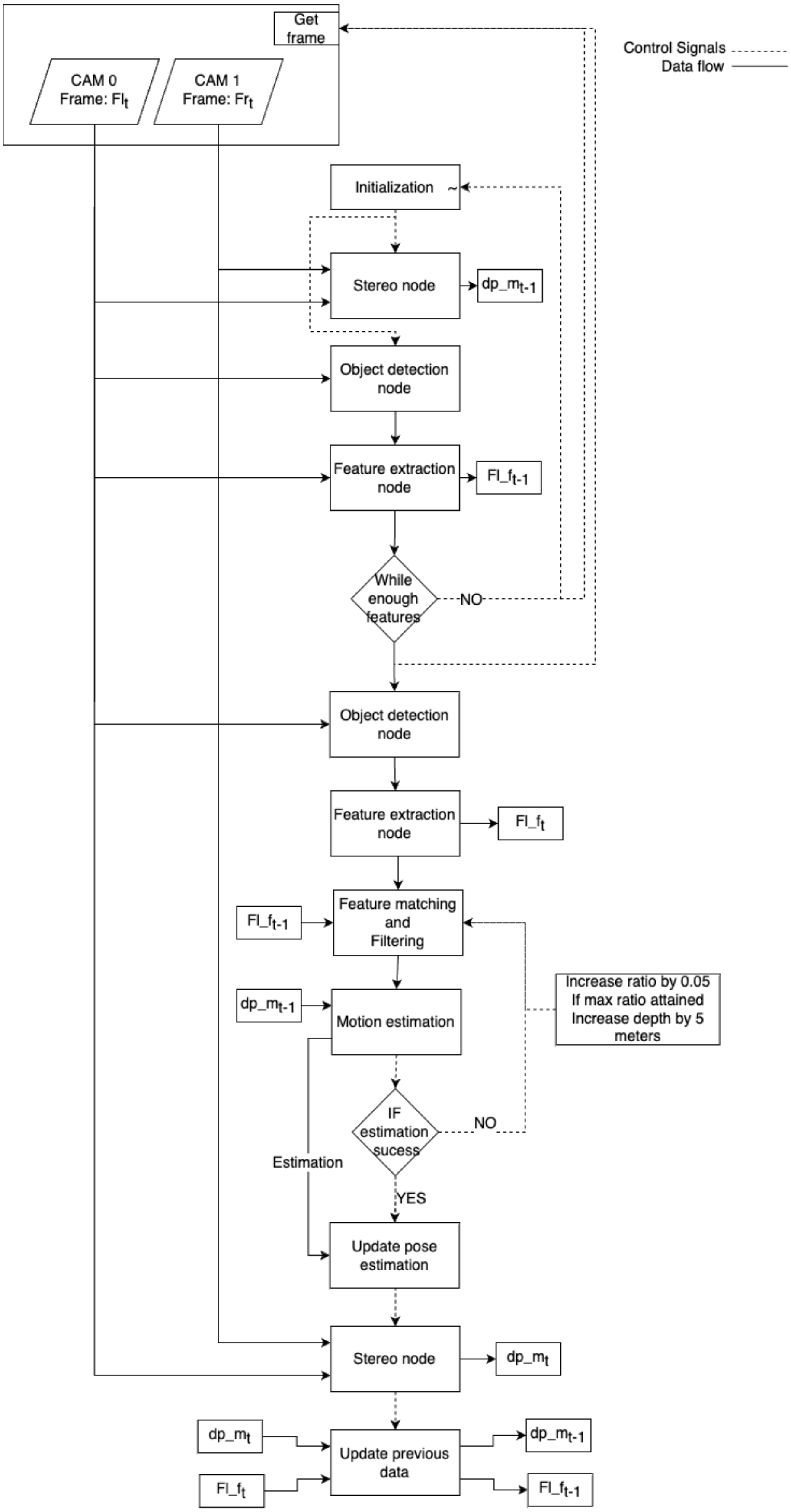

Figure 3.9 Pipeline of proposed visual odometry method

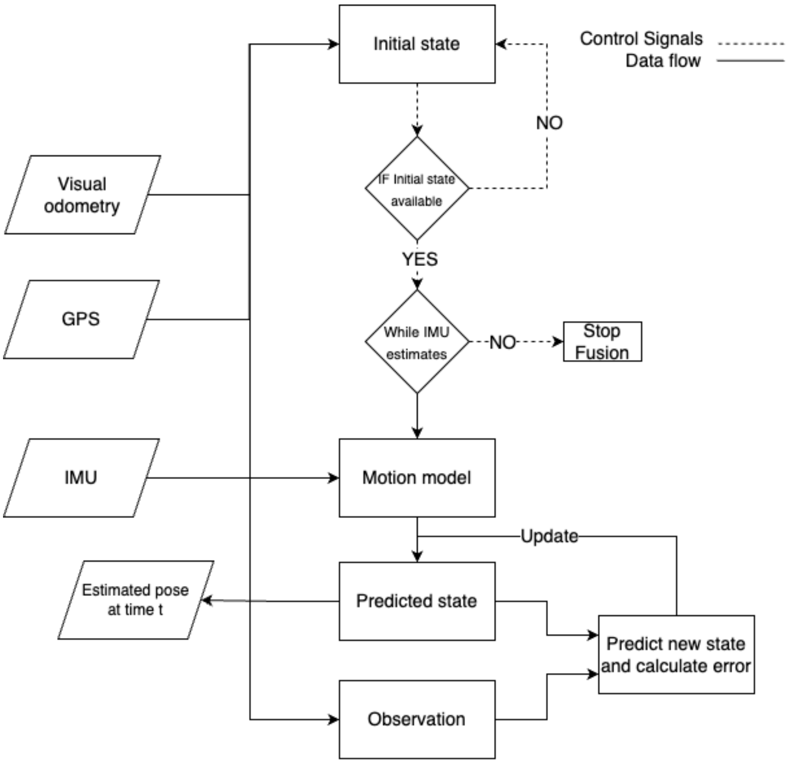

Figure 3.10 Proposed pipeline of the fusion system

3.3 Experiment setup

The dataset that we would initially use to test our proposed system is the KITTI dataset as mentioned earlier. There are totally 11 sequences in the odometry benchmarking subset of the KITTI dataset. This dataset contains ground truth information in the format of pose estimation. This will allow us to compare our system output directly to the provided ground truth without any additional computation or conversions.

Out of the 11-sequence provided, we will use 4 sequences for our experiments. These are Sequence 01, 03, 04 and 09. As we have described in the introduction of this thesis that the aim of this thesis is to have a system that is able to track its location in a GNSS denied area for up to 400 meters. Three of the sequence selected has a traversed path that is longer than 400 meters. The choice of the sequence is for the following reasons:

1) The selection of sequence represents different environments. For example, sequence 01 is from a highway, 04 is on the main street on a city. Both 03 and 09 are urban environments. The selection range encompasses highway scenario for low texture scene, urban areas have higher texture therefore the method is expected to demonstrate higher accuracy as depth and features are widely

available and are more accurate. The city area represents a combination of both these environments.

2) Sequence 01 is considered a very difficult sequence for state-of-the-art system such as ORB-SLAM 2 [2]. The authors also acknowledge that this sequence is a difficult sequence since a part of this sequence does not contain trackable features that are close to the agent and acknowledged most method fail in this sequence. Especially being a low feature scenario and our system being based on feature-based approach, we will demonstrate the performance of our method in this scenario.

3) Sequence 04 is a straight traversed path in a city setting. This sequence is approximately 400 meter long.

4) Sequence 03 and 09 contains curves that would prove difficult for a visual odometry method. The difficult corners in these sequences would explain the robustness of the method.

5) Sequence 01 and 04 are in dynamic situations. With other dynamically moving objects in the scene. Our proposed method must be able to handle this situation and show improved performance compared to method without dynamic considerations.

Figure 3.11 and 3.12 shows the traversed path in two dimensions for sequence 01, 03, 04 and 09 respectively as given. By the KITTI odometry dataset. The plot is based on the coordinate system of the camera frame. Conversion of the same to a different coordinate system will yield in a different view.

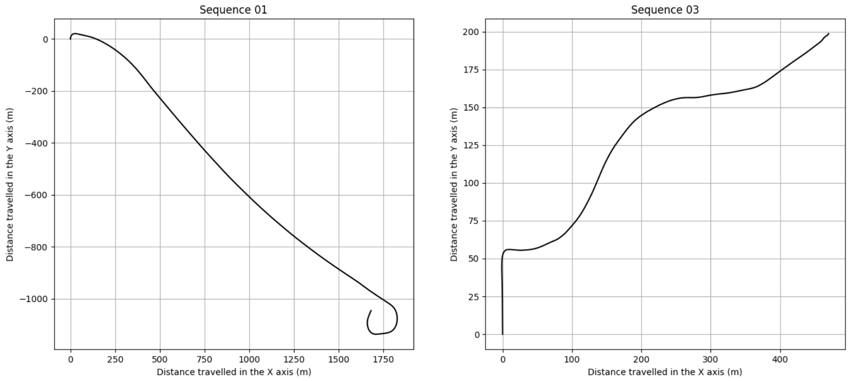

Figure 3.11 Left shows the sequence 01 and right is the trajectory of the sequence 03

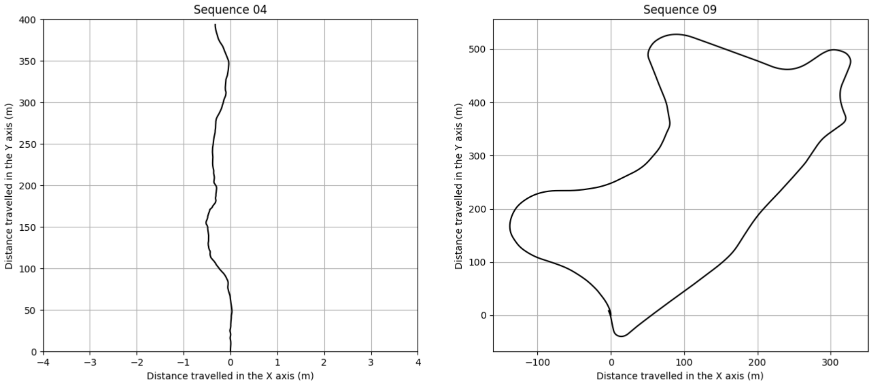

Figure 3.12 Left shows the sequence 04 and right is the trajectory of the sequence 09

The following experiments are planned to demonstrate and test the systems performance.

1) **Experiment 1**: Only visual odometry system without the fusion component.

   a. Computation on all sequences in stereo mode with object detection and depth filtering functionality. In essence the entire proposed system is tested. KITTI metrics will be computed in these settings and compared with other state of the art systems.

   b. Without object detection node in order to verify the effect of dynamic objects in the scene on the proposed system. Will be tested on sequence 01 and 03. As sequence 01 has vehicles which are dynamic and sequence 03 has vehicles which is static mostly.

   c. Without filtering of the depth map to verify the performance of the system with better quality depth estimation. Will be tested on sequence 01 and 03. Sequence 01 is a low texture scenario and sequence 03 being a high texture scenario.

   d. Performance of the system in monocular mode. Using 2D-to-2D motion estimation algorithm. Will be tested on sequence 03 and 04.

2) **Experiment 2**: Evaluation of the application of the fusion node

   a. Fusion of visual odometry estimates with IMU estimates and comparison with our results without fusion and ground truth.

b. Fusion of visual odometry estimates with IMU estimates in the scenario where 25%, 50% and 90% of the visual odometry estimate is randomly dropped. This should demonstrate the robustness of the system.

The fusion will be performed for the sequence 01, 04 and 09. The IMU raw data for sequence 03 is not provided in the KITTI dataset.

The next set of experiment described as **experiment 3** will be performed on the dataset that that we collected. This is the dataset collected using our setup as described in section 3.1.2. We managed to collect data from two scenarios. The first scenario is on hallway (as show in the Figure 3.13) that spans approximately 30 meters and the second one spans approximately 250 meters in an open parking space. In both cases there were not any measure of a ground truth. For the indoor scene in the hallway the setup was held by hand for the duration of the experiment as the person walked in a relatively straight line. In the other experiment at the parking lot, an initial plan was scaled and measured according to google maps and the plan was executed using visual cues (landmarks and road markings) to ensure that the collected data could be compared with some form of ground truth.

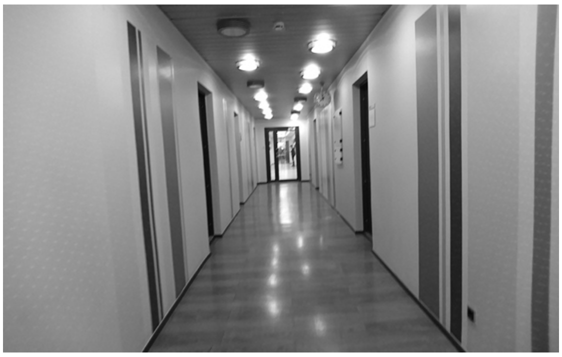

Figure 3.13 Hallway in which the data was collected

The inexistence of a real ground truth does not allow us to have a fair comparison. Nevertheless, the experiment would show the applicability of the system in a completely different environment. It would even describe the system's ability when different hardware is used other than the one tailored for the system like the KITTI dataset. Figure 3.14 shows the google map outline of the path that was traversed for the data collected in the parking space.

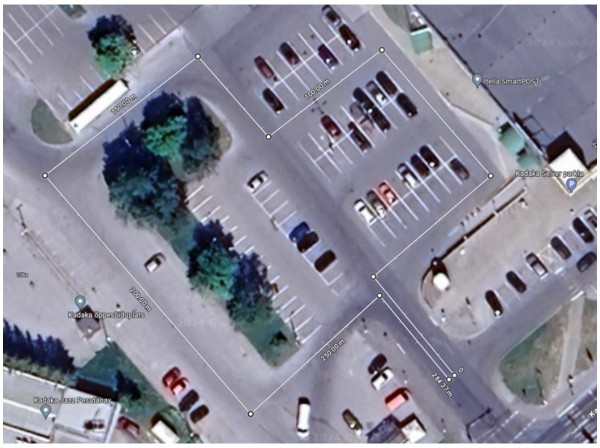

Figure 3.14 Traversed path outlined in google maps for the experiment in the outdoor parking space.

4 RESULTS

The proposed system was implemented on Python. All experiments were carried out on a MacBook pro M1 pro with 16GB of RAM. We acknowledge that such critical system such as visual odometry if based on a language like C++ would have faster execution time. In our case Python allows for easier integration with the existing system and faster development. A test and debugging tool is implemented on the system, that allowed for easier representation of all the major process of the system at a glance. Figure 4.1 shows an output of this tool. On Figure 4.1 camera view and the object detection is shown on the top left part of the image. The top right image displays the initial matching. The middle-left image displays the filtered matches followed by bottom left which displays the depth information color coded. The bottom left part of the image depicts the trajectory. Black represents the ground truth and blue represents the estimated trajectory in real-time.

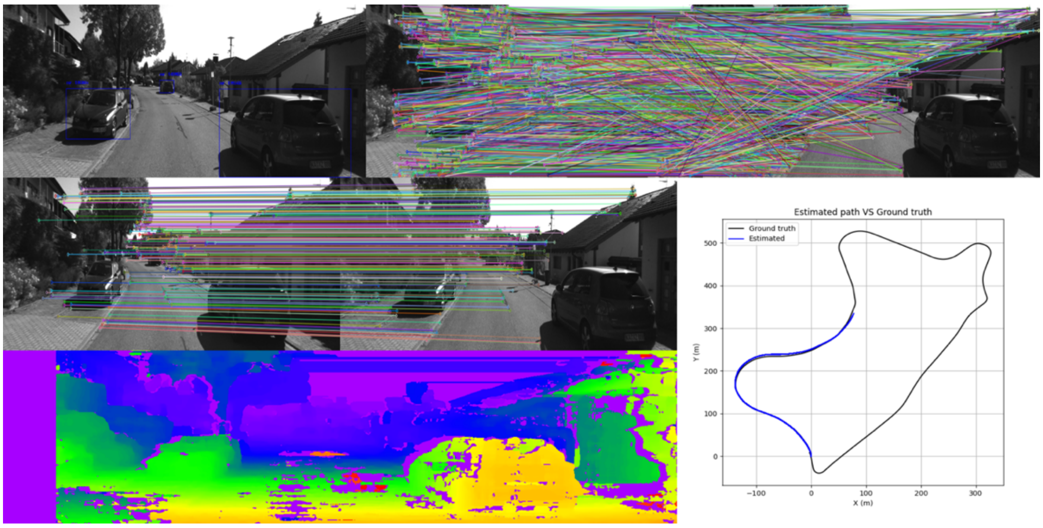

Figure 4.1 Output of the developed tool to debug the proposed method

4.1 Performance of the visual odometry method

Experiment 1.a: The proposed visual odometry method in its entirety (with all nodes and filters) was used to observe the performance of the method. As discussed in the Chapter 2 we would use the tool proposed by [2] in order to calculate the error metrics. The data to compare other methods is also obtained from the works of [2]. The authors in this work have meticulously collected data from some of the other state of the art methods through employment of said methods on sequences from the KITTI odometry dataset. Our results for all four sequence 01, 03, 04 and 09 is shown below in the Figure 4.2. The ground truth trajectory is represented as “GT” in black and the trajectory estimate from our method is given in green color on the Figure 4.2.

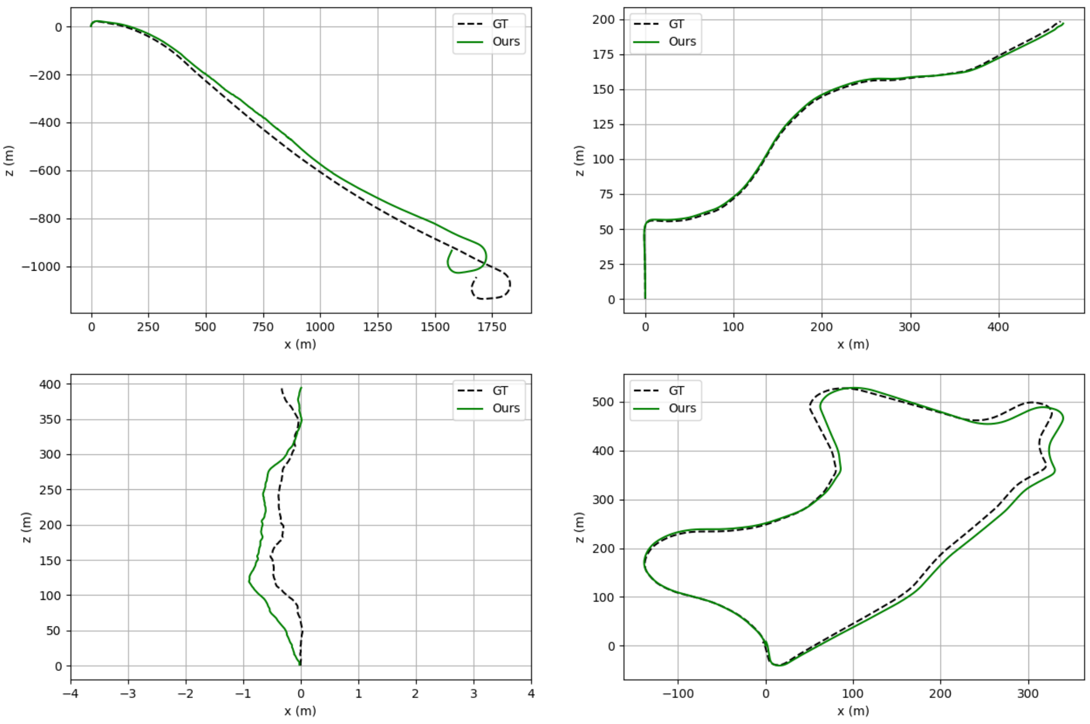

Figure 4.2 Top left shows the sequence 01, top right: sequence 03. Bottom left: sequence 04 and bottom right: sequence 09.

The results from the sequence are compared with other state of the art systems results [2] in the following Tables 4.1, 4.2, 4.3 and 4.4. The Tables compares translational error in %, Rotational error in deg/100m, ATE and RPE in meters.

Table 4.1 Error metrics of different state of the art [2] and our result on sequence 01

Method	**Error Metric**			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
SfM-Learner	22.41	2.79	109.61	0.660
Depth-VO-Feat	23.78	1.75	203.44	0.547
SC-SfM-Learner	27.09	1.31	**85.90**	0.888
ORB-SLAM2 no LC	107.57	0.89	502.20	2.970
ORB-SLAM2 LC	109.10	**0.45**	508.34	3.042
VISO2	61.36	7.68	494.60	1.413
DF-VO stereo	40.02	0.47	342.71	0.854
Ours (delta)	**8.77** (0)	1.54 (+1.09)	178.26 (+92.36)	**0.264** (0)

For sequence 01 (Table 4.1) our method outperforms in terms of translational error in percentage and RPE in meters compared to all other state of the art compared with. In terms of translational error our method demonstrates an error percentage of 8.77 percentage. 15.01 percentage lower than Depth-VO Feat. ORB-SLAM 2 based method with and without loop closure demonstrates an error percentage 13 times more than our proposed method. ATE metric however demonstrates that our system is outperformed by both SfM-Learner and SC-SfM-Learner. SC-SfM-Learner demonstrates an ATE in meters 92.36 less than our proposed method.

Table 4.2 Error metrics of different state of the art [2] and our result on sequence 03

Method	**Error Metric**			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
SfM-Learner	12.56	4.52	8.42	0.077
Depth-VO-Feat	15.76	10.62	21.34	0.168
SC-SfMLearner	9.22	4.93	10.21	0.059
ORB-SLAM2 no LC	0.97	**0.19**	**0.94**	0.031
ORB-SLAM2 LC	**0.91**	**0.19**	1.02	0.038
VISO2	30.21	2.21	52.36	0.226
DF-VO stereo	2.22	0.30	1.96	0.021
Ours (delta)	1.52 (+0.61)	0.77 (+0.58)	3.30 (+2.36)	**0.016** (0)

For sequence 03 (Table 4.2) our proposed method outperforms in the metric RPE compared to all other state of the arts. ORB-SLAM 2 based method demonstrates lower metrics in terms of translational error, rotational error and ATE in meter. The difference in translational error in percentage being 0.61 compared to ORB-SLAM 2 with loop closure. However, our method demonstrates a translational error in percentage of 0.7 compared to DF-VO stereo. For ATE both ORB-SLAM 2 and DF-VO demonstrate better performance than our method by 2.36 meters and 1.34 meters less than our method respectively.

Table 4.3 Error metrics for different state of the art [2] and our result on sequence 04

Method	**Error Metric**			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
SfM-Learner	4.32	3.28	3.10	0.125
Depth-VO-Feat	3.14	2.02	3.12	0.095
SC-SfMLearner	4.22	2.01	2.97	0.073
ORB-SLAM2 no LC	1.30	0.27	1.30	0.078
ORB-SLAM2 LC	1.56	0.27	1.57	0.081
VISO2	34.05	1.78	38.33	0.496
DF-VO stereo	**0.74**	**0.25**	**0.70**	0.026
Ours (delta)	0.95 (+0.21)	0.87 (+0.62)	3.20 (+2.50)	**0.019** (0)

Table 4.4 Error metrics for different state of the art [2] and our result on sequence 09

Method	**Error Metric**			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
SfM-Learner	11.32	4.07	26.93	0.103
Depth-VO-Feat	11.86	3.60	52.12	0.164
SC-SfMLearner	7.64	2.19	15.02	0.095
ORB-SLAM2 no LC	9.30	0.26	38.77	0.128
ORB-SLAM2 LC	2.88	0.25	8.39	0.343
VISO2	18.06	1.25	52.62	0.284
DF-VO stereo	2.07	**0.23**	**7.59**	0.044
Ours (delta)	**1.98** (0)	0.79 (+0.56)	13.76 (+6.17)	**0.019** (0)

For sequence 04 (Table 4.3) and 09 (Table 4.4) our method demonstrates superior local performance given by RPE in meter. Given the global metric ATE our method demonstrate 2.50 meter more than the DF-VO method for sequence 04 and 6.17 meter more than the DF-VO system for sequence 09. ORB-SLAM 2 with and without loop closure also outperforms our method for sequence 04 by an average of 1.76 meters in ATE. In terms of translational error in percentage Our method outperforms ORB-SLAM 2 methods for both sequences. For sequence 04 the translational error in percentage is 0.21% more than DF-VO stereo method. For sequence 09 our method outperforms all other state of the art in terms of translational error in percentage.

Experiment 1.b: In this experiment we will set all other nodes of the system function active except for the object detection node as explained in Section 3.2.2. to test the effect of the dynamic object on the visual odometry system. Key "objdet" on the Figure 4.3 represents object detection.

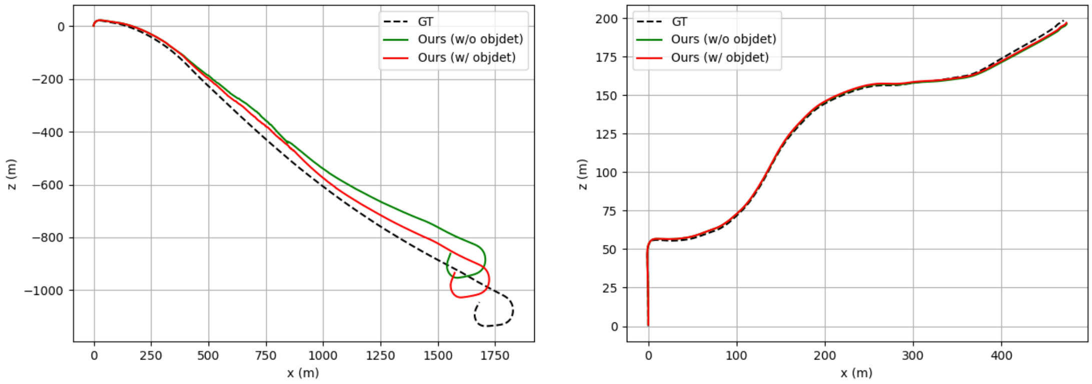

Figure 4.3 Left graph and right graph of ground truth and the proposed system on sequence 01 and 03 respectively.

Table 4.5 Result of with and without dynamic object detection for sequence 01

Method	Error Metric			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
Ours w/o object detection	11.99	2.21	201.36	0.324
Ours w/ object detection	**8.77**	**1.54**	**178.26**	**0.264**

Table 4.6 Result of with and without dynamic object detection for sequence 03

Method	Error Metric			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
Ours w/o object detection	**1.49**	0.77	3.33	0.017
Ours w/ object detection	1.52	0.77	**3.30**	**0.016**

For sequence 01 (Table 4.5) using object detection node to exclude dynamic object shows an improvement of 3 percentage in terms of translational error compared to results without object detection, and over 22 meter in terms of ATE. An improved metrics of RPE is also noted. Rotational error is also improved for sequence 01. For sequence 03 (Table 4.6) there is no improvement in terms of translational error in percentage. The improvement in terms of ATE and RPE in meter is 0.3 and 0.001 for this sequence. This is however negligible and doesn't show an increased performance for this sequence.

Experiment 1.c: We will also perform evaluation on the effect of the depth filter, in our case the median filter that was used. Like the previous evaluation scenario all other nodes including the object detection node are active except for the filtering of depth in node as mentioned in Section 3.2.4. The result of this experiment is visualized by Figure 4.4.

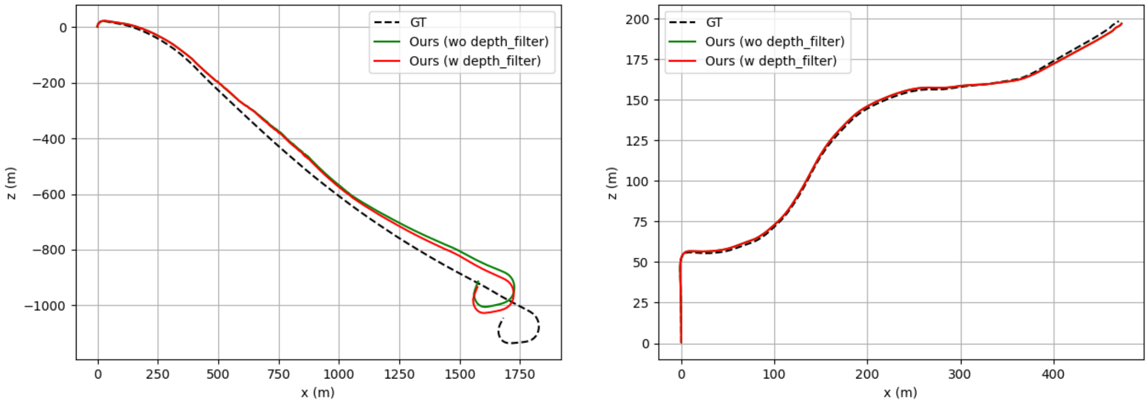

Figure 4.4 Left graph and right graph of ground truth and the proposed system on sequence 01 and 03 respectively.

Table 4.7 Result of with and without depth filter for sequence 01

Method	**Error Metric**			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
Ours w/o depth filter	9.34	1.59	186.76	0.272
Ours w/ depth filter	**8.77**	**1.54**	**178.26**	**0.264**

Table 4.8 Result of with and without depth filter for sequence 03

Method	**Error Metric**			
	Translational error %	**Rotational error (deg/100m)**	**ATE (m)**	**RPE (m)**
Ours w/o depth filter	1.53	**0.76**	3.31	0.016
Ours w/ depth filter	**1.52**	0.77	**3.30**	0.016

For both sequence 01 (Table 4.7) and 03 (Table 4.8), the depth filtering node has shown to improve performance in terms of translational error in percentage. For sequence 01

0.57 percentage for translational error is improved in comparison to metrics when depth filtering is not performed. ATE is also improved by 8.5 meter for sequence 01. Increase in terms of rotational error and RPE is also noted for sequence 01 by 0.05 and 0.008 respectively. For sequence 03 translational error in percentage and ATE in meter is improved by 0.01 percentage and 0.01 meters respectively. This increase however is not as significant as for the sequence 01.

Experiment 1.d: The performance of our system in monocular mode, that is using the 2D-to-2D estimation method is evaluated next. The result of thesis experiment is visualized by Figure 4.5.

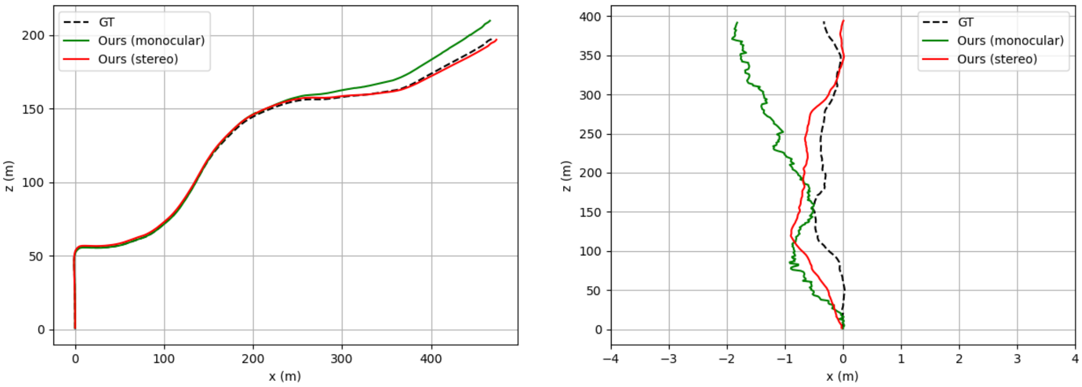

Figure 4.5 Left graph and right graph of ground truth and the proposed system on sequence 03 and 03 respectively.

Table 4.9 Results for monocular and stereo mode for sequence 03

Method	Error Metric			
	Translational error %	Rotational error (deg/100m)	ATE (m)	RPE (m)
Ours (monocular)	5.21	2.60	36.79	0.032
Ours (stereo)	**1.52**	**0.77**	**3.30**	**0.016**

Table 4.10 Results for monocular and stereo mode for sequence 04

Method	Error Metric			
	Translational error %	Rotational error (deg/100m)	ATE (m)	RPE (m)
Ours (monocular)	**0.93**	**0.80**	**1.66**	**0.032**
Ours (stereo)	0.95	0.87	3.20	0.019

For sequence 03 (Table 4.9) the stereo mode outperforms the monocular mode across all metrics. The significant difference being Translational error and ATE by 3.6 percentage and 33.46m respectively. For sequence 04 (Table 4.10) however the monocular mode demonstrates improved metrics, the difference being not as significant as for the sequence 03.

4.2 Performance of the fusion system

The coordinate system has an effect on ATE. The method of calculating ATE is the Euclidian distance between two observations in the same coordinate system. Since we have transformed our data from the camera coordinate system to the IMU coordinate system for the KITTI dataset, this will have an effect on the ATE computed. If the transformation is applied and the transformation is purely translational then the ATE is not affected. However, if the transformation also contains rotational then there is a change in the geometry of the data compared to the other coordinate system. Hence the computed ATE for the next set of experiment may be different and cannot be directly compared with the experiment 1. For these purposes ATE for all sequence available for this evaluation will be computed.

Experiment 2.a: fusion of visual odometry and IMU estimates in its entirety. The metric absolute trajectory error and relative pose error in meters is computed for estimates from the fusion system with respect to the ground truth. Additionally, the metric is also compared for visual odometry estimates with respect to the ground truth. ES-EKF represents the estimates form the fusion system and VO represents the estimate from the proposed method for visual odometry on Figure 4.6.

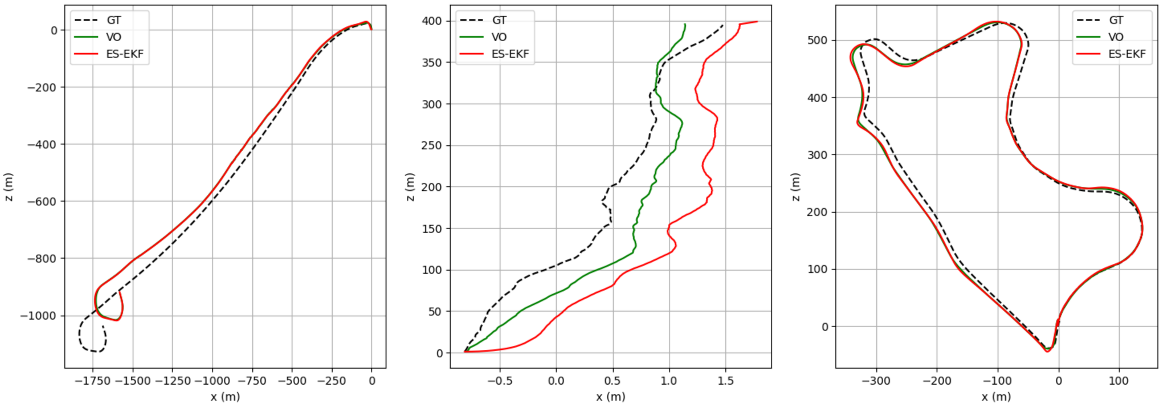

Figure 4.6 Estimate from the fusion system for sequence 01, 04 and 09 respectively (from left to right)

The Tables 4.11, 4.12 and 4.13 compares the results of ATE and RPE in meters for trajectory estimates from the fusion system with respect to the ground truth and the VO trajectory estimate with respect to the ground truth.

Table 4.11 Error metrics for comparison of fusion method on sequence 01

Trajectory	**Error metric**	
	ATE (m)	**RPE (m)**
ES-EKF w.r.t GT	154.792	0.279
VO w.r.t GT	154.756	0.224

Table 4.12 Error metrics for comparison of fusion method on sequence 04

Trajectory	**Error metric**	
	ATE (m)	**RPE (m)**
ES-EKF w.r.t GT	4.636	0.159
VO w.r.t GT	2.953	0.014

Table 4.13 Error metrics for comparison of fusion method on sequence 09

Trajectory	**Error metric**	
	ATE (m)	**RPE (m)**
ES-EKF w.r.t GT	15.606	0.068
VO w.r.t GT	15.323	0.013

ATE for visual odometry estimate with respect to the ground truth is 154.756, 2.953 and 15.323 meters for sequence 01 (Table 4.11), 04 (Table 4.12) and 09 (Table 4.13) respectively. With the fusion system we find a negligible increase in ATE and RPE of 0.036 and 0.055 meters respectively for the sequence 01. For sequence 04 we find that ATE and RPE increased by 1.683 and 0.145 meters. For sequence 09 just as for sequence 01 the ATE and RPE is shows a slight increase of 0.283 and 0.055 meters respectively.

Experiment 2.b: fusion of visual odometry and IMU estimates, with random dropping of visual odometry estimates.

For Sequence 01:

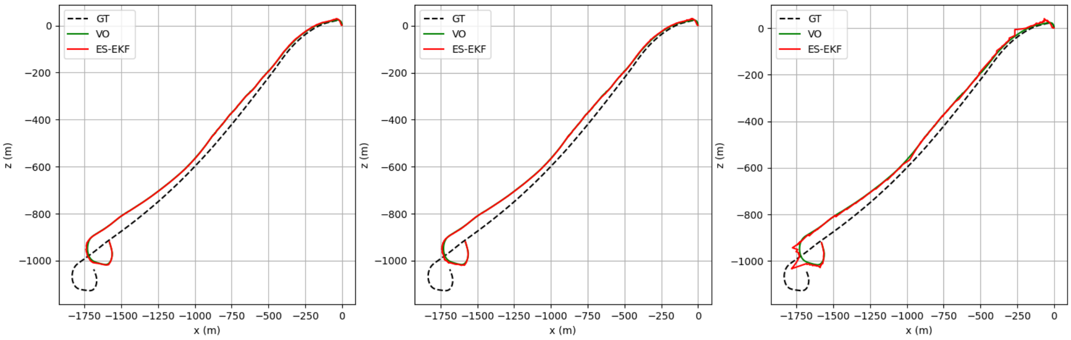

Figure 4.7 Sequence 01 at different dropout rate of 25%, 50% and 90% respectively (from left to right)

Table 4.14 Computed metrics for sequence 01 depicting scenarios of dropout

Dropout %	Trajectory	Error metric	
		ATE (m)	**RPE (m)**
No dropout	ES-EKF w.r.t VO	3.255	0.259
	VO w.r.t GT	154.756	0.224
25	ES-EKF w.r.t VO	3.721	0.315
	ES-EKF w.r.t GT	154.702	0.334
50	ES-EKF w.r.t VO	4.534	0.427
	ES-EKF w.r.t GT	154.518	0.394
90	ES-EKF w.r.t VO	17.290	0.912
	ES-EKF w.r.t GT	153.925	0.808

For sequence 01 (Table 4.14) ATE and RPE for fusion system estimate with respect to visual odometry increase as the drop out percentage increases. Fusion estimate with respect to ground truth in the metric of ATE improves at each drop out level which is not consistent with the expected result which is a degradation of the result. However, RPE shows an increase in error for fusion estimates compared with the ground truth which is consistent with the expected result. The increase in ATE for fusion estimate with respect to visual odometry estimate from 25 percentage to 50 percentage is 0.813 percentage. The increase in ATE of fusion estimate with respect to visual odometry system of 50 percentage dropouts compared to a no dropout scenario is 1.279 meters. A dropout of 90 percentage is however the results that yields the most erroneous ATE and RPE compared to another dropout scenario.

For Sequence 04:

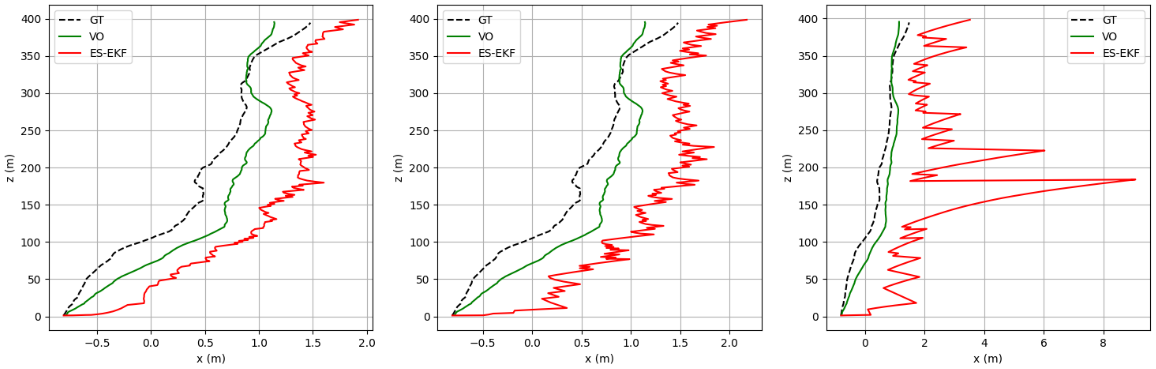

Figure 4.8 Sequence 04 at different dropout rate of 25%, 50% and 90% respectively (from left to right)

Table 4.15 Computed metrics for sequence 04 depicting scenarios of dropout

Dropout %	Trajectory	Error metric	
		ATE (m)	RPE (m)
No dropout	ES-EKF w.r.t VO	3.053	0.162
	VO w.r.t GT	2.953	0.014
25	ES-EKF w.r.t VO	3.297	0.177
	ES-EKF w.r.t GT	4.792	0.174
50	ES-EKF w.r.t VO	4.863	0.262
	ES-EKF w.r.t GT	5.938	0.258
90	ES-EKF w.r.t VO	8.549	0.416
	ES-EKF w.r.t GT	9.137	0.412

A similar comparison as for the sequence 01 performed on sequence 04 (Table 4.15) indicates the following results. Fusion estimate with respect to visual odometry system shows an increase in ATE and RPE all throughout the dropout scenarios. The increase in ATE from a dropout of 0 percentage to a dropout of 50 percentage is approximately 1.8 meters. Whereas the increase in ATE from a 25-percentage dropout to a 50-percentage dropout is 1.566 meters. Like for the sequence 01 the increase in error in terms of ATE at 90 percentage is approximately 2 times compared to 50 percentage. The relative pose error for all dropout scenario for fusion estimate with respect to visual odometry and ground truth shows an minor increase.

For Sequence 09:

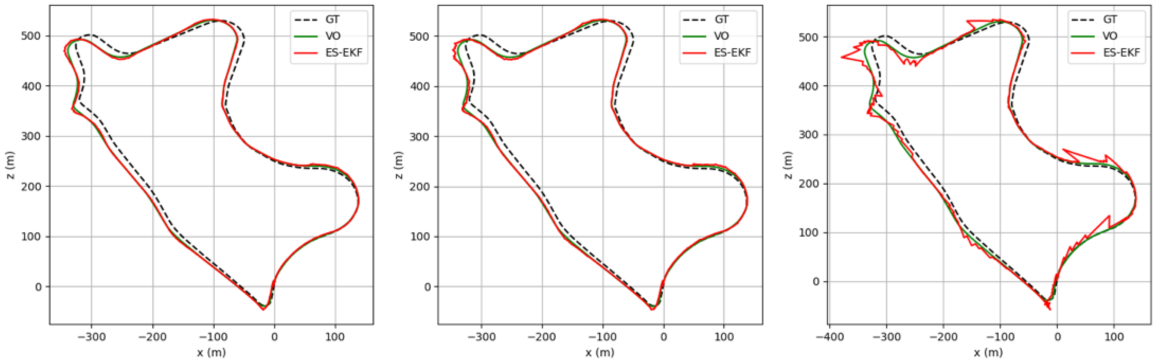

Figure 4.9 Sequence 09 at different dropout rate of 25%, 50% and 90% respectively (from left to right)

Table 4.16 Computed metrics for sequence 09 depicting scenarios of dropout

Dropout %	**Trajectory**	**Error metric**	
		ATE (m)	**RPE (m)**
No dropout	ES-EKF w.r.t VO	1.831	0.065
	VO w.r.t GT	15.323	0.013
25	ES-EKF w.r.t VO	2.222	0.116
	ES-EKF w.r.t GT	15.714	0.118
50	ES-EKF w.r.t VO	2.767	0.173
	ES-EKF w.r.t GT	15.874	0.175
90	ES-EKF w.r.t VO	9.140	0.552
	ES-EKF w.r.t GT	18.689	0.552

For sequence 09 (Table 4.16) we see similar result as for sequence 04. Fusion estimate with respect to visual odometry system shows an increase in ATE and RPE all throughout the dropout scenarios. The increase in ATE from a dropout of 0 percentage to a dropout of 50 percentage is approximately 0.936 meters. Whereas the increase in ATE from a 25-percentage dropout to a 50-percentage dropout is 0.54 meters. The increase in error in terms of ATE at 90% is approximately 3 times compared to 5%. The relative pose error for all dropout scenario for fusion estimate with respect to visual odometry and ground truth shows a minor increase.

4.3 Performance of the method on our data

Figure 4.10 depicts the **Experiment 3 scenario 1**, indoor hallway scenario. Figure 4.10 depicts the traversed path in black and the estimated path in red.

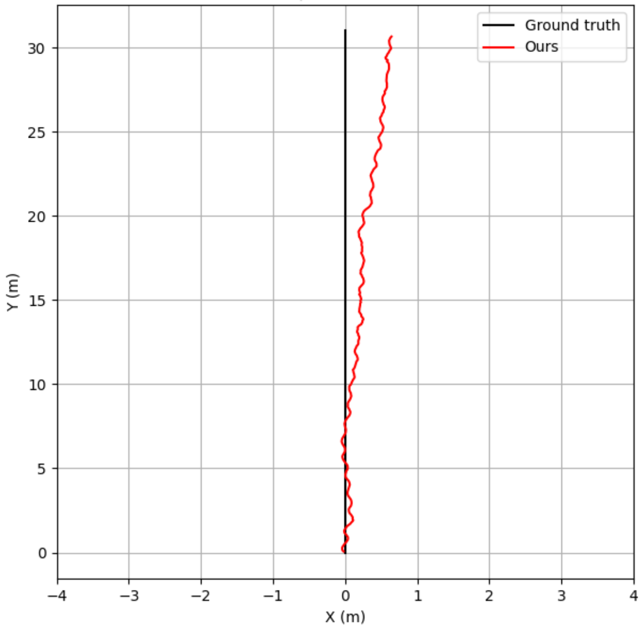

Figure 4.10 Ground truth in black and estimated trajectory in red for indoor hallway scenario

Table 4.17 Computed metric for the custom scenario

Trajectory	**Error Metrics**	
	ATE (m)	**RPE (m)**
Ours	1.51	0.08

Experiment 3 scenario 2 was not successful. The explanation to which will be discussed in the next Chapter.

5 ANALYSIS

The results presented in above section demonstrates our methods capability. To better understand it lets analyze the results.

5.1 Analysis of the visual odometry method

The results of **experiment 1.a**, represents the performance of the method. The results from this experiment are detailed in Table 4.1, 4.2, 4.3 and 4.4. Sequence 01 is considered as a difficult sequence, our method compared to the ORB-SLAM 2 method did not require multiple runs to estimate the trajectory [2] for this sequence. This sequence is hard because of the low texture environment the sequence was recorded in. For sequence 01 on Figure 4.2 we can understand the graphical representation of the result might not be acceptable even though the metrics are in favor of the estimates. Visual odometry is an incremental type estimator; an erroneous during the estimation at the beginning of the trajectory has resulted in the progression of the error throughout the next estimate demonstrating the inherent problem of visual odometry method. Even though the later estimate could be correct the addition to previous erroneous estimate has deviated the estimated trajectory.

While SC-SfM-Learner shows better performance in terms of ATE, SC-SfM-Learner is a machine learning based visual odometry method. The authors have mentioned that the pose network and single view depth estimation network used in this method have been trained on a part of the KITTI dataset [22]. Specifically, the authors mention that KITTI dataset sequence 00 to 08 are used for training of the pose network and 28 sequence from the raw KITTI dataset has been used to train the single view depth estimation network. Based on which we can safely highlight the fact that our method did not rely on any previous knowledge in its estimation like the SC-SfM-Learner or the SfM-Learner. Machine learning based visual odometry method when trained, given the inherent ability of deep learning models to understand and learn does not allow for a fair comparison in our case. As its clearly understood that sequence 01 was used in training of the method. If we compare our method against other known non machine learning based techniques such as ORB-SLAM 2 with/without loop closure and VISO2 then we find that our proposed method demonstrates clear advantage over these methods. ORB-SLAM 2 with loop closure which is a SLAM based techniques is even outperformed by our method.

For sequence 03 we find that our method outperforms other methods when the metric relative pose error in meters is compared. Whereas other methods such as ORB-SLAM 2 with loop closure and without loop closure demonstrates better performance on the metrics of translational error in %, rotational error % and ATE. DF-VO stereo a machine learning based visual odometry method also demonstrates its robustness in this sequence. Like the SC-SfM-Learner, DF-VO stereo method is also trained using several instances of KITTI dataset. Notably sequence 03 was used in training parts of the proposed method of DF-VO [2].

On sequence 04 and 09 our method demonstrates superior local performance measured by the metric RPE (m) than all other methods compared with. When comparing the ATE for sequence 04 between our method and DF-VO method we find that DF-VO method outperforms our method but given that the training sequence used to train this method contained the sequence 04, we can argue that our method has demonstrated its robustness with no prior knowledge. The performance of ORB-SLAM 2 based method has shown to be superior to our method in this sequence which highlights the ability of the method to be well suited for these sequences than our method. It is however important to note that ORB-SLAM 2 based method uses advanced concept such as keyframe insertion which could aid in a better estimation. For sequence 09 the DF-VO method shows similar performance to our method. This highlights the ability of the machine learning based method in this case DF-VO to perform well in situation with rich texture and changing environment's, demonstrating the advantages of machine learning based method, because this sequence was not used to train the DF-VO stereo method. However, our method outperforms the DF-VO stereo method in translational error, demonstrating that without even complicated approaches such as machine learning based method, geometric method can still perform relatively well.

If we now adhere to the goal set out for this method as described in the Chapter 1 introduction in which we specify the method must demonstrate robustness for a cumulative distance of 400 meters, analyzing the results then for ATE up until 400 meters gives us the following result. The red line highlights the threshold of 3 meters as the goal of ATE.

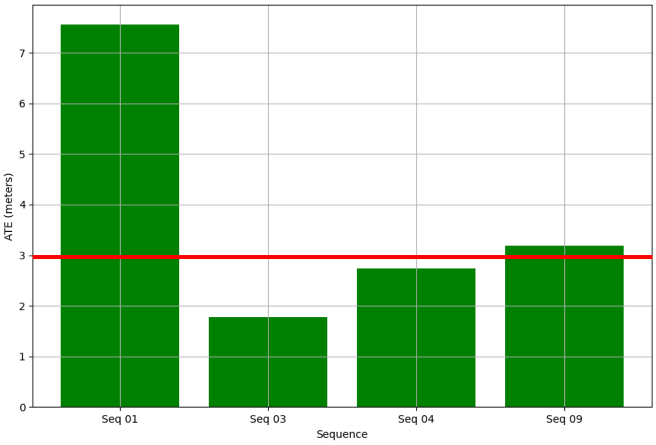

Figure 5.1 ATE for up to 400 meters for sequence 01, 03, 04 and 09 on our method

The definition of robustness of the method lies on its ability to approximate the trajectory with the lowest possible error. In this case the ATE. From Figure 5.1 we find that our method has demonstrated its robustness in sequence 03 followed by sequence 04, 09 and 01 respectively. Sequence 01 shows the highest ATE error of 7.5 meters. Followed by sequence 09 which shows an error of 3.1 meters and sequence 04 of 2.7 meters. Sequence 03 has an error of 1.7 meters. Our method can perform robustly in different environment's if the error threshold remains within the range of the above results. For example, GPS estimation of position in outdoor environment using off the shelf GPS enabled devices demonstrates accuracy within a 4.9 meters radius according to [42]. While high end devices can boost centimeter level precision, the high-end devices are expensive and does not guarantee its estimation precision in indoor areas or GPS denied environment's. By the standard of the goal set out which is an ATE of less than 3 meters we find that our method is well able to achieve these results for sequence 03 and 04. In sequence 09 we differ by only 0.1 meters for ATE compared to 3 meters ATE which is negligible. However, for sequence 01 we do not meet the requirement of ATE be less than 3 meters. Without the fusion method we can achieve the goals only using the visual odometry method.

In **experiment 1.b** we demonstrate the advantage of using dynamic object detection functionality of our method. With all metrics for sequence 01 shows an improvement on all metrics computed with the objected detection node. However, the improvement is not as significant for 03 as for 01. This is explained by understanding the sequence itself. While sequence 01 consisted of lot of moving cars sequence 03 on the other hand contained fewer moving cars. Our system in its current setup does not differentiate between moving and nonmoving dynamic objects in the scene. A stationary dynamic object in the scene is valuable to the system because of the additional feature points that can be extracted from the object. However, excluding all possible dynamic object does not show added benefits and warrants a different approach. Dynamic moving object as explained in section 3.2.2 induces error in our estimate. Figure 5.2 and 5.3 depicts the system with and without the object detection node for sequence 01 and 03 respectively. From which we can clearly understand that the amount of feature information present in an object such as a car and difference in their context of moving and not moving being significant. The red box highlights the features being identified and tracked/not tracked on the dynamic objects in the scene.

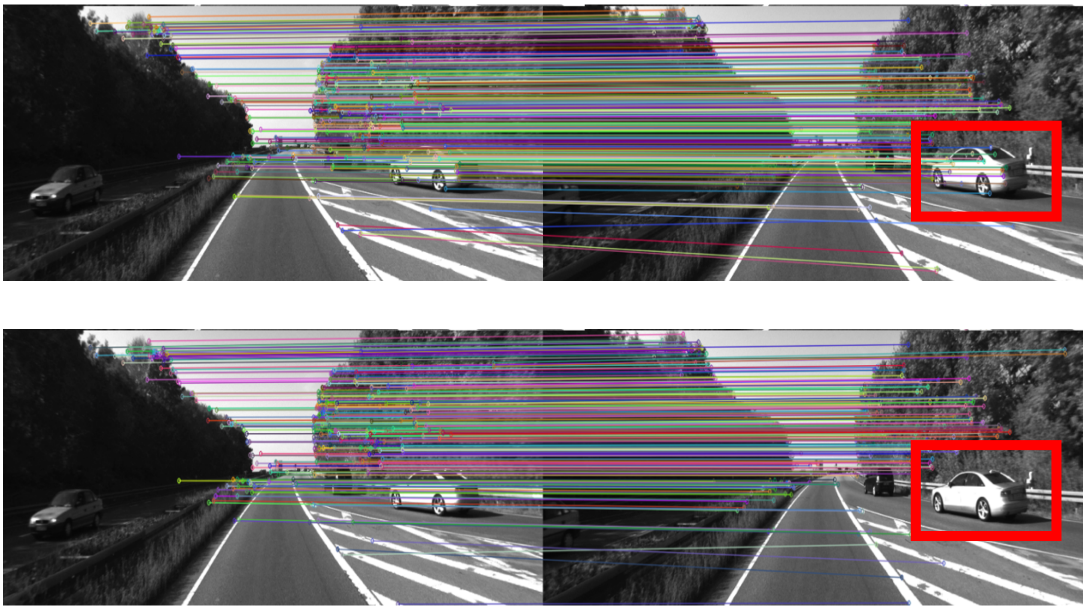

Figure 5.2 Top image shows the system without dynamic object detection node. The bottom image demonstrates the system with the dynamic object detection.

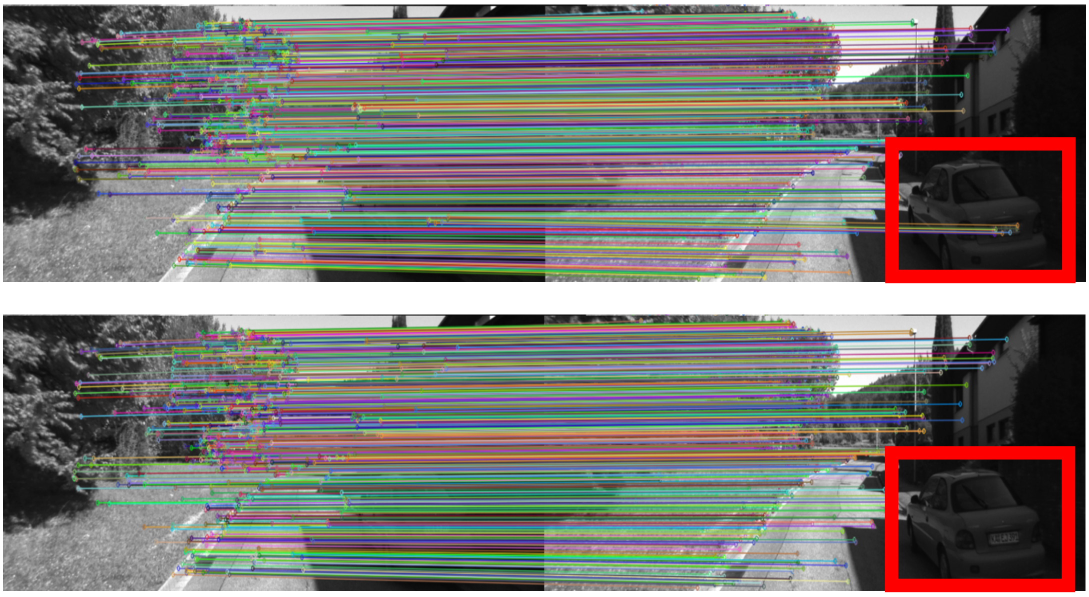

Figure 5.3 Top image shows the system without dynamic object detection node. The bottom image demonstrates the system with the dynamic object detection

Similarly **experiment 1.c** demonstrates the effect of depth filter on methods estimates. Where using depth filter, we can improve the methods estimations. For sequence 01 the improved results are because its mostly in a highway with large portion of the frame being open sky. There is very less objects in the surrounding and most of the scene has a consistent texture, unlike the scene in sequence 03 which is an urban setting with lots of trees and rough edges.

Depth sensing by stereo vision is suspectable to the issue of feature less environments. In a feature less environment the estimates of depth are less reliable than the converse. By addition of a filter, we smooth out to an extent the depth estimation. Figure 5.4 shows the depth map computed for an instance in sequence 01 with and without the depth filtering node. However, the smoothing effect can have negative effects as well specially on the edges as the type of filter used in the system is not an edge preserving filter.

Therefore, if an estimation of depth is accurate and the location of this estimation is on an edge, its effected by the filter which changes the estimation based on other nearby estimations. The red box on Figure 5.4 demonstrates the improvement in depth estimate in localized areas.

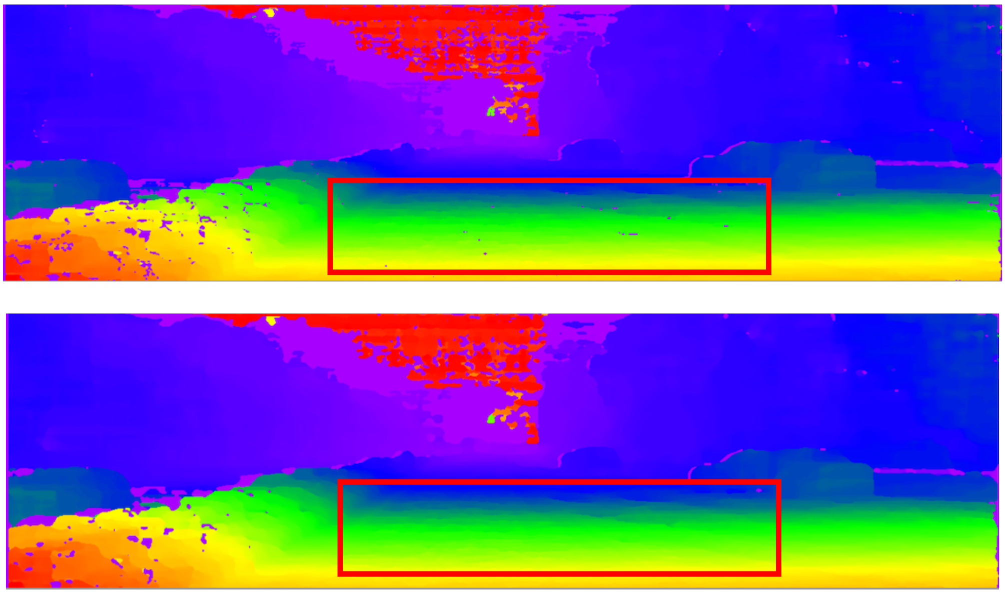

Figure 5.4 Top image shows the system without depth filtering. The bottom image demonstrates depth map with filtering enabled

In **experiment 1.d** we demonstrate the capability of our system to work in either stereo or monocular mode. The monocular mode estimates are better for sequence 04 as depicted by Table 4.9, whereas the estimate by the stereo mode is better for sequence 03 depicted by Table 4.10. The quantitative difference is not as important because, while the system is capable of performing in either mode there is a requirement for ground truth of translation between estimation in monocular mode. The inherent issue of scale ambiguity in monocular visual odometry technique in this case is bypassed by providing the scale explicatively to the system for each estimate from the available ground truth. This experiment highlights in this case the difference in accuracy and reliability of motion estimation algorithm, 2D to 2D algorithm for monocular mode and 2D to 3D estimation algorithm for stereo mode as all other parameters of the system are kept the same. We find from the results from Table 4.1.8 that stereo visual odometry outperforms the monocular visual odometry in our setting for sequence 03. Whereas for sequence 04 we see the opposite by a small margin.

Both **experiments 1.b** highlights the tradeoff in accuracy and computational efficiency. Dynamic object detection is CPU intensive tasks. While the addition of this system has reliably decreased error it has however increased the computational time for each pose estimation. We have found out that on the test system setup we averaged 9 pose estimation per second whereas with the dynamic object detection node we averaged 4.5 pose per second. Yes, if the dynamic object detection node computation was performed on a GPU or TPU would have yielded better results. However, given a system with only CPU resources available the tradeoff can be significant. Being able to estimate pose at 9 pose per second is significant when considering that the KITTI dataset was collected at 10 frames per second. Therefore, ensuring real time capabilities. However, the system was tested on a full-fledged computer in this case an M1 MacBook pro. The computational time required on edge devices such as jetson or raspberry pi is more interesting.

5.2 Analysis of the fusion system

As explained in Chapter 3 the fusion system is not going to yield a better estimate in comparison to the ground truth because of the type of the fusion system developed. Rather the system will provide increased stability in certain situation. We will understand the results from the experiments and conclude the usefulness of the fusion system in our proposed method.

Experiment 2.a describes the fusion of visual odometry estimate from our visual odometry method in stereo mode with inertial measurement estimates provided by the KITTI dataset. ES-EKF represent the estimation from the fusion system. From the result alone we can conclude that as described the fusion system estimate does not yield in a better estimate than the visual odometry system. For example, in sequence 04 the fusion estimate yielded significantly more erroneous result than visual odometry alone.

With the establishment of results from experiment 2.a we proceeded with experiment 2.b which better explains the application of this current fusion system. Table 4.14, 4.15 and 4.16 explains the results of this experiment for sequence 01, 04 and 09 respectively. In this experiment random estimates from the visual odometry system were not considered. And the non-consideration of the number of estimates were described in percentages. We find that for sequence 01, for all cases of dropout there is a slight improvement of the ATE in meters. However, the logic dictates a degradation or an increase in the error. The marginal increase can be better understood by looking at the Figure 4.7, Because ATE computes the difference in Euclidian distance between the estimate and ground truth even though the estimate is wrong, being closer to the ground truth improves the ATE metrics. However, the RPE which is a local metric shows the steady increase in error corresponding to an increase in dropout percentage.

We better understand the actual increase in the ATE and RPE if we look at the results of the fusion estimate with respect to the visual odometry estimates. As the drop out percentage increase the ATE and RPE also increases with respect to the visual odometry estimates. Here in this case the application of the fusion system is highlighted. Even in a very adverse condition when half of all estimates is discarded the fusion system is still able to maintain the trajectory estimate with relatively low error estimates. This yield potential benefits in cases when the visual odometry system additionally outputs a reliability metric for a given estimate and based on the metric the fusion system can better decide how to use the visual odometry estimate.

For sequence 04 and 09 we see the same trend as for 01, however in these cases ATE is significantly increasing at each dropout level. We observe that even at 90% drop out the ATE has increased by only 3.366 meters for sequence 09. But we can see that from Figure 4.9 that this is not viable as at certain places the error is as high as 20 meters. This effect is profoundly visualized by Figure 4.8. Where a drop out of 90% shows how much the estimate from fusion system veer off when solely relied on IMU estimates only.

The fusion system in conclusion from this experiment clearly demonstrates the viability of the system in adverse events. The utility of this fusion system and its applicability context is best explained when there is a situation where a few estimates from the visual odometry system is not available. Might be because of the errors in the visual odometry system stemming from different reasons such as low feature points to track motion or failure in any other subsystem. In such events the fusion system can reliably track motion. This is necessary as mentioned in Chapter 2 both the visual odometry system and inertial only odometry system are incremental estimator. In the case of visual odometry here, if an estimation failed or is deemed less reliable than the next estimate made when added to the previous incorrect estimate yields more erroneous result. In such situation the fusion system can be a reliable source of estimate. In conclusion with the fusion system the goal can be attained however there is added error when considering all the IMU estimates and 100 percentage of visual odometry estimates

5.3 Analysis of the method on our data

Experiment 3 scenario 1 from Table 4.17 we understand that the method demonstrated an ATE of 1.51 meters, in this scenario. However, the cumulative distance is only 32 meters approximately. The error is too large for such a short distance when compared to the results form experiment 1. Analyzing the depth map and the scenario can better explain our results. Figure 5.5 shows the depiction of the depth map for an instance from the scenario.

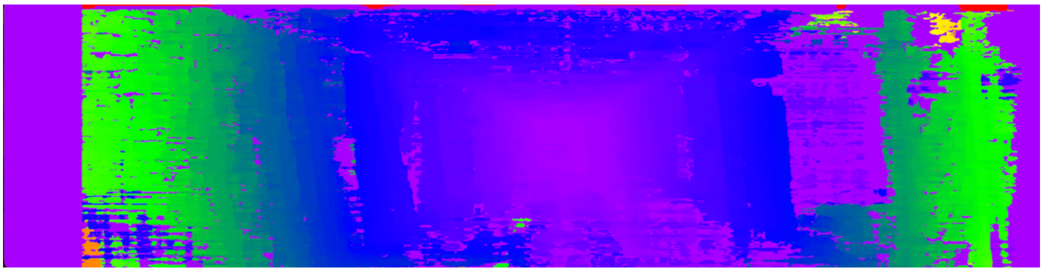

Figure 5.5 Depth map of an instance of the scenario 1 in our data

From Figure 3.13 we can understand that the environment is relatively texture less as both sides are wall. This scenario was collected using the system proposed in 3.1.2. The camera used in this scenario is an OAK D pro system with a baseline of 0.07 meters and active stereo. We can see that even with active stereo the depth map is inconsistent. A lower baseline for the stereo system reduces the range of depth estimation. For comparison the stereo baseline used in KITTI dataset is 0.54 meters. Having a larger baseline effectively ensures a depth estimation at longer ranges and more accuracy at shorter ranges. From the Figure 4.10 we can visually see that the computed trajectory is very close to the approximated ground truth. Even though ATE is large the local metric RPE is 0.08 meters. With a better stereo setup, we can expect a better result even at longer distance.

Scenario 2 of experiment 3 as described in Section 3.3, was limited by the baseline of the stereo system. Given that the scenario was set up in an outdoor environment in a parking space the baseline of the stereo system was too small essentially rendering most of the feature identified in the scenario to be discarded because the depth estimation yielded incorrect results. Those that were filtered out and satisfied the depth criteria were only a few and the proposed method was not able to successfully estimate the motion. This also highlights a potential disadvantage of the system in low feature or texture less environment. However, having a large baseline distance for the stereo setup allowed for more inclusion of feature points. Based on both scenarios we can conclude that a larger baseline between the camera in a stereo setup would yield better results, as larger baseline would yield in better depth estimation.

6 DISCUSSION

The development of the proposed visual odometry method took into consideration several key challenges. The proposed method is dedicated as solution to the issue of GNSS denied localization which is almost always an issue mostly present in an indoor setting. Outdoor environment with a clear sky allows for GNSS based localization. However, GNSS based localization is proving to be a difficult choice given the rise in the use of GNSS jamming systems or rise in more city structure such as bridges and subway where GNSS does not work as intended. To solve this issue while ensuring minimal infrastructural changes and ensuring scalability is the main reason to choose visual odometry system. As described in detail in Chapter 2 use of BLE, Wi-Fi, UWB or RFID is possible. However, these radio-based methods are greatly affected by several factors such as line of sight and the accuracy of these system is influenced by the number of deployed supporting system in the area. As a rule of thumb since these technologies are effectively meant to be used within tens of meters for range estimation to increase the range and also increase the accuracy requires more of this device to be set up. This task is tedious as it requires infrastructural changes and is dependent on the environment. The scalability of such system is low, and the cost can be modeled using a linear relationship to accuracy and range.

As an alternative visual-inertial odometry system is proposed. The proposed system assumes the following:

1) The availability of a stereo camera setup or a monocular camera setup. For the later translational information is also required by the means of other sensor such as wheel odometry or using complicated systems and assumption.

2) Knowledge of the start position. Visual odometry is an incremental type of estimator. And the motion between two frames is computed and there is no way for the system alone to understand the start location. To acquire the start position in certain cases in indoor settings, beacon-based systems can be used.

The advantages of the proposed system in comparison to beacon based or radio-based technology are as follows:

1) Proposed system is not susceptible radio interference, multi path effect and line of sight requirements.

2) The accuracy of the system depends on the software component and to an extent the camera setup. High-definition camera can capture detailed images which could allow for better estimation of motion. The scalability is not a problem for

visual odometry based system as those agents that are required to track their location are only fitted with a camera setup and computing unit. There is no requirement of any infrastructural changes required as in the radio-based system.

3) The system is robust against occlusion and invariant to presence of dynamic moving objects in the scene.

4) The estimation is done locally and does not require any server node setup.

The disadvantage of the proposed system are as follows,

1) Is affected by environment low in texture.

2) Inability of the system to work in dark environment.

3) Requires more computational resource.

The above-mentioned factors such as scalability and cost impede the ability of radio-based systems application in different applications. Visual odometry based system suffers from the limitation of knowing the start position which in this case must come from a radio-based system or other system such as GPS or visual marker setups. This indicates that a fully autonomous system for a given environment will most probably combine the radio-based system or visual markers with visual odometry system. In which case radio-based system are not required in large quantities and a minimum of three beacons would suffice for the triangulation and estimation of the start position for the visual odometry system. Industry such as maritime industry could greatly benefit from this setup. Maritime logistics or logistics in general is a task which can be extensively improved with automation. Different equipment's used to carry goods in a logistic environment indoor if tracked would allow for optimization and improving the efficiency of the task.

However, the nature of the system involves the use of cameras to capture visual data which could be a problem where privacy is required or enforced. In these cases, visual odometry system that are offline like the proposed method can be employed. However, for machine learning based visual odometry method, the lack of necessary data for training from the specific environment would be an issue. Requirement of authorities in the application of visual odometry systems are low and more open. Modern vehicles come with camera in any way such as a driving assistant or digital mirror. The use of information from these cameras in estimation of motion is therefore only an issue of the software and not the issue of regulation mostly.

SUMMARY

6.1 Conclusion

In this thesis a visual-inertial odometry based method was proposed to overcome the limitation of localization in GNSS denied areas. Specifically, a geometric feature-based hybrid visual odometry method coupled with an error state extended Kalman filter has been proposed and implemented. The implemented system clearly demonstrated robustness and reliability when tested in various scenarios. The proposed system demonstrated superior performance to state of the art system such as ORB SLAM 2 and DF-VO in KITTI sequences 01, which known for its challenging environment, with translational error in percentage being 98.9% and 31.25% lower than ORB-SLAM 2 without LC and DF-VO respectively. Additionally, our method demonstrates superior results in terms of RPE for all tested Sequence 01, 03, 04 and 09 compared to other state of the art systems as depicted in experiment 1.a. The fusion system also demonstrated its robustness in scenarios where visual odometry estimates were discarded randomly. Even in such adverse conditions the fusion system was able to ensure steady estimation of the trajectory although with an increase of ATE in the estimation. With a drop out of 25% of the visual odometry estimates, the fusion system was able to predict the trajectory with an ATE of 1.839 meters more than the ATE without any dropout which is 2.953 meters for sequence 04 as presented in results of experiment 2.b. We also demonstrated that for up to 400 meters our proposed method can predict the trajectory with an ATE of less than 3 meters for most scenarios tested. The proposed method clearly achieved the goals set out by the thesis. We also demonstrate that the proposed system can be used with custom hardware setup without any additional requirement in changes to the method.

The goal was limited to a small area of 400 meters depicting indoor environment or applications such as maritime logistic environment or warehouses. The proposed method is a concept that could be improved on. Further improvements must be made for the application of the method, this include but is not limited to real time processing requirements on edge devices. Integration requirements in existing projects and development time consideration are reasons behind the existing implementation of the method in language such as Python. Provided good and satisfactory results were achieved, the proposed method still can be improved on as discussed in the next section.

6.2 Future works

As extensively discussed by analysis of the results the current proposed method can be improved in several ways. While the current method in its current setup can demonstrate reliable performance, improvements can still be made. As discussed currently the method lacks the ability to distinguish between static and dynamic object. As a result, all possible dynamic object in each frame is excluded. Static objects in the scene are a source of information for the method which can yield in a better estimate. To circumvent this issue an object detection and filtering node with optical flow-based classification of moving object is proposed to overcome the existing limitation.

The proposed method uses a geometric based feature detector SIFT. Argument as for the inherent benefits such as repeatability, simplicity of application, robustness over other geometric system is the primary reason for the use of geometric based detector. Machine learning based detector however could prove useful in custom application. In a scenario where the visual inertial system is deployed in the same environment, machine learning based feature detectors can be trained to yield better accuracy.

Moving forward as an improvement to the proposed method a dynamic weighted estimate setup is proposed, in which the visual odometry estimate is given a score, based on which the fusion system decides the variance of the visual odometry estimates enabling a dynamic Kalman fusion system. The system can also discard the visual odometry estimate and replace it with an estimate computed from the IMU. Such a system could prove reliable in several situation and would improve the reliability of the system.

It’s imperative that machine learning based techniques are becoming more prominent. However, this thesis demonstrates the robustness of a primarily geometric based method. Moving forward a hybrid approach seems the most viable in terms of overcoming the disadvantage of both geometric based and machine learning based methods. For example, the stereo node uses a geometric based computation for the estimation of depth map in the proposed method. This system can be replaced with a machine learning based stereo network which could demonstrate better depth estimation even in low texture scenarios. Reimplementation of several computation in the proposed method to be computed on a GPU would decrease the computation requirement and would enable real time applications. The method could be tailored for use with hardware such as jetson Orin nano, taking use of the dedicated GPU which would allow for the proposed method to run on a edge device in real time.

KOKKUVÕTE

Selles lõputöös pakuti välja visuaal-inertsiaalne odomeetriapõhine süsteem, et ületada lokaliseerimise piirangud GNSS-i keelatud piirkondades. Täpsemalt on välja pakutud ja rakendatud geomeetriliste tunnustepõhine hübriidne visuaalse odomeetria meetod koos veaoleku Extended Kalmani filtriga. Rakendatud süsteem näitas erinevate stsenaariumide korral testimisel selgelt robustsust ja töökindlust. Kavandatav süsteem näitas KITTI järjestustes 01 paremat jõudlust kui nüüdisaegsed süsteemid, nagu ORB SLAM 2 ja DF-VO, mis on tuntud oma keerulise keskkonna poolest, kusjuures translatsiooniviga oli 98,9% ja 31,25% madalam kui ORB-SLAM 2 ilma vastavalt LC ja DF-VO. Lisaks näitab meie meetod paremaid tulemusi RPE osas kõigi testitud järjestuste 01, 03, 04 ja 09 puhul võrreldes teiste nüüdisaegsete süsteemidega, nagu on kujutatud katses 1.a. Liitmise süsteem näitas oma tugevust ka stsenaariumides, kus visuaalse läbisõidu hinnangud jäeti juhuslikult kõrvale. Isegi sellistes ebasoodsates tingimustes suutis liitmise süsteem tagada trajektoori püsiva hindamise, kuigi hinnangus ATE suurenes. Kui visuaalse läbisõidu hinnangutest 25% langes, suutis liitmise süsteem ennustada trajektoori 1,839 meetri võrra suurema ATE-ga kui ATE ilma väljalangemiseta, mis on 2,953 meetrit järjestuse 04 puhul, nagu on näidatud katse 2.b tulemustes. Samuti näitasime, et kuni 400 meetri ulatuses suudab meie pakutud meetod enamiku testitud stsenaariumide puhul ennustada trajektoori ATE-ga alla 3 meetri. Väljapakutud meetod saavutas selgelt lõputöös püstitatud eesmärgid. Samuti näitame, et pakutud süsteemi saab kasutada kohandatud riistvara seadistusega ilma meetodi muutmiseks täiendavate nõueteta.

Eesmärk piirdus väikese 400-meetrise alaga, mis kujutas sisekeskkonda või rakendusi, nagu merelogistika keskkond või laod. Kavandatud meetod on kontseptsioon, mida saaks täiustada. Meetodi rakendamiseks tuleb teha täiendavaid täiustusi, sealhulgas, kuid mitte ainult, servaseadmete reaalajas töötlemise nõuded. Olemasolevate projektide integreerimisnõuded ja arendusaja arvestamine on põhjused süsteemi olemasoleva juurutamise taga sellistes keeltes nagu Python.

LIST OF REFERENCES

[1] https://www.statista.com/topics/3573/autonomous-vehicle-technology/#editorsPicks (Accessed 19-04-2024)

[2] H. Zhan, C. S. Weerasekera, J. -W. Bian and I. Reid, "Visual Odometry Revisited: What Should Be Learnt?," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 4203-4210, doi: 10.1109/ICRA40945.2020.9197374.

[3] Leitch SG, Ahmed QZ, Abbas WB, Hafeez M, Laziridis PI, Sureephong P, Alade T. "On Indoor Localization Using WiFi, BLE, UWB, and IMU Technologies." *Sensors (Basel)*. 2023 Oct 20;23(20):8598. doi: 10.3390/s23208598.

[4] Kim Geok, Tan, Khaing Zar Aung, Moe Sandar Aung, Min Thu Soe, Azlan Abdaziz, Chia Pao Liew, Ferdous Hossain, Chih P. Tso, and Wong Hin Yong. 2021. "Review of Indoor Positioning: Radio Wave Technology" *Applied Sciences* 11, no. 1: 279. https://doi.org/10.3390/app11010279

[5] Aqel, M.O.A., Marhaban, M.H., Saripan, M.I. *et al.* "Review of visual odometry: types, approaches, challenges, and applications." *SpringerPlus* 5, 1897 (2016).

[6] Nistér, D., Naroditsky, O. and Bergen, J. (2006), "Visual odometry for ground vehicle applications." *J. Field Robotics*, 23: 3-20.

[7] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 3946-3952, doi: 10.1109/IROS.2008.4651147

[8] Cumani A (2011) "Feature localization refinement for improved visual odometry accuracy." *Int J Circuits Syst Signal Process* 5(2):151–158

[9] Houssem-Eddine Benseddik, Oualid Djekoune, Mahmoud Belhocine. "SIFT and SURF Performance Evaluation for Mobile Robot-Monocular Visual Odometry," in *Journal of Image and Graphics*, pp. 70-76, 2014.

[10] O. Naroditsky, X. S. Zhou, J. Gallier, S. I. Roumeliotis and K. Daniilidis, "Two Efficient Solutions for Visual Odometry Using Directional Correspondence,"

in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 818-824, April 2012, doi: 10.1109/TPAMI.2011.226

[11] Yunliang Jiang, Yunxi Xu, Yong Liu. "Performance evaluation of feature detection and matching in stereo visual odometry," in *Neurocomputing*, vol. 120, pp. 380-390, 2013. doi: 10.1016/j.neucom.2012.06.055

[12] Parra, I., et al. "Robust visual odometry for vehicle localization in urban environments," in *Robotica*, vol. 28, no. 3, pp. 441–452, 2010. doi: 10.1017/S026357470900575X

[13] Aqel, M.O.A., Marhaban, M.H., Saripan, M.I. *et al.* "Review of visual odometry: types, approaches, challenges, and applications." *SpringerPlus* 5, 1897 (2016).

[14] M. Maimone, Y. Cheng, L. Matthies. "Two years of Visual Odometry on the Mars Exploration Rovers," in *Journal of Field Robotics*, vol. 24, no. 3, pp. 169-186, 2007. doi: 10.1002/rob.20184

[15] F. Xu, X. Liu, Y. Cui, M. Yan, and Z. Lai, "Comparison of Image Feature Detection Algorithms," *2022 9th International Conference on Dependable Systems and Their Applications (DSA)*, Wulumuqi, China, 2022, pp. 723-731, doi: 10.1109/DSA56465.2022.00103.

[16] Shashi Poddar, Rahul Kottath, Vinod Karar, "Evolution of Visual Odometry Techniques," in *Recent Advances in Computer Vision*, 2018. doi: 10.1007/978-3-030-03000-1_13

[17] D. Scaramuzza and R. Siegwart, "Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles," in *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1015-1026, Oct. 2008, doi: 10.1109/TRO.2008.2004490.

[18] Jiabin Wang, Faqin Gao. "Improved visual inertial odometry based on deep learning," in *Journal of Physics: Conference Series*, vol. 2078, no. 1, pp. 012016, 2021. doi: 10.1088/1742-6596/2078/1/012016

[19] T. Zhou, M. Brown, N. Snavely and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," *2017 IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 6612-6619, doi: 10.1109/CVPR.2017.700

[20] S. Wang, R. Clark, H. Wen and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 2043-2050, doi: 10.1109/ICRA.2017.7989236

[21] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal and I. M. Reid, "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 340-349, doi: 10.1109/CVPR.2018.00043

[22] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, & Ian Reid. (2019). "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video." doi: 10.48550/arXiv.1908.10553

[23] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal and I. M. Reid, "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 340-349, doi: 10.1109/CVPR.2018.00043

[24] A. Geiger, J. Ziegler and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, 2011, pp. 963-968, doi: 10.1109/IVS.2011.5940405

[25] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255-1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103

[26] J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611-625, 1 March 2018, doi: 10.1109/TPAMI.2017.2658577

[27] Kim, Pyojin, Jungha Kim, Minkyeong Song, Yeoeun Lee, Moonkyeong Jung, and Hyeong-Geun Kim. 2022. "A Benchmark Comparison of Four Off-the-Shelf

Proprietary Visual–Inertial Odometry Systems" *Sensors* 22, no. 24: 9873. https://doi.org/10.3390/s22249873

[28] Andreas Geiger, et al. "Vision meets robotics: The KITTI dataset," in *The International Journal of Robotics Research*, vol. 32, pp. 1231 - 1237, 2013. doi: 10.1177/0278364913491297

[29] Maddern, W., et al. "1 year, 1000 km: The Oxford RobotCar dataset," in *The International Journal of Robotics Research*, vol. 36, 2016. doi: 10.1177/0278364916679498

[30] H. Kim, P. Kim and H. J. Kim, "Moving object detection for visual odometry in a dynamic environment based on occlusion accumulation," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 8658-8664, doi: 10.1109/ICRA40945.2020.9196767.

[31] Zhang, X., Yu, H., Zhuang, Y.: A robust RGB-D visual odometry with moving object detection in dynamic indoor scenes. *IET Cyber-Syst. Robot*. 1–10 (2023). e12079. doi: 10.1049/csy2.12079

[32] Q. Ye, C. Dong, X. Liu, L. Gao, K. Zhang, and X. Chen, "A Visual Odometry Algorithm in Dynamic Scenes Based on Object Detection," *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, Chengdu, China, 2022, pp. 465-470, doi: 10.1109/PRAI55851.2022.9904100.

[33] D. DeTone, T. Malisiewicz and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, USA, 2018, pp. 337-33712, doi: 10.1109/CVPRW.2018.00060.

[34] Yi, K.M., Trulls, E., Lepetit, V., Fua, P. (2016). „LIFT: Learned Invariant Feature Transform." *In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9910. Springer, Cham*. doi: 10.1007/978-3-319-46466-4_28

[35] D. Mistry, A. Banerjee. "Comparison of Feature Detection and Matching Approaches: SIFT and SURF," in *GRD Journals- Global Research and Development Journal for Engineering*, vol. 2, pp. 7-13, 2017.

[36] F. Xu, X. Liu, Y. Cui, M. Yan, and Z. Lai, "Comparison of Image Feature Detection Algorithms," *2022 9th International Conference on Dependable Systems and Their Applications (DSA)*, Wulumuqi, China, 2022, pp. 723-731, doi: 10.1109/DSA56465.2022.00103.

[37] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection," in *ArXiv*, vol. abs/2004.10934, 2020. doi: 10.48550/arXiv.2004.10934

[38] P. -E. Sarlin, D. DeTone, T. Malisiewicz and A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 4937-4946, doi: 10.1109/CVPR42600.2020.00499

[39] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 91–110 (2004). doi: 10.1023/B:VISI.0000029664.99615.94

[39] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," in *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80-92, Dec. 2011, doi: 10.1109/MRA.2011.943233

[40] Joan Sola. "Quaternion kinematics for the error-state Kalman filter," in *ArXiv*, vol. abs/1711.02508, 2015. doi: 10.48550/arXiv.1711.02508

[41] https://www.coursera.org/lecture/state-estimation-localization-self-driving-cars (Accessed 19-04-2024)

[42] https://www.gps.gov/systems/gps/performance/accuracy/ (Accessed 19-04-2024)