Tightly coupled GNSS/map-based monocular vision in vehicle positioning

Menglin Pang

TU Delft

Tightly coupled GNSS/map-based monocular vision in vehicle positioning

by

Menglin Pang

to obtain the degree of Master of Science

at the Department of Geoscience and Remote Sensing,

Delft University of Technology

Student number: 4500245

Project duration: June 1, 2017 – October 31, 2018

Thesis committee:

Prof. dr. ir. Ramon Hanssen,	TU Delft, Department of Geoscience and Remote Sensing
Dr. ir. W. (Winnie) Daamen,	TU Delft, Department of Transport & Planning
Dr. ir. Christian Tiberius,	TU Delft, Department of Geoscience and Remote Sensing, supervisor
Dr. ir. Peter de Bakker,	TomTom N.V.

TU Delft

Abstract

Knowing 'where I am' is always essential and a prior to answer for a moving vehicle. Among numerous onboard sensors, a GNSS receiver for single-frequency Precise Point Positioning and camera are competitive due to the fairly lower cost and the potential to provide a lot of useful information.

However, due to the degraded GNSS solution performance in city valleys, a tight integration is considered combining the two sensors at the observation level, ie. processing the GNSS ranges and the vision measurements in the image of the camera. The availability of High Definition Maps (HD Maps) aids vehicle positioning by providing extra information on the environment. In this project, landmark positions are retrieved through vision and the HD map, and can complement GNSS in city valleys. Additionally, the project focuses on building the mathematical model for the integration of observed landmark position (using a single camera, considering the ease of implementation and cost) and GNSS measurements, analyzing the performance as well as the feasibility for vehicle positioning. The project emphasizes the feasibility study of the proposed mathematical model, which is flexible and capable of using all available input automatedly, and providing a position solution with the best precision.

The uncertainty in the available landmark positions (for instance errors in the HD maps) is handled in two different ways: one is to include the landmark position coordinates as measurements into the model, the other one projects the uncertainty onto the measurements in the camera image. The latter method turns out to be much more efficient. To integrate vision and GNSS measurements, a conversion between an ECEF (earth-centered, earth-fixed coordinate frame), typically used for GNSS, and a world coordinate frame for the camera measurements, is required. A position offset between the GNSS antenna and camera is considered, since the camera lens center does not coincide with the GNSS antenna center. In the simulation and experiment, an extended integration is also presented and discussed which leaves out the position offset, for instance when the GNSS antenna is very close to the camera, which can further improve the redundancy and lower the computational load.

From the simulation and experiment, we conclude that the integration model is able to produce a position solution when one of the sensors is unable to produce a position solution and the other one still can; the extended integration model is able to produce a position solution even when both sensors individually fail to produce a position solution. Among these scenarios, the one when GNSS fails and vision operates, the integration model can produce a position solution within a quarter of a meter in local horizontal coordinates,

and the GNSS measurements do not contribute much to the position solution. Compared to the integration model, the extended integration improves the model by reducing or eliminating the (typically heavy) correlation between the estimates, in particular those for the camera-antenna position offset, the GNSS receiver clock error and the vertical coordinate. Under the same scenarios, the extended integration improves the standard deviation in vertical coordinate and receiver clock error, within a quarter of a meter and one-third of a meter respectively.

Further study is recommended in the direction of applying full image processing procedures to obtain more realistic vision measurements, to include GNSS carrier phase observations to replace the current GNSS positioning based on Precise Point Positioning, in order to have a position solution of similar quality as the vision part. The dimension of system gets larger when carrier phase measurements (phase ambiguities) are added as well as two additional rotations for a camera; the extra rotations introduce a significant amount of nonlinearity in the model. A larger model with increased nonlinearity may call for an alternative model formulation.

Preface

First, I would like to show my sincere gratitude to my daily supervisor, Dr.ir. Christian Tiberius, who provided valuable suggestions and guidance throughout this thesis work by well-scheduled meetings, which enabled me to solve the problems and proceed in time. Thanks for the guidance and timely technical support of providing SF-PPP data processing goes to Dr. ir. Peter de Bakker, who always gave me detailed answers and patient explanations to my SF-PPP related questions. A big thank goes to Dr.ir. Winnie Daamen, who helped me to arrange necessary facilities in the Green Village for the experiment. On the field experiment day, thanks for help go to Mr. Edwin Scharp and Ir. Han Dun for setting up the facilities and taking down necessary measurements, and good company while collecting the measurements.

Menglin Pang

Delft, September 2018

Contents

List of Figures

List of Tables

1

Introduction

In this Chapter, the background of the study is presented, followed by the motivation for the chosen study subject. A general introduction to the structure of the thesis is given to arrive at the answer to the research question.

1.1. Background

Due to a massive need for transportation and higher requirements for safety and convenience, the improvement of vehicle positioning in urban areas is critical to intelligent-transportation-related technologies. However, these technologies involve various aspects and require intensive collaboration from multiple disciplines. Among them, vehicle positioning is one of the key parts, which is the basis to answer the questions of 'where am I' and 'where to go'.

To meet the higher accuracy and reliability of such a vehicle positioning system, multiple sensors are installed to obtain sufficient measurements from the vehicle's surroundings and to ensure that the system is capable of handling complicated environment conditions.

GNSS (Global Navigation Satellite System) is widely used in vehicle navigation and localization on account of its low cost and relatively high precision, and more satellite constellations such as the incoming Galileo and BeiDou navigation system allow the system to have higher precision, accuracy and availability. The method in this thesis computes the solution from corrected range measurements between the known satellite positions and unknown receiver position, however, this technique heavily depends on satellite visi-

bility and the quality of captured signals.

Several alternative positioning techniques have been developed in recent years, one of them is visual odometry. Given an initial camera position, this technique can compute increments with respect to a previous position from a series of real-time images to yield current position, however, the incremental nature of position solution inevitably leads to a system drift, see Brubaker et al. [5]. The possibility to further exploit information from vision system intrigues many related studies in recent years. The techniques in digital image processing are developed and well maintained in some open-source packages such as openCV and openGL. These packages can provide easier access, better performance and faster computation power in the balance of realizing the image processing techniques' functionality. However, a vision system is subject to weather conditions, color of the sky, shadows, sunshine, the angles between sunlight and camera with respect to the traffic signs' normal, and sunlight reflected by close-by facades, see Soheilian et al. [31]. Its application is also restricted when there is a lack of objects.

Because of the properties of these two sensors (GNSS and a camera), one can conclude that a position solution provided by GNSS has better performance in rural areas while it behaves poorer in urban areas; a vision system, which depends heavily on detected objects, is more capable of producing position solutions in urban areas. In order to ameliorate the limitations by using these two sensors individually, an integration of the two sensors can handle more situations and is supposed to outperform the two sensors operating individually. Besides, the GNSS receiver is able to estimate attitude estimation with the aid of visual measurements, and the integration system is attractive since its cheap and portable, see Chen et al. [8].

1.2. Motivation

Current integration for vehicle positioning falls into two main categories, loosely coupled integration and tightly coupled integration system. The former one requires an external GNSS processor to track the satellite measurements and produce position information. However, GNSS observations are sufficient to produce accurate position information only when signals are good, which means that number of available satellites is at least 4, multipath effect is not strong and the satellite geometry is good. In most city areas, although more and more navigation satellites are coming into view, these requirements can still not always be met. Therefore, a tightly coupled integration system, which makes use of GNSS range measurements can help improve the application in city areas, when the available satellites are insufficient to calculate a position solution.

By reason of the scaling problem of the single camera model, one cannot determine distance between the camera and the object from the images. One common approach to compensate the scaling problem is to use multiple cameras on the vehicle with relatively different pose. From vision observations (features

extracted from camera images), the stereo vision can determine the positions of the features with reference to the camera position by bundle adjustment. However, this technique requires extra information such as GPS or Ground Control Points (GCPs) to obtain the absolute scale, otherwise the solutions will drift since Visual Odometry provides the movement of the vehicle by relative measurements from two or more sequentially captured images. Additionally, due to fewer available satellites in a city canyon and the lower the reliability of the vision system, the integration of GNSS and vision is more advantageous when the system suffers longer GNSS outages, see e.g. Schreiber et al. [30].

Due to the improvement of availability and quality of the digital map, the informative map can aid low-cost autonomous driving in most cases, see e.g. Tao and Bonnifait [33]. One approach under this application is to solve the above scaling problem by adding known positions of the landmarks matched from a digital map and captured by a single camera. As we mentioned before, the position produced by GNSS model is rather restrictive in urban areas, compared with the former approach, this approach can also reduce the vision computation when sufficient external information is retrieved from the database (e.g. HD maps) and is stronger at dealing with the situation when GNSS fails.

GNSS has poor position solutions in urban areas which can be improved by integrating with an affordable and portable sensor, a single camera. In order to improve the availability and accuracy of vehicle positioning, in this thesis project, we boost the application further in urban areas by deploying a model in a tightly coupled manner, which uses raw measurements from GNSS and therefore, and has the possibility to produce an acceptable positioning solution with satellites number less than four with the aid of pre-surveyed landmark positions.

1.3. Research questions

The goal of this project is to develop a tightly coupled integration model of GNSS and monocular vision (single camera), with an emphasis on feasibility of monocular vision model in the application of vehicle positioning (or eventually assisted or automated driving). To simplify the problem, several assumptions (vision measurements are returned by object recognition, position of landmarks are retrieved by data association and the road is flat within a near distance) were made to develop a suitable integration model in order to keep the project manageable, and the integration model was tested with a limited field experiment. The research questions are: what is the performance of the newly developed GNSS/Vision integrated model (by testing it with simulated and real data)? Can it work under extreme settings (when the two sensors both fail or one of the sensors fails)?

To answer the main research question, first we need to derive the monocular vision model and the integration model.

- How to derive a Monocular vision model to determine the camera pose with the aid of measured landmark position coordinates?

  The airborne photogrammetry model can extract the ground information from images by exploiting the projection of the positions on images to positions in the real world. Building upon the concept of airborne photogrammetry, the monocular vision model for the automotive application is explained in Section 3.2, which requires one extra rotation to enable a vehicle-mounted camera to look forward.

- How to design a tight integration model for single-frequency Precise Point Positioning (GNSS) and monocular vision model?

  The observations entering in a tightly coupled integration model are at range level, which enables wider application in urban area, when GNSS or vision failed to provide a position solution individually. The observations of the individually failed sensor can increase the integration redundancy, and can possibly provide more validation on the estimates. To combine the two models, the unknowns in the integration model are carefully chosen to link the unknowns from the two models. The concept of a tight integration model and its properties are demonstrated in Section 2.4.

However, the observations are not perfect in practice, it is important to study how these uncertainties propagate into the results.

- How to account for the effect of observation uncertainties? How will the measurements errors (feature identification error and landmark position error) affect the result?

  The pixel measurements and GNSS raw measurement uncertainties (error of vision data processing procedures and error of satellite range measurements) are included in the stochastic model, and we use two methods to account for the effect of landmark position uncertainties. Method I includes the uncertainties effect by updating the stochastic model of the vision model in the iterative process, Method II considers the landmark positions as observations and therefore, their effect is included in a constant stochastic model of the vision measurements in the iterative process. Faults or anomalies in the measurements can be detected through statistical testing and the testing performance is given by the Minimal Detectable Bias, which determines the threshold of bias of each observation that the system is able to detect under certain hypothesis, and when the observation bias is beyond the threshold the model of current hypothesis is able to detect it. When the MDB has a large value, this also means that the other observations in the system have little validation capability of this observation, so this observation can contribute to a biased estimation solution.

- How to express/measure performance (what quantities are of interest)?

The quantities are required to evaluate the performance of the model tested by simulated data and experimental data. The quantities are: precision, reliability and correlation coefficients. Conclusions and recommendations are based on these quantities as well.

1.4. Thesis overview

To improve the GNSS positioning performance in urban areas, one way is to combine it with other sensors, in this thesis project, we choose a portable and affordable sensor, namely a single camera, and integrate these two sensors in a tight manner with the aid of the information retrieved from HD maps to enlarge applications in the future. To simplify the problem, several assumptions are made and a forward-looking monocular vision models and integration models are developed. The model performance are analyzed by using simulated data and tested by experimental data. These contents are described, derived and analyzed in six chapters in total.

The first chapter contains the research background and motivation for this thesis project, and the derivation of the research questions according to the limitations and the assumptions.

The second chapter presents a general description of the chosen sensors, GNSS and Vision respectively, then related work with an emphasis on a map-based integration model is presented, as well as the elaboration of the complete procedure for vision-aided GNSS positioning in the application.

The third chapter first gives a description of the positioning method of GNSS, namely single-frequency Precise Point Positioning, and the derivation of a monocular vision model and the integration model based on the assumptions of Chapter 1, in the context of the vision-aided GNSS positioning application stated in Chapter 2. The properties of the different models are also discussed. The Chapter also introduces the quantities of interest used in the performance evaluation in Chapters 5 and 6.

The fourth chapter presents the optimal performance of the monocular vision model, the SF-PPP model and the integration model, its performanceis evaluated by the simulated data, which gives more insights in the models properties and performance.

The fifth chapter includes the results from the models tested by experimental data. Different from the previous chapter, the results are computed through full data processing procedures, the model performances comply with those in Chapter 4.

The sixth chapter answers the research questions of chapter 1 and draws conclusions for the thesis project, recommendations are proposed for further researches and applications.

2

Research approaches

In this chapter, a brief introduction of the chosen sensors, GNSS and a vision system (camera) is presented, and a literature review of current researches on GNSS/Vision integration methods is presented, with an emphasis on GNSS/Vision integration in the application of map-based vehicle positioning, which is the context of the thesis project.

2.1. Introduction to Global Navigation Satellite System

GNSS refers to the constellations of satellites which generate signals in space and provide positioning, navigation and timing (PNT) information to global users who receive the signal. The first and well-known GNSS, GPS was launched by United States Air Force for military use and was fully operational in 1995 and also available for civil use. GLONASS, sponsored by the Russian Federal Space Agency has been restored since 2010 and is the second fully operational GNSS. The European Union Galileo and Chinese Beidou are both expected to be fully operational around 2020. Besides, the wide-area augmentation system (WAAS), the Japanese Multifunctional Satellite-Based Augmentation System (MSAS), Indian GPS-Aided GEO-Augmented Navigation System (GAGAN) and the European Geostationary Navigation Overlay Service (EGNOS) are compatible with GPS signals and their services can meet the International Civil Aviation Organization (ICAO) standards on accuracy, integrity availability and continuity, see e.g. Camacho-Lara [7]. Low-cost receivers are widely employed based on GPS and GLONASS , in the wake of complete functionality of Galileo and Beidou. The (positioning) services will be enhanced in terms of availability and continuity due to more available satellite observa-

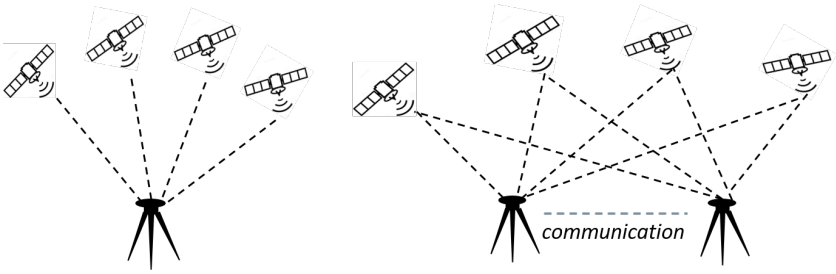


Figure 2.1: Illustration of Single point positioning (SPP, left) and Differential GPS (DGPS, right). SPP uses single receiver while DGPS uses two or more receivers to produce positioning solutions. The black dash lines indicate signals used to produce code measurements from satellites to the receivers. The vector between the base and the rover is typically referred to as a baseline.

tions.Therefore the synchronizations and error of combining satellite observations from multiple constellations are critical for the interoperability of these services to further improve the performance in surveying, agriculture, mining, construction and academic research by using more available satellite observations.

GNSS positioning techniques are divided in three categories: absolute point positioning (e.g. SPP in fig.2.1), relative positioning (e.g. DGPS in fig.2.1) and precise point positioning (introduced in the next section). Absolute positioning is achieved by the 'intersection' of at least four satellite pseudorange measurements towards a single receiver along with the navigation broadcast message. It meets an accuracy level of several meters, and can be used in car-navigation, hydro-graphic surveying and dredge positioning.

The idea of relative positioning is to setup one or more base stations at known positions to provide corrections to another receiver which occupies an unknown position. The overall corrections are obtained by the base station by exploiting its accurately surveyed position. Applying the corrections at the spot with unknown position, a relative position with respect to the base station is obtained for the unknown receiver position by this method. With a real-time data link over Internet or radio link, and the availability of Continuously Operating Reference Stations (CORS), Real Time Kinematic (RTK) can provide a positioning solution using carrier phase observations with centimeter accuracy, where a network of base stations allows more advanced estimation of the various error sources within certain baseline.

Precise Point Positioning (PPP), is based on the observations collected by a single receiver, eliminates the limitations brought by setting up two or more receivers, and is able to produce positioning solutions with centimeter-level accuracy as well.

2.1.1. Introduction to GNSS positiong approaches: PPP and SF-PPP

Precise Point Positioning (PPP) is a positioning technique which utilizes un-differenced pseudorange and carrier phase measurements with the aid of GNSS data products from a global network of reference stations providing satellite orbits and clocks. In practice, a dual or multi-frequency receiver is employed to mitigate the ionospheric delay by linear combination of observations from different frequencies, see e.g. Elsobeiey [15]. Therefore, by providing satellite orbit and clock products, position solutions from this service achieved by the

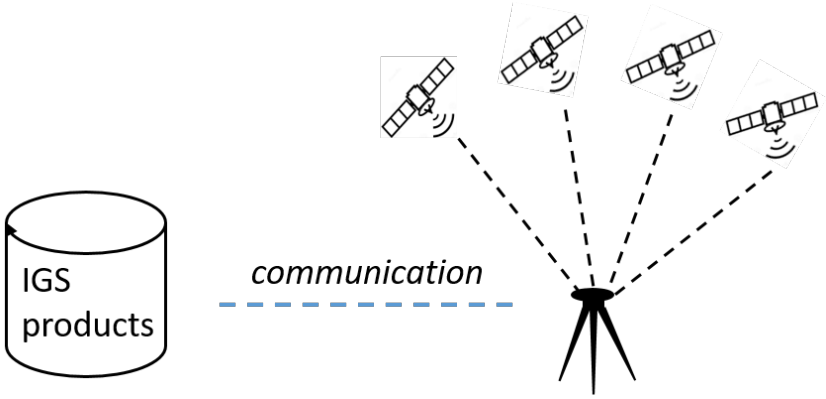


Figure 2.2: Illustration of PPP. The black dash lines indicate signals used to produce code and carrier phase measurements from satellites to the receivers, the blue dash line is the wireless communication between receiver and IGS products.

measurements from a single receiver potentially are with a centimeter-level accuracy. However, the solution needs to converge before the technique stabilizes and meets a specific accuracy, by forming an ionosphere-free linear combination of different frequencies to remove the first-order ionospheric delay errors, and this takes typically half an hour or more, see e.g. Cai et al. [6]. Besides, the noisy pseudorange observations that are used for the ambiguity initial values and the slow change of satellite geometry, lead to high correlation between the ambiguity and position estimates, and are also responsible for the long convergence time, see e.g. Galala et al. [17]. This positioning technology is appealing due to its wider coverage, ease of practice (no need to deploy a local base station), improved accuracy (by applying IGS products), and cheap price (need only one receiver and consumer-grade antenna). However, the collection of corrections is a must and it enables PPP solutions to achieve higher accuracy, which offers more consistency than position solutions in a regional reference frame.

In order to ameliorate the discontinuity in kinematic applications and improve the accuracy of PPP solutions, one way to improve the drawbacks is to provide more corrections to the positioning method. By eliminating satellite phase bias and precise slant ionospheric delays respectively, PPP-AR and PPP-RTK are able to achieve up to sub-centimeter level accuracy, which is comparable to RTK's accuracy without a baseline constraint to the reference network, see e.g. Wang et al. [38] and Odijk et al. [23]. According to GPS Solutions [32], PPP-AR is fully kinematic with an accuracy of sub-centimeter level in horizontal directions. The additional constellations also contributes to the reliability and continuity of PPP performance. By taking more available observations, PPP is able to handle more complicated environments. Another approach is to integrate PPP with other sensors that can be complementary during PPP outages in order to maintain system performance, see e.g. Gao et al. [18].

Instead of eliminating the ionospheric delay from the measurements, single-frequency PPP corrects for the ionospheric delay using external sources, which enables single-frequency PPP to outperform PPP during the first 10 minutes with decimeter accuracy, see van der Marel and de Bakker [36]. Due to its low cost and

reasonable accuracy, the positioning technique can meet functional requirements at lane level, especially in the applications of vehicle positioning and mapping, see e.g. Bakker et al. [4]. Detailed discussions with regard to the properties and implementation of SF-PPP are given in Section.3.1.

2.1.2. Summary

Multi-constellations provide more satellite observations which is beneficial to perform satellite positioning services. Within a certain distance, the corrections for errors in satellite observation are common when the two or more receivers are tracking the same satellites. Based on this concept, Differential GPS requires two or more receivers within a certain distance (hence, local infrastructure). The single receiver positioning method, PPP is attractive by its easier setup and wider coverage, SF-PPP reduces the cost further by employing a low cost receiver. The method depends on corrections from external sources, the error and resolution of the external sources are responsible for accuracy and availability of the positioning solutions.

2.2. Introduction to Vision

GNSS can provide absolute position information individually, and vision sensors can function as both a relative positioning and absolution positioning sensor, and it can be used to evaluate vehicle motion by post-processing procedures, see Gruyer et al. [19]. Vision systems in navigation applications are based on the concept of Photogrammetry, which extracts position information about topographic points from photographs taken by a camera on a airborne platform in combination with image processing techniques. As an absolute positioning technique, vision needs ground control points to estimate the camera pose. As a relative positioning technique, it tracks the same features from the images taken over a different time interval and uses previous absolute position information to estimate its pose.

A vision system has the potential to extract more information and perform more functionalities from the scene with flexible implementation methods: monocular vision (single camera) and stereo vision (two or more cameras). The local features extracted from the image are pixels of interest that are subject to environment changes, and they can be as many as possible. Currently, based on the concept of photogrammetry, monocular vision gains researchers' attention due to its cheap price, small size, and easy implementation. Stereo vision aims to form multi-view geometries of related images. This implementation also has more flexibility in mounting these cameras on a vehicle with different viewing angles, which gives the vision system wider coverage of the scene, and enables it to avoid blind spots and can extract more surrounding information, especially when the field of vision of some of the cameras is blocked. Traditionally, the images captured by multi-camera have to intersect to determine the camera pose and extract motion information. With the aid of GCPs, the vision can be used as an absolute positioning sensor as well. This method eliminates the

error of intersecting the images, but the error of matching the GCPs from the images rises, Like the former method, its implementations are flexible and ersatile. Recently, some researchers uses cameras as an absolute positioning technique by forming lateral cameras and local point of view without intersection between the images, thereby lane detection is more robust since the lane shape is less subjective to the environment changes, and the implementation can be completed by low-cost cameras, see Gruyer et al. [19].

The fast development in computer vision enables practitioners to retrieve more information with better performance and higher computation speed from the images. However, as a passive detection technique, a vision system is highly subjective to the light condition and visibility of the environment, as well as the shape and the texture of the objects, especially when the vehicle is moving, hence, it is necessary to combine other positioning techniques to gain a more stable overall performance and to eliminate system drift. The low cost and fast developing image processing techniques intrigue many researchers' interest worldwide to explore the potential in the application of automated driving.

The following sections describe the principle of photogrammetry. In order to obtain the camera pose, one first needs to link the object in the image to reality (projection model), however, due to the imperfection of the manufactured lens, corrections for lens distortion are required to reduce the error of the object measurement in the image.

2.2.1. Projection model

As one can see from fig.2.3, an object QP lying on the ground plane is captured by the image plane as $Q'P'$. The image plane in viewing position is to simplify this lateral inversion to form the similarity to the ground plane, in case of pre-surveyed landmarks. The image measurement of the object is expressed as $t' = \frac{c}{H_g} t$ when c, H_g, t are known.

2.2.2. Intrinsic, extrinsic parameters and distortions

The extrinsic parameters describe the pose of the camera, namely the orientation ω, ψ, κ in Fig.3.2, the position coordinates on a geo-referenced map X, Y, Z, and the focal length f, if it is unknown or the camera is focused on near sight instead of infinity. In the implementation of this project, the focal length is assumed to be constant and the camera is focused at infinity. The intrinsic parameters encompass focal length, image sensor, position of the principal point, and lens distortion. The focal length describes the distance between the lens and the image sensor when the object is in focus. The principal point is the central point of image plane and is computed as half of the image size in pixels. Lens distortion describes the phenomenon that straight lines in reality are shown as curves in the image, which is caused by the imperfection of optical lens manufacturing. Once the photo is taken, distortion correction is needed to adjust the image before taking

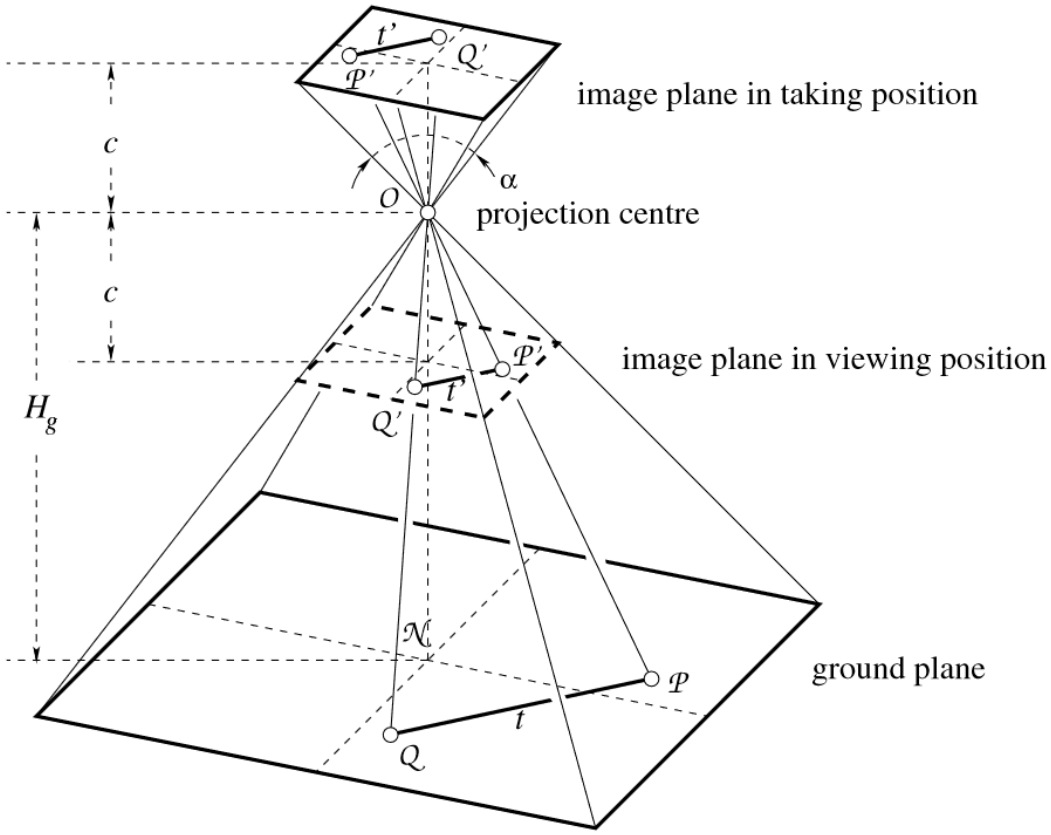


Figure 2.3: The camera is looking towards the nadir point N and its image plane is strictly parallel with the ground plane in the example of Förstner and Wrobel [16]. H_g is the vertical distance between the projection centre O and the nadir point N. Principal distance c indicates the vertical distance from the projection centre to image plane at taking/viewing position and it is equivalent to the focal length f when the camera focuses at infinity. α is the viewing angle at the diagonal of image plane.

	Factors	DGPS	PPP	RTK
Single camera	Position accuracy	+/-	+	++
	Cost	+/-	-	--
	Complexity	--	-	--
	Coverage	+	++	+/-
Multi cameras	Position accuracy	+/-	+	++
	Cost	+/-	-	--
	Complexity	---	--	---
	Coverage	++	+++	+

Figure 2.4: Table of optional combinations for GNSS/Vision tight integration, the scale varies from — (very bad) to +++(very good). The combinations are evaluated for the cost of the hardware and installation, position accuracy, complexity and coverage of the tightly integrated system.

pixel measurements.

2.2.3. Summary

The vision positioning system is based on the projection model of photogrammetry, which links the information retrieved from the image and the corresponding information in reality. The technique intrigues intensive researches in recent years due to the improvement of computer vision and image processing tools which can provide more information of the surroundings compare to other positioning techniques, and to meet higher accuracy requirements in future applications.

2.3. GNSS/Vision Integration System

Current approaches to integrate GNSS/Vision are divided into two categories: loose and tight integration. Under each of these approaches, there are many combinations of different techniques of each sensor and various processing filters in the implementation. RTK (Real-time Kinematic) of GPS is able to provide a real-time solution with centimeter accuracy, of a moving object. For a vision system, with the aid of a high precision digital map, positions of certain landmarks which the vehicle captured by its vision system can be retrieved from a pre-surveyed (previously surveyed) database, and to be used in vehicle pose estimation. A vision system can also work as a relative positioning sensor, namely visual odometry. Tie points extracted between images from the vision system can be corrected by absolute position of vehicle from GNSS or the estimate from a previous time interval, which can significantly increases the redundancy of the process to improve vehicle position estimation.

Comparisons between different combinations of GNSS and vision positioning techniques in terms of position accuracy, cost, complexity (computational load for time and space) and coverage of the tight integration are presented in Figure.2.4.

The Extended Kalman filter (EKF) is widely used in navigation due to its straight forward design and multi-

sensor fusion by its iterative structure. However, the EKF may suffer from a large error in the estimated mean and covariance due to the 'first-order' linearized system, since it achieves only first-order (Taylor series expansion) accuracy, while a UKF (Uncented Kalman Filter) tackles the problem by finding the true mean and covariance by carefully selecting the samples. A UKF can handle any nonlinear problem by capturing up the observation equation up to the 3rd order (Taylor series expansion), see Wan et al. [37]. Besides, a particle filter is naturally more suitable and flexible to handle a nonlinear system. Active selection of sufficient particles can ensure any kind of measurement distribution is captured, however, which also introduces extra uncertainty to the model by optimal selection of data due to its changing distribution.

2.3.1. Loosely coupled GNSS/Vision

Loosely coupled GNSS/Vision utilizes the position, velocity and time (PVT) output from processed GNSS data and that from processed vision data. This approach relies on independent performance of two stand-alone sensors, therefore, sufficient observations for each individual sensor. The easy acquisition of positioning service or user equipment for commercial GNSS service and utilization of a mature computer vision tool box requires less knowledge in processing and configuring the sensors, since the solutions are obtained without access to the data processing and internal sensors, which enables a fast the implementation.

2.3.2. Tightly coupled GNSS/Vision

Instead of processing position solutions from both sensors, tightly coupled GNSS/Vision makes use of raw measurements from both sensors. As we mentioned before, GNSS measurements have limitations in urban areas: the geometry of satellites, sufficient satellites (at least four) and the interference of multipath damp GNSS service from producing an accurate solution. This integration method can improve the availability of positioning service by exploiting raw measurements when the above situation occurs, and therefore increase the continuity of position estimation in mobile navigation.

2.3.3. Summary

The basic difference of the two integration strategies is the input data type, velocity and position, raw observations for loose and tight integration respectively. The acquisition of different data type and complexity requirements on the software determine which integration method to choose. In loose integration, GNSS and Vision can operate separately as navigation systems. Loosely coupled integration becomes adequate when the requirement for the consistency and accuracy are less stringent. Tight integration can improve availability of position estimation when the vehicle suffers from satellite outage and multipath propagation in urban areas, but it requires more complicated software and computation power to achieve this.

2.4. Related work

Due to the cost, ease of implementation, and the larger potential to improve the positioning solution especially in urban area, in this thesis project, we are interested in exploring the **tightly coupled GNSS** and **monocular vision** with the aid of **pre-surveyed landmark positions** (previously surveyed landmark positions). In this section, a brief introduction of related researches and findings is presented.

Tightly coupled GNSS/Vision

Instead of using position output, tightly coupled implementation blends raw observations from available GNSS satellites with vision data, which enables the integration to produce a solution with limited visibility and bad satellite geometry.

Choi et al. [9] develops an algorithm that can switch to Vision aided positioning when GPS based positioning is not available; in Won et al. [41]'s work, an integrated vision/INS navigation system based on GNSS observations is developed and tested, and they found the positioning problem is solvable when the number of satellites equals at least three. According to their INS/Vision model, the system is always rank deficient by three, since the system can only determine the relative distance between the features and the sensor. By introducing extra GNSS observations, the integrated model becomes full rank, and what's more, the system observability is further improved and the state vector of the system is solvable when the number of available satellites are less than four when the clock bias is known or well calibrated. Aumayer et al. [3] compared simulated results by using Kalman Filter (KF) and Least Squares (LS) with GPS positions and GNSS measurements, and they found the tightly integrated system (using GNSS) by KF shows advantages in outliers detection capabilities. However, the design of this approach is more complicated than loosely coupled GNSS/Vision, therefore, larger computation power and longer computation time is required.

Tightly coupled GNSS/Vision with pre-surveyed landmark positions

One way to enhance the tightly coupled integration model is by adding pre-surveyed landmark positions, which allows a single camera to capture and compute position estimates in a single epoch, The approach improves the solvability by adding scale to the vision system, which enables the vision system to act as an individual positioning sensor. Besides, using landmarks also facilitates the feature matching process by high precision due to their simple geometric form, lower storage volume, and lower matching errors of their shape and texture, see Qu et al. [28]. Note that the map-based vehicle localization assumes that map uncertainties need to be better than a certain threshold, to be of interest, which means the position information retrieved from the maps are ground truth or with limited uncertainties, see Cui et al. [11].

The close range landmark aided positioning approach is attractive by featuring high resolution in vision measurements since close-by landmarks are easily included in the image even in case of the slightly mis-

pointing camera.The assumption of a flat-horizontal road (only considering heading changes), as we take in this research, is sufficiently safe when considering close-range vision measurements. Besides, the landmarks captured within short distances are likely to have less occlusions, which enables less uncertainties in vision processing, see Peixoto et al. [26].

KIM and N.-H [20] show the feasibility of integrating GNSS range measurements from the visible satellites with precisely known image measurements on landmarks. From their test results, they confirm that a stereo vision aided system enables accurate and precise positioning, and the integrated system performs better than GNSS with the same amount of measurements. Kim and Hwang [21] present a novel model integrating inertial and vision measurements by projecting precisely known landmark positions on to the camera plane, and the results are quite precise, with simulated data and van tests. The model is realized by the difference between the landmark focal plane measurements and projected ones by Inertial Navigation System (INS) estimates on the focal plane, therefore, the system enables to work even with a few landmarks. Qu et al. [28] generated a geo-referenced landmarks database and for visual odometry, integrated with features extracted from selected images to reduce the computational cost, and they found the proper density of the landmark database is critical to compensate the accumulated errors in visual odometry. Won et al. [40] improves the positioning solution by a selective integration of raw measurements from different sensors, which is determined by the selective index aiming to recognize a poor environment with increased error. The augmented integration model outperforms especially when satellites and feature points form poor geometries. Cui et al. [11] developed a real-time positioning method by fusing GPS and vision-based lane mark detection in urban areas, which can achieve an accuracy of centimeter level by producing fix solutions when GPS outages occur. Once the detected lane marking results are cross-validated with the provided road shape prior. This method subsequently refines the localization by the corresponding GPS positions of the detected lane markings.

Localization based on geo-referenced traffic signs

Besides the derivation and design of the integrated model, to make the application in map-based vehicle localization complete, it is also essential to include the landmark selection from and matching with a landmark database (a high-precision digital map).

External geo-referenced database generation: The database contains precise 3D landmark positions and additional information to facilitate the matching procedure. The Navigation Data Standard (NDS) is a popular format for high precision map databases, but it is not the optimal option for researchers to solve the vehicle self-localization problem, since the database needs to provide reliable information of dynamic road information in real time, see Zang et al. [42] and Gruyer et al. [19]. To satisfy these requirements, researchers have unique ways to create a map database to facilitate their own experiment. Zang et al. [42] included land boundaries, occupancy grids and road signs, which can help self-localization system by providing land boundary

shape priority, distance from the camera to the surroundings, orientation of the road surface, and spatial location of the road sign respectively. In the experiment of Cui et al. [11], the researchers first use satellite images and manually segmented and labeled lane markings. The road shape prior is extracted, then they scale the aerial image and add landmark positions by GPS-RTK as the ground truth. However, landmarks look quite different in ground-based imagery and aerial imagery, and the problem aggravates in dynamic environments. To address this problem, recent progress on image processing algorithms mitigates this challenge, see Wang et al. [39]. Further discussion of this subject is beyond this thesis project.

Detection: Since a vision system is subject to more constraints, the landmark detector needs to locate them in the image within a certain computation load and accuracy. What's more, the forward looking camera can yield variations caused by curvature of the road, which further reduces the landmark detector accuracy, see Gruyer et al. [19].

Matching: Matching contains global feature identification from the image and links the 2D observations to geo-referenced database. According to Qu et al. [28], matching the 3D landmarks with their 2D position in the image includes two steps, first to select a collection of candidates from the 3D landmark database by model constraints, and second, to identify the detected 2D traffic sign in the image with the selected landmark candidates and keep the 3D landmark when its re-projected position is close to the nearest 2D traffic sign. Tao and Bonnifait [34] uses pre-stored orientation information of lane markings to search for the segment that has a heading closest to that of the vehicle. In Cui et al. [11], the shape of road prior/or extra environmental information is used to improve the accuracy of detection or pixel measurement in the image, by using sorted GPS points through map matching and the landmarks in the shape of white bars can be detected by searching for certain intensity. In this research project for simplicity, landmark image-measurements were done by human operation (pointing and reading the corresponding pixel positions in the image).

2.4.1. Summary

Recent studies show that the tightly coupled GNSS/Vision has the potential to improve the position solution performance in urban areas. However, the research methods vary differently due to the assumptions, chosen sensors, processing methods, and preparation for the database. The full procedure of an automated GNSS/Vision positioning system is introduced, which is the context of this thesis work. The proposed mathematical model based on single camera and single frequency PPP approach guarantees cheap cost in the applications. With the aid of HD maps, the monocular vision model can also work as a absolute positioning sensor alone. Compared with multi-cameras within the application of SLAM, its implementation is easier and it requires less computational load.

3

GNSS, vision and integration model

In this chapter, the functional and stochastic model of SF-PPP is explained in Sect.3.1, the functional and stochastic model of monocular vision model is elaborated in Sect.3.2, the integration model of the above two models is presented in Sect.3.3 as well as the description of the processing procedure in Sect.3.4 (the input and output for each model is presented in fig.3.1), and the conception, computation and interpolation of statistical properties (standard deviation, correlation coefficient and MDB) is presented in Sect.3.5.2.

3.1. Description of SF-PPP model

In addition to the general description in Chapter 2, a detailed explanation of SF-PPP, its properties and mathematical model are presented in this Chapter. SF-PPP is characterized by the following properties:

First, in the absence of a base station, implementation of PPP enables users to save time and cost in a broader application area. Single frequency PPP edges the cost lower by replacing an expensive dual-frequency receiver with a single-frequency one. The eliminated limitation of the communication between rover and the base station enables the service to work anywhere on the Earth's surface with an accuracy of sub-decimeter in combination of single-frequency GPS/Galileo PPP, see Afifi and El-Rabbany [1].

Second, besides the cost reduction, the prime advantage of SF-PPP is that the observation noise is $\frac{1}{3}$ of the one of the ionosphere-free combination: to form the ionosphere-free combination of observations from a dual-frequency receiver, coefficients are specified to remove ionospheric errors, whose amplification effect

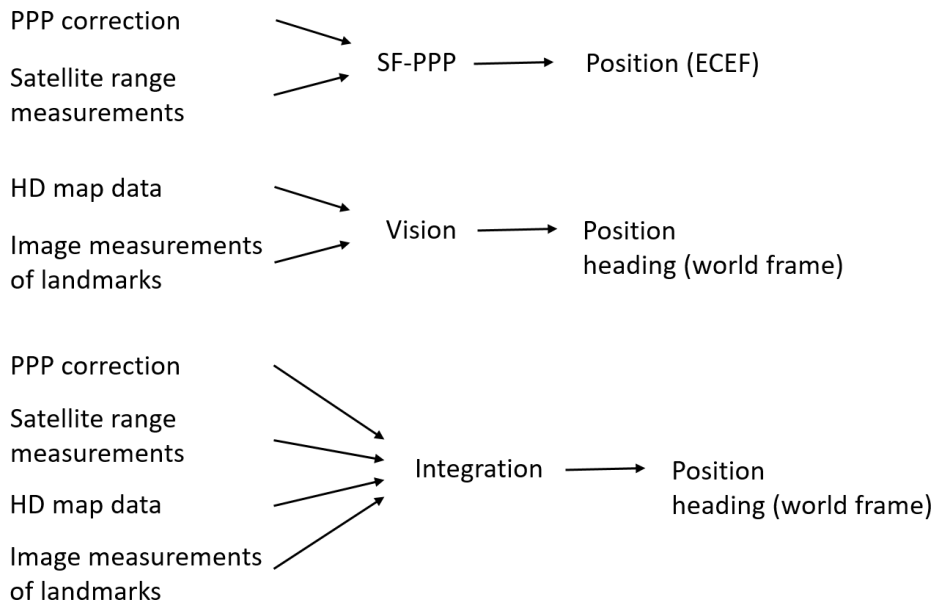


Figure 3.1: The illustration of the SF-PPP, vision and integration model in terms of their input and output.

on observation noise is computed by:

$$\sigma_{ionosphere-free} = \sqrt{a^2\sigma_1^2 + b^2\sigma_2^2} \tag{3.1}$$

where a, b are the selected coefficients, $\sigma_{ionosphere-free}, \sigma_1, \sigma_2$ are the measurement noise from ionosphere-free combination and dual-frequency observations, $\sigma_{ionosphere-free}$ is related to the receiver frequency, and is about 3.23 in unit of length (relative to L1) from GPS observations, see Petovello [27].

Third, SF-PPP would lead to faster convergence. Since the ambiguity from the ionosphere-free combination needs to be solved recursively while suffering larger measurement noise from Eq.3.1, and one more unknown zenith tropospheric delay needs to be estimated as well, processing of dual frequency PPP requires 20-40 min to converge, which means that SF-PPP can outperform PPP at the initial stage, seevan der Marel and de Bakker [36].

Additionally, the improvement of precision and convergence time due to the emergence of other constellations, enhanced processing models, together with the above promising properties, enable SF-PPP to become an extensively used positioning approach in both scientific researches and industries.

GPS errors sources and their effects are: satellite orbit (2m), satellite clock (2-5m), ionosphere (5-30m), troposphere (0.5-5m), multipath (1-10m), and receiver noise (1-3m). Among them, the ionospheric delay contributes significantly in PPP or SF-PPP applications, see Andrei et al. [2] and Pan et al. [24]. The most straightforward and accessible way of including the ionospheric error is to utilize ionospheric coefficients computed by best fitting IGS ionosphere product with Klobuchar style Schaer [29]. Another method adopts

the ionospheric-free code and phase combination and it raises the complexity of the system, since the ambiguity is unknown, a longer convergence time is required till the float ambiguities stabilize, see Andrei et al. [2]. IGS also generates GIMs (Global Ionosphere Models) with a temporal and spatial resolution of 2(hours) × 5°(longtitude) × 2.5°(latitude) which imposes restrictions on real-time application at a local-level. Hence, interpolation or filters are considered to obtain desirable spatial and temporal corrections. The last approach is to make use of satellite augmented system (SBAS; these services can provide real-time corrections and not only ionospheric delays but also satellite clock and orbits at the expense of lower accuracy than those from the post-processing services, see Li et al. [22]. However, the corrections from external services have to be modeled or estimated to achieve high precision positioning results, the position estimation model derived from the interpolated corrections is of higher risk to be rank-deficient and slower to reach convergence, see Li et al. [22]

The implementation in our experiment is based on de Bakker and Tiberius [12]'s model with several considerations: an un-differenced GNSS model; the satellite differential code bias and ionospheric delay are available from CODE products; satellite orbit and clock errors can be obtained by IGS Real Time Service (RTS) over the Internet; the pseudorange hardware delay is absorbed by both receiver and satellites clock offsets, and carrier phase hardware delay is absorbed by the float ambiguities. The observation quations of pseudorange and carrier phase:

$$p = ||r^s - r_r|| + t_r - t^s + n + \iota + e_p$$
$$\phi = ||r^s - r_r|| + t_r - t^s + n - \iota + \lambda A + e_\phi$$

where r^s and r_r are the satellite positions and receiver position, t_r and t^s are the receiver and satellite clock bias respectively, ι is the ionospheric delay n is the tropospheric delay, A is the ambiguity in cycles and λ is the wave length of carrier phase. e_p and e_ϕ are the random noises of pseudorange and carrier phase measurements. The computed observations of the functional model is:

$$\Delta p = p - p_0$$
$$\Delta\phi = \phi - \phi_0$$

which is calculated by the difference between the observed pseudorange and carrier phase and computed ones by inputting initial guesses (approximate values for the unknown receiver position). The positioning

algorithm with k visible satellites is:

$$\begin{bmatrix} \Delta p^k \\ \Delta \phi^k \\ \hat{a}^{k,-} \end{bmatrix} = \begin{bmatrix} -h_G^k & u^k & \delta & \\ -h_G^k & u^k & \delta & I^k \\ & & & I^k \end{bmatrix} \begin{bmatrix} \Delta r_r \\ \Delta t_r \\ b \\ a^k \end{bmatrix} + \epsilon_G \tag{3.2}$$

the increments of receiver position and receiver clock bias $\Delta r_r, \Delta t_r$ are the differences between the unknown parameter values and the initial guess, ϵ_G contains measurement noise, b is the intersystem bias (clock effect between GPS and GLONASS for instance), $\hat{a}^-$ is the ambiguity estimate from the previous epoch to increase the model's solvability. h_G is the unit direction vectors pointing from the receiver to satellites, u is a column of ones, δ is 1 for one constellation and 0 for the reference one, and I is an identity matrix.

$$[\Delta p^k, \Delta \phi^k, \hat{a}^{k,-}]^T \approx N(E([\Delta p^k, \Delta \phi^k, \hat{a}^{k,-}]^T), Q_{p,\phi,a})$$

Since the satellite clock and orbit error, ionospheric error and tropospheric error are evaluated from external products, their uncertainty should be included in stochastic model. In combination of their effects and the observation uncertainty, the stochastic model is:

$$Q_{p,\phi,a} + J_{p,\phi,a} Q_{r^s,t^s,n,\iota} J_{p,\phi}^T = \begin{bmatrix} Q_{pp} & Q_{p\phi} & \\ Q_{\phi p} & Q_{\phi\phi} & \\ & & Q_{\hat{a}\hat{a}}^- \end{bmatrix}$$

where $Q_{p,\phi,a}$ contains observation uncertainty, $Q_{r^s,t^s,n,\iota}$ includes the uncertainty of the parameters obtained from external services, $J_{p,\phi,a}$ is the Jacobian of observation equation with respect to these parameters. The measurement noises are proportional to $\frac{1}{\cos\theta_{zenith}}$ to describe the signal loss mainly due to the main lobe of the receiver antenna gain pattern, the longer path length through the atmosphere by larger zenith angle (or lower elevations) and other propagation errors, e.g. multipath. In the implementation, the measurement noise values: $\sigma_p = 0.3m, \sigma_\phi = 0.02m$ are obtained by personal communication with Dr. Ir. P.F.de Bakker.

3.2. Monocular vision model design

The experiment set-up is designed as Fig.3.5, a number of landmarks with pre-surveyed positions within angle of view are captured by a single camera, therefore, the camera pose can be derived from inputting these position measurements in the world frame in combination of the following central projection model. To simplify the problem, we consider only heading changes in the world frame (flat road assumption), and

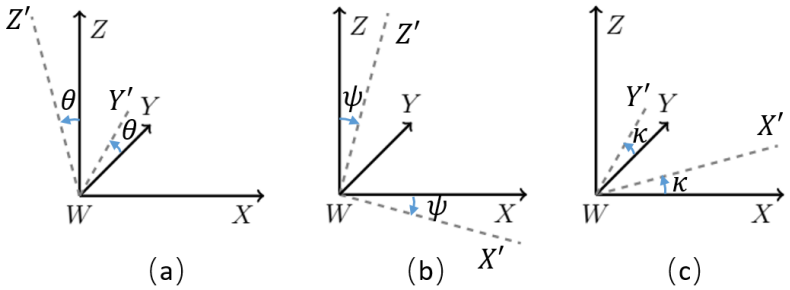


Figure 3.2: Elementary rotations with respect to x axis (a), y axis (b) and z axis (c). The positive directional changes of the three rotations are defined as θ,ψ,κ, by rotating the solid one to the dashed one.

assume the camera lens is without distortions.

The classical photogrammetry camera frame on a airborne platform has the same orientation as the world frame (black; Fig. 3.3), looking down to the ground plane Fig. 2.3. Different from the airborne photogrammetry model, the camera in vision based navigation is placed in front of the vehicle and once its heading is aligned, the camera is viewing along the Y axis (black; Fig. 3.3) of the world frame, which then aligns the opposite of the z axis of the camera frame (orange; Fig. 3.3). The world frame and camera frame are conformed as Fig. 3.3:

In the following monocular vision model, we will consider the camera is mounted directly under the GNSS receiver with a height offset and only the headings κ change in the world frame, or ψ under the camera frame, that is, assuming the vehicle is turning on a horizontal plane (e.g. local level) that is perfectly parallel with the map (X-Y coordinates).

3.2.1. Introduction to 3D Rotation

Elementary rotation

To represent a rotation in 3D space, an arbitrary rotation is achieved by a concatenation of three elementary rotations. The elementary rotations with respect to x-axis, y-axis and z-axis when viewing along the axis towards their origins are (see Fig.3.2):

$$R_x(\omega)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\omega & -\sin\omega\\ 0 & \sin\omega & \cos\omega\end{bmatrix}, R_y(\psi)=\begin{bmatrix}\cos\psi & 0 & \sin\psi\\ 0 & 1 & 0\\ -\sin\psi & 0 & \cos\psi\end{bmatrix}, R_z(\kappa)=\begin{bmatrix}\cos\kappa & -\sin\kappa & 0\\ \sin\kappa & \cos\kappa & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3.3}$$

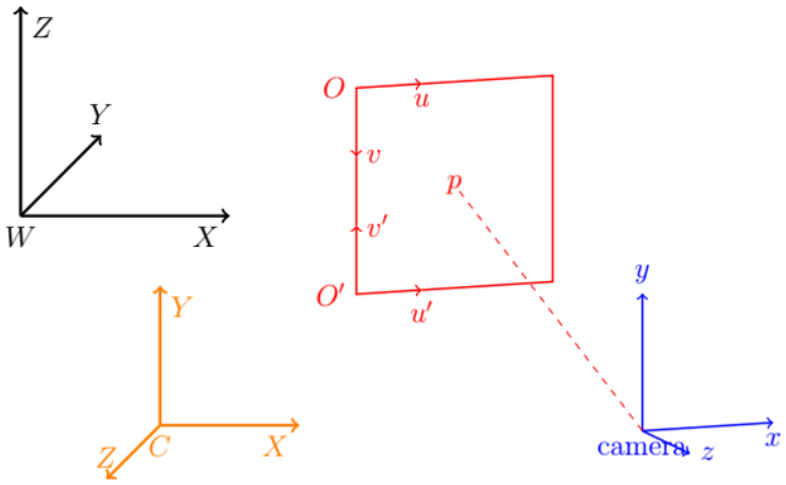


Figure 3.3: World frame (black), camera frame (orange), camera (blue) and image plane O (represents the digitally stored image) and shifted origin O'(the image plane in read parallel to the x-y axes with the u-axis aligned with the x-axis and v-axes with the y-axis) geometry. The rotations follow the right-hand rule: the angle rotation is positive when viewed along the axis towards its origin Fig.3.2. Principle point p is the projected camera center on the image plane, distance between them is f (focal distance).

Concatenated rotation

Based on the above model and the assumption we made, there are many ways to express the concatenated elementary rotations. These methods fall into two main categories, either rotate the camera to the world frame or rotate the world frame to the other. Here we only present the later approach: when the camera is fixed, and we rotate the world frame towards the camera frame and then express the camera within the rotated world frame, see Förstner and Wrobel [16], then the concatenated elementary rotations can be written as:

$$^{C}R_{W} = R_{x}(\omega)^{T} R_{z}(\kappa)^{T}$$

where $^{C}R_{W}$ is the concatenated rotations from the world frame to camera coordinates. First spin the world frame around z axis of its frame with angle of κ, and then rotate with respect to its rotated x axis by $\omega = \frac{\pi}{2}$, by pre-multiplying the elementary rotations from the right to left:

$$^{C}R_{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ 0 & 0 & 1 \\ \sin\kappa & -\cos\kappa & 0 \end{bmatrix}$$

So the projection of a world point (X, Y, Z) under the camera coordinate system $(x, y, -f)$ is achieved by

the conversion of the two frames ${}^{C}R_{W}$:

$$\begin{bmatrix} x \\ y \\ -f \end{bmatrix}_{C} = \lambda {}^{C}R_{W} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z0 \end{bmatrix}_{W}$$

where λ is the scale from image coordinates to world coordinates, (X, Y, Z) and (X_0, Y_0, Z_0) are the pre-surveyed landmark position and camera position coordinates in world frame respectively, the column on the left-hand side is the corresponding landmark position in the camera coordinate system. After multiplication of the right-hand side, and divide the first two rows by the third one, one can rewrite the observation equations $[H_x, H_y]$ as:

$$x = -f \frac{(X - X_0)\cos\kappa + (Y - Y_0)\sin\kappa}{(X - X_0)\sin\kappa - (Y - Y_0)\cos\kappa} \tag{3.4}$$

$$y = -f \frac{(Z - Z_0)}{(X - X_0)\sin\kappa - (Y - Y_0)\cos\kappa} \tag{3.5}$$

The other way under this approach is to first rotate with respect to the x axis of the world frame by $\frac{\pi}{2}$, then rotate the rotated frame with its y axis with the same angle κ. Under this sequence, the concatenated rotations are:

$${}^{C}R_{W} = \begin{bmatrix} \cos\kappa & 0 & -\sin\kappa \\ 0 & 1 & 0 \\ \sin\kappa & 0 & \cos\kappa \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ 0 & 0 & 1 \\ \sin\kappa & -\cos\kappa & 0 \end{bmatrix}$$

Compare ${}^{C}R_{W}$ from the two methods, one can see the two concatenated rotations yield the same solutions. However, features' camera coordinates, x and y cannot be obtained directly from the image. Conversions are needed after obtaining the image coordinates of features are obtained, by shifting the pixel coordinate origin O to the O' and adding the translation from O' to image coordinate origin p. From the above geometry fig.(3.4), point $r(u_r, v_r)$ lying on the image can be converted in camera frame coordinates $r(x_r, y_r)$ by:

$$x_r = u_r - u_p \tag{3.6}$$

$$y_r = v_p - v_r \tag{3.7}$$

(u_p, v_p) is the principal point coordinates in pixels, one can obtain from the half value of the digitally stored image size in pixels. Substitute the left-hand-side of equation (3.4) and (3.5) by (3.6) and (3.7), one obtains the

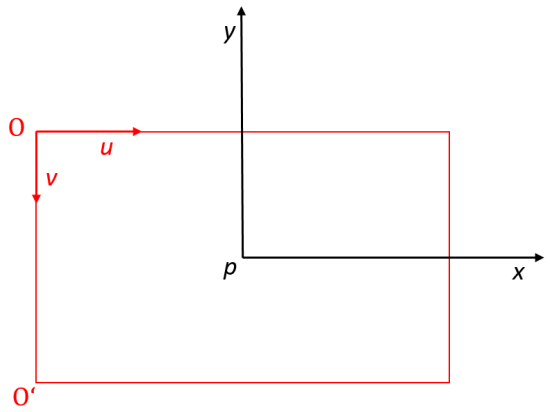


Figure 3.4: Vertical view of image plane. Camera frame orientation (black), image frame orientation, the digitally stored image (O) and shifted one (O'). In this figure, principal point is overlapped with the camera origin. Camera x-y axis is parallel with image u-v axis.

following two equations that link feature measurements from the images (u, v) to their positions in reality,

$$u = -f\frac{(X-X_0)\cos\kappa + (Y-Y_0)\sin\kappa}{(X-X_0)\sin\kappa - (Y-Y_0)\cos\kappa} + u_p \tag{3.8}$$

$$v = f\frac{(Z-Z_0)}{(X-X_0)\sin\kappa - (Y-Y_0)\cos\kappa} + v_p \tag{3.9}$$

In further implementation, they are $[H_u, H_v]$ in abbreviation.

$$u = H_u(\kappa, X_0, Y_0, Z_0, X, Y, Z, u_p)$$

$$v = H_v(\kappa, X_0, Y_0, Z_0, X, Y, Z, v_p)$$

To simplify the above equation, in following sections, we abbreviate the camera pose parameters $[\kappa, X_0, Y_0, Z_0]^T$ as P, landmark position coordinates $[X, Y, Z]^T$ as G, m denotes the feature number, u_p and v_p are known.

3.2.2. Functional model of monocular vision

The goal of the functional model is to obtain camera pose information by pixel measurements from pictures of the landmarks with available position coordinates. In the experiment, we proposed two approaches to function model design, applicable in different application scenarios.

As one can see from the observation equations (3.8) and (3.9), the functional model pixel measurement with respect to the camera pose is expressed as:

$$y = \begin{bmatrix} u^m \\ v^m \end{bmatrix} = H(P,G) + \epsilon_V = \begin{bmatrix} H_u^m(P,G) \\ H_v^m(P,G) \end{bmatrix} + \epsilon_V$$

the system is nonlinear in unknown camera pose $P = [\kappa, X_0, Y_0, Z_0]^T$. To solve the nonlinearity, one way is to transform the system into a linear one by the first derivative term of the Taylor series approximation,

$$\Delta y = y - H(P_0, G) \approx \frac{\partial H(P_0, G)}{\partial P} \Delta P + \epsilon_V$$

$$\Delta P = P - P_0$$

y are the observed pixel measurements, ϵ_V is random errors, P is the unknown camera pose, G is given as known landmark coordinates and P_0 is the initial guess for the camera pose, then repeat the above two equations until the 'weighted sum of square' from the updated term ΔP_0 of this iteration is smaller than a certain threshold (the stop criterion), the iterative process is explained in Section.3.4. The OMC (observed minus computed) equation in an iterative routine is expressed as, with i the iteration number:

$$\Delta y_{i-1} = y - H(P_{0,i-1}, G)$$

In practice, the pixel measurements obtained from the object recognition or feature identification procedure, and the pre-surveyed landmark positions measurements are not perfect because of the accuracy of the equipment and object identification error. These uncertainties are accounted for the following two methods. By inspecting how these uncertainties are included in the stochastic model and propagated through the functional model, suggestions can be made for future application accordingly.

Method I

Method I assumes that the unknowns for monocular vision model are the four parameters of the camera pose:

$$\Delta P = [\Delta\kappa, \Delta X_0, \Delta Y_0, \Delta Z_0]^T$$

In this method, the available landmark position coordinates are not considered to contribute to the functional model directly, instead, they are included in pixel measurements directly to form corrected pixel measurements, and their effect is accounted in an update term in addition to Q_{yy}, the pixel measurement uncertainty, refer for a detailed explanation to Section.3.2.3. So the linearized functional model of m captured landmarks in the image in function of camera pose of Method I is:

$$\begin{bmatrix} \Delta u^m \\ \Delta v^m \end{bmatrix} \approx \begin{bmatrix} \frac{\partial H_u^m(P_0,G)}{\partial P} \\ \frac{\partial H_v^m(P_0,G)}{\partial P} \end{bmatrix} [\Delta P] + \epsilon_V \tag{3.10}$$

$$[\Delta u^m, \Delta v^m]^T \approx N(E([\Delta u^m, \Delta v^m]), Q_{yy}) \tag{3.11}$$

where m is the index of feature measurement, $\Delta y = [\Delta u^m, \Delta v^m]^T$ are the OMC of pixel measurements of the m^{th} feature, $[\frac{\partial H_u^m(P_0,G)}{\partial P}, \frac{\partial H_v^m(P_0,G)}{\partial P}]^T$ is the corresponding Jacobian matrix J_P of $[H_u, H_v]$ from the previous section equation (3.8) and (3.9) with respect to the unknowns P of Method I.

Method II

Different from accounting for the landmark position coordinates uncertainty through the pixel measurements by adding an additional term to Q_{yy}, the other method is more direct: besides the pixel measurements, the functional model includes the OMC of available landmark positions coordinates as well. Therefore, instead of updating the variance matrix of the corrected pixel measurements in each iteration, the landmark position uncertainty $[\sigma_X, \sigma_Y, \sigma_Z]$ is directly included as observation uncertainty in Q_{yy} and it is constant for the iterative optimization process. ΔP is the increment of the unknown parameters of camera pose, the increments of unknowns for Method II are:

$$\begin{bmatrix} \Delta P \\ \Delta G^m \end{bmatrix} = [\Delta\kappa, \Delta X_0, \Delta Y_0, \Delta Z_0, \Delta X^1, \Delta Y^1, \Delta Z^1, \ldots, \Delta X^m, \Delta Y^m, \Delta Z^m]^T$$

ΔG^m denotes m landmark position coordinates increment, G_0 is the initial guess for landmark position coordinates, Δg_0^m is the difference between the observed and computed landmark position coordinates, ϵ_g is the random error,

$$\Delta g^m = g^m - G_0^m = I\Delta G^m + \epsilon_g$$
$$\Delta G^m = G^m - G_0^m$$

The unknowns for this method are $[\kappa, X_0, Y_0, Z_0, X^1, Y^1, Z^1, \ldots, X^m, Y^m, Z^m]^T$, by this method, the given landmark position coordinates are treated as observations, therefore, at the end of iteration, one can also obtain estimators on likely improved landmark positions at the expense of larger computational cost and

slower converging time. The functional model for the second method is:

$$\begin{bmatrix} \Delta u^1 \\ \Delta v^1 \\ \vdots \\ \Delta u^m \\ \Delta v^m \\ \Delta g^1 \\ \vdots \\ \Delta g^m \end{bmatrix} = \begin{bmatrix} \frac{\partial H_u^1(P_0,G_0)}{\partial P} & \frac{\partial H_u^1(P_0,G_0)}{\partial G^1} & 0_{1\times 3(m-1)} \\ \frac{\partial H_v^1(P_0,G_0)}{\partial P} & \frac{\partial H_v^1(P_0,G_0)}{\partial G^1} & 0_{1\times 3(m-1)} \\ & \vdots & \\ \frac{\partial H_u^m(P_0,G_0)}{\partial P} & 0_{1\times 3(m-1)} & \frac{\partial H_u^m(P_0,G_0)}{\partial G^m} \\ \frac{\partial H_v^m(P_0,G_0)}{\partial P} & 0_{1\times 3(m-1)} & \frac{\partial H_v^m(P_0,G_0)}{\partial G^m} \\ 0_{3\times 4} & I_{3\times 3} & 0_{3\times 3(m-1)} \\ & \vdots & \\ 0_{3\times 4} & 0_{3\times 3(m-1)} & I_{3\times 3} \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta G^m \end{bmatrix} + \epsilon_V$$

$$[\Delta u^m, \Delta v^m, \Delta g^m]^T \approx N(E([\Delta u^m, \Delta v^m, \Delta g^m]^T), Q_{yy,MethodII})$$

3.2.3. Stochastic model of monocular vision

The pixel measurement noise is represented by σ_u, σ_v, in our experiment setting, the feature position measurements are independent in the image plane, and the u-axis coordinate and v-axis coordinate of one feature are also independent.

$$Q_{yy} = I_m \otimes \begin{bmatrix} \sigma_u^2 & \\ & \sigma_v^2 \end{bmatrix}$$

The landmark measurements uncertainty is represented as $[\sigma_X, \sigma_Y, \sigma_Z]$ in the experiment design and implementation, the measurements in the three directions are assumed to be independent. According to the illustration plot of Fig.3.5, there are m landmarks captured by the camera, the pre-surveyed landmark coordinates have independent uncertainty in each direction of the world frame,

$$Q_{xx} = I_m \otimes \begin{bmatrix} \sigma_X^2 & & \\ & \sigma_Y^2 & \\ & & \sigma_Z^2 \end{bmatrix}$$

Method I

To take the landmark position measurements uncertainty into consideration, one way is to project the uncertainty of the landmark position into the pixel measurements uncertainty Q_{yy}. With the variance propagation law, the projected variance for the linearized functional model is $J_G Q_{xx} J_G^T$. To update the observation uncer-

tainty in the iterative process:

$$\Delta Q_{yy,i-1} = Q_{yy} + J_{G,i-1} Q_{xx} J_{G,i-1}^T$$

where $J_G = [\frac{\partial H_u^m(P_0,G)}{\partial G}, \frac{\partial H_v^m(P_0,G)}{\partial G}]^T$ is the Jacobian matrix of collinear equations (3.8) and (3.9) with respect to the landmark position $[X^m, Y^m, Z^m]$, the observation uncertainty $\Delta Q_{yy,i-1}$ updates for each iteration, Q_{xx} represents the given landmark position coordinates uncertainty, and the reasoning to form $\Delta Q_{yy,i-1}$ as adding the two terms up, is elaborated in the Section Comments below.

Method II

The observations in Method II include pixel measurements as well as available landmark position coordinates. The uncertainty of observations is propagated through the iterative optimization process by the weight matrix W which is computed by the inverse of the variance matrix, different from Method I, the weight matrix remains constant throughout the iterative optimization process:

$$Q_{yy,MethodII} = \begin{bmatrix} Q_{yy} & \\ & Q_{xx} \end{bmatrix} \tag{3.12}$$

Comments for the two methods

Applying a linear approximation to solve the nonlinear problem up to the first order derivative of Taylor expansion on equation (3.8) and (3.9), the functional model is expressed as the OMC of pixel measurements on the left-hand side as a function of unknown parameters increment. In Method I, the functional model is described as equation (3.13), which can be interpreted as the pre-surveyed landmark position coordinates will contribute to the correcte pixel measurements through the function model. Therefore, subtracted $\frac{\partial H}{\partial G}\Delta G$ from pixel measurements, the corrected pixel measurement is a function of camera pose increment ΔP,

$$\Delta y - \frac{\partial H}{\partial G}\Delta G = \frac{\partial H}{\partial P}\Delta P + \epsilon_V \tag{3.13}$$

In Method II, the pixel measurements are a function of the camera pose and landmark position coordinates. The camera pose and landmark position coordinates both contribute to the functional model of pixel measurements,

$$\Delta y = \begin{bmatrix} \Delta u^m \\ \Delta v^m \end{bmatrix} = \frac{\partial H}{\partial P}\Delta P + \frac{\partial H}{\partial G}\Delta G + \epsilon_V$$

Note that for both methods, they all start from the following full model which is also the functional model of Method II: we take into account the landmark position coordinates as observation measurements, and the pixel measurements are a function of both the camera pose and landmark position coordinates. So the OMC of landmark position coordinates are also included on the left-hand side. The unknowns are camera pose P as well as the landmark position coordinates G,

$$\begin{bmatrix} \Delta y \\ \Delta g^m \end{bmatrix} = \begin{bmatrix} \frac{\partial H^m}{\partial P} & \frac{\partial H^m}{\partial G} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta G^m \end{bmatrix} + \epsilon_V \tag{3.14}$$

the corresponding observation variance matrix of Method II is shown as 3.12.

In Method II, the landmark position coordinates alter as the process goes on since they serve as unknown parameters, according to the model design, the redundancy brought by pixel measurements may improve the coordinates of the landmark positions. Multiply Eq.3.14 on both sides with an invertible matrix $\begin{bmatrix} I & -\frac{\partial H^m}{\partial G} \\ 0 & I \end{bmatrix}$ in order to transform the equation without information loss, one can obtain,

$$\begin{bmatrix} \Delta y - \frac{\partial H^m}{\partial G} \Delta g^m \\ \Delta g^m \end{bmatrix} = \begin{bmatrix} \frac{\partial H^m}{\partial P} & \oslash \\ \oslash & I \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta G^m \end{bmatrix} + \epsilon_V$$

When one is not explicitly interested in landmark position coordinates estimates G and noticing that the observed landmark position coordinates Δg^m would not contribute to the camera pose increment ΔP , and corrected pixel measurements would not contribute to the landmark position coordinates increment ΔG^m, which is equivalent to $g = G$, one can thereby rewrite the above equation as Eq.3.13 by canceling the corresponding I column and row, then the above model coincides with Eq.3.13 of Method I. Based on this, the pre-surveyed landmark position coordinates are measured with sufficiently small error. Applying the variance propagation law to the multiplied matrix,

$$\begin{bmatrix} I & -\frac{\partial H^m}{\partial G} \\ 0 & I \end{bmatrix} \begin{bmatrix} Q_{yy} & \oslash \\ \oslash & Q_{xx} \end{bmatrix} \begin{bmatrix} I & -\frac{\partial H^m}{\partial G} \\ 0 & I \end{bmatrix}^T = \begin{bmatrix} Q_{yy} + \frac{\partial H^m}{\partial G} Q_{xx} \frac{\partial H^{m,T}}{\partial G} & -\frac{\partial H^m}{\partial G} Q_{xx} \\ -Q_{xx} \frac{\partial H^{m,T}}{\partial G} & Q_{xx} \end{bmatrix}$$

since we are only interested in the uncertainty brought by the landmark position coordinates into the corrected pixel measurements, the observation variance matrix for Method I is,

$$\Delta Q_{yy,methodI} = Q_{yy} + \frac{\partial H}{\partial G} Q_{xx} \frac{\partial H^T}{\partial G}$$

The prime advantage of Method I is the fast computation speed and the potential to handle more data, due to the reduction of the functional model dimensionality and smaller observation noises introduced to the

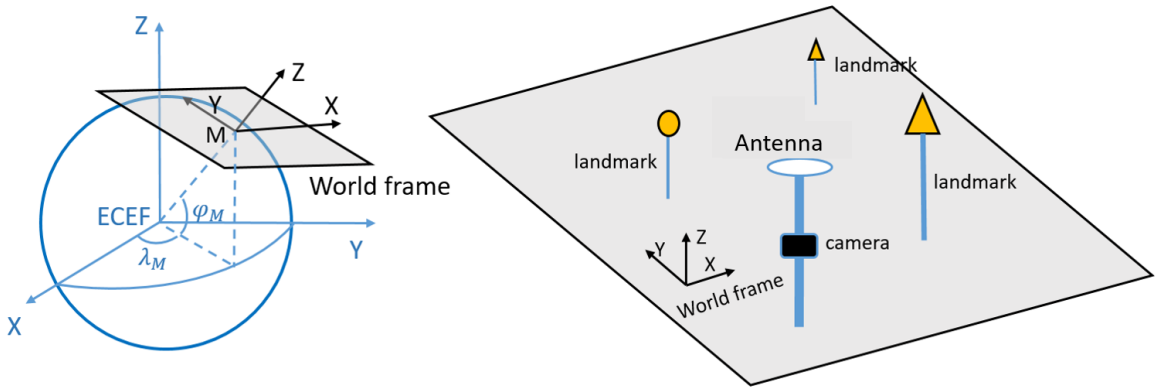


Figure 3.5: On the left: conversion from $[X, Y, Z]$ (ECEF) to $[X, Y, Z]$ (world frame). On the right: visualization of GNSS/Monocular vision set up in world frame. Landmarks are all standing in front of camera within viewing angle. The world frame is converted based on M occupying geographic coordinates $[\varphi_M, \lambda_M, h_M]$, M is the center of this local topocentric reference coordinate system.

optimization process because of less observations. It's worth to mention that the solutions from the two methods are exactly equivalent when we deal with an originally linear problem but may deviate in case of a nonlinear problem as we have.

3.3. Integration model design

The simple illustration of experiment setup is described in Fig.3.5, we assume the GNSS receiver and camera are mounted together with a translation offset in the world frame, and is expressed as a vector t_0 in the following implementation. In the experiment, we assume this offset only accounts for a height difference between these two sensors, that is, they occupy the same X, Y in the world frame but a different Z. All landmarks are standing in front of the camera, and the camera is able to capture them with its default focal length and viewing angle. P_r and $[X_0, Y_0, Z_0]^T$ in P_v are GNSS receiver and camera position coordinates in the world frame respectively, in this section, P_v replaces $[X_0, Y_0, Z_0]^T$ to represent camera position coordinates in world frame.

$$P_r = t_0 + P_v \tag{3.15}$$

3.3.1. Functional model of GNSS/Monocular Vision integration

The input for the integration model includes satellite positions r^s, landmark positions G (for Method II), pixel measurements $[u, v]$ from the single camera, GNSS range measurements $[p, \phi]$, height offset t_0 (optional), initial values for the GNSS receiver position in the local reference frame $P_{r,0}$, the camera position initialization $P_{v,0}$ is not necessarily known and it can be computed from $P_{r,0}$ in combination with the height offset using the linear relation explained in equation (3.15), $r_{r,0}$ can be computed from $P_{r,0}$ and conversions below. The satellite positions and satellite clock offsets can be derived from the navigation message, the position of the

landmarks are measured by, for instance, a total station with reference to the world frame. The idea is to transform the nonlinear system by a first order Taylor expansion and combine the functional model of Method I Eq.3.11 and SF-PPP Eq.3.2. The total OMC of the integration model $[\Delta p^k, \Delta\phi^k, \hat{a}^-, \Delta u^m, \Delta v^m]^T$ is a function of the increments of unknowns $[\Delta P_r, t_0, \Delta\kappa_0, \Delta t_r, b, a]^T$.

To combine the unknowns in the integration model, the following relations are used to bind the unknowns from two sensors through P_r in the integration model:

$$\Delta P_{v,0} = \Delta P_{r,0} - t_0$$

$$\Delta P_{r,0} = P_r - P_{r,0}$$

The vehicle position coordinates are in the local topocentric coordinate system, and conversions from the ECEF frame to the local world frame ${}^E C_N$ are included in the functional model for the GNSS part to produce vehicle pose estimates from the observations of both sensors. The following integration model is based on SF-PPP and the vision model from Method I:

$$\begin{bmatrix} \Delta p^k \\ \Delta\phi^k \\ \hat{a}^{k,-} \\ \Delta u^m \\ \Delta v^m \end{bmatrix} = \begin{bmatrix} -h^k_{G,r_r}{}^E C_N & & & u^k & \delta^k & \\ -h^k_{G,r_r}{}^E C_N & & & u^k & \delta^k & I^k \\ & & & & & I^k \\ h^m_{V,P_v} & -h^m_{V,P_v} & h^m_{V,\kappa} & & & \\ h^m_{V,P_v} & -h^m_{V,P_v} & h^m_{V,\kappa} & & & \end{bmatrix} \begin{bmatrix} \Delta P_r \\ t_0 \\ \Delta\kappa \\ \Delta t_r \\ b \\ a^k \end{bmatrix} + \epsilon$$

Where $\Delta P_r, \Delta t_r, b, a$ denote the GNSS receiver position in world frame, the receiver clock bias, the intersystem bias, ambiguities of the receiver, h_G and h_V indicate vectors for GNSS and vision respectively, and they are the set of unit vectors from the receiver to the satellites and the Jacobian of features measurements on the image with respect to the parameters specified in its subscript, to link with the expression in 3.2.2, $[h^m_{V,P_v}, h^m_{V,\kappa}] = \frac{\partial H^m(P_0,G)}{\partial P}$. Different from the GNSS functional model, the vision model has one orientation parameter κ to be estimated, u is a column of ones, δ is 1 for one GNSS constellation and 0 for the reference one, I is an identity matrix. ϵ is the total measurement noise of GNSS and vision. By the conversion matrix ${}^E C_N$ from the world frame to the ECEF (WGS-84) frame, one can link the GNSS receiver position and camera position together, especially for the later iterative optimization process, and combine into common unknown ΔP_r in the integration model.

$$\Delta r_r = {}^E C_N \Delta P_r$$

$${}^E C_N^{-1} \Delta r_r = \Delta P_r$$

r_r is the receiver position in the ECEF frame from the GNSS observation functional model 3.2. The conversion matrix ${}^{E}C_N$ is computed as:

$$
\begin{aligned}
{}^{E}C_N &= R_z(-(\pi/2+\lambda_M))R_x(-(\pi/2-\varphi_M)) \\
&= \begin{bmatrix} -\sin\lambda_M & -\sin\varphi_M\cos\lambda_M & \cos\varphi_M\cos\lambda_M \\ \cos\lambda_M & -\sin\varphi_M\sin\lambda & \cos\varphi_M\sin\lambda_M \\ 0 & \cos\varphi_M & \sin\varphi_M \end{bmatrix}
\end{aligned}
$$

where $[\varphi_M, \lambda_M, h_M]^T$ are the geographic coordinates of the origin of the topocentric coordinate system M. To express M in ECEF coordinates, the following transformation is needed:

$$
\begin{aligned}
{}^{E}X_M &= (\bar{N}+h)\cos\varphi_M\cos\lambda_M \\
{}^{E}Y_M &= (\bar{N}+h)\cos\varphi_M\sin\lambda_M \\
{}^{E}Z_M &= (\bar{N}(1-\varepsilon^2)+h_M)\sin\varphi_M
\end{aligned}
$$

ε is the eccentricity, $\bar{N}$ is the radius of the curvature of the ellipsoid, in this case WGS-84. Since the receiver position P_r is a relative vector with regard to the center of map, the receiver position in ECEF r_r is written as:

$$
r_r = {}^{E}M + {}^{E}C_N P_r \tag{3.16}
$$

One can rewrite the integration model of SF-PPP and vision for Method II as well:

$$
\begin{bmatrix} \Delta p^k \\ \Delta\phi^k \\ \hat{a}^{k,-} \\ \Delta u^m \\ \Delta v^m \\ \Delta g^m \end{bmatrix} = \begin{bmatrix} -h^k_{G,r_r}\,{}^{E}C_N & & & u^k & \delta^k & & \\ -h^k_{G,r_r}\,{}^{E}C_N & & & u^k & \delta^k & I^k & \\ & & & & & I^k & \\ h^m_{V,P_v} & -h^m_{V,P_v} & h^m_{V,\kappa} & & & & h^m_{V,G} \\ h^m_{V,P_v} & -h^m_{V,P_v} & h^m_{V,\kappa} & & & & h^m_{V,G} \\ & & & & & & I^m \end{bmatrix} \begin{bmatrix} \Delta P_r \\ t_0 \\ \Delta\kappa \\ \Delta t_r \\ b \\ a^k \\ \Delta G^m \end{bmatrix} + \epsilon
$$

Although the Vision system has the potential to produce measurements at a higher frequency, the model holds for all Vision measurements for one single GNSS epoch when the vehicle is static or with carefully synchronized sensors on a moving vehicle.

3.3.2. Stochastic model of GNSS/Monocular Vision integration

To account for the satellite clock and orbit error, ionospheric error, tropospheric error and landmark position coordinates error from external sources or surveying in the integrated GNSS/Monocular model, stochastic deductions for the two methods are introduced, in order to further study the performance of the integration model in a realistic way. The stochastic models presented in the section contribute to the optimization process by giving weights to the observations in the update term, and they need to be computed per iteration.

Method I

The observations for this method are: pseudorange, carrier phase, ambiguity estimate, and two-dimension pixel measurements. Apply variance propagation law in combination of observation uncertainty, the stochastic model of Method I is:

$$Q_{p,\phi,a,u,v} + J_{p,\phi,a,u,v} Q_{r^s,t^s,n,\iota,XYZ} J^T_{p,\phi,a,u,v} = \begin{bmatrix} Q_{pp} & Q_{p\phi} & & & \\ Q_{\phi p} & Q_{\phi\phi} & & & \\ & & Q^-_{\hat{a}\hat{a}} & & \\ & & & Q_{uu} & Q_{uv} \\ & & & Q_{vu} & Q_{vv} \end{bmatrix}$$

where $Q_{p,\phi,a,u,v}$ contains observation uncertainty, $Q_{r^s,t^s,n,\iota,XYZ}$ includes the uncertainty of relevant parameters obtained from external services, the corresponding parameters are specified in the subscript. $J_{p,\phi,a,u,v}$ is the Jacobian of observation equation with respect to these parameters.

Method II

The observations for this method are extended by the landmark position coordinates, so its effect is included in the observation uncertainty directly. The stochastic model for this method is derived based on the same approach as Method I, with respect to the same parameters besides the landmark position coordinates.

$$Q_{p,\phi,a,u,v,g} + J_{p,\phi,a,u,v} Q_{r^s,t^s,n,\iota} J^T_{p,\phi,a,u,v} = \begin{bmatrix} Q_{pp} & Q_{p\phi} & & & \\ Q_{\phi p} & Q_{\phi\phi} & & & \\ & & Q^-_{\hat{a}\hat{a}} & & \\ & & & Q_u & \\ & & & & Q_v \end{bmatrix}$$

The symbols have the same meaning as with Method I, but observation uncertainty is extended with landmark position coordinates uncertainty, and the Jacobian is computed by one less parameter.

3.4. Data processing in model realization

To solve the nonlinear problem, an iterative process is considered by updating the estimates in a series of steps, which can be computed by different estimation methods. In this section, the adoped data processing methods and concept are explained, with an emphasis on weighted least-squares and Gauss-Newton iteration.

3.4.1. Least-squares estimation

A functional model contains a set of observations in y, which are linear in a set of parameters of interest x by the design matrix A and measurement error ϵ.

$$y = Ax + \epsilon$$

Inconsistency of an undetermined system arises when there are more equations than necessary and it failed to produce a unique solution. Least-squares estimation solves this inconsistency by searching for the solution that is as close as possible to the observed measurements, that is, hits the minimal sum of squared error, which can be defined as:

$$\hat{x} = \arg\min_x (y - Ax)^T (y - Ax)$$

The least-squares solution can realize the above constraint by the computation from y and A:

$$\hat{x} = (A^T A)^{-1} A^T y$$

The above least-squares solution is based on equal weight for all observations. If we know prior information about the observables, that is, some of them are more precise than the others, we can include the information into weight matrix W, which is the so called weighted least-squares (WLS):

$$\hat{x} = (A^T W A)^{-1} A^T W y$$

and its corresponding objective function of minimize changes into:

$$\hat{x} = \arg\min_x (y - Ax)^T W (y - Ax)$$

3.4.2. Nonlinear optimization

The nonlinear optimization is an iterative procedure aiming to search for a solution that produces a minimum or maximum value of the corresponding target function. In the procedure, step size, initial guess and search direction are critical to the convergence speed and reliability of the solution.

General procedure for monocular vision model processing

One approach to realize nonlinear least-squares is to use Gauss-Newton (GN) iteration, which is adopted in processing both simulation and real test data in implementation of vision, GNSS and integration model. The method is based on linearization, which is to transform the nonlinear problem by the approximation of first order Taylor expansion, and minimize the second order term. In Section.3.2.2 we already introduced the functional modesl for the two methods, the following optimization procedure is based on Method I is monocular functional model, and one can replace the unknowns and functional model to obtain the optimization for Method II or make some adaptations for it to obtain integration model optimization.

$$P_i = P_{i-1} + \Delta\hat{P}_{i-1}$$

To get the best linear unbiased estimator, we input weight matrix $W = [\Delta Q_{yy_{i-1}}]^{-1}$ into the iterative process, the impact of pixel measurements uncertainty is propagated through each iteration, until the stop criterion is met. The updated term $\Delta\hat{P}_{i-1}$ is computed by WLS to the linearized system:

$$\Delta\hat{P}_{i-1} = (J_{P,i-1}^T W J_{P,i-1})^{-1} J_{P,i-1}^T W \Delta y_{i-1}$$

$J_{P,i-1}$ is the Jacobian matrix of (3.8) and (3.9) with respect to the unknowns in P. Given the initial values for the unknown P, the scheme iterates until the stop-criterion is met:

$$\Delta\hat{P}^T Q_{PP}^{-1} \Delta\hat{P} < stop\ \ criterion$$

Procedure for integration model processing

The processing procedure of the integration model follows the same rule as we explained with the in above monocular vision model. Since the initialization and observations from the two sensors are in different frames, the procedure to process the integration model involves more conversions and constraints from the experiment design shown in Fig.3.5. Since the h_G, h_V are computed with the receiver position in the ECEF frame r_r and the camera position in the world frame P_v respectively, conversion steps are inserted at the end

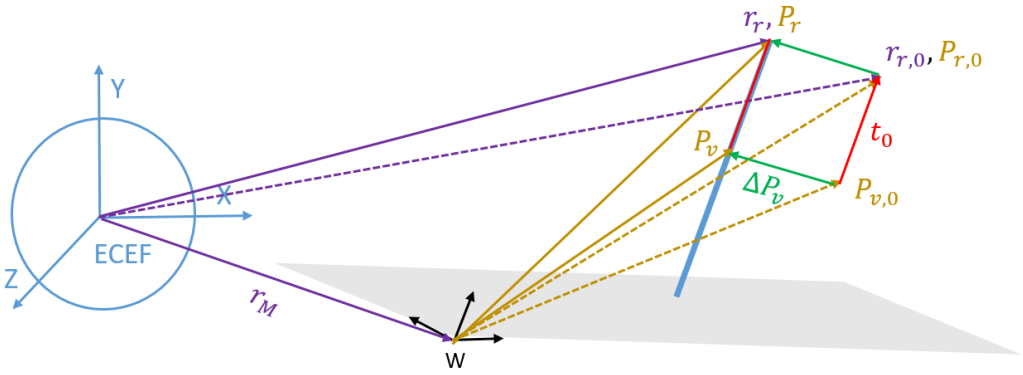


Figure 3.6: The illustration plot of the relationship of unknown position parameters of antenna and camera position. The antenna is mounted with an height offset t_0 above the camera in the world frame. P_v is chosen as a basis in the integration model to link the camera position P_v and GNSS antenna position. $P_v, P_{v,0}$ are the camera position and its initialization in the world frame, $r_{r,0}$ and $P_{r,0}$ are the initial receiver position and they are computed from $\Delta P_v, P_{v,0}$ together with r_M and t_0.

of each iteration and prepare for the Jacobian computation of the next iteration:

$$\Delta \hat{r}_{r,i-1} = {}^{E}C_N \Delta \hat{P}_{r,i-1}$$

$$\Delta \hat{P}_{v,i-1} = \Delta \hat{P}_{r,i-1}$$

Then update the estimate $r_{r,i}$ and $P_{v,i}$ to get prepared for the computation of the Jacobian in next iteration:

$$r_{r,i} = r_{r,i-1} + \Delta \hat{r}_{r,i-1}$$

$$P_{v,i} = P_{v,i-1} + \Delta \hat{P}_{v,i-1}$$

Before the iteration begins, the integration model also needs to convert the initial value $P_{r,0}$ to $r_{r,0}$ and $P_{v,0}$ in order to compute h_V and h_G to begin the iteration. Further, the height offset t_0 is of several decimeters and in the implementation the initial value for it is set to 0.

Selection of stop criterion

To end the iteration and obtain the estimators, the selection of the stop criterion is an open problem and affects the estimation quality, especially for the second method of the monocular vision functional model. The stop criterion is a threshold to control the 'sum of weighted squares' a successful estimation process should reach.

$$sum\ of\ weighted\ square = \Delta \hat{P}^T Q_{PP}^{-1} \Delta \hat{P} \approx \sum_{i=1}^{n} \frac{\Delta \hat{P}_i^2}{\sigma_{\hat{P}_i}^2} \tag{3.17}$$

$Q^{-1}_{\hat{P}\hat{P}}$ is the normal matrix,

$$Q^{-1}_{\hat{P}\hat{P}} = J^T_{P,i-1} W J_{P,i-1} \tag{3.18}$$

One can see the value of stop criterion depends on the number of estimators. To have a general intuition, the stop criterion can be rewritten as an approximation on equation (3.17) right hand side. When we take the square root of the stop criterion divided by the number of parameters, one can rewrite the above equation as:

$$\Delta\hat{P}_i \approx \sqrt{\frac{sum\ \ of\ \ weighted\ \ square}{n}\sigma^2_{\hat{P}_i}}$$

For Method I in monocular vision, we set the stop criterion as 1, the number of estimators are always 4; while for method 2, the value changes into 4 for 4 features case, because the number of estimators is 16.

$$\sqrt{\frac{1}{4}\sigma^2_{\hat{P}_i}} = \sqrt{\frac{4}{16}\sigma^2_{\hat{P}_i}} = \frac{1}{2}\sigma_{\hat{P}_i}$$

This setting is to meet the general standard for both methods, therefore, the change in the estimate between two consecutive iteration steps can meet half of its assumed precision. On the other hand, we would like to see the process improves the estimators' precision, however, for Method II of monocular vision, it is computationally heavy to have a stricter criterion. Gauss-Newton iteration is able to solve nonlinear problems when the computed Jacobian is full rank and the search direction is suitable for a line search. The method has fast convergence speed when it is in vicinity of the solution, and its efficiency relies on an accurate initial guess, see CROEZE et al. [10].

3.5. Description of performance parameters

To evaluate the performance of the above model, several quantities of interest are defined which give further insights in the model and interpretation of results. In this section, these quantities are introduced by conception, computation and interpretation.

Precision shows the influence of the propagated measurement noise spread in result, correlation coefficient describes the dependence of any of the two estimates, and reliability shows the capability of detecting faults and anomalies in the measurements by statistical testing.

3.5.1. Precision and correlation coefficients

The inverse of the normal matrix of equation (3.18), $Q_{\hat{P}\hat{P}}$, is symmetric, and contains variance on the diagonal and covariances between the estimators elsewhere. Suppose there are two estimates $\hat{P}_1, \hat{P}_2$ of unknowns, to obtain precision and correlation coefficients, $Q_{\hat{P}\hat{P}}$ should be processed as: taking the square root of the diagonal which are standard deviations $\sigma_{\hat{P}_1}, \sigma_{\hat{P}_2}$ of the estimates; omit lower triangular matrix and divide the non-diagonal elements which is the covariance $C(\hat{P}_1, \hat{P}_1)$ by the corresponding standard deviations. The correlation coefficient of two estimates:

$$\rho(\hat{P}_1, \hat{P}_2) = \frac{C(\hat{P}_1, \hat{P}_2)}{\sigma_{\hat{P}_1}\sigma_{\hat{P}_2}}$$

$$-1 \leq \rho(\hat{P}_1, \hat{P}_2) \leq 1$$

3.5.2. Minimal detectable bias

Minimal detectable bias can describe the internal reliability that determined by matching a certain probability of false alarm and detection, see De Jong and Teunissen [14]; de Bakker et al. [13].

In the previous monocular vision model, we use all coordinates information of features/landmarks extracted from the image plane and the real world, however, there might be some misidentification when either detecting features from the image plane, or retrieving landmark position coordinates from the digital map. Under such situations, we take the alternative hypothesis so as to account for these blunders' effect (error of misidentification), or when the observations are not fit for the current hypothesis (misspecification).

H_0: there is no blunders in observation set-up, or model misspecification.

H_a: the functional model includes blunders or misspecification in observation set-up

The goal of the MDB is to find a corresponding value of the noncentrality parameter λ that meets the false alarm and misdetection probability (the purple and green area in Figure.3.7 respectively) of the two hypothesis. The false alarm α for type I error we set in this experiment as 0.005, as well as the one for type II error β. There are two ways to find the desirable noncentrality parameter λ, either fix α and look for the λ which gives the error probability β that meets our requirement under a certain degree of freedom (in our case, the size of the error), or keep β fixed, and from which λ the probability of type I error complies with the threshold.

The alternative hypothesis is:

$$\Delta y = \frac{\partial H}{\partial P}\Delta P + \frac{\partial H}{\partial G}\Delta G + C_y \bigtriangledown + \epsilon_V$$

C_y describes the dimension of $\bigtriangledown$. In the following implementation, we keep β fixed to compute λ. Since the

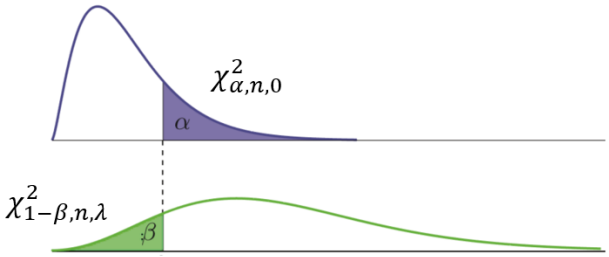


Figure 3.7: Noncentral distribution: computation of type I error probability α and type II error probability β. The probability related to a critical value describes the area of the upper tail of the probability distribution.

feature identification error largely depends on the shape of the global feature, resolution of image, and light conditions, we present two ways to design the H_a model, first consider a feature the additional parameters is 1-vector c_y, when a feature can be clearly identified in one direction, while the measurement of the other direction might be faulty by Teunissen et al. [35]:

$$|\nabla_j| = \sqrt{\frac{\lambda}{c_y^T Q_{yy}^{-1} Q_{\hat{e}\hat{e}} Q_{yy}^{-1} c_y}}$$

with index $j = 1,\ldots,2m$. The variance of residuals $Q_{\hat{e}\hat{e}}$ is computed by the Jacobian of the camera pose parameters by substituting its initial guess P_0,

$$Q_{\hat{e}\hat{e}} = Q_{yy} - J_{p_0}(J_{p_0}^T Q_{yy}^{-1} J_{p_0})^{-1} J_{p_0}^T$$

then find λ recursively until the computed critical value meets the false alarm and the misdetection probability of the H_0 and H_a hypothesis.

$$\chi^2_{\alpha,n,0} = \chi^2_{1-\beta,n,\lambda} \tag{3.19}$$

n is related to the dimension of the error, when ∇ is 1-vector, n = 1. The noncentrality parameter λ has a direct effect on error ∇_y which can contribute to the corresponding observation:

$$\nabla_{y,j} = c_y \nabla_j$$

When we identify the shape of a landmark in the image to obtain its pixel measurements, a fault may happen and the pixel measurements affect in both directions. The additional parameters is 2-vector C_y, since when the global detection procedure misidentified the feature in the image, the measurements are wrong in both

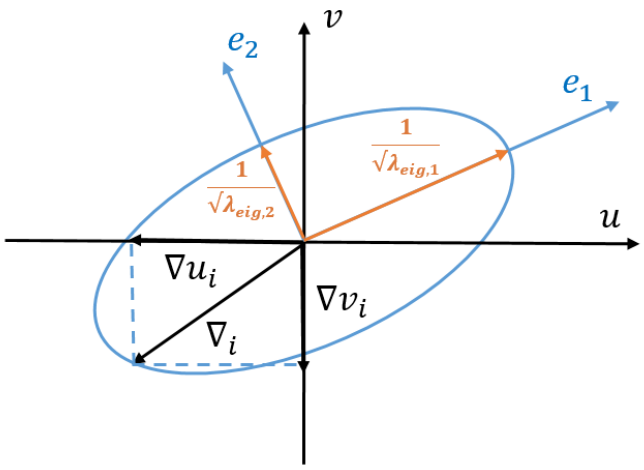


Figure 3.8: The relations between parameters in above equation in Fig.3.21, e_1, e_2 define the direction of major and minor axis of ellipse, the length of the axis are $\frac{1}{\sqrt{\lambda_{eig,1}}}, \frac{1}{\sqrt{\lambda_{eig,2}}}$. A vector ∇_i projects the 2-vector MDB solution with the length of $[\nabla u_i, \nabla v_i]$ on direction of u, v respectively.

u and v, while the measurements of different features are independent. When the observation uncertainty is known, and λ is computed from 3.19 one can obtain size of the error ∇_i:

$$\lambda = \nabla_i^T C_y^T Q_{yy}^{-1} Q_{\hat{e}\hat{e}} Q_{yy}^{-1} C_y \nabla_i \tag{3.20}$$

However, since C_y is 2-dimensional and we could not obtain ∇_i directly from 3.20 which is a 2×1 vector, since it contains two unknowns, $\nabla u_i, \nabla v_i$, we apply the eigenvalue decomposition on the positive definite matrix $A = C_y^T Q_{yy}^{-1} Q_{\hat{e}\hat{e}} Q_{yy}^{-1} C_y$, the decomposition can be described as:

$$A = [e_1 \;\; e_2] \begin{bmatrix} \lambda_{eig,1} & \\ & \lambda_{eig,2} \end{bmatrix} \begin{bmatrix} e_1^T \\ e_2^T \end{bmatrix}$$

e_1, e_2 contains the unit direction vectors corresponding to the major and minor axis of the tilted ellipse. Since we have two unknowns in ∇_i, according to the principal axis theorem, matrix A from equation (3.20) can be represented in fig.3.8 as an ellipse $\nabla_i^T A \nabla_i = \lambda$ (for a fixed value of λ), and substitute by the above equation, the ellipse function is obtained,

$$\frac{(e_1^T \nabla_i)^2}{\frac{1}{(\sqrt{\lambda_{eig,1}})}^2} + \frac{(e_2^T \nabla_i)^2}{\frac{1}{(\sqrt{\lambda_{eig,2}})}^2} = \lambda \tag{3.21}$$

A point lies on the ellipse in fig.3.8, when it satisfies the above equation, and the vector from the center of

the ellipse to the point is $\bigtriangledown_i$, the range of the magnitude of vector $\bigtriangledown_i$ is:

$$\sqrt{\frac{\lambda}{\lambda_{eig,1}}} < \sqrt{\bigtriangledown u_i^2 + \bigtriangledown v_i^2} < \sqrt{\frac{\lambda}{\lambda_{eig,2}}}$$

Since the properties of global feature detection is unknown, we assume the 2-vector MDB can fall into any point onf the ellipse, within the range in the above equation.

3.5.3. Condition number

The condition number is computed by the largest eigenvalue divided by the smallest one of a functional matrix, it states how sensitive the function is subject to changes or perturbations. In later performance analysis, the condition number is labeled as Γ.

4

Simulation and results

This chapter contains the results from the simulation, which gives more insights into the model performance and properties beside those analyzed in Chapter 5, and provides recommendations for the real experiment. The Chapter begins with how to conduct a simulation of the particularly designed landmark geometries and the projection of their pixel coordinates on the image. The monocular vision model demonstrated in Chapter 3 will be tested with simulated data, which contains simulated landmark positions in reality and in the image plane. Theoretical analysis is carried out to evaluate the performance of the monocular vision model, the GNSS model and the integration model from the selected scenarios in this chapter. Additionally, an extended integration model is proposed based on the performance of the integration model, and it is evaluated by the theoretical analysis as well.

4.1. Set-up of simulation

The simulation is based on the data acquisition procedure from a field experiment, with the consideration of different scenarios by forming four types of features geometry in the image. Constants (like focal length, image size and true camera pose) and four distributed landmark position geometries are chosen to simulate the corresponding pixel coordinates (forward model). Estimation of the unknown parameters is achieved by the inverse model, the quantities of interest of the estimators are computed using the true camera pose.

4.1.1. Set-up for focal length, image size and truth

The following camera parameters are set as:

Focal length: $4.9mm$

Image size: 3000×4000 pixels

True camera pose: $\kappa = 0.20rads, X_0 = 2.800m, Y_0 = -0.500m, Z_0 = 1.500m$

Pixel and landmark measurement uncertainty: $\sigma_u = \sigma_v = 2.5[pixels], \sigma_X = \sigma_Y = \sigma_Z = 0.050m$

The focal length and image size are obtained from the manufacturer of the camera, the pose of the camera is close to, but different from the map origin. The uncertainty in vision measurements is small (optimal object recognition is assumed, as well as a distortion-free image). The landmark position error which is set as the same level of HD maps error.

Since observations from the image are all in unit pixels, a conversion is needed to express the focal length in pixels:

$$\text{ppi} = \frac{L_c}{L} = \frac{\sqrt{w_c^2 + h_c^2}}{L}$$

ppi is the abbreviation of pixel per inch, w_c, h_c are the width and height the physical dimension of the image or sensor in pixels, L_c, L are the diagonal size expressed in inches of the image or sensor and the display respectively.

4.2. Simulating landmark positions

Landmark geometry is critical to the position solution, to explore the best performance of the position solution and provide insights for the real experiment. We consider four scenarios with particularly distributed landmark geometries, by inspecting their geometries both in reality Fig.4.1 and in the image plane Fig.4.3: in the 'normal' scenario, the landmarks are well distributed on the image plane; in the 'cluster' scenario, the landmarks are crowed together; in the 'line up in x direction' scenario, the landmarks are aligned around the image center horizontally; in 'line up in y direction', the landmarks are standing vertically at the central line of the image. The scenarios are selected by simulating the extreme cases, and real cases should fall between the cases. By setting up four geometries, one can inspect performance and properties of the quantities of interest, and further, make suggestions for the map database construction.

However, to generate the landmark position coordinates in reality more reasonably, several constraints are considered: landmarks are all standing in front of the camera within a certain angle of view; the simulated pixel coordinates computed from the constants and simulated landmarks must within the image size; and some landmarks are of the same height.

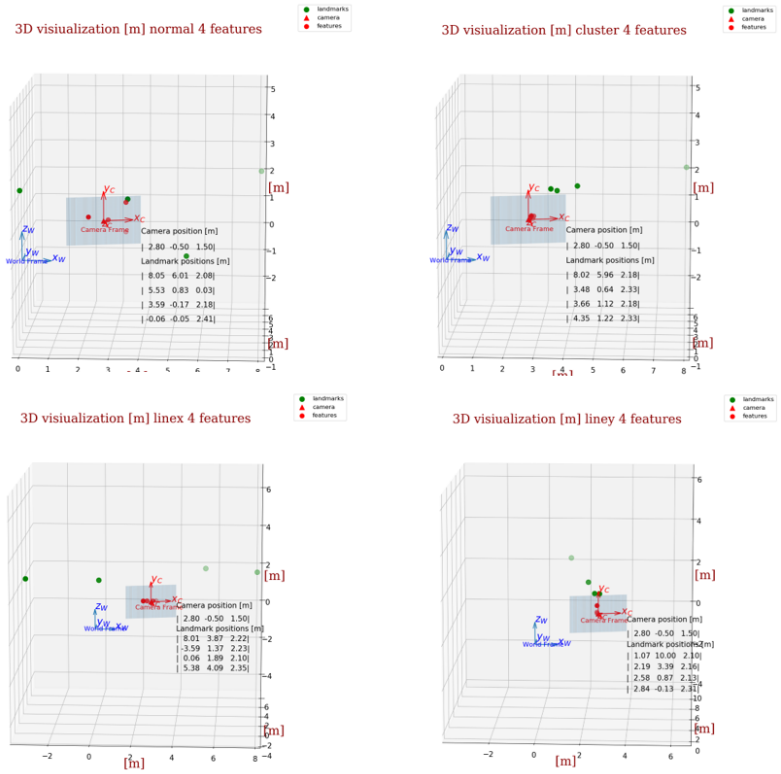


Figure 4.1: 3D plot of four landmark geometries in reality. The green dots are landmarks in the world frame (blue), red ones are the projected features of the green ones on image plane in the camera frame (red).

All the geometries are obtained by changing landmark coordinates in X and Y, leaving the coordinates in Z as constant. Scenarios with less landmarks or feature cases are obtained by discarding the first landmark and then the first and second landmark, the changed geometry in reality is included in Appendix.A.

4.3. Simulating pixel coordinates

After setting up constants and landmark positions by constraints from different geometries, a series of functions are called to compute pixel coordinates of the projected landmarks on the image by forward modeling. In Fig.4.2, the forward model is to use landmark position coordinates and constants to produce simulated pixel coordinates, the inverse model employs pixel coordinates measurements and constants to estimate the unknowns, and to produce results of the simulated data as well as the data collected from a real experiment, which is elaborated in Chapter 5.

4.3.1. Function flow chart to simulate pixel measurements

From Fig.4.2, the pixel coordinates are computed using the forward model, its conversion to the image plane is not necessary in the experiment of simulated landmark position coordinates and pixel coordinates. The following results are computed by simulated pixel coordinates through the inverse model. It is worth not-

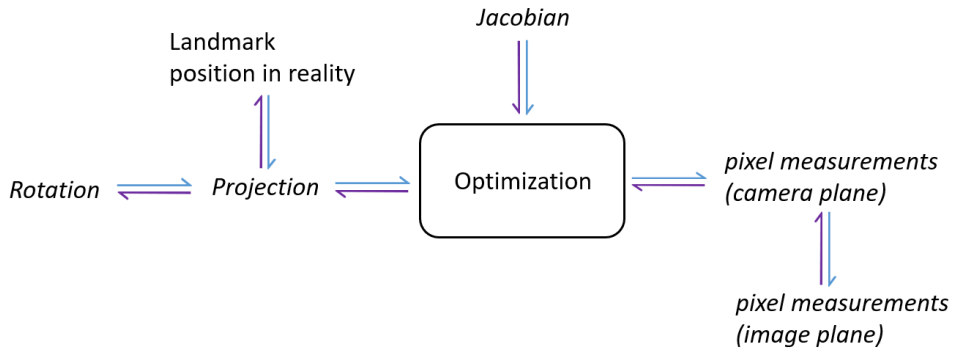


Figure 4.2: Process of forward model (blue arrow) and inverse model (purple arrow). The defined functions are set in italics.

ing that in Chapter.5, the observations are obtained in the image plane so conversions are indispensable to enter the inverse model and carry out parameter estimation. Fig.4.2 displays the flow of functions, and the description for each function is:

$Jacobian$: compute Jacobian matrix of the target functions with respect to the unknowns;

$Projection$: project landmark position coordinates in reality onto camera plane;

$Rotation$: computation of concatenated rotations from world frame to camera coordinates;

$pixel\ \ measurements$: conversion of coordinates of camera plane to the image plane or the other way around.

4.4. Results and discussion

The results from this chapter are based on theoretical analysis, by inspecting the MDBs, correlation coefficients and standard deviations of the estimators.

Theoretical analysis aims to reveal the optimal performance of testing and estimation that can be achieved in the monocular vision model with different simulated geometries. The MDBs are computed by 1-vector and 2-vector cases as we explained in Section.3.5.2, by varying from 4 features to 3 features under each scenario, because the minimal detectable bias computation terminates at 2-feature cases since there is no redundancy (4 measurements and 4 unknown parameters to estimate). The precision and coefficients are interpreted from the variance matrix $Q_{\hat{P}\hat{P}}$ which is computed from J_{P0}, W, the deduction and steps are explained in Section.3.5.1 and Section.3.4 respectively.

The experimental analysis is based on an iterative procedure with an initial guess for the camera pose, by varying from 4 features to 2 features under each scenario, this full processing operation is adopted in Chapter.5.

To further interpret these approaches, one can see from Figure.4.4, different from the experimental anal-

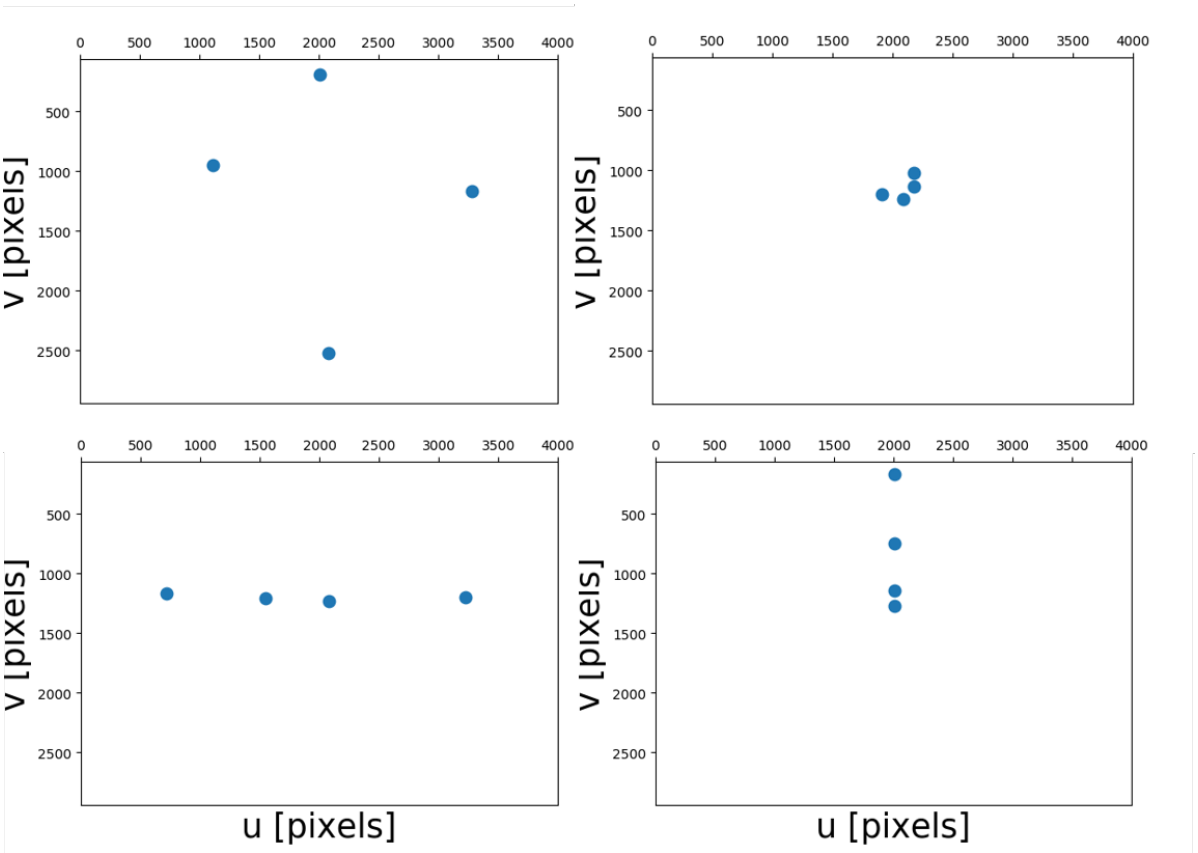


Figure 4.3: Geometry of 4 selected features on the image plane of four scenarios, with the origin of the image plane on the top left. Top left: normal, top right: cluster, bottom left: line up in x, bottom right: line up in y.

Group	Theoretical analysis	Experimental analysis
Input	• True camera pose	• Initial camera pose
$Q_{\hat{P}\hat{P}}$	• Method I: $W = \left(Q_{yy} + J_{g,0}^{T} W J_{g,0}\right)^{-1}$ Method II: $W = Q_{yy}^{-1}$ • $Q_{\hat{P}\hat{P}} = J_{P,0}^{T} W J_{P,0}$	• Method I: $W_{i-1} = \left(Q_{yy} + J_{g,i-1}^{T} W J_{g,i-1}\right)^{-1}$ Method II: $W_{i-1} = Q_{yy}^{-1}$ • $\Delta y_{i-1} = y - H(P_{i-1}, G)$ • $\Delta\hat{P}_{i-1} = \left(J_{P,i-1}^{T} W_{i-1} J_{P,i-1}\right)^{-1} J_{P,i-1}^{T} W_{i-1} \Delta y_{i-1}$ • $P_i = P_{i-1} + \Delta\hat{P}_{i-1}$ • $Q_{\hat{P}\hat{P},i} = J_{P,i-1}^{T} W_{i-1} J_{P,i-1}$

Figure 4.4: Comparison between theoretical analysis and experimental analysis: the input and procedure.

Group	Vision(Method I)	Vision(Method II)	GNSS	Integration	Extended Integration
MDB	• 1-vector and 2-vector • 4 feature • 3 feature	• 1-vector and 2-vector • 4 feature • 3 feature	• 1-vector • 8 satellites • 5 satellites	• 1-vector • All measurements • 3 satellites, 2 features	• 1-vector • All measurements • 2 satellites, 2 features
$Q_{\hat{P}\hat{P}}$	• 4 feature • 3 feature • 2 feature	• 4 feature • 3 feature • 2 feature	• 8 satellites • 5 satellites	• All measurements • 2 satellite 2 features • 4 satellite 1 feature	• All measurements • 3 satellite 1 feature

Figure 4.5: Illustration of the different input for the vision models (Method I and Method II), GNSS and integration model for the theoretical analysis. One feature implies two pixel measurements in the vision model, one satellite denotes its pseudorange measurement that enters GNSS and Vision model. The input measurements are chosen by the redundancy of the model, MDB requires at least one redundancy, the estimates can be determined without redundancy.

ysis that the theoretical analysis is iteration- free and uses the true camera pose as the input, while the later one uses an initial camera pose as the input to an iterative process to compute the estimators.

Figure.4.5 shows the different sources of input used to produce the results in Chapter 4, the results are presented in this order for each model.

4.4.1. Vision

Method I

The deduction of Method I is described in Section.3.2.2: the pixel measurements are the computed observations and the unknowns are X_0, Y_0, Z_0, κ. We use the true camera pose as the input in the theoretical analysis to show the optimal performance of quantities of interest of the model as well as the estimators.

Reliability According to the explanation on Section.3.5.2, the MDB depends on the chosen level of significance, the design matrix J, vector c or C, and the observation variance matrix Q_{yy}. It gives the minimal value of bias that can be detected by satisfying the probability of false alarm α and misdetection error β under two hypothesis. If $\bigtriangledown$ has a larger value, it means that the performance of validation of the corresponding observation is poor, and the observation has the potential for the system to produce seriously biased estimators (effect of the MDB on the parameters). It can also be interpreted that, if the observation fault is beyond the MDB, the system is able to detect it under the current hypothesis. For 2-vector MDB (only computed for vision), the larger range and values indicate the corresponding feature observation is bad. Unlike the 1-vector MDB, the method we proposed in Chapter 3 is unable to describe the fault in which direction in the image.

1-vector MDB computed from 4 features In the 1-vector MDB for 4 features case (table.4.1), Δu_3 in normal geometry and Δv_1 in line up in y direction have larger values.

Table 4.1: Method I: 1-vector MDB for 4 features unit [pixels] by true camera pose without observations uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
Δu_1	19.93	18.76	45.27	54.98
Δv_1	14.04	16.16	16.29	92.53
Δu_2	15.45	22.94	30.44	18.27
Δv_2	23.58	16.03	15.84	15.24
Δu_3	108.90	35.65	15.55	16.58
Δv_3	44.89	16.09	15.31	17.57
Δu_4	18.02	18.76	68.73	15.53
Δv_4	14.17	20.99	17.41	17.02

2-vector MDB computed from 4 features In the 2-vector MDB for the 4 features case (table.4.2), the third feature in normal geometry, fourth feature in line up in x, and first in line up in y direction have larger values and larger ranges, which complies with the conclusion in the 1-vector MDB computation and these values are larger.

Table 4.2: Method I: 2-vector MDB for 4 features unit [pixels] by true camera pose without observations uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
$(\Delta u_1, \Delta v_1)$	(21.26,14.91)	(19.96,17.19)	(49.63,17.26)	(58.49,98.44)
$(\Delta u_2, \Delta v_2)$	(16.43,25.09)	(27.04,16.34)	(32.57,16.83)	(19.43,16.22)
$(\Delta u_3, \Delta v_3)$	(124.46,47.23)	(42.72,16.76)	(16.55,16.29)	(17.64,18.70)
$(\Delta u_4, \Delta v_4)$	(19.24,15.04)	(19.49,23.05)	(99.33,16.07)	(16.53,18.11)

1-vector MDB computed from 3 features In the 1-vector MDB for 3 features case (table.4.3), Δu_3 in normal geometry, Δu_3 and Δu_4 in line up in x direction, Δv_4 in line up in y direction have relatively larger values.

Table 4.3: Method I: 1-vector MDB for 3 features unit [pixels] by true camera pose without observations uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
Δu_2	18.35	23.20	103.83	26.83
Δv_2	35.57	17.34	17.26	17.75
Δu_3	267.45	36.05	236.69	16.58
Δv_3	62.11	17.13	16.45	20.74
Δu_4	21.24	63.80	144.36	43.43
Δv_4	14.34	25.79	16.02	133.74

2-vector MDB computed from 3 features In the 2-vector MDB for 3 features case (table.4.4), the third feature in normal geometry, fourth feature in cluster, all features except for the first one in line up in x and fourth feature in line up in y direction have larger values, the number of larger MDB increases obviously.

Table 4.4: Method I: 2-vector MDB for 3 features unit [pixels] by true camera pose without observations uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
$(\Delta u_2, \Delta v_2)$	(18.54,49.23)	(28.25,17.34)	(114.13,18.34)	(28.54,18.88)
$(\Delta u_3, \Delta v_3)$	(437.76,65.07)	(43.87,17.76)	(305.07,17.48)	(17.64, 22.07)
$(\Delta u_4, \Delta v_4)$	(22.59,15.26)	(303.03,25.53)	(447.70,16.95)	(46.21, 142.29)

Precision and correlation coefficient The diagonal values, for instance in table.4.5, represent the standard deviations of the estimators, whose units are: [degree,m,m,m]. The rest (off diagonal) are the correlation coefficients between the corresponding estimators. Γ_0 is the condition number computed from the Jacobian with respect to the initial value for the camera pose P_0.

$Q_{\hat{P}\hat{P}}$ **computed from 4 features** The unknown parameters computed in the normal geometry can be estimated with good precision, especially a standard deviation of 0.04 degree for the camera azimuth κ, and $0.22m$ for the X_0, Y_0 and Z_0 coordinates of the camera. The correlation coefficients between the unknowns range up to 0.01 which means they are nearly uncorrelated. Under the cluster-like geometry, one can observe that the standard deviation of κ and Y_0 degrade, the correlation coefficients of κ and the rest of other estimators increase, which means the setting is relatively worse in telling κ from the other estimators. Y_0 has the largest standard deviation among the three direction. The line up in x geometry gives degraded standard deviation in κ and X_0, and the correlation coefficients between κ and X_0, κ and Y_0. X_0 has the largest standard deviation in three directions. The line up in y geometry produces results with decent and comparable quantities of interest with regard to the normal geometry. Judging by the condition number Γ_0, system of the cluster geometry is most sensitive to small perturbation to the data, and the easiest to become numerically ill-conditioned.

Table 4.5: The interpreted $Q_{\hat{P}\hat{P}}$ of 4-feature simulated geometries from Method I: true camera pose without observation uncertainties.

	Normal				**Cluster**				**Line up in x**				**Line up in y**			
κ	0.04	0.01	-0.00	-0.00	0.20	0.07	0.15	0.02	0.20	0.12	0.05	0.00	0.04	0.01	0.00	0.00
X_0		0.22	-0.00	-0.00		0.22	0.00	-0.0		0.23	0.00	0.00		0.22	-0.00	-0.00
Y_0			0.22	0.00			0.23	0.01			0.22	0.00			0.22	0.00
Z_0				0.22				0.22				0.22				0.22
Γ_0	12.84				173.78				79.68				29.57			

$Q_{\hat{P}\hat{P}}$ **computed from 3 features** After discarding one feature, all geometries are able to solve the solution of X_0, Y_0, Z_0 with the standard deviation within $0.23m$. For normal geometry, all correlation coefficients are fairly small. The 3-feature cluster geometry, standard deviation of κ increases to 0.22 degree, the correlation coefficients of κ and Y_0, κ and Z_0 slightly decrease, but still fairly low, this is because the features in cluster geometry are very close to each other, the discarded feature can help the system differentiate the estimators from each other. For the line up in x geometry, the standard deviation of κ and Y_0 degrades a little. The standard deviation of κ in line up in y increases. From the condition number, one can see that the normal geometry remains the least to be numerically ill-conditioned of the four geometry settings.

Table 4.6: The interpreted $Q_{\hat{P}\hat{P}}$ of 3-feature simulated geometries from Method I: true camera pose without observation uncertainties.

	Normal				**Cluster**				**Line up in x**				**Line up in y**			
κ	0.05	0.01	0.01	0.00	0.22	0.09	0.12	0.01	0.30	0.14	0.14	0.01	0.08	0.03	0.01	0.00
X_0		0.22	-0.00	-0.00		0.22	-0.00	-0.00		0.23	0.02	0.00		0.22	-0.00	-0.00
Y_0			0.22	0.00			0.23	0.01			0.23	0.00			0.23	0.00
Z_0				0.22				0.22				0.22				0.22
Γ_0	12.32				151.59				109.75				77.60			

$Q_{\hat{P}\hat{P}}$ **computed from 2 features** By inspecting the solutions from the 2-feature normal geometry, the standard deviation of κ degrades to 0.09 degree, although the correlation coefficients of κ and Y_0, κ and Z_0 increase, the values for the normal geometry are fairly small. The 2-feature under cluster geometry produces estimators κ and Y_0 up to 0.22 degree and 0.25 m respectively. The correlation coefficients of κ and Z_0, κ and Y_0, Y_0 and Z_0 increase but these values are still show that the estimators are less correlated. Judging by the quantities of interest, the line up in x direction has worst performance. The standard deviation of κ meets 1.87 degree, and the standard deviation for X_0 and Y_0 are up to 0.3 m. All of the correlation coefficients increase, especially for κ and X_0, κ and Y_0, X_0 and Y_0. The line up in y geometry can have comparable quantities of interest with respect to the normal one. The standard deviation of κ increases to 0.15 degree, the standard deviation of the rest of estimates are below 0.23 m. The condition number increase in all geometries, and line up in x increases significantly.

Table 4.7: The interpreted $Q_{\hat{p}\hat{p}}$ of 2-feature simulated geometries from Method I: true camera pose without observation uncertainties.

	Normal				**Cluster**				**Line up in x**				**Line up in y**			
κ	0.09	0.00	0.05	0.02	0.22	0.08	0.17	0.02	1.87	0.69	0.65	0.04	0.15	0.05	0.01	0.00
X_0		0.22	-0.00	-0.00		0.22	-0.01	-0.00		0.31	0.45	0.03		0.22	-0.01	-0.00
Y_0			0.22	0.00			0.25	0.03			0.30	0.03			0.23	0.01
Z_0				0.22				0.22				0.22				0.22
Γ_0	34.24				199.12				571.97				79.97			

Method II

Different from Method I, which uses pixel measurements to estimate camera pose, the observations and unknowns of Method II also include landmark positions, therefore, initial values for landmark positions are also needed.

Reliability analysis

1-vector MDB computed from 4 features Compared with table.4.1, the values presented by Method II can be one order larger than the those computed by Method I, but one can still see that Δu_3 in normal geometry, Δu_1 and Δv_1 in line up in y have larger MDBs.

Table 4.8: Method II: 1-vector MDB for 4 features unit [pixels] by true camera pose without observation uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
Δu_1	293.76	215.87	399.34	940.07
Δv_1	164.43	180.30	151.57	1498.99
Δu_2	218.78	226.90	270.94	209.19
Δv_2	308.34	154.78	146.27	142.98
Δu_3	1563.13	353.23	137.52	179.37
Δv_3	815.26	144.29	132.90	196.09
Δu_4	266.43	215.63	594.25	288.51
Δv_4	210.53	226.25	125.42	288.01

2-vector MDB computed from 4 features Third feature in normal, fourth feature of line up in x, first feature in line up in y have larger values and ranges.

1-vector MDB computed from 3 features The number of MDBs that have larger values increases. Δu_3, Δv_3 in normal geometry, Δu_3, Δu_4 in line up in x, and Δv_4 in line up in y have relatively larger values, and it has different pattern compare with the 4-feature case.

Table 4.9: Method II: 2-vector MDB for 4 features unit [pixels] by true camera pose without observation uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
$(\Delta u_1, \Delta v_1)$	(315.17,174.49)	(229.74,191.78)	(439.20,160.52)	(1000.17, 1594.81)
$(\Delta u_2, \Delta v_2)$	(232.75,328.09)	(259.16,159.83)	(288.78,155.53)	(222.57, 152.12)
$(\Delta u_3, \Delta v_3)$	(1757.33,855.31)	(435.66,150.33)	(146.35,141.36)	(190.83, 208.63)
$(\Delta u_4, \Delta v_4)$	(293.31,219.51)	(218.37,255.77)	(901.60,131.95)	(306.97, 306.40)

Table 4.10: Method II: 1-vector MDB for 3 features unit [pixels] by true camera pose without observation uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
Δu_2	265.47	230.87	852.21	305.23
Δv_2	498.44	163.96	155.10	168.50
Δu_3	4126.37	359.99	1945.63	188.69
Δv_3	1026.86	153.92	142.09	196.94
Δu_4	303.11	631.50	1185.25	494.19
Δv_4	226.36	260.05	132.46	1269.86

2-vector MDB computed from 3 features Besides the faulty features shown from 1-vector case, fourth feature of cluster also hsow larger error.

Table 4.11: Method II: 2-vector MDB for 3 features unit [pixels] by true camera pose without observation uncertainties.

	Normal	**Cluster**	**Line up in x**	**Line up in y**
$(\Delta u_2, \Delta v_2)$	(262.64,791.77)	(263.39,168.97)	(942.50,164.82)	(324.75, 179.27)
$(\Delta u_3, \Delta v_3)$	(6025.91,1076.97)	(442.01,160.15)	(2620.54,151.03)	(200.75,209.53)
$(\Delta u_4, \Delta v_4)$	(330.49,237.68)	(2735.79,256.95)	(4018.31,140.14)	(525.78,1351.05)

Precision and correlation coefficient

$Q_{\hat{P}\hat{P}}$ **computed from 4 features** Compared with table.4.5, the standard deviation of κ and correlation coefficients degrades significantly. The normal scenario yields results of the best performance, with a standard deviation of 0.76 degree in κ estimation, 7 cm in X_0, Y_0 and 3 cm in Z_0, the correlation between κ and X_0 is close to 1. Then follows the performance of line up in y, with larger correlation coefficients in κ and X_0, X_0 and Y_0, X_0 and Z_0, Y_0 and Z_0. The standard deviation of X_0, Z_0 is lower than 6 cm, Y_0 has a standard deviation of 0.19 m, and 0.55 degree in κ estimation. Line up in x produces Y_0, Z_0 with an acceptable precision below 0.14 m, while X_0 has a precision of 0.24 m and κ has a precision of 1.87 degree. The correlation coefficient of X_0 and κ is nearly 1. The cluster scenario has worst performance among the four, not only the precision of estimators are worse, with 1.84 degree of κ, 0.19 m of X_0, 0.08 m of Z_0 and 0.66 m of Y_0, but also the larger correlation coefficients in κ and X_0, Y_0 and Z_0. From the condition number, one can see that systems for all geometry settings are numerically ill-conditioned compare with the results from Method I.

Table 4.12: The interpreted $Q_{\hat{p}\hat{p}}$ of 4-feature simulated geometries from Method II: true camera pose without observation uncertainties.

	Normal				**Cluster**				**Line up in x**				**Line up in y**			
κ	0.76	0.91	-0.11	-0.07	1.84	0.83	0.45	0.39	1.87	0.99	0.75	0.14	0.55	0.70	0.05	0.00
X_0		0.07	-0.26	-0.1		0.19	-0.11	-0.14		0.24	0.69	0.11		0.06	-0.56	-0.50
Y_0			0.07	0.23			0.66	0.94			0.14	0.31			0.19	0.84
Z_0				0.03				0.08				0.03				0.03
Γ_0	13585.11				85464.73				37438.10				27974.52			

$Q_{\hat{p}\hat{p}}$ computed from 3 features By discarding one feature, results from all geometries degrade, since the correlation coefficients are computed by covariance over standard deviation, some of the correlation coefficients are improved is because of the increased standard deviations. The normal geometry is able to meet an requirement for the estimators in 1.07 degree and 10 cm. Line up in x is capable of obtaining standard deviation of X_0, Y_0, Z_0 within one thirds of a meter while 2.75 degree in κ estimation. Cluster and line up in y have worse standard deviation in Y_0 among the three directions of the local reference frame, which are of 0.67 m and 0.42 m respectively.

Table 4.13: The interpreted $Q_{\hat{p}\hat{p}}$ of 3-feature simulated geometries from Method II: true camera pose without observation uncertainties.

	Normal				**Cluster**				**Line up in x**				**Line up in y**			
κ	1.07	0.85	0.38	0.02	2.06	0.86	0.33	0.25	2.75	0.97	0.87	0.41	0.79	0.67	0.04	0.00
X_0		0.07	0.03	-0.07		0.23	-0.18	-0.23		0.30	0.75	0.31		0.12	-0.67	-0.63
Y_0			0.10	0.23			0.67	0.93			0.33	0.57			0.42	0.90
Z_0				0.03				0.08				0.04				0.07
Γ_0	13368.14				75341.37				50139.31				48830.46			

$Q_{\hat{p}\hat{p}}$ computed from 2 features The 2-feature case table.4.14 shows the same pattern as the 3-feature case but degrades further. Line up in x geometry has the worst performance, the standard deviation of κ, X_0, Y_0 reach 15.42 degree, 1.77 m and 1.59 respectively, with all correlation coefficients close or equal to 1.

Table 4.14: The interpreted $Q_{\hat{p}\hat{p}}$ of 2-feature simulated geometries from Method II: true camera pose without observation uncertainties.

	Normal				**Cluster**				**Line up in x**				**Line up in y**			
κ	1.64	0.29	0.77	0.66	2.13	0.76	0.39	0.35	15.42	1.00	0.99	0.87	1.99	0.79	0.05	0.00
X_0		0.08	-0.30	-0.34		0.24	-0.29	-0.31		1.77	0.99	0.86		0.20	-0.54	-0.55
Y_0			0.29	0.90			1.04	0.96			1.59	0.89			0.58	0.94
Z_0				0.09				0.13				0.08				0.11
Γ_0	32656.24				96558.74				234835.60				55775.98			

Summary

Summary of results from Method I and Method II functional model evaluated by 1-vector and 2-vector MDB: 1-vector and 2-vector MDBs computed from Method I and Method II show similar patterns, and the pattern becomes obscure when the feature number is reduced; Method II's performance is less stable than Method I's by the number of large MDBs , their MDBs show different patterns.

Cluster and line up in x direction have overall smaller MDB based on 1-vector and 2-vector computation.

Less features give larger MDB in both methods, which contributes to less validation between the observations.

The 1-vector MDB has overall smaller MDB than the 2-vector one, it also matches the assumption we applied in developing functional models of Method I and Method II: the pixel measurements uncertainties are independent.

Summary of results from Method I and Method II functional model evaluated by $Q_{\hat{P}\hat{P}}$: Theoretically, the results from Method I and Method II are the same when we deal with a linear problem.

Method I's performance is more stable than Method II's by judging the smaller value of standard deviation and correlation coefficients of the unknowns.

Normal geometry has overall better performance based on evaluation of standard deviation and correlation coefficients of estimators by the theoretical analysis, then followed by line up in y direction, cluster and finally, line up in x.

Normal geometry has overall stabler performance, since it less sensitive to feature number reduction, line up in y geometry has the comparable performance, cluster geometry is subject to the feature changes significantly among all geometry settings.

Each geometry has its own interesting properties so selection of an experiment setup depends on the requirement on the results.

4.4.2. GNSS

To reduce the computational load of the integration model, carrier phase observations are excluded , since one carrier phase observation brings one more ambiguity term to the unknowns when using a single epoch of data. The redundancy of the integration model by taking carrier phase observations remains the same, therefore, including carrier phase as the observations does not contribute to the estimators of interest especially by taking data at the start of SF-PPP.

Reliability The possible faults for GNSS observations are computed by the 1-vector MDB.

MDB computed from all available satellite pseudorange observations. The MDB computed for 8 pseudorange observations varies from 4.56 m to 24.89 m. Among them, the sizes of the errors $\bigtriangledown$ for the second, fourth and fifth pseudorange observations have larger values.

Table 4.15: True camera pose with observations computed from 8 satellites.

Pseudorange	**MDB [m]**
Δp^1	11.21
Δp^2	24.89
Δp^3	10.23
Δp^4	17.63
Δp^5	20.99
Δp^6	5.08
Δp^7	4.56
Δp^8	7.77

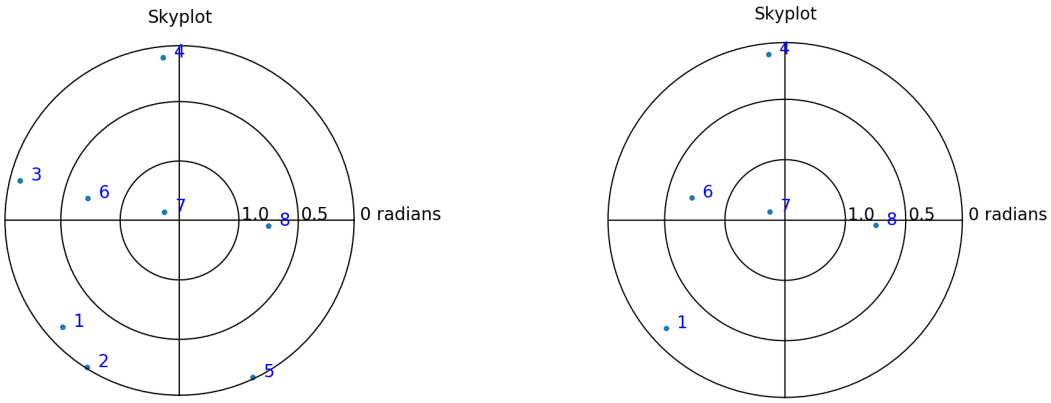
MDB computed from 5 satellite pseudorange observations. The pseudorange observations are selected based on elevations by a descending order. Since the overall input observations are less, the values of the MDB increase, especially for the first one, MDB from table.4.15 increases by more than 2 times. However, the sixth and seventh observation have small values and the increments are small, which means the two observations are more reliable and can be verified with the rest of the input observations.

Table 4.16: True camera pose with observations computed from 5 satellites, selected by descending elevations.

Pseudorange	**MDB [m]**
Δp^1	29.16
Δp^4	30.35
Δp^6	5.59
Δp^7	4.77
Δp^8	10.19

Precision and correlation coefficients

$Q_{\hat{p}\hat{p}}$ **computed from all available and 5 satellite pseudorange observations.** The unit of X_0, Y_0, Z_0, dt_r is [m]. The precision of X_0, Y_0, Z_0, dt_r ranges from 0.78 m to 1.36 m, with relatively high correlation between Z_0 and dt_r. This because the satellites are far away from the Earth surface (local reference frame), and they are all on one side of the receiver, namely above it. Hence they all contribute to the estimation of the local height component (Z-coordinate) through the sine of their elevation angle (all larger than 5 or 10 degrees), and, they all contribute by a coefficient of one to the estimation of the receiver clock error. The impact of the Z-coordinate and the receiver clock error on the range observations is pretty much the same, and thereby



(a) Skyplot of all available satellites.

(b) Skyplot of satellites of top 5 elevations.

Figure 4.6: Skyplot of 8 and 5 satellites at the start of SF-PPP data processing.

these two parameters are hard to separate, leading to a large correlation between them. Ultimately if all satellites were in the local zenith (sine of elevation angle equal to one), it would not be possible to separate them at all. Inspect the satellite geometry from Fig.4.6, one can see the satellite are well distributed, which is reponsible for the fair correlation between the other estimates. By reducing the number of available satellite pseudorange observations, the precision of unknowns all degrades.

Table 4.17: The interpreted $Q_{\hat{P}\hat{P}}$ computed from pseudorange observations of 8 and 5 satellites, X_0, Y_0, Z_0 are transformed into world frame.

	8 satellites				**5 satellites**			
X_0	0.78	0.06	0.36	0.41	0.83	0.05	0.27	0.31
Y_0		1.14	0.07	0.04		1.25	0.06	0.09
Z_0			1.58	0.97			1.91	0.98
dt_r				1.36				1.68

Skyplot Fig.4.6 shows the skyplot of all available satellites over the experiment site at the start of SF-PPP data processing. One the left, one can see the satellites are well distributed but some of them have small elevations.

Estimation of GNSS From table.4.18, position estimation for the start of SF-PPP data processing when using all satellite pseudorange observations has an error 1.10 m in X_0, 1.73 m in Y_0 and -0.94 m in Z_0. When using satellites pseudorange observations of top 5 elevations, only X_0 degrades slighly, this is because the position estimates rely heavily on the observations with largest elevations, which means the less noisy observations. Noisy low elevation observations contribute only slightly to the position estimates.

Summary of GNSS analysis:

Table 4.18: Error of the estimation of vision model, units are [m,m,m].

Scenarios	X_0	Y_0	Z_0
8 satellites	1.10	1.73	-0.94
5 satellites	1.13	1.73	-0.94

Summary of results from the GNSS evaluated by 1-vector MDB: By reducing the number of satellite pseudorange observations, while the number of unknown parameters remain the same, the MDB values increase.

Summary of the GNSS model $Q_{\hat{p}\hat{p}}$ The high correlation coefficients between dt_r and Z_0 is because of the property of GNSS positioning.

By reducing the number of satellite pseudorange observations from 8 to 5, the standard deviation of unknowns degrades by 20%, but they are still within 1.91 m.

4.4.3. Integration

The integration aims to combine the output from the two sensors, SF-PPP of the GNSS receiver and Method I of the vision model by feeding landmark positions of the normal geometry. From the previous section, vision results from Method I by using landmarks in normal geometry has the best and the most stable performance, which means that the following results aim to present the best performance of the integration model.

Reliability The so-called 1-vector MDB is computed for two scenarios: 1. sufficient observations: when satellite observations and pixel measurements are sufficient and both sensors can function individually, therefore, 8 satellite pseudorange observations and 4 detected visual landmarks are considered as observations; 2. insufficient observations: 3 satellite observations and 2 features are regarded as the input for the integration model, which aims to simulate when GNSS fails while vision alone just still functions. Cases in practice should fall in between the two scenarios.

MDB computed for all satellite pseudorange and pixel observations. Compare table.4.15 with table.4.19, the MDB values decrease due to adding the vision system, which improves the estimates performance significantly (table.4.21). What's more, the pattern is clearer by integrating the vision system since the integration can detect errors better in GNSS. The vision MDBs of table.4.1 remain the same due to less contribution of the GNSS system in the integration model.

MDB computed from 3 satellite pseudorange and 2-feature pixel observations. By omitting 5 satellite pseudorange observations and 2 landmark pixel measurements, the MDBs of all observations increase, especially for the vision ones, the value increase up to 3 order larger, since the estimates performance relies

Table 4.19: True camera pose with observations computed from 8 satellites and 4 landmarks.

Pseudorange	**MDB [m]**	**u**	**MDB [pixels]**	**v**	**MDB [pixels]**
Δp^1	7.94	Δu_1	19.93	Δv_1	14.04
Δp^2	24.27	Δu_2	15.45	Δv_2	23.58
Δp^3	9.73	Δu_3	108.90	Δv_3	44.89
Δp^4	9.45	Δu_4	18.02	Δv_4	14.17
Δp^5	19.40				
Δp^6	3.99				
Δp^7	4.44				
Δp^8	3.95				

much on the vision part, and the vision system couldn't be verified by GNSS observations due to the its relatively poor performance compared to vision's. Δp^6 and Δp^8 have the smallest MDB which is different with table.4.16, this is because the GNSS part is much constrained by the more precise vision.

Table 4.20: True camera pose with observations computed from 3 satellites selected from 3 highest elevations and 2 landmarks.

Pseudorange	**MDB [m]**	**u**	**MDB [pixels]**	**v**	**MDB [pixels]**
Δp^6	5.23	Δu_3	6909.59	Δv_3	10665.39
Δp^7	35.43	Δu_4	6771.01	Δv_4	3350.69
Δp^8	4.55				

Precision and correlation coefficients The following results are computed by normal geometry, according to vision simulation results, the normal scenario has the best performance, so that one can explore the optimal performance of integration model. The ways to select pixel measurements and pseudorange observations follow the same rules in previous sections.

Due to the observability of the system, besides the analysis of taking all available measurements into account table.4.21, the integration model we introduced in Chapter 3 can handle cases when GNSS or monocular vision individually fails, which are presented as table.4.23 (vision fails) and table.4.22 (GNSS fails).

$Q_{\hat{p}\hat{p}}$ computed from all available pseudorange observations and pixel measurements. Scenario with sufficient satellite and pixel observations: the unit of $X_0, Y_0, A_0, t_0 dt_r$ is [m], κ is in degree. With sufficient satellite pseudorange and pixel observations, the standard deviation of X_0, Y_0, κ can meet 0.22 m, 0.22m, 0.04 degree as one can see in table.4.21. The standard deviation of Z_0 is 1.48 m, and the correlation coefficients between the three estimators Z_0, t_0, dt_r are close to 1. The system is unable to tell estimators Z_0 from t_0 from each other, as we explained above that GNSS system can not tell Z_0 from dt_r, the integration system is unable to tell the height offset between the camera and the GNSS antenna t_0 from dt_r either, because the antenna is mounted on top of the camera.

Table 4.21: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 8 satellites and pixel observations of 4 features.

	Normal					
X_0	0.22	0.00	0.11	0.10	0.01	0.12
Y_0		0.22	0.01	0.01	-0.00	0.00
Z_0			1.48	0.99	0.00	0.97
t_0				1.49	0.00	0.96
κ					0.04	0.00
dt_r						1.25
Γ_0	11740.13					

$Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 2 satellites and pixel measurements of 2 features. The scenario describes when GNSS fails and vision alone can produce a solution. The $Q_{\hat{p}\hat{p}}$ shows the same pattern as table.4.21, the standard deviations of Z_0, t_0, κ, dt_r degrade further.

Table 4.22: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 2 satellites and pixel observations of 2 features.

	Normal					
X_0	0.22	-0.00	-0.16	-0.16	0.00	-0.16
Y_0		0.22	0.02	0.02	0.05	0.02
Z_0			6.24	1.00	0.00	1.00
t_0				6.25	0.00	1.00
κ					0.09	-0.00
dt_r						5.87
Γ_0	89626.22					

$Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 4 satellites and pixel measurements of 1 feature. The scenario describes when Vision fails and GNSS alone can produce a solution. The standard deviation of X_0, Y_0, κ increase, especially for κ, since it can only be determined from vision. Compare with table.4.22, taking more pixel measurements can improve the standard deviation and correlation coefficients of X_0, Y_0, κ significantly.

Table 4.23: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 4 satellites and pixel observations of 1 features.

	Normal					
X_0	0.91	0.38	0.49	0.50	0.69	0.50
Y_0		3.47	0.84	0.80	0.92	0.84
Z_0			4.60	1.00	0.84	1.00
t_0				4.21	0.82	0.99
κ					23.16	0.85
dt_r						4.06
Γ_0	48194.95					

4.4.4. Extended integration

The integration model we presented has difficulty in differentiate Z_0 from dt_r, and can handle cases when one of the sensors fails. The extended integration model assumes t_0 as given, it is no longer included as one of the unknowns, therefore, this method eliminates the correlation between Z_0 and t_0 and expands its application for when GNSS and vision both fail.

Reliability

MDB computed by all satellite pseudorange and pixel observations. Compare table.4.24 with table.4.19, we observe that the GNSS observations have smaller MDBs by leaving out t_0, GNSS results show that t_0 has strong correlation between Z_0 and dt_r, which results in bigger values in residual variance matrix $Q_{\hat{e}\hat{e}}$. However, MDBs computed by vision remains the same which indicates that the vision observations barely contribute to t_0 estimation.

Table 4.24: True camera pose with observations computed from 8 satellites and 4 landmarks.

Pseudorange	**MDB [m]**	**u**	**MDB [pixels]**	**v**	**MDB [pixels]**
Δp^1	6.97	Δu_1	19.93	Δv_1	14.04
Δp^2	23.58	Δu_2	15.45	Δv_2	23.58
Δp^3	8.63	Δu_3	108.90	Δv_3	44.89
Δp^4	8.36	Δu_4	18.02	Δv_4	14.17
Δp^5	18.59				
Δp^6	3.98				
Δp^7	3.61				
Δp^8	3.92				

MDB computed by 2 satellite pseudorange and 2-feature pixel observations. According to table.4.25, the MDBs of vision increased significantly, the MDBs of the seventh and the eighth satellite pseudorange observations, who are of top 2 high elevation angles, increase slightly compare to the vision result, this is because the vision precision is better than the GNSS's, therefore, vision observations contribute more in unknowns estimation and have less validation from the GNSS observations.

Table 4.25: True camera pose with observations computed from 2 satellites selected from descending elevations and 2 landmarks.

Pseudorange	**MDB [m]**	**u**	**MDB [pixels]**	**v**	**MDB [pixels]**
Δp^7	4.48	Δu_3	11358.61	Δv_3	22592.17
Δp^8	4.48	Δu_4	11635.81	Δv_4	10562.68

Precision and correlation coefficients

$Q_{\hat{p}\hat{p}}$ computed from all available pseudorange observations and pixel measurements. Compare with table.4.5, the first 4 columns have reached the same performance as the vision simulation result, the last column, which describes the correlation coefficients between dt_r and the other unknowns and the standard deviation of dt_r (last one) significantly decline.

Table 4.26: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 8 satellites and pixel observations of 4 features.

	Normal				
X_0	0.21	0.00	0.02	0.01	0.07
Y_0		0.22	0.00	-0.00	-0.02
Z_0			0.22	-0.00	0.49
κ				0.04	0.00
dt_r					0.37
Γ_0	2493.12				

$Q_{\hat{p}\hat{p}}$ computed from 3 satellite pseudorange observations and 1 pixel measurements. This situation describes when the two sensors both fail to produce a positioning solution. Although the extended integration model is able to produce a positioning solution, the result is far off from the truth, the standard deviation of the unknowns are large, especially for Y_0 and κ. The correlation coefficients between the unknowns are close to 1.

Table 4.27: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 3 satellites and pixel observations of 1 feature.

	Normal				
X_0	2.97	0.96	0.95	0.99	0.94
Y_0		18.92	1.00	0.98	0.98
Z_0			3.19	0.97	0.99
κ				18.33	0.96
dt_r					2.13
Γ_0	134474.35				

Summary of results from the integration and the extended integration model evaluated by 1-vector MDB: It is clearer to observe the pattern on the GNSS part because the integration model performance is improved by the vision part, which enables to perform validations of the GNSS peudorange observations better.

The integration and extended integration model can improve the GNSS system internal reliability (decreased MDB values) by reducing the correlation between X_0 and dt_r, Z_0 and dt_r.

Summary of the integration and the extended integration model $Q_{\hat{p}\hat{p}}$ The integration model is able to produce a positioning solution when one of the two sensors fails, theoretically, the extended integration

model is able to produce a solution when the both sensors fail.

The integration model is able to estimate the height offset t_0 of between the camera and antenna, with high correlation coefficients with another two unknowns X_0 and dt_r, since vision has little contribution to the t_0 estimation. In the extended integration, t_0 is treated as a known constant, and its effect is folded in Z_0 estimation.

Vision measurements can significantly improve the precision of estimators, when satellite pseudorange observations are insufficient. Adding satellite pseudorange observations can only slightly improve the precision of the estimators, because the estimators standard deviation obtained by vision is much smaller than by GNSS, which is also verified by KIM and N.-H [20]'s experiment.

Compared with GNSS alone positioning, the integration method can reduce the standard deviation of X_0, Y_0 by 86%; the extended integration method can further reduce the standard deviation of dt_r by 78%, and the correlation between Z_0 and dt_r.

5

Practical experiment and results

Based on the simulation results, a real experiment is designed and performed to experience performance of the developed model closer to a real application. In this chapter, a description of a small field-experiment is given, then followed by visualization, analysis and discussion of the results from the vision, GNSS and integration model with data obtained from experiment. The method used to evaluate the model performance is based on the experimental analysis.

5.1. Experiment setup

Based on the prototype of the experiment design in Fig.3.5, the GNSS antenna is mounted on top of the camera with an height offset, the geometry of the landmarks is chosen as the normal one according to the simulation results, and are measured by total station, which is also the origin of the local map, different but close to the camera. In order to determine the local north precisely, one need an extra GNSS receiver to compute the north correction angle, which describes the difference between the measured north and the corrected north. The facilities used in the experiment include: total station, reflectors, camera, GNSS receiver (2 high-end; 1 automotive type). The overview of the field experiment is shown in Fig.5.1.

Several steps need to be followed sequentially to obtain and process the measurements:

1. download IGS real-time corrections for SF-PPP;
2. take down weather conditions; picture of surroundings; experiment setup;
3. take measurements of two high-end GNSS receivers, to determine the origin of the world frame in the

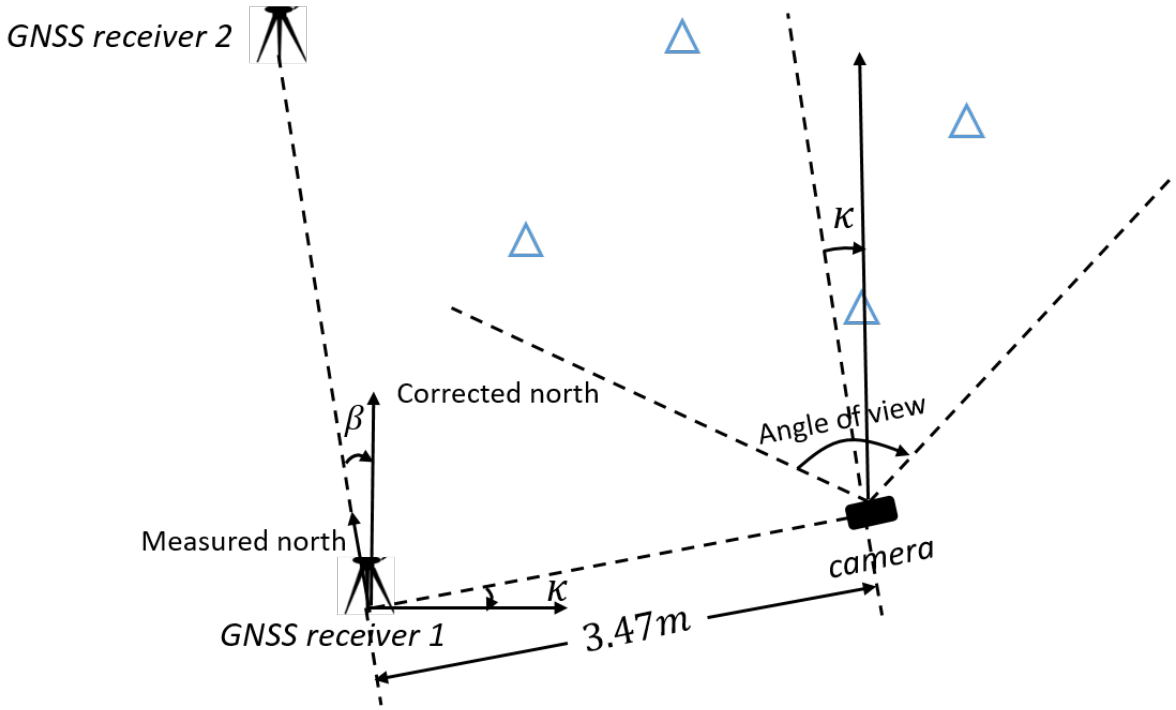


Figure 5.1: Design of experiment setup. GNSS receiver 1 occupies the origin of the world frame, its north can be related to the ECEF frame by measuring the relative position between the two GNSS receivers. Once the position measurements of two high-end GNSS receiver are successfully collected, move GNSS receiver 2 to the camera position, and swap GNSS receiver 1 with the total station. The camera is set up locally horizontal and is aligned with respect to the world frame surveyed with the total station and is around 3.5 m away from the GNSS receiver 1, κ is obtained by total station. The landmarks (blue) are placed within the angle of view and a maximum range of 8 m.

ECEF frame, and to determine the azimuth of the baseline between the two in ECEF. Measure camera position and κ by total station;

4. set landmarks within the camera's angle of view, preferably in normal or line up in y geometry, with one on the ground. Measure their positions using the total station in world frame, take several pictures;

5. on the same camera position but different height, mount low-cost GNSS antenna, take measurements for 30 minutes, also log the data.

Constants set-up

Constants of the camera parameters from manufacturer:

Focal length: $4.9mm$

Image size: 3000×4000 pixels

True camera pose: $\kappa = 3.48rads, X_0 = -3.192m, Y_0 = -1.360m, Z_0 = 1.408m$ (used for validation)

The pixel measurements are obtained manually on an image with distortion, and the landmark positions are measured by the total station. The uncertainties for pixel and landmark measurement are: $\sigma_u = \sigma_v = 20[pixels], \sigma_X = \sigma_Y = \sigma_Z = 0.05m$

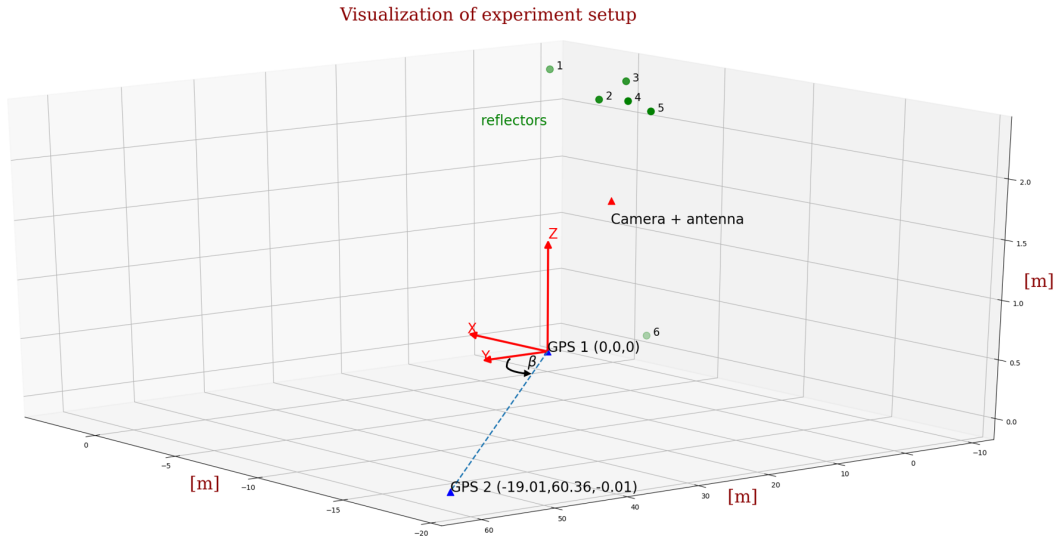


Figure 5.2: Visualization of reflectors (landmarks, as green dots), GNSS receiver (GPS1 and GPS2), and camera in the field experiment. β describes the angle from measured north and corrected north in the world frame (red), so that the Y-aixs in the world frame is pointing North (according to ECEF).

Visualization

The reflectors' coordinates, camera coordinates and camera orientation κ in the world frame are measured by the total station occupying the same position as GNSS1 by aligning the north with respect to GNSS2. The north correction angle β is computed by initializing the null direction of the total station at GNSS1 by pointing to GNSS2 in the world frame. Fig.5.2 shows the visualization of the two GNSS receivers, camera and reflectors of the experiment site after applying the north correction angle β. Fig.5.4 is the picture of reflectors taken by the camera, the pixel measurements are obtained by manually selecting the geometric center of the reflectors in the image.

5.2. Results and discussion

5.2.1. Vision

The results in this chapter are evaluated by applying experimental analysis presented in Figure.4.4 on the experimental data in the vision, GNSS, and integration model.

From the previous results, Method I has better and more stable performance over Method II, therefore, the performance of the vision and integration model is based on Method I in this chapter.

Reliability analysis The MDBs are computed by varying the number of reflectors from 6 to 3 reflectors as the input pixel measurements to the vision model. The reduced number of reflectors is achieved by removing the reflectors coordinates and pixel measurements from the start of their label sequence.

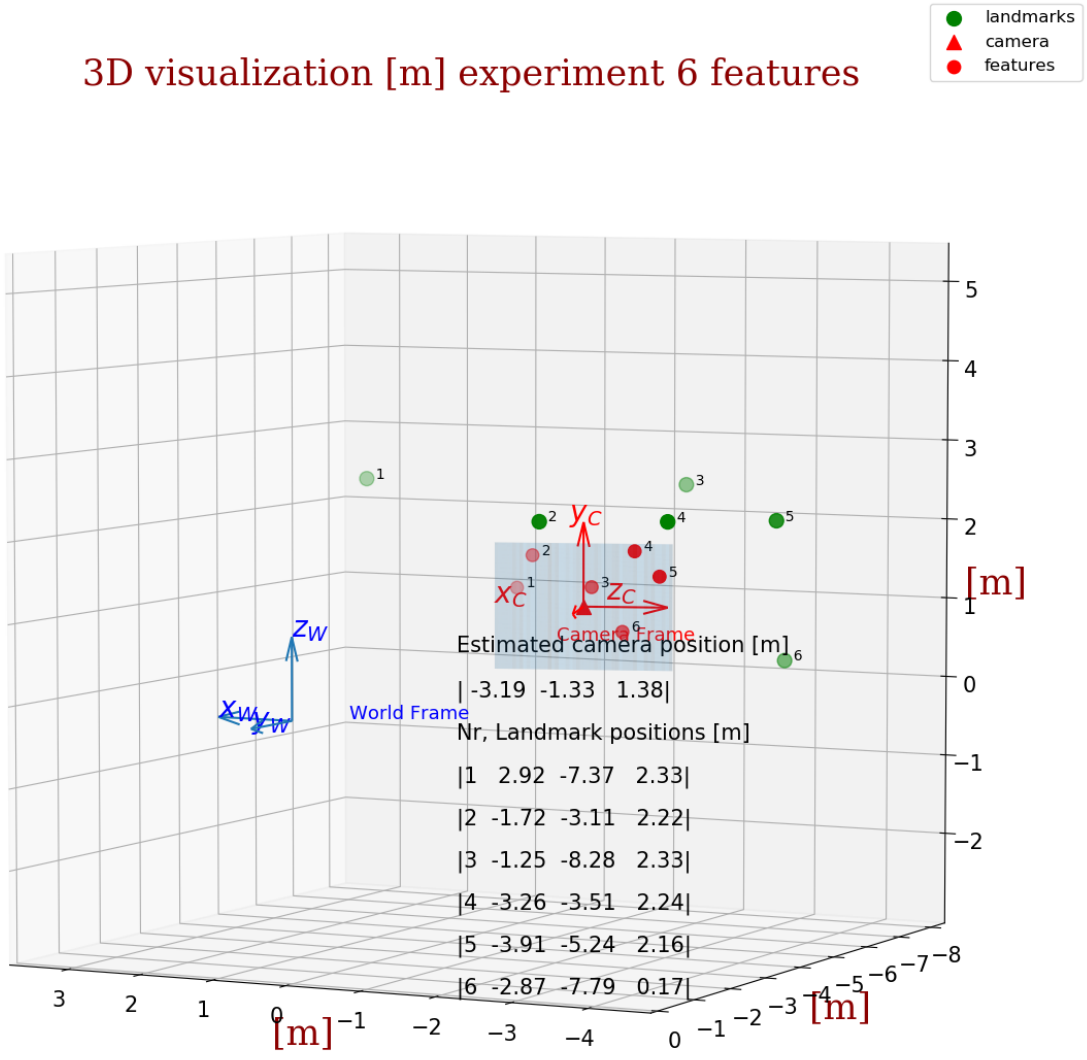
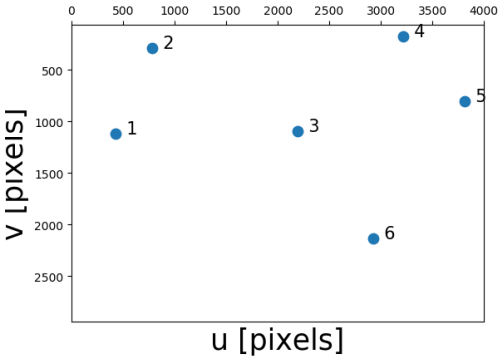


Figure 5.3: 3D plot of landmarks (reflectors) of the experiment site by viewing at camera orientation. The blue one is the world frame, the red one is the camera frame, the reflectors (green) are mapped in reality, on the image plane (grey) and presented in table sequentially.



(a) Visualization of experiment setup.

(b) Image taken by single camera

Figure 5.4: Visualization of reflectors (landmarks) in the experiment setup, simulated pixel measurements (a) and pixel measurements from camera (b). In (a), the simulated pixel measurements are computed here for visualization from reflectors coordinates in world frame and camera pose. (b) shows the feature geometry in the image, the pixel measurements are obtained by manually selecting the geometric center of the reflectors in the image. The experiment was carried out at TU Delft; Green Village.

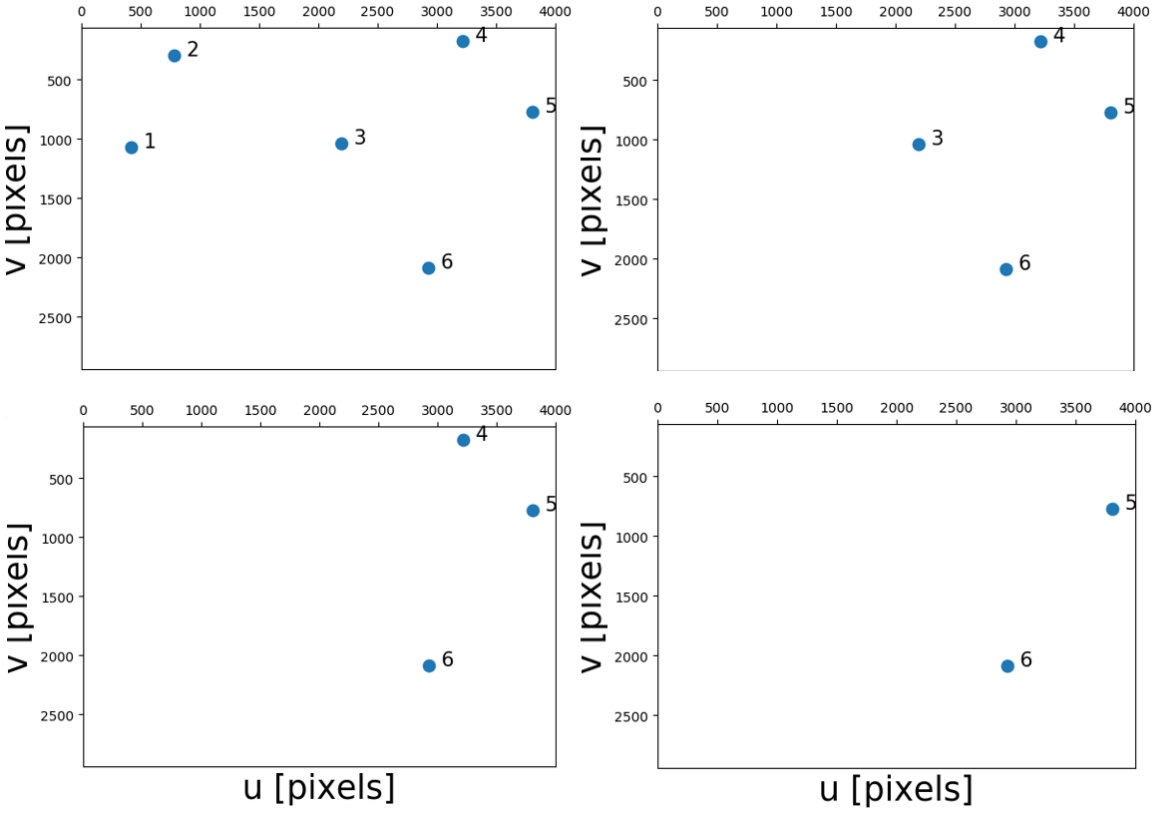


Figure 5.5: Reflectors geometry on image plane. The results are computed for different scenarios by varying the number of reflectors. From top left, top right, bottom left to bottom right are: 6, 4, 3 and 2 features in the image plane

Group	Vision (Method I)	GNSS (start of the filter)	GNSS (filter converged)-Appendix C	Integration	Extended Integration

MDB	• 1-vector and 2-vector
	  • 6 feature
	  • 5 feature
	  • 4 feature
	  • 3 feature
		• 1-vector
		  • 13 satellites
		  • 5 satellites
			• 1-vector
			  • 14 satellites
			  • 5 satellites
				• 1-vector
				  • All measurements
				  • 3 satellites, 2 features
					• 1-vector
					  • All measurements
					  • 2 satellites, 2 features

$Q_{\tilde{P}\tilde{P}}$	• 6 feature • 4 feature • 3 feature • 2 feature	• 13 satellites • 5 satellites	• 14 satellites • 5 satellites	• All measurements • 2 satellite 2 features • 4 satellite 1 feature	• All measurements • 3 satellite 1 feature

Figure 5.6: Illustration of the different input for the vision model, the GNSS and the integration model for the experimental analysis. One feature implies two pixel measurements in the vision model, one satellite denotes its pseudorange measurement. The results of GNSS (filter converged) are included in Appendix C but are discussed in this Chapter, as a comparison with the results of GNSS (start of the filter).

From Table.5.1, Δu_2 has larger values in the 1-vector MDB when having 6 and 5 reflectors in the view. As the number of reflectors reduced to 4 and 3, Δu_4, Δv_4 and Δv_6 are rising since less available pixel measurements can verify it.

Table 5.1: 1-vector MDB computed by varying reflector numbers from 6 to 3. Unit: [pixels]

	6 reflectors	**5 reflectors**	**4 reflectors**	**3 reflectors**
Δu_1	92.38			
Δv_1	56.21			
Δu_2	162.63	181.86		
Δv_2	75.20	81.20		
Δu_3	58.43	70.12	75.25	
Δv_3	55.27	55.52	58.88	
Δu_4	86.36	96.90	195.87	232.94
Δv_4	65.96	67.00	140.78	155.60
Δu_5	63.10	65.04	66.56	72.07
Δv_5	57.69	58.88	62.26	64.49
Δu_6	58.27	65.41	67.09	93.65
Δv_6	57.83	59.17	84.08	106.78

For the 2-vector MDB, Table.5.2 shows a similar trend with the 1-vector MDB but they are presented now in pairs. The two tables show the fifth measurements can always be verified best by the left pixel measurements when gradually omitting pixel measurements.

Table 5.2: The 2-vector MDB for pixel measurements computed by varying the number of reflector from 6 to 3. Unit: [pixels]

	6 reflectors	**5 reflectors**	**4 reflectors**	**3 reflectors**
$(\Delta u_1, \Delta v_1)$	(98.42,59.77)			
$(\Delta u_2, \Delta v_2)$	(180.29,79.34)	(202.85,85.62)		
$(\Delta u_3, \Delta v_3)$	(62.17,58.80)	(74.70,59.01)	(80.21,62.57)	
$(\Delta u_4, \Delta v_4)$	(92.16,70.05)	(103.13,71.27)	(245.80,139.96)	(261.81,161.84)
$(\Delta u_5, \Delta v_5)$	(67.49,61.10)	(69.24,62.61)	(71.79,65.47)	(77.30,68.17)
$(\Delta u_6, \Delta v_6)$	(62.90,60.68)	(69.98,62.67)	(71.37,89.46)	(89.81,135.78)

Precision and correlation coefficient analysis The precision and correlation coefficients are computed for different scenarios by changing the number of reflectors on the input. Units of the estimates κ, X_0, Y_0, Z_0 are [degree, m, m, m].

From table.5.3, when having 6 reflectors in the image, the estimates can reach the best performance with an standard deviation of 0.13 degree in κ, and 0.22 m in X_0, Y_0, Z_0. The estimates can have acceptable accuracy of estimates when the number of reflectors is reduced to 3, with fair correlation coefficients between the estimates. When the number of reflectors reduces to 2, the standard deviation of estimates degrades to 0.59 degree in κ, and 0.23 m in X_0 and Y_0, while Z_0 can have a standard deviation of 0.22 m, with relatively high correlation between κ and X_0. Under the experiment setting, the X_0, Y_0, Z_0 can be solved within 0.23 m by regardless the number of reflectors, while the standard deviation of κ is more subject to the number of the input vision observations.

Table 5.3: The interpreted $Q_{\hat{x}\hat{x}}$ computed from measurements of 6,4,3,2 reflectors. Unit: [degree, m, m, m]

	6 reflectors				**4 reflectors**				**3 reflectors**				**2 reflectors**			
κ	0.13	-0.03	-0.01	-0.00	0.19	-0.05	-0.02	0.01	0.22	-0.05	0.01	-0.00	0.59	-0.23	-0.12	0.02
X_0		0.22	-0.00	0.00		0.22	0.00	-0.00		0.22	0.00	-0.00		0.23	0.04	-0.01
Y_0			0.22	-0.00			0.23	-0.00			0.23	-0.01			0.23	-0.01
Z_0				0.22				0.22				0.22				0.22
Γ	9.57				21.62				21.70				38.33			

Estimation of vision To determine the actual estimation error, the true camera pose values of Section 5.1 are used. According to Table.5.4, by reducing the number of reflectors of vision, the position error is below 0.16 m, the accuracy of κ is more subject to the change of number of reflectors.

Table 5.4: Error of the estimation of vision model, units are [degree,m,m,m].

Scenarios	κ	X_0	Y_0	Z_0
6 reflectors	-0.10	0.00	0.03	-0.02
5 reflectors	-0.19	0.02	0.15	-0.06
3 reflectors	0.04	0.01	0.14	-0.06
2 reflectors	-0.86	0.09	0.16	-0.07

5.2.2. GNSS

The GNSS results are computed from the start and the end of the SF-PPP data processing, which are included in this Chapter and in Appendix C respectively. To analyze and visualize the results of SF-PPP data processing, one needs to deal with the Earth tide effect during the experiment time, since SF-PPP position solutions are computed in ITRF2008 and the receiver is subject to Earth tides motion in this frame. It is more intuitive to have the mean position solution, with the Earth tide effect removed by a correction. In the experiment,

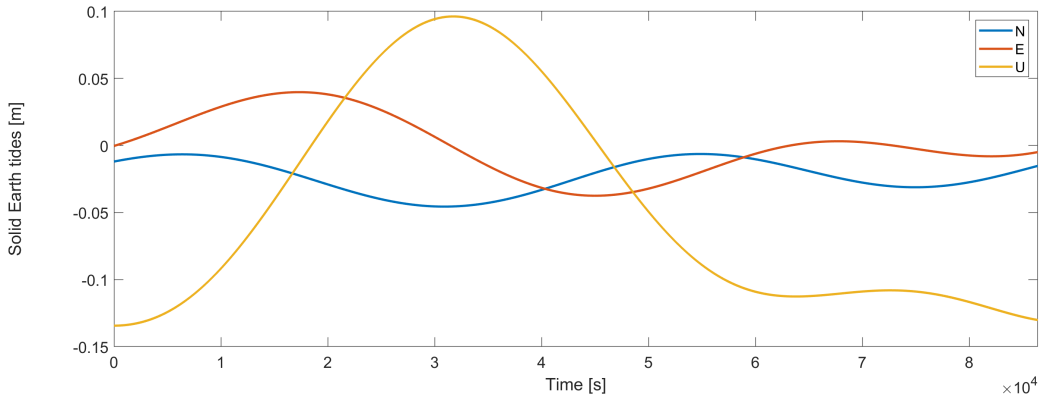


Figure 5.7: Position variation (X_0, Y_0, Z_0) caused by earth tide in ITRF2008 on July 25^{th} for DLF1 permanent GNSS receiver in Delft.

the GPS and total station measurements are in ETRF2000, and to convert the SF-PPP position solutions of camera in the world frame, some transformations and correction computations are required and is explained in Appendix.B.

Solid earth tide correction For precise point positioning, it is necessary to include the solid earth tide effect which describes the response of the Earth to lunisolar gravitational attraction. Fig.5.7 shows the position variations caused the solid earth tide, and the variations can reach +/-13 cm in one day.

Frame transformation To transform the global position solutions from SF-PPP to the world frame, two steps are required in our case: 1. ITRF2008 to ETRF2000 2. ETRF2000 to world frame. The first one is computed by the position offset of the permanent reference station DLF1 in the two reference frames, since the DLF1 provides corrections to GNSS1 and GNSS2 and the baseline is short, the offset is regarded as a constant vector and can be applied to GNSS1, GNSS2 and the SF-PPP position solutions. The reference frame transformation is briefly described in Appendix.B. The second step follows the same procedure and is elaborated in Chapter 3, the GNSS1 position is set as the origin of the world frame, and ECEF coordinates are converted into the world frame.

After removing the solid earth tide effect during the experiment time and applying transformations to the SF-PPP solutions, Fig.5.8 shows the position error of SF-PPP position solutions (carrier phase measurements are included here). One can see that the filter converges at 1500 s, and a gap occurs at 3400 s, the gap is probably caused by some new satellite observations that came into the observations at 3400 s and then the filter converges at 4800 s again.

Reliability

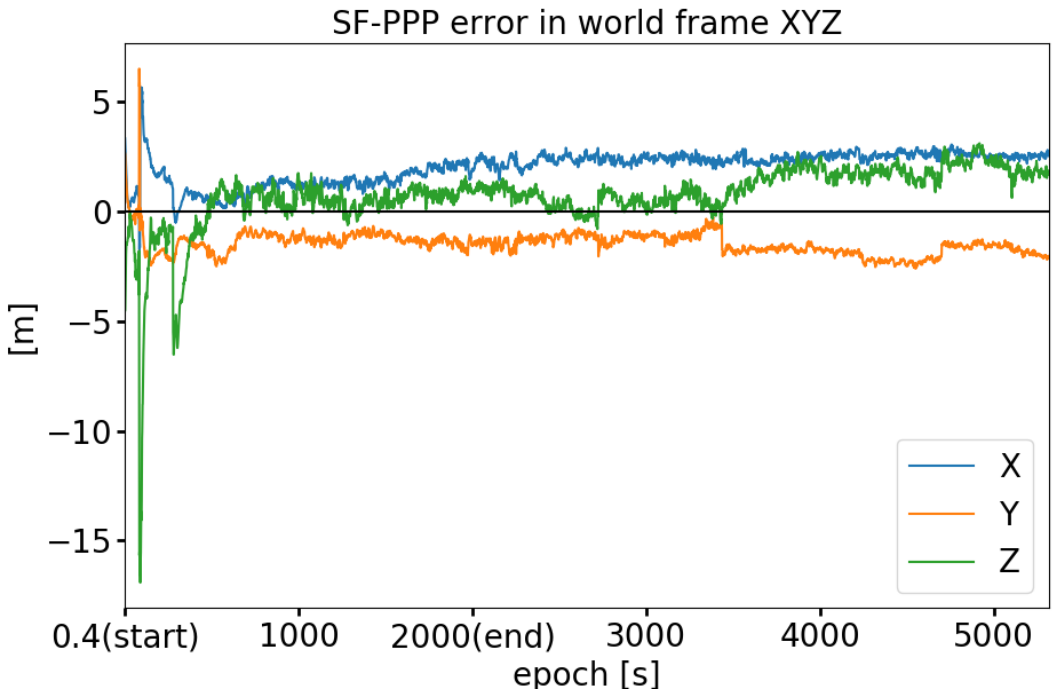
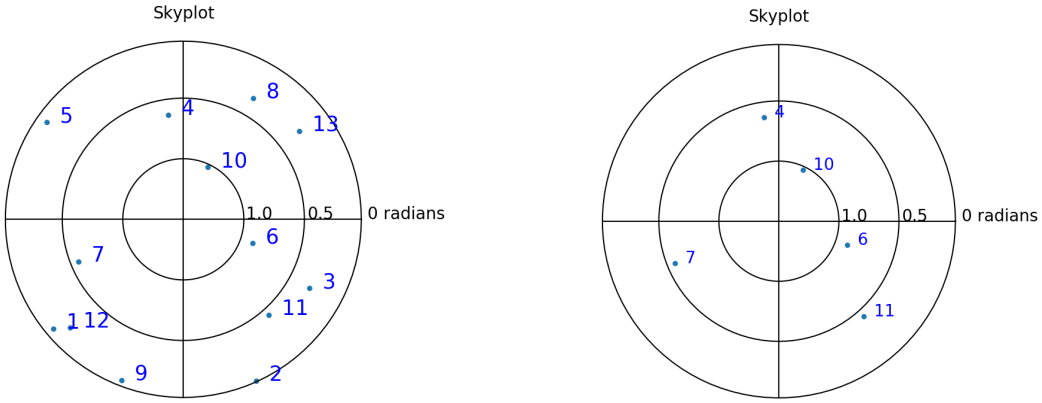


Figure 5.8: Position error (X_0, Y_0, Z_0) under world frame processed by SF-PPP. Results in this chapter are based on data of epoch 0.4s and those in Appendix C are based on data of epoch 2000s.

MDB computed for all available satellite pseudorange observations. From table.5.5, the second and ninth pseudorange observations have larger MDB values. The MDB computed for the observations when only the 5 with highest elevations are selected, especially for the seventh pseudorange observation, which is one order larger than the others. Table.C.1 in Appendix C is computed based on the SF-PPP data after filter convergence, the MDB values increase as the number of the input pseudorange observations decreases, the trend of MDB is different by the changing elevations which have changed during the filter converging time.

Table 5.5: MDBs computed for 13 satellites and 5 satellites by using the experimental data (at the start of SF-PPP data processing).

Pseudorange	**MDB [m]**	**MDB [m]**
Δp^1	11.39	
Δp^2	27.78	
Δp^3	7.48	
Δp^4	5.21	26.16
Δp^5	10.35	
Δp^6	5.26	10.76
Δp^7	6.46	295.18
Δp^8	6.64	
Δp^9	32.14	
Δp^{10}	9.02	10.19
Δp^{11}	12.19	27.50
Δp^{12}	17.51	
Δp^{13}	14.61	



(a) Skyplot of all available satellites.

(b) Skyplot of satellites of top 5 elevations.

Figure 5.9: Skyplot of 13 and 5 satellites at the start of SF-PPP data processing.

Precision and correlation coefficients

$Q_{\hat{p}\hat{p}}$ computed from 13 and 5 pseudorange observations. The $Q_{\hat{p}\hat{p}}$ computed with all available pseudorange observations shows that X_0 has the smallest standard deviation which is 0.64 m, Z_0 has the largest standard deviation which is 1.79 m. Besides the high correlation of Z_0 and dt_r, the high correlation coefficients of X_0 and dt_r is caused by the fact that the satellites are concentrated on one side in the skyplot (namely on the right-hand, or East side). By reducing the pseudorange observations down to 5, the standard deviation and correlation coefficients all degrade, especially for the standard deviation and correlation between X_0 and Z_0, the selection based on elevation causes the satellites to be even more concentrated on one side in the skyplot. Table.C.1 in Appendix C also shows the same pattern but with different values, however, after the selection of top 5 elevations as input, the satellites are largely clustered on one side in the skyplot.

Table 5.6: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 13 and 5 satellites, X_0, Y_0, Z_0 have been transformed into the world frame.

	13 satellites				**5 satellites**			
X_0	0.64	0.02	0.20	0.89	1.41	0.05	0.76	0.76
Y_0		0.69	0.04	0.00		1.00	0.07	0.08
Z_0			1.79	0.97			8.21	1.00
dt_r				1.35				6.93

Skyplot Fig.5.9 shows the skyplot of all available satellites over the experiment site at the start of SF-PPP data processing. On the left, one can see the satellites are well distributed, on the tight, the seventh satellite is far away from the others, whose MDB shown in table.5.5 has a larger value.

Estimation of GNSS From table.5.7, position estimation of the start of SF-PPP data processing when using all satellite pseudorange observations has a error of 3.07 m in X_0, 1.66 m in Y_0 and 2.91 m in Z_0. When using

satellites pseudorange observations of top 5 elevations, Z_0 degrades significantly, since the selection reduces the variation in vertical direction, and it increases the difficulty for GNSS to estimate Z_0 together with the receiver clock error from the selected pseudorange observations.

Table 5.7: Actual error of the estimation of GNSS model, units are [m,m,m].

Scenarios	X_0	Y_0	Z_0
13 satellites	3.07	1.66	2.91
5 satellites	2.44	4.62	-17.70

5.2.3. Integration

Reliability

MDBs computed for all available observations from both sensors in integration model Compare table.5.8 with table.5.1, Δu_1, Δu_3, Δu_5 and Δu_6 decrease, and the values of Δu_2 and Δu_4 are larger. The increased redundancy caused by combining GNSS and vision results improved validation of the vision observations. Compare table.5.8 with table.5.5, the MDBs are smaller since the vision solution is more precise and reliable than the GNSS solution. One can observe the same by comparing table.C.4 in Appendix C with the other two tables.

Table 5.8: MDBs computed for all available observations from both sensors.

Pseudorange	**MDB [m]**	u	**MDB [pixel]**	v	**MDB [pixel]**
Δp^1	10.56	Δu_1	65.54	Δv_1	54.75
Δp^2	27.42	Δu_2	175.13	Δv_2	70.03
Δp^3	5.44	Δu_3	59.48	Δv_3	54.79
Δp^4	4.10	Δu_4	119.58	Δv_4	73.56
Δp^5	9.53	Δu_5	66.28	Δv_5	58.61
Δp^6	4.53	Δu_6	60.44	Δv_6	56.62
Δp^7	4.17				
Δp^8	5.15				
Δp^9	31.94				
Δp^{10}	8.98				
Δp^{11}	11.86				
Δp^{12}	17.21				
Δp^{13}	14.30				

MDB computed for 3 satellites (3 highest elevations ones selected) and 2 landmarks By leaving out 10 pseudoranges of lower elevations, the MDBs of the vision part increase by to 2 of magnitude,which means the integration model relies heavily on the vision system.

Precision and correlation coefficients

Table 5.9: MDB computed for 3 satellites of 3 highest elevations and 2 landmarks.

Pseudorange	**MDB [m]**	**u**	**MDB [pixels]**	**v**	**MDB [pixels]**
Δp^4	28.94	Δu_5	6534.39	Δv_5	18115.06
Δp^6	7.25	Δu_6	3910.00	Δv_6	9623.59
Δp^{10}	9.67				

$Q_{\hat{p}\hat{p}}$ computed by all available observations from both sensors. The integration model can have a position solution with a standard deviation of 0.21 m in X_0 and Y_0, 1.77 m in Z_0 and t_0, 0.13 degree in κ and 1.34 m in dt_r. The system has difficulty in telling Z_0 from t_0 and $d_t r$, whose correlation coefficients are close to 1. Compared with the 13-satellite case in table.5.6, the dependence between X_0 and dt_r is reduced, this is because by adding the vision system under the experiment setting, the vision has better geometry and variations in X-Y than X-Z in the world frame, hence, the integration model can better tell X_0 from dt_r, but behaves worse in telling Z_0 from dt_r and t_0.

Table 5.10: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 13 satellites and pixel observations of 6 features.

	13 satellites and 6 features					
X_0	0.21	0.00	0.07	0.07	-0.03	0.06
Y_0		0.21	0.01	0.01	-0.01	-0.00
Z_0			1.76	0.99	-0.00	0.97
t_0				1.77	-0.00	0.97
κ					0.13	-0.00
dt_r						1.34
Γ_0	15792.98					

$Q_{\hat{p}\hat{p}}$ computed for limited situations, when one of the sensors fails. When GNSS fails, but vision operates, the estimates have better performance than the 5-satellites GNSS only case presented in table.5.6, except for the precision of Y_0. The estimates have acceptable precision, however, estimation of Z_0 is 'contaminated' by GNSS observations. When GNSS works but vision fails, all quantities of interest degrade significantly, especially for Z_0, t_0, dt_r, which are around 18.07 m, with the coefficients between the estimates close to 1, except for Y_0 with the others. Table.C.7 in the Appendix C has better performance for the second scenario, which means the second scenario heavily on the quality of satellite pseudorange observations.

Estimation of integration From table.5.12, the position error of the integration model increases significantly when GNSS operates but vision fails. Insufficient vision measurements lead to poor results of all unknowns, especially for κ, X_0 and Y_0, satellite pseudorange observations selected by top elevations further reduces the accuracy of Z_0.

Table 5.11: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 3 satellites and pixel observations of 2 features, and 4 satellites and 1 feature.

	3 satellites and 2 features						**4 satellites and 1 features**					
X_0	0.23	0.04	0.10	0.10	-0.23	0.09	2.89	-0.55	0.95	0.95	-0.99	0.95
Y_0		0.23	-0.16	-0.16	-0.13	-0.16		1.27	-0.57	-0.58	0.45	-0.57
Z_0			7.98	1.00	-0.00	1.00			17.80	1.00	-0.93	1.00
t_0				7.99	-0.00	1.00				18.07	-0.93	1.00
κ					0.59	-0.00					11.12	-0.93
dt_r						6.97						15.17
Γ_0	156718.40						101039.32					

Table 5.12: Error of the estimation of integration model, units are [degree,m,m,m].

Scenarios	κ	X_0	Y_0	Z_0
13 satellites and 6 features	-0.04	0.32	0.18	1.23
3 satellites and 2 features	-0.87	0.05	0.22	-6.66
4 satellites and 1 feature	13.32	3.43	6.43	54.50

5.2.4. Extended integration

Reliability

MDBs computed for all available observations from both sensors of the extended integration model. Compare table.5.13 with table.5.8 and table.5.5, the MDBs for the GNSS part are further slightly improved by the one more redundancy of the extended integration model, the vision part generally stay changed (the largest values dropped).

Table 5.13: MDBs computed for all available observations from both sensors in the extended integration model.

Pseudorange	**MDB [m]**	u	**MDB [pixel]**	v	**MDB [pixel]**
Δp^1	9.42	Δu_1	69.43	Δv_1	54.98
Δp^2	26.70	Δu_2	146.95	Δv_2	69.14
Δp^3	5.17	Δu_3	59.82	Δv_3	54.99
Δp^4	4.02	Δu_4	102.28	Δv_4	72.27
Δp^5	8.43	Δu_5	66.66	Δv_5	58.92
Δp^6	3.69	Δu_6	60.61	Δv_6	57.57
Δp^7	4.16				
Δp^8	4.97				
Δp^9	31.60				
Δp^{10}	8.74				
Δp^{11}	11.85				
Δp^{12}	17.03				
Δp^{13}	14.21				

MDB of the extended integration model, computed by 2 satellite observations and 2 features. From table.5.14, we can see the values of $\Delta p^6, \Delta p^{10}$ are closer to those in table.5.5, the integration model provides poorer validation between vision observations since the integration model depends heavily on the vision

measurements, and the integration model is well constrained by the vision, GNSS observations can be validated from the integration model.

Table 5.14: MDB of the extended integration model, computed by 2 satellite observations and 2 features.

Pseudorange	**MDB [m]**	u	**MDB [pixel]**	v	**MDB [pixel]**
Δp^6	9.16	Δu_5	10897.59	Δv_5	41909.13
Δp^{10}	9.16	Δu_6	6903.33	Δv_6	19782.18

Precision and correlation coefficients

$Q_{\hat{P}\hat{P}}$ **computed from all available observations.** Leaving out t_0 not only eliminates the dependence between t_0 and Z_0, dtr, but also leads to less correlation between Z_0 and dt_r, which means by the aid of vision system, the integration can better differentiate Z_0 from dt_r. The standard deviation of X_0, Y_0, Z_0 is below 0.22 m, and a standard deviation of 0.13 degree and 0.34 m in κ and dt_r estimation respectively.

Table 5.15: The interpreted $Q_{\hat{P}\hat{P}}$ computed from pseudorange observations of 13 satellites and pixel observations of 6 features.

	13 satellites and 6 features				
X_0	0.21	-0.00	0.01	-0.03	-0.03
Y_0		0.21	0.00	-0.01	-0.05
Z_0			0.22	-0.00	0.48
κ				0.13	-0.00
dt_r					0.34
Γ_0	3212.18				

$Q_{\hat{P}\hat{P}}$ **computed from pseudorange observations of 3 satellites and pixel observations of 1 feature.** The performance of table.C.12 in Appendix C is better than table.5.16 since the integration model relies much on the quality of satellite pseudorange observations. It shows that it is possible that when both sensor fail individually, the integration still can to produce a precise and reliable position solution when the input observations are good.

Table 5.16: The interpreted $Q_{\hat{P}\hat{P}}$ computed from pseudorange observations of 3 satellites and pixel observations of 1 feature.

	3 satellites and 1 feature				
X_0	34.01	1.00	-1.00	-0.88	-1.00
Y_0		23.37	-1.00	-0.86	-1.00
Z_0			7.33	0.87	1.00
κ				1.03	0.00
dt_r					16.82
Γ_0	312623.93				

Estimation of extended integration Compare table.5.17 with table.5.12, the extended integration model further improves the accuracy of Z_0 by using all available observations from both sensors. The estimation computed by two sensors failing at the same time degrades dramatically. Compared with the accuracy estimation error by using the end of SF-PPP data processing filter and the same model, values are smaller but still far away from the truth.

Table 5.17: Actual error of the estimation of integration model, units are [degree,m,m,m].

Scenarios	κ	X_0	Y_0	Z_0
13 satellites and 6 features	-0.04	0.32	0.18	0.49
3 satellites and 1 features	-1675.62	103.98	72.61	-23.83

5.3. Summary

Vision observations can significantly improve the GNSS position solutions since the vision solutions are of higher precision, they contribute more in the integration model.

The height offset t_0 between the SF-PPP antenna and camera is estimated from the GNSS model and the vision model, the camera orientation κ is estimated from the vision model.

The position estimates produced by the integration model is related to the performance of SF-PPP model. The position estimates produced by the later one outperform since it indirectly improves the performance of SF-PPP model by reducing correlation between Z_0 and dt_r, and eliminating correlation between t_0 and dt_r, t_0 and Z_0.

As we only take pseudorange observations in the integration model, data from the start of SF-PPP filter performs similarly as the data from upon convergence, since filter convergence improves the position solution by keeping ambiguities from the carrier phase observations constant, which are not included the integration and extended integration model.

The integration algorithm has the potential to be open to any GNSS positioning technique. So the current results can be improved by replacing SF-PPP with a RTK algorithm.

6

Conclusions and recommendations

The goal of this thesis project is to explore, by combining an affordable monocular vision system, Single-frequency Precise Point Positioning (SF-PPP) GNSS positioning and previously measured landmark positions, the performance of a GNSS/Vision integrated model, and the feasibility of the application of the integration model by considering different scenarios.

6.1. Answers to the research questions

How to derive a Monocular vision model to determine the camera pose with the aid of measured landmark position coordinates? The forward looking monocular vision model is derived from an airborne photogrammetry model, which describes the projection of world points onto the image plane. The camera frame can be linked to the world frame by just two rotations. First, we need to re-orientate the forward looking vision model in the world frame, and apply an extra rotation to describe the camera orientation. Since we are more interested in vehicle movement in the horizontal X-Y plane, the concatenated rotations are limited to include the frame re-orientation and heading changes in the world frame.

How to design a tight integration model for SF-PPP and monocular vision? The tight integration model estimates the unknowns from observations of the two sensors, therefore, its design matrix includes the Jacobian or derivative of the observations equations with respect to the unknowns. After derivation of the forward looking model, the vision Jacobian matrix is computed and inserted in the integration model. The SF-PPP

(GNSS) one is computed using for instance the known satellite positions and also inserted in the integration model. However, as SF-PPP unknown parameters are expressed in ECEF frame (earth-centered, earth-fixed), and vision's parameters are computed in a local world frame, a conversion matrix between the two frames is necessary to be inserted in the SF-PPP model to have position estimates eventually in the world frame in the integration model. Additionally, a position offset between the GNSS antenna and camera (lens) is considered in the integration model, because it is difficult to accurately determine the camera and antenna center in practice. As we mount the GNSS antenna directly on top of the camera, the position offset is simplified as a height offset t_0. An extended integration model is also tested by simulated and experimental data, by leaving out the height offset t_0, as in practice the GNSS antenna may be very close to the camera (e.g. with 1 dm).

How to account for the effect of landmark position errors? How will the measurements errors (feature identification error and landmark position error) affect the result? Chapter 3 presents two methods to account for the landmark position error through the landmark-camera geometry. Method I projects the landmark position error by error propagation law into vision observation uncertainty and the uncertainty needs to be recomputed and updated in each iteration (for the non-linear model); Method II includes landmark positions as observations, and its uncertainty is directly included as observation uncertainty and remains constant through iterations. In the simulation and experiment test, the landmark position or High Definition Maps (HD Maps) error is set as $\sigma_X \equiv \sigma_Y \equiv \sigma_Z \equiv$0.05 m (with sufficient landmarks), and the pixel error as $\sigma_u = \sigma_v =$20 pixels, then the standard deviation of the camera position estimates is below 0.22 m. From both tests, the results show that a larger landmark position error contributes to a larger standard deviation of the estimates, the pixel error only has a minor impact on this, see table.6.1.

Table 6.1: Standard deviation of position estimation by varying pixel error (unertainties) and landmark position error (uncertainty).

Scenarios	$\sigma_X \equiv \sigma_Y \equiv \sigma_Z$[m]	$\sigma_u = \sigma_v$[pixels]	std [m]
1 (default)	0.05	20	0.22
2	0.003	20	0.06
3 (simulation)	0.05	2.5	0.22
4	0.003	2.5	0.05

How to express/measure performance (what quantities are of interest)? The model performance by simulated data and experimental data are evaluated by the precision of the estimates, correlation coefficients between the estimates, and Minimal Detectable Bias (MDB) for outliers (faults in the measurements). Precision describes the error of estimates computed from the current model, correlation coefficient presents how well the model can differentiate the contribution of the measurements between the estimates, and the MDB describes to what extent an incidental fault in a measurement can be detected, by means of statistical testing on the data. However, vision observations can be contaminated by landmark misidentifications, since

a potential pixel error is unknown, we evaluate vision performance by 1-vector MDB and 2-vector MDB, the former one means the pixel error is independent in the two directions when a pixel measurement is obtained, the later one indicates that the coordinates in the two directions of a pixel measurement are both faulty at the same time in the image, for instance when a feature identification fault is made. We use theoretical analysis and experiment analysis to evaluate the model performance, the former one aims to show the optimal performance by using the true camera pose as the initial values, the later one aims to show the performance of the complete procedure of estimating the unknowns.

6.2. Conclusions

What is the the performance of GNSS/Vision integrated model? What are limitations of the integration model, can it work under extreme settings? When the pre-surveyed landmark positions (or error of HD map) are accurate at the 5-cm-level (standard deviation), the integration model is able to estimate X_0, Y_0 within a standard deviation of 0.22 m, Z_0, the antenna-camera offset t_0 within 1.77 m and heading κ within 0.13 degree, with fair correlation coefficients between the unknowns, except for the ones of Z_0 and t_0, Z_0 and dt_r, t_0 and dt_r. These unknowns are quite similarly related to the observations and it is difficult for the integration model to differentiate the observational data over these parameters. We also investigated the model performance by leaving out offset t_0, which is the so-called extended integration model. The model improves the standard deviation of Z_0 and dt_r compared to the integration model, by eliminating the correlation between Z_0 and t_0, t_0 and dt_r. The integration model is able to produce the position solution when one of the two sensors fails: when GNSS operates and vision fails, the integration can barely improve the GNSS solution, neither accuracy or precision of the estimates are acceptable, this is because the vision measurements are insufficient to estimate κ; including t_0 also aggregates the correlation between t_0 and Z_0, X_0, all these relies on the GNSS measurements which can barely produce a position solution; when GNSS fails and vision operates, the integration can improve the GNSS solution significantly, with acceptable accuracy and precision of the estimates. The extended integration model is theoretically capable of handling more complicated situations: when the two sensors both fail, however, the performance is poor in this extreme case.

6.3. Recommendations

With the aid of pre-surveyed landmarks in the image and the external processing products of the two sensors (GNSS pre-processing, object recognition and data association), the integration model can improve the GNSS positioning solution in urban areas. In this condition, the work of the thesis project can provide a more accurate position solution much faster than the one provided by SF-PPP. The work can be regarded as part of a fully automated GNSS/vision positioning procedure, who breaks the limitations of the two chosen sen-

sors individually and is more robust in challenging environments. The full procedure can contribute to the vehicle positioning part in automated driving which emphasizes more on sensing and analyzing the environment and planning for the optimal route. Based on the assumptions and limitations of this thesis project, we propose the following recommendations:

Refine the vision part of the application: Although the project is done in the context of map-based vehicle positioning, the work mainly focuses on the integration performance analysis after the pre-processed data from both sensors is obtained. The image processing of the vision part is missing and can be added to have the a complete and automated procedure. Therefore, correcting for lens distortion and applying object detection for the landmarks to obtain image observations can improve the current image observations and contribute positively to the integration. For the monocular vision model derivation, we apply a horizontally flat road assumption which means that heading changes are taken into account. Further improvement can include full rotations of the camera, which is critical for vehicle positioning on non-flat road or in dynamic environment, since the vision measurements are sensitive to angular changes.

Refine the GNSS part of the application: As one can see, the errors in vision observations are much smaller than the GNSS observations in our experiment, so the integration model is largely constrained by the vision model, and the GNSS model through SF-PPP has little contribution to the estimates. One way to improve this is to add carrier phase observations, which can increase the computation load of the integration model but enable GNSS to contribute more. Therefore, the integration model is potentially able to produce estimates better than the position solution produced by any one of the sensors. The other way is to use another GNSS positioning technique, for instance, RTK (Real-time Kinematic) can provide positioning solutions with centimeter accuracy in real time and can improve the integration performance significantly, though at a larger infrastructure demand.

Use more powerful optimization methods and mathematical model: The dimension of the integration increases when we include more satellite observations (pseudorange and carrier phase) and pixel measurements. The nonlinearity also increases when we consider all rotations (not only for heading κ, but also ψ and θ) of the camera, and what's more, the Jacobian might be difficult to compute and current symbolic computation can take longer time. Then one needs to call for an alternative mathematical model formulation to deal with the rising problems in matrix dimension and nonlinearity. The integration model in this research is meant for vehicle single epoch positioning and can work on a snap-shot basis. For real-time positioning in dynamic application, one can use one of the filters (Extended Kalman filter, Unscented Kalman filter, Particle filter or etc.), which enables the integration to make predictions for next time step. In the simulation and experiment, we put sufficient landmarks to study the influence of their geometry and number of landmarks

on the position solution from the vision and the integration model. In reality, landmarks are sparse and random Zang et al. [42], and we may have to handle two more rotations and therefore a sufficient number of landmarks needs to be present for the method to work. In urban areas, GNSS pseudorange multipath errors can contaminate the observations but likely a lot of landmarks are available for vision to operate and provide a reliable position solution.

A

Landmark visulization in reality and in image plane of simulation

3D plot and projected features on the image plane by reducing the number of features of the simulated vision data. Fig.A.1 in Chapter 4 shows the 3D plot of 3 features; fig.A.2 shows the 3D plot of 2 features; fig.A.3 shows the projected 3 features on the image plane; fig.A.4 shows the projected 2 features on the image plane.

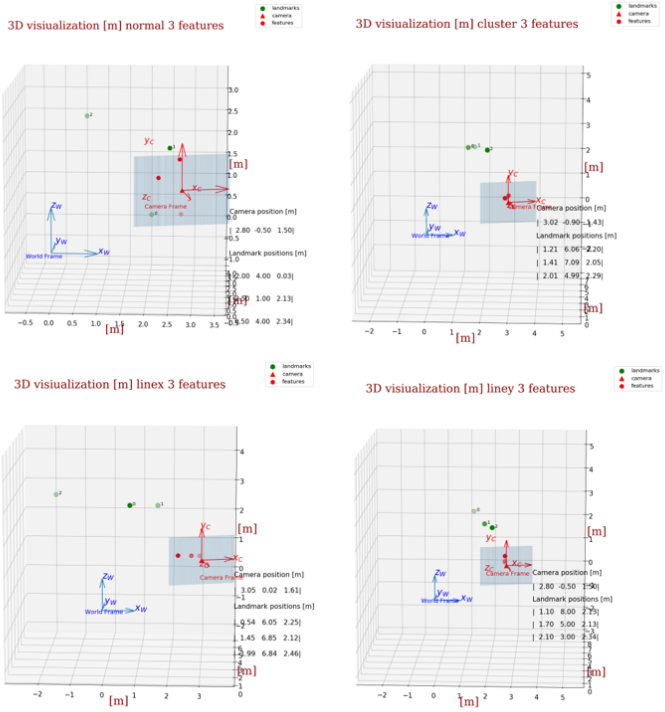


Figure A.1: 3D plot of landmark geometries with 3 features in reality. The green dots are landmarks in the world frame (blue), red ones are the projected features of the green ones on image plane (grey) in the camera frame (red).

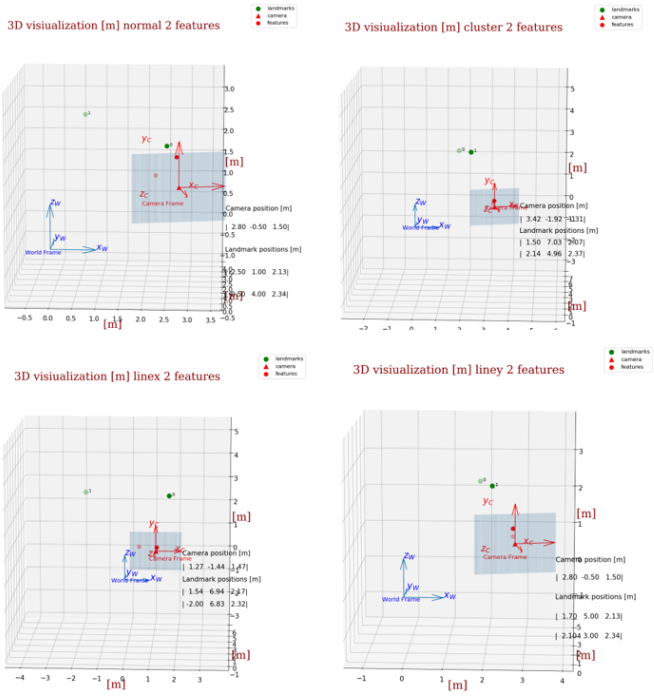


Figure A.2: 3D plot of landmark geometries with 2 features in reality. The green dots are landmarks in the world frame (blue), red ones are the projected features of the green ones on image plane (grey) in the camera frame (red).

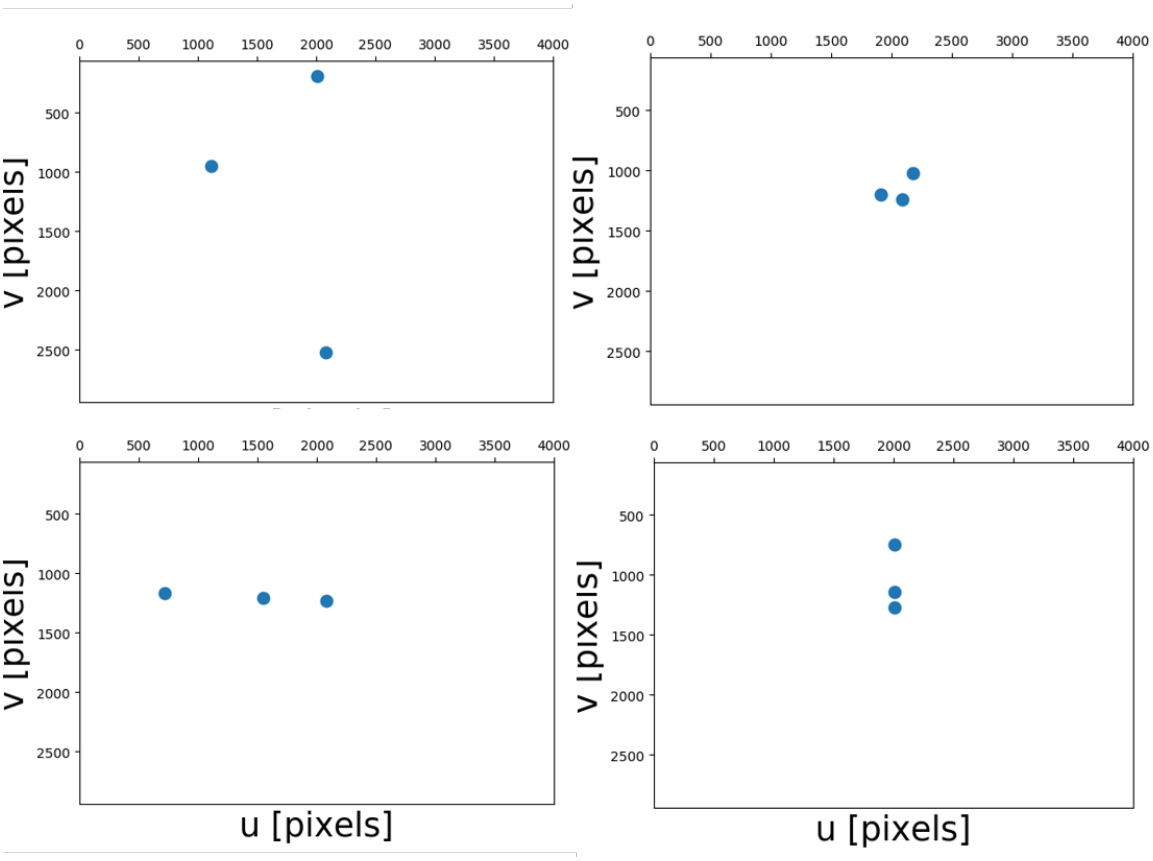


Figure A.3: Geometry of 3 selected features on the image plane of four scenarios, with the origin on the top left. From the top left, top right, bottom left to bottom right are: normal, cluster, line up in x direction and line up in y direction scenarios.

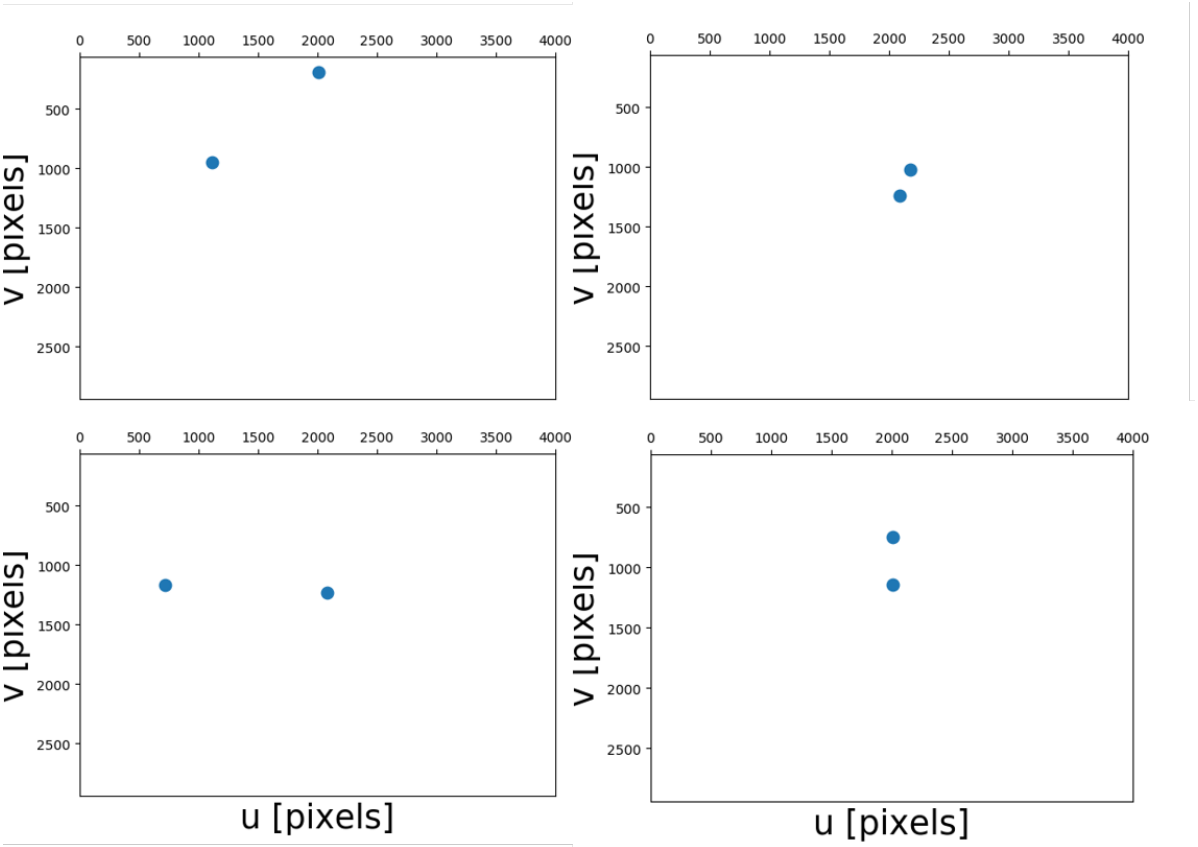


Figure A.4: Geometry of 2 selected features on the image plane of four scenarios, with the origin on the top left. From the top left, top right, bottom left to bottom right are: normal, cluster, line up in x direction and line up in y direction scenarios.

B

Transformation from ITRF2008 to ETRF2000

The position of DLF1 (permanent GNSS receiver at TU Delft) in the ETRF2000 can be found at: http://www.epncb.oma.be/_productsservices/coordinates/crd4station.php?station=DLF100NLD, the transformation is service provided by http://www.epncb.oma.be/_productsservices/coord_trans/. We apply the latest updated DLF1 position in ETRF2000, and we only need to configure the current frame, target frame and the experiment date. The conversions and intermediate results are shown in table.B.1. The transformation is needed as SF-PPP is providing positions in ITRF2008.

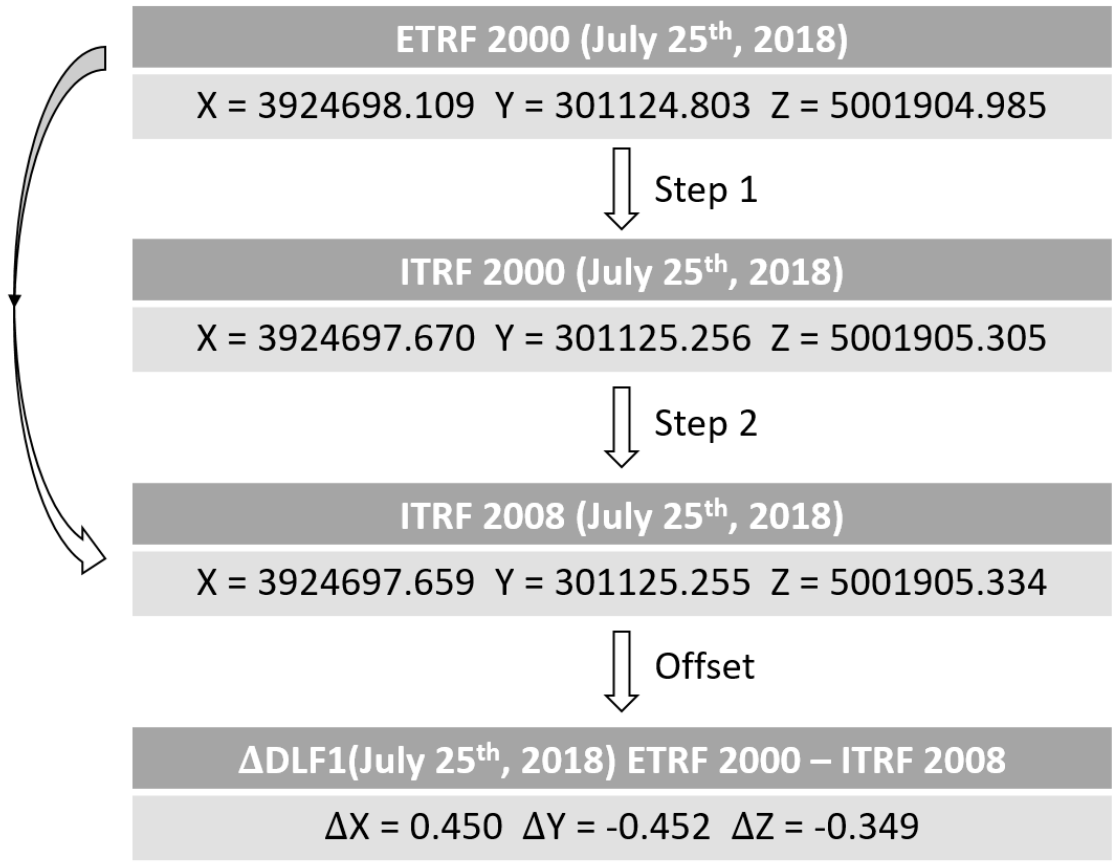


Figure B.1: Computation of ETRF2000 and ITRF2008 offset for the day of the experiment, July 25^{th}, 2018. The first step is to transform DLF1 position from ETRF2000 to ITRF2000, the second step is to transform from DLF1 position from ITRF2000 to ITRF2008, by adding the offset to SF-PPP position solutions.

C

Experiment results by using GNSS-measurement after the convergence of SF-PPP

The results of GNSS and integration model calculated by fully converged SF-PPP solutions in the experiment are presented.

C.0.1. GNSS

SF-PPP

Reliability

Precision and correlation coefficients

Skyplot

Estimation of GNSS

C.0.2. Integration

Reliability

Table C.1: MDB computed from 14 satellites and 5 satellites by using the experimental data (SF-PPP filter converged).

Pseudorange	**MDB [m]**	**MDB [m]**
Δp^1	13.22	
Δp^2	5.81	15.86
Δp^3	5.04	49.86
Δp^4	4.16	21.35
Δp^5	11.03	
Δp^6	8.12	
Δp^7	12.17	
Δp^8	38.73	
Δp^9	27.29	
Δp^{10}	9.89	9.96
Δp^{11}	47.00	
Δp^{12}	35.63	
Δp^{13}	9.60	94.54
Δp^{14}	10.77	

Table C.2: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 14 and 5 satellites, X_0, Y_0, Z_0 are transformed into world frame.

	14 satellites				**5 satellites**			
X_0	0.68	0.02	0.41	0.01	2.02	0.68	0.09	0.16
Y_0		0.91	0.17	0.14		2.41	0.43	0.37
Z_0			1.46	0.97			6.08	1.00
dt_r				1.21				5.53

Precision and correlation coefficients

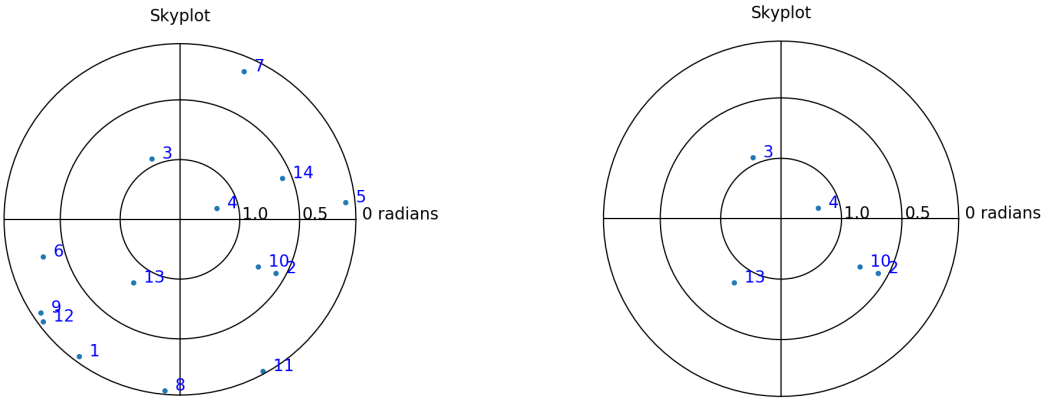
Estimation of integration

C.0.3. Extended integration

Reliability

Precision and correlation coefficients

Estimation of integration



(a) Skyplot of all available satellites.

(b) Skyplot of satellites of top 5 elevations.

Figure C.1: Skyplot of 14 and 5 satellites at the end of SF-PPP data processing.

Table C.3: Actual error of the estimation of vision model, units are [m,m,m].

Scenarios	X_0	Y_0	Z_0
14 satellites	2.41	-1.60	-6.61
5 satellites	-12.83	-22.28	-18.81

Table C.4: MDB of the integration model, computed by all available observations from both sensors.

Pseudorange	**MDB [m]**	u	**MDB [pixel]**	v	**MDB [pixel]**
Δp^1	12.26	Δu_1	65.54	Δv_1	54.75
Δp^2	4.12	Δu_2	175.15	Δv_2	70.03
Δp^3	3.86	Δu_3	59.48	Δv_3	54.79
Δp^4	4.10	Δu_4	119.59	Δv_4	73.56
Δp^5	9.86	Δu_5	66.28	Δv_5	58.61
Δp^6	5.54	Δu_6	60.44	Δv_6	56.62
Δp^7	8.16				
Δp^8	38.50				
Δp^9	27.04				
Δp^{10}	9.65				
Δp^{11}	46.83				
Δp^{12}	35.44				
Δp^{13}	9.25				
Δp^{14}	10.55				

Table C.5: MDB computed from 3 satellites selected from 3 highest elevations and 2 landmarks.

Pseudorange	**MDB [m]**	**u**	**MDB [pixels]**	**v**	**MDB [pixels]**
Δp^3	6.59	Δu_5	12127.80	Δv_5	13002.96
Δp^4	17.57	Δu_6	5371.33	Δv_6	6907.89
Δp^{13}	10.54				

Table C.6: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 14 satellites and pixel observations of 6 features.

	14 satellites and 6 features					
X_0	0.21	-0.00	0.02	0.02	-0.03	0.00
Y_0		0.21	0.04	0.04	-0.01	0.03
Z_0			1.67	0.99	-0.00	0.97
t_0				1.68	-0.00	0.96
κ					0.13	-0.00
dt_r						1.20
Γ_0	24161.45					

Table C.7: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 3 satellites and pixel observations of 2 features, and 4 satellites and 1 feature.

	3 satellites and 2 features						**4 satellites and 1 features**					
X_0	0.23	0.03	0.11	0.11	-0.22	0.11	2.73	0.64	0.57	0.60	0.41	0.56
Y_0		0.23	-0.03	-0.03	-0.11	-0.04		2.47	0.38	0.41	-0.42	0.36
Z_0			14.23	1.00	-0.02	1.00			13.91	1.00	0.22	1.00
t_0				14.24	-0.02	1.00				14.25	0.22	1.00
κ					0.57	-0.00					3.35	0.23
dt_r						13.52						12.96
Γ_0	284528.24						110709.75					

Table C.8: Actual error of the estimation of integration model, units are [degree,m,m,m].

Scenarios	κ	X_0	Y_0	Z_0
14 satellites and 6 features	-0.23	0.25	-0.07	-6.49
3 satellites and 2 features	-0.89	0.12	-0.08	14.23
4 satellites and 1 feature	128.89	-19.57	24.30	-64.70

Table C.9: MDB of the extended integration model, computed by all available observations from both sensors.

Pseudorange	**MDB [m]**	u	**MDB [pixel]**	v	**MDB [pixel]**
Δp^1	11.29	Δu_1	69.43	Δv_1	54.98
Δp^2	4.11	Δu_2	146.98	Δv_2	69.14
Δp^3	3.64	Δu_3	59.82	Δv_3	54.99
Δp^4	3.52	Δu_4	102.28	Δv_4	72.27
Δp^5	8.86	Δu_5	66.66	Δv_5	58.92
Δp^6	5.16	Δu_6	60.61	Δv_6	57.57
Δp^7	7.26				
Δp^8	38.22				
Δp^9	26.75				
Δp^{10}	9.64				
Δp^{11}	46.57				
Δp^{12}	35.15				
Δp^{13}	9.22				
Δp^{14}	10.55				

Table C.10: MDB of the extended integration model, computed by 2 satellite observations and 2 features.

Pseudorange	**MDB [m]**	u	**MDB [pixel]**	v	**MDB [pixel]**
Δp^3	4.35	Δu_5	3941.30	Δv_5	11316.76
Δp^4	4.35	Δu_6	2899.53	Δv_6	5489.35

Table C.11: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 14 satellites and pixel observations of 6 features.

	14 satellites and 6 features				
X_0	0.21	-0.00	0.00	-0.03	-0.05
Y_0		0.22	0.01	-0.01	-0.02
Z_0			0.22	-0.00	0.52
κ				0.13	0.00
dt_r					0.34
Γ_0	3095.34				

Table C.12: The interpreted $Q_{\hat{p}\hat{p}}$ computed from pseudorange observations of 3 satellites and pixel observations of 1 features.

	3 satellites and 1 feature				
X_0	2.25	0.43	-0.66	0.80	-0.56
Y_0		2.37	-0.84	-0.19	-0.86
Z_0			0.54	-0.16	0.89
κ				5.44	0.35
dt_r					0.97
Γ_0	12516.68				

Table C.13: Actual error of the estimation of integration model, units are [degree,m,m,m].

Scenarios	κ	X_0	Y_0	Z_0
14 satellites and 6 features	-0.21	0.27	-0.02	0.30
3 satellites and 1 feature	146.99	9.23	22.93	2.64

Bibliography

[1] Akram Afifi and Ahmed El-Rabbany. An Improved Model for Single-Frequency GPS/GALILEO Precise Point Positioning. 6(May):7–21, 2015. ISSN 2150-850X, 2150-8526. doi: 10.4236/pos.2015.62002. URL http://www.scirp.org/journal/pos{%}5Cnhttp://dx.doi.org/10.4236/pos.2015.62002{%}5Cnhttp://creativecommons.org/licenses/by/4.0/.

[2] C-O. Andrei, R Chen, H Kuusniemi, M Hernandez-Pajares, J M Juan, and D Salazar. Ionosphere Effect Mitigation for Single-frequency Precise Point Positioning. *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, pages 2508–2517, 2009. URL http://www.ion.org/search/view{_}abstract.cfm?jp=p{&}idno=8664.

[3] Bernhard M Aumayer, Mark G Petovello, and Gérard Lachapelle. Development of a Tightly Coupled Vision/GNSS System. *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2014)*, (September 2014):8–12, 2014.

[4] P F De Bakker, V L Knoop, C C J M Tiberius, and B Van Arem. Mapping Motorway Lanes and Real-Time Lane Identification with Single-Frequency Precise Point Positioning. In *Proceedings of the European navigation conference (ENC)-GNSS*. Nederlands Instituut voor Navigatie, 2014.

[5] Marcus A. Brubaker, Andreas Geiger, and Raquel Urtasun. Map-based probabilistic visual self-localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):652–665, 2016. ISSN 01628828. doi: 10.1109/TPAMI.2015.2453975.

[6] Changsheng Cai, Yangzhao Gong, Yang Gao, and Cuilin Kuang. An approach to speed up single-frequency PPP convergence with quad-constellation GNSS and GIM. *Sensors (Switzerland)*, 17(6), 2017. ISSN 14248220. doi: 10.3390/s17061302.

[7] Sergio Camacho-Lara. Current and Future GNSS and Their Augmentation Systems. In Joseph N. Pelton, , Scott Madry, , and Sergio Camacho-Lara, editors, *Handbook of Satellite Applications*, pages 617–654. Springer New York, New York, NY, 2013. ISBN 978-1-4419-7671-0. doi: 10.1007/978-1-4419-7671-0_25. URL https://doi.org/10.1007/978-1-4419-7671-0{_}25.

[8] Xiao Chen, Weidong Hu, Lefeng Zhang, Zhiguang Shi, and Maisi Li. Integration of low-cost GNSS and monocular cameras for simultaneous localization and mapping. *Sensors (Switzerland)*, 18(7):1–18, 2018. ISSN 14248220. doi: 10.3390/s18072193.

[9] Ji Hoon Choi, Yong Woon Park, Jae Bok Song, and In So Kweon. Localization using GPS and VISION aided INS with an image database and a network of a ground-based reference station in outdoor environments. *International Journal of Control, Automation and Systems*, 9(4):716–725, 2011. ISSN 15986446. doi: 10.1007/s12555-011-0413-y.

[10] Alfonso CROEZE, Lindsey Pittman, and Winnie REYNOLDS. Solving Nonlinear Least-Squares Problems With the Gauss-Newton and Levenberg-Marquardt Methods. *Math.Lsu.Edu*, pages 1–16, 2012. URL https://www.math.lsu.edu/system/files/MunozGroup1-Paper.pdf.

[11] Dixiao Cui, Jianru Xue, and Nanning Zheng. Real-Time Global Localization of Robotic Cars in Lane Level via Lane Marking Detection and Shape Registration. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1039–1050, 2016. ISSN 15249050. doi: 10.1109/TITS.2015.2492019.

[12] Peter F. de Bakker and Christian C.J.M. Tiberius. Real-time multi-GNSS single-frequency precise point positioning. *GPS Solutions*, 21(4):1791–1803, 2017. ISSN 15211886. doi: 10.1007/s10291-017-0653-2.

[13] Peter F. de Bakker, Hans van der Marel, and Peter J.G. Teunissen. The Minimal Detectable Bias for GNSS Observations with a Single Receiver Setup and a Geometry-Free Model. (1), 2008.

[14] K. De Jong and P. J.G. Teunissen. Minimal Detectable Biases of GPS observations for a weighted ionosphere. *Earth, Planets and Space*, 52(10):857–862, 2000. ISSN 18805981. doi: 10.1186/BF03352295.

[15] Mohamed Elsobeiey. Precise point positioning using triple-frequency GPS measurements. *Journal of Navigation*, 68(3):480–492, 2015. ISSN 14697785. doi: 10.1017/S0373463314000824.

[16] Wolfgang Förstner and Bernhard P. Wrobel. *Photogrammetric Computer Vision*, volume 11. 2016. ISBN 978-3-319-11549-8. doi: 10.1007/978-3-319-11550-4. URL http://link.springer.com/10.1007/978-3-319-11550-4.

[17] M.A. Galala, M.R. Kaloop, M.M. Rabah, and Z.M. Zeidan. Improving precise point positioning convergence time through TEQC multipath linear combination. *Journal of Surveying Engineering*, 144(2):1–11, 2018. ISSN 07339453. doi: 10.1061/(ASCE)SU.1943-5428.0000250.

[18] Zhouzheng Gao, Maorong Ge, Wenbin Shen, Hongping Zhang, and Xiaoji Niu. Ionospheric and receiver DCB-constrained multi-GNSS single-frequency PPP integrated with MEMS inertial measurements. *Journal of Geodesy*, 91(11):1351–1366, 2017. ISSN 14321394. doi: 10.1007/s00190-017-1029-7.

[19] Dominique Gruyer, Rachid Belaroussi, and Marc Revilloud. Accurate lateral positioning from map data and road marking detection. *Expert Systems with Applications*, 43:1–8, 2016. ISSN 09574174. doi: 10.1016/j.eswa.2015.08.015. URL http://dx.doi.org/10.1016/j.eswa.2015.08.015.

[20] C.-H. PARK KIM and N.-H. Precise and Reliable Positioning Based on the Integration of Navigation Satellite System and Vision System. *International Journal of Automotive Technology*, 15(1):81–87, 2014. ISSN 1229-9138. doi: 10.1007/s12239âĹŠ014âĹŠ0009âĹŠ7. URL http://link.springer.com/article/10.1007/s12239-012-0027-2.

[21] Youngsun Kim and Dong Hwan Hwang. Vision/INS Integrated Navigation System for Poor Vision Navigation Environments. *sensors*, 16(10):531–535, 2016. ISSN 15987833. doi: 10.3390/s16101672.

[22] Liang Li, Chun Jia, Lin Zhao, Jianhua Cheng, Jianxu Liu, and Jicheng Ding. Real-time single frequency precise point positioning using SBAS corrections. *Sensors (Switzerland)*, 16(8), 2016. ISSN 14248220. doi: 10.3390/s16081261.

[23] Dennis Odijk, Peter J. G. Teunissen, and Baocheng Zhang. Single-frequency integer ambiguity resolution enabled GPS precise point positioning. *Journal of Surveying Engineering*, 138(4):193—-202, 2012. ISSN 0733-9453. doi: 10.1061/(ASCE)SU.1943-5428.0000085. URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)SU.1943-5428.0000085.

[24] Lin Pan, Xiaohong Zhang, Jingnan Liu, Xingxing Li, and Xin Li. Performance Evaluation of Single-frequency Precise Point Positioning with GPS, GLONASS, BeiDou and Galileo. *Journal of Navigation*, 70 (03):465–482, 2017. ISSN 0373-4633. doi: 10.1017/S0373463316000771. URL https://www.cambridge.org/core/product/identifier/S0373463316000771/type/journal{_}article.

[25] J F Pascual-S´anchez. Introducing Relativity in GNSS. Technical report, Dept. Matem´atica Aplicada, Facultad de Ciencias, Universidad de Valladolid, Valladolid, Spain, EU, 2005.

[26] Rafael Peixoto, Derenzi Vivacqua, Massimo Bertozzi, Pietro Cerri, Felipe Nascimento Martins, and Raquel Frizera Vassallo. Self-Localization Based on Visual Lane Marking Maps : An Accurate Low-Cost Approach for Autonomous Driving. 19(2):582–597, 2018.

[27] Mark Petovello. GNSS Solutions : GPS jamming and linear carrier phase combinations. *InsideGNSS*, 2009.

[28] Xiaozhi Qu, Bahman Soheilian, and Nicolas Paparoditis. Vehicle localization using mono-camera and geo-referenced traffic signs. *Intelligent Vehicles Symposium (IV), 2015 IEEE*, (Iv), 2015.

[29] Stefan Schaer. Global Ionosphere Maps Produced by CODE, 2012. URL http://aiuws.unibe.ch/ionosphere/.

[30] Markus Schreiber, Hendrik Konigshof, Andre Marcel Hellmund, and Christoph Stiller. Vehicle localization with tightly coupled GNSS and visual odometry. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2016-Augus(Iv):858–863, 2016. doi: 10.1109/IVS.2016.7535488.

[31] Bahman Soheilian, Nicolas Paparoditis, and Bruno Vallet. Detection and 3D reconstruction of traffic signs from multiple view color images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 77:1–20, 2013. ISSN 09242716. doi: 10.1016/j.isprsjprs.2012.11.009. URL http://dx.doi.org/10.1016/j.isprsjprs.2012.11.009.

[32] GPS Solutions. Ppp-ar. URL https://www.gps-solutions.com/pppar.

[33] Zui Tao and Philippe Bonnifait. Tightly Coupling GPS with Lane Markings for Autonomous Vehicle Navigation. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[34] Zui Tao and Philippe Bonnifait. Sequential data fusion of GNSS Pseudoranges and Dopplers with map-based vision systems. *IEEE Transactions on Intelligent Vehicles*, 8858(c):1–11, 2017. doi: 10.1109/TIV.2017.2658185.

[35] P.J.G. Teunissen, D.G. Simons, and C.C.J.M. Tiberius. *Probability and Observation Theory*. Department of Geoscience and Remote Sensing Faculty of Civil Engineering and Geosciences Delft University of Technology P.O. Box 5048 2600 GA Delft The Netherland, Delft, The Netherlands, 2009.

[36] Hans van der Marel and Peter F. de Bakker. GNSS Solutions: Single- versus Dual- Frequency Precise Point Positioning. *InsideGNSS*, 2012.

[37] Eric A Wan, Rudolph Van Der Merwe, and N W Walker Rd. The Unscented Kalman Filter for Nonlinear Estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000 (AS-SPCC).*, Lake Louise, Alberta, Canada, 2000.

[38] Kan Wang, Amir Khodabandeh, Peter J G Teunissen, and Nandakumaran Nadarajah. Satellite-Clock Modeling in Single-Frequency PPP-RTK Processing. *Journal of Surveying Engineering*, 144(2):1–13, 2018. doi: 10.1061/(ASCE)SU.1943-5428.0000252.

[39] Xipeng Wang, Steve Vozar, and Edwin Olson. FLAG: Feature-based Localization between Air and Ground. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3178–3184, 2017. ISSN 10504729. doi: 10.1109/ICRA.2017.7989360.

[40] Dae Hee Won, Eunsung Lee, Moonbeom Heo, Seung-woo Lee, Jiyun Lee, Jeongrae Kim, Sangkyung Sung, and Young Jae Lee. Selective Integration of GNSS , Vision Sensor , and INS Using Weighted DOP Under GNSS-Challenged Environments. 63(9):2288–2298, 2014.

[41] Dae Hee Won, Eunsung Lee, Moonbeom Heo, Sangkyung Sung, Jiyun Lee, and Young Jae Lee. GNSS integration with vision-based navigation for low GNSS visibility conditions. *GPS Solutions*, 18(2):177–187, 2014. ISSN 15211886. doi: 10.1007/s10291-013-0318-8.

[42] Andi Zang, Zichen Li, David Doria, and Goce Trajcevski. Accurate Vehicle Self-Localization in High Definition Map Dataset. *Proceedings of the 1st ACM SIGSPATIAL Workshop on High-Precision Maps and Intelligent Applications for Autonomous Vehicles - AutonomousGIS '17*, pages 7–10, 2017. doi: 10.1145/3149092.3149094. URL http://dl.acm.org/citation.cfm?doid=3149092.3149094{%}0Ahttps://doi.org/10.1145/3149092.3149094.