Fusion of imaging and inertial navigation system for navigation

TIMOTHEÉ MAZIOL
Fusion of imaging and inertial navigation system for navigation

TIMOTHEÉ MAZIOL
FUSION OF IMAGING AND INERTIAL NAVIGATION SYSTEM FOR NAVIGATION

ROYAL INSTITUTE OF TECHNOLOGY (KTH) – STOCKHOLM, SWEDEN

ECOLE CENTRALE DE NANTES – FRANCE

TIMOTHEE MAZIOL

2018, Feb. 5th – 2018, Jul. 27th

Under the supervision of Valentin Demange (Safran Electronics & Defense).
University tutors: Xiaoming Hu (KTH) and Olivier Kermorgant (Ecole Centrale de Nantes).

DROITS DE PROPRIÉTÉ INDUSTRIELLE
Les informations contenues dans ce document sont la propriété de Sagem Défense Sécurité et diffusées à titre confidentiel dans un but spécifique. Le destinataire assure la garde et la surveillance de ce document et convient qu’il ne sera ni copié ni reproduit en tout ou partie et que son contenu ne sera révélé en aucune manière à aucune personne, excepté pour répondre au but pour lequel il a été transmis. Cette recommandation est applicable à toutes les pages de ce document.

PROPRIETARY RIGHTS
This document contains information of a proprietary nature to Sagem Défense Sécurité company and is submitted in confidence for a specific purpose. The recipient assumes custody and control and agrees that this document will not be copied or reproduced in whole or in part, nor its contents revealed in any manner or to any person except to meet the purpose for which it was delivered. This legend is applicable to all the pages of this document.
Page left blank intentionally.
CONFIDENTIALITE DU RAPPORT DE STAGE
AUTORISATION DE DIFFUSION ET EXPLOITATION

Le présent rapport établi par : Timothée MAZIOL
École/Organisme : KTH Royal Institute of Technology
A l’issue de son stage/apprentissage/thèse du : 05/02/2018 au 27/07/18
Dans l’établissement Safran Electronics & Defense d’Eragny,
Au sein du service/programme/GE : CED-NS
A été validé par : Valentin DEMANGE

Les informations qu’il contient ne relève pas du patrimoine scientifique et technique ou ont déjà été publiées et peuvent donc être diffusées et exploitées en totalité ou en partie, hors de l’organisme d’appartenance du stagiaire, avec l’autorisation de Safran Electronics & Defense.

Responsable Safran du stage
Nom : DEMANGE Valentin
Date : 11/07/2018
Visa d’autorisation

Rédacteur du rapport de stage
Nom : MAZIOL Timothée
Date : 11/07/2018
Visa :

Sûreté Safran
Nom : 23/07/2018 - VU
Date :
Visa :

Christine LETSCHER
Officier de Sécurité

Version du 29 Septembre 2017
ACKNOWLEDGMENTS

This project has been at the initiative of Safran Electronics & Defense on the site located at Eragny in France and it was a wonderful experience for learning and develop my skills. I would like to express all my gratitude to all the employees I had the opportunity to meet there, the engineers, managers, co-interns… and especially to the people that worked in the same department as me.

This experience would not have been that fruitful and enlightening without the support of my company supervisor, Valentin DEMANGE. His patience, understanding, great teachings and sympathy guided me well through this internship period.

I would also like to address my gratitude to Emmanuel R. who took time out to support me, follow my project and who placed his confidence in my work.

I would also like to thank Olivier KERMORGANT, my supervisor from l'Ecole Centrale de Nantes for his help for the well-conduction of this internship.

Finally, I would like to thank Xiaoming HU for its great support and understanding all along this master thesis.

This internship led to my recruitment at Safran Electronics & Defense and I am very glad to continue to be a part of a department that gathers such wonderful people.
DOCUMENT CONTROL

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>NAME</th>
<th>POSITION</th>
<th>DATE</th>
<th>SIGNATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Established by</td>
<td>Sagem Défense Sécurité</td>
<td>Timothée Maziol</td>
<td>Intern</td>
<td>19/07/2018</td>
</tr>
<tr>
<td>Approved by</td>
<td>Safran Electronics &amp; Defense</td>
<td></td>
<td>Engineer</td>
<td></td>
</tr>
</tbody>
</table>

HISTORY OF THE DOCUMENT MODIFICATIONS

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modification Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td></td>
<td>Creation</td>
</tr>
</tbody>
</table>

CONTACT

Timothee Maziol  
Safran Electronics & Defense DS  
Etablissement d’Eryagny  
Avenue du Gros Chêne  
BP 10051  
95612 Cergy-Pontoise Cedex

Valentin Demange

© Propriété de Sagem Défense Sécurité - Reproduction et Divulgation interdites / Disclosure and copy prohibited.
# CONTENTS

| DOCUMENT CONTROL | ................................................................. | 5 |
| HISTORY OF THE DOCUMENT MODIFICATIONS | ................................................................. | 5 |
| CONTACT | ................................................................. | 5 |
| ABBREVIATIONS | ................................................................. | 8 |
| ABSTRACT | ................................................................. | 9 |
| 1. INTRODUCTION | ................................................................. | 10 |
| 2. INDUSTRIAL CONTEXT | ................................................................. | 11 |
| 2.1. SAFRAN | ................................................................. | 11 |
| 2.1.1. Global presentation | ................................................................. | 11 |
| 2.1.2. Key figures | ................................................................. | 11 |
| 2.2. SAFRAN ELECTRONICS & DEFENSE | ................................................................. | 12 |
| 2.3. EXCELLENCE CENTER FOR NAVIGATION SYSTEMS (CESN) | ................................................................. | 13 |
| 3. LITERATURE REVIEW | ................................................................. | 14 |
| 3.1. DIFFERENT APPROACHES OF FUSION | ................................................................. | 14 |
| 3.1.1. Extended Kalman Filter (EKF) | ................................................................. | 14 |
| 3.1.2. Unscented Kalman Filter (UKF) | ................................................................. | 15 |
| 3.1.3. Particular Filter | ................................................................. | 15 |
| 3.1.4. Nonlinear optimization | ................................................................. | 15 |
| 3.2. IMPROVEMENT OF VISION-COUPLED EKF | ................................................................. | 16 |
| 3.2.1. Self-calibration IMU - Camera | ................................................................. | 16 |
| 3.2.2. Observability Constrained Visual Inertial Navigation System | ................................................................. | 16 |
| 3.3. VISUAL ODOMETRY | ................................................................. | 17 |
| 3.3.1. Match pixels from different images | ................................................................. | 17 |
| 3.3.2. Bundle Adjustment algorithm | ................................................................. | 18 |
| 3.3.3. The scale factor | ................................................................. | 18 |
| 4. THE SIMULATOR | ................................................................. | 19 |
| 4.1. PRESENTATION OF THE SIMULATOR | ................................................................. | 19 |
| 4.1.1. Simulation of the Pure Inertial Navigation System (INS) | ................................................................. | 19 |
| 4.1.2. Simulation of the Vision observations | ................................................................. | 19 |
| 4.1.3. Modifications of the simulator | ................................................................. | 20 |
| 4.1.4. Simulations for different trajectories | ................................................................. | 21 |
| 4.2. ADDITION OF THE STIMULATION | ................................................................. | 22 |
| 5. STIMULATION | ................................................................. | 23 |
| 5.1. PRESENTATION OF THE DATA | ................................................................. | 23 |
| 5.2. INERTIAL-ONLY NAVIGATION | ................................................................. | 23 |
| 5.2.1. Barometer measurements | ................................................................. | 23 |
| 5.2.2. Missing frames | ................................................................. | 23 |
| 5.2.3. Lack of static alignment | ................................................................. | 24 |
5.2.4. Gradient descent method to estimate the angles between the body (vehicle) and the IMU .....24
5.3. ADDITION OF THE SIMULATED VISION MEASUREMENTS ...........................................26
6. VISION ALGORITHM .............................................................................................................27
6.1. PARAMETERS OF CALIBRATION ......................................................................................27
6.2. APPLICATION ON THE IMAGES OF THE HELICOPTER ................................................27
6.2.1. The data sets .....................................................................................................................27
6.3. OPEN-SOURCE DIRECT SPARSE ODOMETRY (DSO) VSLAM .....................................28
6.4. PROCESSING OF THE VISION OUTPUTS ........................................................................29
6.4.1. Frames ...............................................................................................................................29
6.4.2. Processing steps ................................................................................................................30
6.4.3. Scale Factor .......................................................................................................................30
7. HYBRIDISATION .......................................................................................................................33
7.1. THE DATA SET .....................................................................................................................33
7.2. STIMULATION OF THE INERTIA-ONLY NAVIGATION ..................................................34
7.3. VISION ALGORITHM ..........................................................................................................34
7.3.1. Direct-Sparse Odometry (DSO) VSLAM .............................................................34
7.3.2. Bundle Adjustment with inertial constraint .................................................................35
7.4. RESULT OF HYBRIDISATION ............................................................................................36
7.4.1. Results with Vision from DSO VSLAM .........................................................................36
7.4.2. Results with vision from Bundle Adjustment ...............................................................38
7.5. ANALYSIS OF THE RESULTS ..........................................................................................38
8. CONCLUSION AND FURTHER WORK ..................................................................................41
9. REFERENCES ..........................................................................................................................42
APPENDIX 1. THE KALMAN FILTER .........................................................................................44
A1.1. DEFINITION.........................................................................................................................44
A1.1.1. Model of the system ........................................................................................................44
A1.1.2. Concept of the Kalman filter ..........................................................................................44
A1.2. THE EXTENDED KALMAN FILTER ..................................................................................45
APPENDIX 2: FRAMES DEFINITIONS .....................................................................................46
A2.1. INERTIAL FRAME [I] ..........................................................................................................46
A2.2. EARTH CENTERED EARTH FIXED FRAME [T] .................................................................46
A2.3. LOCAL GEOGRAPHIC FRAME [G] ..................................................................................47
A2.4. BODY FRAME [B] .............................................................................................................48
A2.5. CAMERA FRAME [C] ........................................................................................................48
A2.6. IMU FRAME [M] ...............................................................................................................49
A2.7 OTHER CONVENTION ......................................................................................................49
APPENDIX 3. BUNDLE ADJUSTMENT ALGORITHM .................................................................50
# ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEDSN</td>
<td>Centre of Excellence for the Development of Navigation Systems</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filtering</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman Filtering</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measure Unit</td>
</tr>
<tr>
<td>IRS</td>
<td>Inertial Reference System</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
</tr>
<tr>
<td>SLAM</td>
<td>Simultaneous Localisation And Mapping</td>
</tr>
<tr>
<td>VSLAM</td>
<td>Visual SLAM</td>
</tr>
<tr>
<td>DSO</td>
<td>Direct Sparse Odometry</td>
</tr>
<tr>
<td>FOV</td>
<td>Field Of View</td>
</tr>
<tr>
<td>IFOV</td>
<td>Instantaneous Field Of View</td>
</tr>
</tbody>
</table>
ABSTRACT

Abstract – This report presents the work conducted at Safran Electronics & Defense in the context of my master thesis-internship. Vision-hybridisation with inertial systems has great potential in navigation systems. This master thesis investigates this potential and evaluates it on real data of trajectories for one specific hybridisation: loosely-coupled via an Extended Kalman Filter. The vision hybridisation is developed in order to limit the drift of position of the state-of-the-art GPS-hybridisation when the GPS is not available. This thesis presents hybridisation tests on different trajectories that were either simulated from scratch or reconstructed from real vehicle route (stimulation). Using real data for research work brings technical challenges. Some are presented, as well as the proposed solutions to tackle them.


Résumé – Ce rapport présente le travail réalisé à Safran Electronics & Defense lors mon stage de fin d'études. L'hybridation inertie-vision a un grand potentiel dans le domaine des systèmes de navigation. Cette thèse de master a pour but d'évaluer ce potentiel à partir de données d'essais réels pour un type d'hybridation en particulier: hybridation lâche par filtre de Kalman étendu. Cette hybridation a pour but de limiter la dérive des centrales inertielles lorsque l'hybridation GNSS n'est pas possible. Cette thèse présente différents tests d'hybridation par la vision sur des trajectoires qui sont soit complètement simulées soit des rejou d'essais réels. Rejouer des données d'essais dans le cadre de la recherche amène des difficultés. Cette thèse en présente un certain nombre ainsi que les solutions qui ont été proposées.
1. INTRODUCTION

Self-localisation systems are at the core of many applications and reliable systems open new horizons of innovation. The applications are very diverse: self-localisation of autonomous vehicle, localisation of cargo ships to optimize their routes, self-localisation of unmanned air vehicles (UAV) and many military applications such as missiles targeting, soldiers’ equipment…

Safran Electronics & Defense provides high technology inertial navigation systems for air, land and sea vehicles. These devices aim to deliver the position and orientation of the mobile at any time, in any conditions and anywhere on Earth. To reach this objective, they undergo constant improvements to face the hardest conditions.

Today’s inertial navigation systems are commonly fused with a GNSS to compensate inertial deviations and maintain a high accuracy over time. However, GNSS is not 100% reliable: the vehicle can lose the GNSS signals because of tunnel, jammed signal, satellite failure... and GNSS accuracy can just not be enough for some applications such as autonomous vehicles.

Computer vision as a navigation tool has already been introduced in the robotics field with the Simultaneous Localisation and Mapping (SLAM) and more recently with the Visual-SLAM (VSLAM), which uses real cameras.

This report presents an evaluation of a specific hybridisation of inertial systems with vision by using real data (IRS data and images) extracted from real time trajectories. The hybridisation is done in two steps: first, images are used by a VSLAM algorithm in order to get a first estimation of the trajectory using only vision. This first estimation of position is only “proportional” to the true navigation, as it lacks the estimation of a scale factor. As it will be explained further, using only images from one single camera does not give any information on how much distance the vehicle covered. The second step is the implementation of this estimation as measurements in an EKF to improve the navigation computed by integrating the inertial measurements.

This project is part of a broader development project: the autonomous vehicle. The autonomous vehicle requires very accurate localisation information, by using not too expensive sensors. Inertial Measurement Unit is very interesting because it works independently of the environment of the vehicle and provides measurements at a high frequency. Camera is a good solution because it is affordable and has the potential to provide accurate observations. Indeed, humans mostly use their vision to locate themselves in their environment, so vision for vehicles seems like a perfect match. Without vision, one can only rely on organs such as utricle and vestibule of the ears that sense the motion of the body, similarly to an IMU. To continue the analogy with the human body, the addition of vision would be like to allow someone to use his eyes to locate himself.

However, to go from images to motion estimation is a complex task. To use vision optimally for in-motion systems is a big challenge in today’s research but has a very big potential. Camera sensor errors are badly handled in the vision algorithms, and so the parameters of the camera need to be known perfectly to obtain good estimation of the trajectory by vision only.

This thesis is addressed to people that know the basic notions of Kalman Filtering and navigation systems. A brief cover of the basic knowledge required to fully understand this thesis is presented in the appendices.
2. INDUSTRIAL CONTEXT

2.1. SAFRAN

2.1.1. Global presentation

Safran is a leading international high-technology group with three core businesses: aerospace propulsion, aircraft equipment and defense.

- **AEROSPACE PROPULSION**
  - SAFRAN AIRCRAFT ENGINES
    - Engines for commercial and military aircraft. Electric propulsion and propulsion systems for satellites and orbital vehicles.
  - SAFRAN HELICOPTER ENGINES
    - Turboshaft engines for civil and military helicopters, auxiliary power units (APUs), starting and propulsion systems for missiles, target drones and unmanned aerial vehicles (UAVs).
  - SAFRAN CERAMICS
    - Center of excellence in high-temperature composite materials. Development of ceramic technologies.
  - SAFRAN AERO BOOSTERS
    - Low-pressure compressors for aircraft engines. Equipment for aircraft and spacecraft. Test cells and equipment for engine testing.
  - ARNAGROUP
    - Commercial and military space launchers, propulsion systems and related equipment, products and services.

- **AIRCRAFT EQUIPMENT**
  - SAFRAN LANDING SYSTEMS
    - Aircraft landing and braking systems. Control and monitoring systems.
  - SAFRAN TRANSMISSION SYSTEMS
    - Mechanical power transmission systems for commercial and military airplanes and helicopter engines. Mechanical components for airplane and helicopter propulsion systems.
  - SAFRAN ELECTRICAL & POWER
    - Electric power systems for the aerospace market. Engineering solutions for aerospace and other sectors.
  - SAFRAN NACELLES
    - Complete nacelle systems for aircraft engines (civil aircraft and business jets through to the largest airliners). Composite materials for aerostructures.
  - SAFRAN VENTILATION SYSTEMS
    - Ventilation equipment for civil and military aircraft.

- **DEFENSE**
  - SAFRAN ELECTRONICS & DEFENSE
    - Equipment and systems in optoelectronics, avionics, navigation, electronics and critical software for civil and defense applications.

Its global presence allows the Group to build industrial and commercial relationships with the world’s leading prime contractors and operators, while delivering fast, local service to customers around the world. Comprising a number of companies, the Safran group holds, alone or in partnership, world or European leadership positions in its core markets.

Safran develops, produces and markets engines and propulsion systems for civil and military airplanes and helicopters, ballistic missiles, launch vehicles and satellites. It also provides a wide range of systems and equipment for civil and military airplanes and helicopters.

Operating in the optronic, inertial guidance, electronics and safety-critical software markets, Safran offers today’s armed forces a complete range of optronic, navigation and optical systems and equipment for use in the air, on land and at sea.

2.1.2. Key figures

Operating worldwide, the Group has more than 65,000 employees and a revenue of 16.5 billion euros in 2017.

The aerospace, defence and electronics markets are global. Safran is an international group, operating in more than 50 countries through local facilities that guarantee customers close support and quick responsiveness. The
Group continues to invest in its industrial facilities around the world, as well as opening major new production plants in Europe, thus ensuring the long-term viability of its skills bases in their traditional areas of employment.

![Safran - worldwide](image)

Figure 1: Safran worldwide.

Working alone or in partnership, Safran holds world or European leadership positions in its core markets. In the Aerospace core, Safran is the world leader in commercial aircraft engines (mainline jets with more than 100 seats, in partnership with GE), in helicopter turbine engines, in landing gears, in wheels and carbon brakes, in aircraft wiring and in helicopter flight controls. Considering the Defence activity, it is number 1 in Europe in inertial navigation, optronic systems and tactical drones.

### 2.2. **SAFRAN ELECTRONICS & DEFENSE**

Safran Electronics & Defense, a high-tech company in the Safran group, holds world or European leadership positions in optronics, avionics, electronics and safety-critical software for both civil and military markets. Safran Electronics & Defense is the No. 1 company in Europe for inertial navigation systems (INS) used in air, land and naval applications. It is also the world leader in helicopter flight controls, high-performance space optics. It is also the European leader in optronics and tactical UAV systems. Operating across the globe through the Safran group, Safran Electronics & Defense and its subsidiaries employ 7,850 people in Europe, Southeast Asia and America.

The Avionics division’s know-how in inertial navigation and civilian and military avionics is renowned worldwide. Its navigation, guidance and pointing systems are used in numerous countries’ airplanes, surface ships, submarines, missiles, armoured vehicles and land weapon systems. Its avionics equipment and systems – onboard information and piloting systems, cockpit displays, multifunction screens, etc – are in use in the largest airplane and helicopter programs. Lastly, its information and mission preparation systems are used by numerous armed forces.
2.4. EXCELLENCE CENTER FOR NAVIGATION SYSTEMS (CESN)

As a subdivision, the CESN is specialized in the navigation systems; especially in inertial navigation.

European leader in inertial navigation, Safran Electronics & Defense offers a wide range of inertial systems and hybrid systems for submarines and surface ships, land vehicles, aircrafts and helicopters. Safran Electronics & Defense has eighty years of expertise in gyroscopes and in mastering most technologies: mechanical gyroscopes, fiber optic, laser, hemispherical resonant, vibration, etc. Its expertise has allowed, for example, to be selected for the development of the inertial navigation system of the Airbus A400M or the Rafale F1.

Safran Electronics & Defense constantly mobilizes significant resources in R&D to develop tomorrow’s technologies, and to provide today the best products to its customers. Its 100% proprietary technologies cover all areas of navigation. Thanks to their experience of managing large programs in this specialty, the R&D teams of the CESN imagine and design equipments and systems for the aviation and the land vehicles. This centre includes most of the Research and Development Unit of the Defence & Security division of Safran Electronics & Defense and hosts 700 people working on the site of Eragny near Cergy-Pontoise, in Paris’ area.

Hemispherical Resonator Gyros (HRG)

BlueHalo®

Inertial navigation systems

Sigma-40XP
3. LITERATURE REVIEW

This literature review has two objectives:
- Introduce the basics of localisation systems of a vehicle
- Produce a state of the art of the more recent literature on the specific topic of the loose fusion of inertia (IMU) and vision (camera).

The first objective has been completed by reading documents published within the company. To achieve the second one, a deep research of academic literature has been conducted. The methods of visual odometry have not been at the centre of the research, but the methods to couple the output of those visual odometry algorithms, i.e. rotation and translation minus a scale factor, with the inertia measurement of the IMU (accelerometers and gyro measurements).

Different methods of fusion exist. The most used is the Extended Kalman Filter (EKF), which is the filter used for this project. However, this method has its drawbacks, mainly because of the non-linearities of the system. Indeed, the linearization of the system gives only an approximation of the solution, and the system is linearized around an initial estimation of the state. If the initial approximation is wrong, the solution may diverge. In our case, it's the pose that is estimated via the IMU that is used as initial estimation. Besides, the Gaussian distribution of the measurements and states of the filter is only assumed but it is questionable for the system.

3.1. DIFFERENT APPROACHES OF FUSION

3.1.1. Extended Kalman Filter (EKF)

In the vast majority of the scientific papers the inertia-vision coupling is done via EKF.

However, there are small differences in the architecture of the filter. The first one is the implementation of the visual measurements: it can be loose or tight. In both case, the images are processed in order to extract some features of each images. If it is loose, the images are further processed to get estimations of the pose of the camera by vision algorithms that are discussed further in this chapter. These estimations are then implemented in the EKF as measurements. For a tightly-coupled visual-inertial fusion, the 2D position on the images are given as the measurements. The 3D position of these feature points are actually states of the filter and are estimated, as it is done in paper [1]. A loosely-coupled visual-inertial fusion requires more pre-processing on the image and generally, several images are required to obtain an estimation of the pose (measurement for the EKF), whereas when it is tight, one image corresponds to one (set of) measurement and so one update of the filter. However a tight fusion leads to larger computational load for the EKF which is currently a problem for its implementation in a real-time system.

Paper [2] discusses the implementation of the EKF for a drone. It gives two different approaches to the EKF: the first one is said "direct": the states of the filter are position, speed, attitude, biases of the sensors… whereas for the second approach, said "indirect", the states of the filter are the error of the position, speed, attitude and biases of the sensors. Indeed, the pose of the vehicle is first estimated at high frequency by the UMI measurements integration and, at a lower frequency, the EKF estimates the errors made by doing this first integration. This second method is the one used for the thesis.

For the large part of the EKF that are presented in the scientific literature, other states are being estimated. For example in paper [3], where the approach is "direct", the distance and the rotation between the UMI and the camera are states of the EKF. This calibration is discussed later in this review.
3.1.2. Unscented Kalman Filter (UKF)

In paper [4] is proposed the UKF to estimate the classical states: pose, speed, biases of the sensors and position of the feature points (tight fusion). The filter first creates several vectors \( \chi \) called sigma points "close" to the state vector of the filter. The vectors are close in the sense that they are in the space delimited by the covariance of the state vector. A specific algorithm is used to choose these sigma points which are presented in paper [4]. For each sigma points is associated a weight matrix \( \varpi \). These sigma points are propagated in the system as for the usual Kalman Filter. The a priori state estimate \( \hat{x}^- \) and its associated covariance matrix \( P^- \) are calculated by taking the weighted mean of the sigma points \( \chi_i \) (with weight for the mean \( \varpi_0 \)) and the sum of the weighted covariances (with weight \( \varpi_0 \)) as follow:

\[
\hat{x}^- (t) = \sum_{i=0}^{2N} \varpi_i \chi_i (t) \\
P^- (t) = \sum_{i=0}^{2N} \varpi_i \left( \chi_i (t) - \hat{x}^- (t) \right) \left( \chi_i (t) - \hat{x}^- (t) \right)^T
\]

Concerning the update step, when a measurement arrives, the predicted measurement vector for each sigma points is determined. As for the a priori state estimate, the final predicted measurement vector is the sum of the weighted predicted vectors determined from the sigma points. The expression of the Kalman gain is different from the usual Kalman Filter too; it considers all the covariance of the predicted measure vectors and the covariances between these vectors.

Having the a priori state estimate, the a priori covariance matrix estimate, one predicted measurement vector and a Kalman gain, the remain steps of the update phase are similar to the ones of the regular Kalman Filter.

In [5] is discussed the interest of the UKF over the EKF. The two Kalman filters are both tested for the localisation of the end-effector of an industrial manipulator. The computational cost is slightly larger when using the UKF and the accuracy is similar to the solution obtained by using the EKF. Several articles emphasizes that using UKF is difficult to implement in an embedded system because of its big computational cost.

3.1.3. Particular Filter

In papers [6] and [7], a particular Filter is used for the inertial-visual fusion. Without going in details, the main advantage is that there are no needs to assume any Gaussian distribution for the state variables. But as the UKF, the main problem is its large computational cost because several calculations are executed in parallel (in a similar way as the UKF). The two studies apply the Particular Filter for systems evolving in 2 dimensions, which decrease the computational cost.

3.1.4. Nonlinear optimization

Paper [8] challenges the use of filters for inertial-visual vision. The authors argue that a non-linear optimization approach gives a better accuracy for a similar computing cost. For the filters, the integration of the inertial measurements are first used to estimate the localisation of the vehicle and visual measurements update this estimation.

Paper [9] describes such an approach. The goal is to minimize the following cost function:

\[
J(x) := \sum_{i=1}^{I} \sum_{j=1}^{K} \sum_{J \in J(i,k)} e_i^{j,k} W_i^{j,k} e_i^{j,k} + \sum_{k=1}^{K-1} e_k^{k} W_k^{k} e_k^{k}
\]
The term $e^{i,j,k}_r$ correspond to the projection error of the feature point $j$ on the image $k$ of camera $i$. It is defined as follow:

$$e^{i,j,k}_r = z^{i,j,k} - h_i\left(T_{Ci}^k T_{SW}^k W^T\right)\text{measure argument of the optimization}$$

$z^{i,j,k}$ is the position (in terms of pixels) of the feature point $j$ on the image $k$ of the camera $i$. $h_i(\cdot)$ is the projection model function that projects the feature point $j$: 3D position $c^{i,j}$ into its 2D (pixel) position on the image of the camera. The position of the feature point and the rotation between the world frame [W] and the frame of the inertial sensors [S] are arguments of this optimization. This error term will be discussed later in the visual odometry part.

The term $e^k_s$ corresponds to the inertial errors between the predicted pose of the vehicle at time $k+1$ (obtained with the IMU model and measurements) and the real pose of the vehicle that is to be optimized.

The two errors terms are weighted by weight matrices $W^{i,j,k}_r$ and $W^k_s$ that are related to the covariance matrix of the state and the error model of the measurements.

This method, briefly described here, has been performed on real data in paper [9] and compared to a more usual filtering approach. The conclusion is that nonlinear optimization gives better accuracy but it comes with a bigger computational cost.

### 3.2. IMPROVEMENT OF VISION-COUPL ED EKF

#### 3.2.1. Self-calibration IMU - Camera

Paper [4] describes a method to estimate the exact pose of the camera relatively to the IMU. The method consists in first measuring "by-hand" the distance and orientation between the IMU and the camera as a first approximation. Then, these position and orientation variables become states of the Kalman Filter (Unscented in the experiment of the paper). The states are not propagated but only updated, based on the visual odometry and inertial measurements. The experiment results show that this relative position and orientation is better than the initial measurement.

#### 3.2.2. Observability Constrained Visual Inertial Navigation System

Papers [10] and [4] prove some observability properties of the visual inertial system. Paper [29] demonstrates that a system using only one of the two sensors: IMU or camera is not observable. For example the calibration of the camera relatively to other parts of the vehicle cannot be estimated only using one single sensor. But with the coupling of the two sensors with enough dynamics, the calibration camera-IMU, inertial biases and the gravity vector can be estimated.

Papers [11] and [12] show that with EKF being linearized at estimated points that are not exact, it can create inconsistency in the system, i.e. the system gains some wrong extra observability. Paper [12] proposes to constrain the observability of the system by modifying the Jacobian matrix. To do so, the kernel of the observable function of the nonlinear system is calculated in order to identify the non-observable modes, which are the translations of the \{vehicle + feature points\} and its rotation around the vector of gravity. The kernel of the linearized observable function (matrix of observability) is then constrained to keep the same kernel as in the nonlinear system. This is
done by modifying the propagation and observation matrices. It has been proven that it reduces the risk of inconsistency in the system (variances to small).

### 3.3. VISUAL ODOMETRY

#### 3.3.1. Match pixels from different images

The first step of visual odometry is to be able to match successive images of same image flow. In other words, it is necessary to be able to recognize the displacement of pixels, whether a specific set of pixels or all the pixels, on the image. Paper [16] presents the two main methods.

The first one is the tracking of feature points on the image (“feature-based” method). The idea is to extract a finite number of feature points easily recognizable on the images. There exist many algorithms that can be used to detect these points, such as the Features from Accelerated Segment Test (FAST algorithm) presented in paper [17]. It detects "corners" on the image by looking at the difference of intensity between one pixel and the pixels directly next to it. If the difference is larger than a specific threshold, the pixel becomes a feature point 2D projection. Then these points are tracked on the images.

The second one, presented in paper [18] is the "Direct" method. It is based on the intensity of the pixels in the whole image. For two successive images, let be:

- $J$ and $I$ two functions that map the photometric intensity of pixels of respectively the 1st image and the 2nd image. They are functions of the pixel position.
- $\mathbf{x}$ the 2D position coordinates of a pixel in the first image
- $\mathbf{v}$ the displacement model function of the pixels from the 1st image to the 2nd.

We then have the following relation:

$$J(x, y) = I(x + u(x, y), y + v(x, y)).$$

For pixel displacements small enough, we can write: $I_x u + I_y v + I(x, y) - J(x, y) = 0$, with $I_x$ and $I_y$ the spatial derivatives respectively in the direction of $x$ and $y$, at the point $(x, y)$.

The displacement functions are modeled as affine functions: $u = a_1 + a_2 x + a_3 y$ and $v = a_4 + a_5 x + a_6 y$.

By combining these equations, it results a cost function representing the photometric error to minimize on the coefficients of the displacement model:

$$E(a_1, ..., a_6) = \sum_{(x,y)} (I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I(x, y) - J(x, y))^2$$

By getting the coefficients of the displacement model, every pixel displacement is known on every frame.

Papers [16] and [19] proposes a new method that is said to be 2 times faster than the previous methods.
3.3.2. Bundle Adjustment algorithm

The algorithm is presented in Appendix 3.

In paper [15] is proposed a bundle adjustment algorithm for a system with 2 cameras (left and right) and with the addition of an inertial constraint. The "modified" cost function is as follow:

\[
(R, t) = \arg \min \left( \sum_{i=1}^{N} \left\| l_i - \pi^l(P_i; R, t) \right\|^2 + \left\| r_i - \pi^r(P_i; R, t) \right\|^2 \right) + (1 - w) \left\| l_{imu} - (R, t)^T \right\|^2
\]

\( R \) and \( t \) are the rotation and translation of the camera between two moment. \( P_i \) is the 3D position of the feature point \( i \), \( l_i \) and \( r_i \) are the 2D positions of this on the images of respectively the left and right cameras. \( \pi^l \) and \( \pi^r \) are the camera projection models of the cameras. They are functions of the positions of the feature points and the estimated position and rotation of the camera. The term \( \left\| l_{imu} - (R, t)^T \right\|^2 \) corresponds to the inertial constraint. \( l_{imu} \) is the rotation and translation calculated via the IMU.

It is very similar to the nonlinear optimization problem presented in the first part of this review, except here the inertia is viewed as a constraint in order to have a more accurate vision estimation.

3.3.3. The scale factor

Only using a single camera to estimate the whole rotation and translation of a camera is impossible. Indeed, by seeing a certain object on the image, it is impossible to know its true size. As for a pixel, even if we know the angular resolution of the camera, an image does not give any information about the distance between the camera and the surface covered by the pixel.

Different methods exist to estimate this scale factor:

Paper [14] proposes the use of the inertia as a constraint in the vision algorithm as presented previously, to estimate the scale factor.

If the optic system is constituted of more than 1 camera and the general pose of the cameras are accurately known, it is possible to use triangulation to get this scale factor.
4. THE SIMULATOR

For this thesis, a simulator created in Safran Electronics & Defense has been reused and adapted. This simulator integrates IMU acceleration and angular rate in order to compute a whole trajectory. Then it is possible to activate different options of sensor fusion: GNSS, barometer, tight vision and loose vision. All these sensors are simulated, as for the IMU accelerations and angular rates. These simulated sensors are integrated via an EKF.

The purpose of doing simulations is to validate the loose integration of measurements from vision sensors with the inertial measurements by using an EKF, and compare it to other measurements such as GNSS. This simulator has then been adapted to be able to work on real trajectories with real measurements (IMU increments and images).

4.1. PRESENTATION OF THE SIMULATOR

4.1.1. Simulation of the Pure Inertial Navigation System (INS)

A software, not in the simulator, creates "perfect" IMU measurements (angular rates and accelerations in the three directions), at high frequency (100Hz), that correspond to a trajectory that has been giving as input. This perfect IMU measurements (with the time of this measurements) are the only input when doing simulations.

These perfect increments are integrated, using dead reckoning method, to obtain the trajectory of reference, which is the perfect simulated trajectory, i.e. perfect speed, position and attitude (roll, pitch, yaw).

To get the simulated, stochastic errors are added to the perfect ones that simulated the sensor error of the IMU. To get the Pure INS trajectory, these simulated increments are integrated as for the perfect ones.

4.1.2. Simulation of the Vision observations

4.1.2.1. Loose integration

For loose integration, the observations are rotations and translations (minus the scale factor) between images. The images are not simulated, the rotations and translations are simply simulated by using the trajectory of reference and calculating the rotation and translation during one Kalman cycle (3.84 seconds). Gaussian errors are added to these observations to get simulated observations. Moreover, a small misalignment of a Gaussian distributed angle (mean 0 and standard deviation of 10 µrad) between the camera and the body of the vehicle is simulated.

4.1.2.2. Tight integration

For tight integration, the observations are feature points 2D position on images. However, because for tight integration the 3D positions of feature points are also estimated in the EKF, the 3D feature points are created as well as their projection on the images (2D pixel coordinates) according to the trajectory of the vehicle. This type of fusion is not studied further in this thesis.
4.1.2.3. Other sensors

The GNSS is loosely coupled, i.e., the GPS observations are the positions of the vehicle. The GNSS and barometer observations are simply simulated by using the reference positions: altitude, longitude and altitude (only the latitude for the barometer) and normally distributed errors.

4.1.3. Modifications of the simulator

4.1.3.1. The covariance matrix properties

The simulator has known several updates by different people. The last version added the tight vision coupled. A quick comparison between the two previous versions showed some other modifications.

One is the addition of the Joseph Formula for the calculation of the updated covariance matrix:

$$P_{k+1,k+1} = (I - KH^T)P_{k+1,k}(I - KH^T)^T + KRK^T$$

instead of the usual equation

$$P_{k+1,k+1} = (I - KH^T)P_{k+1,k}.$$

Using this formula assures that the covariance matrix remains symmetric and positive during this step. The covariance matrix can lose these properties with the first formula because of numerical computation. It is not rare in EKF for navigation because the covariance matrix is often ill-conditioned. The reason is that its eigenvalues are very different. For example, by considering the diagonal initial covariance matrix, the variance of the position is around 10⁶ larger than the attitude variance.

However this formula may not be sufficient to keep the properties of the covariance matrix for long simulation. That's why a modification of the EKF has been implemented: the Bierman filter.

Without going in details, the covariance matrix is decomposed as $P = U. D. U^T$, $U$ being an upper triangular matrix and $D$ a diagonal matrix. Then all the updates of the covariance matrix are made by using $(U, D)$ instead of $P$ which assures that the covariance matrix always remain symmetric.

4.1.3.2. Control of random number generation

Stochastic functions and errors are used to simulate real conditions. For example, the simulator receive as input "perfect" IMU measurements. These measurements are integrated to get the trajectory of reference. To get simulated IMU measurements that can be used for the computation of the trajectory, random errors (with variances given as parameters) are added to the perfect IMU measurements to simulate the errors of the sensor.

It is then required to have the same seed for each simulation in order to compare results. The best solution is however to run several simulations (with different seeds) in order to apply the Monte Carlo method but it increases drastically the computational time.
4.1.4. Simulations for different trajectories

The simulator has been tested on different trajectories that have been created independently. The trajectories and the IMU measurements are created and used as input of the simulator. Three trajectories are tested: a racetrack that simulates the trajectory of a car, an airplane flight and a helicopter flight. An important point is to adapt the parameters of the simulator for each trajectory, because the vehicles are quite different and so are the trajectories. The parameters are for example the sensor errors of the IMU (way smaller for the helicopter and airplane than for the car), as well as the initial values of the variance of position, attitude and speed. For example, the racetrack trajectory having a smaller duration and the speed of the car being way smaller, the corresponding initial variances must be smaller.

<table>
<thead>
<tr>
<th>Trajectories</th>
<th>Duration (s)</th>
<th>RMS horizontal position Pure Inertia (m)</th>
<th>RMS horizontal position PI+Vision (m)</th>
<th>RMS horizontal position PI+GNSS (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Racetrack</td>
<td>1000</td>
<td>42</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Airplane flight</td>
<td>5000</td>
<td>2293</td>
<td>480</td>
<td>21</td>
</tr>
<tr>
<td>Helicopter flight</td>
<td>5500</td>
<td>1222</td>
<td>228</td>
<td>14</td>
</tr>
</tbody>
</table>

RMS means Root Mean Square errors, it is the square root of the mean of all the square of the errors (saved every 4s here). RMS horizontal position means it is the 2-norm of the RMSs of latitude and longitude. This is a good indicator to understand how accurate an estimation is.

The results show that vision improves well the inertia only navigation but does not reach the performance of the GNSS coupled navigation. The performances depend on the choice of parameters of the IMU and errors of the measurements.

These simulations also show the response to initial errors as seen on the racetrack plot. The GNSS coupled navigation retrieves quickly the trajectory of reference contrary to the vision-coupled navigation. This comes from the fact that vision only measures displacements and rotation, not absolute pose and so it cannot rectify initial errors of position and attitude.
4.2. ADDITION OF THE STIMULATION

A stimulation is a simulation where the input data (IMU increments, GPS measurements, images…) have been recorded during real time trajectories, i.e. comes from a field test. For simulations, the inputs are simply perfect IMU increments whereas for stimulations, they are the real IMU increments, the sensors measurements (GNSS, barometer…) and the trajectories computed by the IRS in real time during the motion of the vehicle. In the data used there are 2 navigation estimates: one with only the INS and barometer and the other one with INS and barometer and GNSS. As a consequence, the structure of the simulator needs to be slightly changed.

The navigation of reference becomes the real time GPS-coupled navigation. However, one first objective is to retrieve the pure inertia navigation calculated in real-time using only the IMU increments and so for that specific step, the navigation of reference is the real time pure Inertia navigation.

The sensors measurements must be well formatted to be integrated in the simulator. One difficulty is that each measurement is associated with a timestamp, and the sensors may have different frequencies. It requires extra attention to "synchronise" all these measurements with the different cycles of the simulator (integration of the navigation, EKF cycles…).

Finally, each set of real data brings specific difficulties that need to be tackled. The implementation of helicopter data presented in the next part will show some of them.
5. STIMULATION

5.1. PRESENTATION OF THE DATA

The data are measurements of the real IRS output of a helicopter during its flight. The data used for the stimulation for simulating the inertia-only navigation consists in: IMU increments, GNSS-coupled and inertia-only navigation computed in the embedded system during the flight, timestamps of the data.

5.2. INERTIAL-ONLY NAVIGATION

The first step to do the stimulation is to validate the integration of the inertial increments. As the inertial-only navigation that has been calculated in real time is given in the data, the integration of the increments in the simulator should logically give the same navigation. However it is not the case, the position error reaches more than 10 km at the end of the simulation compared to the real time inertia-only navigation calculated during the flight of the helicopter. Several reasons have been investigated and different solutions have been proposed to tackle these issues.

5.2.1. Barometer measurements

In addition to the measurements of the IRS, the barometer measurements are used in the so-called "Inertia-only" navigation. The GPS measurements of altitude are used instead to replace the barometer to stabilize the vertical direction.

The "purely inertial" localisation always needs an additional observation in order to stabilize the vertical direction. The reason is that a sensor error of the accelerometer on the z direction causes a divergence. For example if the z-accelerometer has a bias of 50 µg, the estimated altitude will then be larger than the actual value. The consequence is that the gravity estimated at the estimated altitude will be smaller than the actual gravity on the vehicle. As a consequence, during the next integration of the increment, the gravity term that will be removed from the z-accelerometer measurement will be smaller, so the IRS will estimate a larger (upper) acceleration in the z-direction than the real acceleration. And so the altitude error will increase more and more.

5.2.2. Missing frames

During the extraction of the data, 108 frames were missing. This means that 108 sets of IMU increments are missing. These frames are missing in the same time interval, between 2200s and 2500s of flight. These creates big errors of navigation during and after that time interval. This issue is tackled by replacing the missing values by means of interpolation. Two interpolations were tested: the first one calculates the mean value between the previous increment and the following one. The second interpolation takes the mean of the 5 previous and 5 following values. The second method was more successful. As it can be observed on the following figure, it improves the simulated navigation a lot.

However, the interpolation is not good enough as some errors remain. It has been then decided to limit the simulation of the flight up to 2200 seconds so that no missing frames would disturb the simulation.
5.2.3. Lack of static alignment

The recording of the data begins when the helicopter is already flying. It is an issue because every Inertial Reference System is supposed to do a static alignment. A static alignment is done during a static phase before any movement of the vehicle. The principle is that it computes the navigation via an EKF by using zero velocity as a measurement of the EKF. In static, the only force acting on the vehicle is the gravity. By identification on the accelerometer measurements, it is easy to retrieve the pitch and roll of the vehicle. Then by identifying the projection of the rotation of the Earth on the gyrometer measurements, it is possible to retrieve both the yaw and the latitude of the vehicle. Moreover, the alignment phase shows some correlations between sensor errors and errors of navigation. Finally, some vehicle does a "double-alignment" which is two consecutive alignments with the IMU being turned of 180° between the two alignments. This method permits to estimate some sensor errors. However it is quite uncommon to do a double alignment.

Because no data are available during the static phase, the static alignment cannot be simulated. This is one argument to not have any initial navigation errors.

5.2.4. Gradient descent method to estimate the angles between the body (vehicle) and the IMU

The navigation errors of the Pure Inertia solution computed by the simulator were larger than the ones of the Pure Inertia navigation calculated in real time during the flight of the helicopter. One reason would be a problem regarding the harmonization of the IMU frame regarding the body frame. For simulations, the two frames are perfectly aligned but for real trajectories, they are usually not. The angles are measured before the flight but unfortunately, the values of these angles were not available. The angles are commonly defined as yaw, pitch and roll between the body frame [b] and the IMU frame [m]:

\[
T_{bm} = Rot_z(\text{roll}).Rot_y(\text{pitch}).Rot_z(\text{yaw})
\]

Without these angles, the navigation errors obtained by stimulations were quite large.

A gradient descent method has been used to estimate these angles. Gradient descent method is an optimization tool that use gradients of a specific cost function to decide in which direction should the parameters evolve so that it minimizes the cost function. The cost function method used here were the quadratic sum of the integrated errors of yaw, pitch and roll (to go from the geographic frame [g] to the body frame [b]) between the reference navigation and the one computed by doing the stimulation:

The parameters to estimate and so the variables of the cost function were the yaw, pitch and roll between [b] and [m] as presented previously.

At each iteration k is calculated the gradient (using numerical approximation) at a certain point \(\bar{x}_k\). The gradient being directed towards increasing values of the cost function, the point of the next iteration is

\[
\bar{x}_{k+1} = \bar{x}_k + \frac{d\bar{x}}{d\bar{x}}
\]

with

\[
\frac{d\bar{x}}{d\bar{x}} = -\gamma . \text{gradient}.
\]

The initial value is set to \([0,0,0]\) and after more than 70 iterations, the final value gives a yaw harmonization angle of 1.39 mrad, pitch harmonization angle of 0.05 mrad and a roll harmonization angle of 1.23 mrad.
This method is not optimal, as it can only detect local minimums. A first idea of where is the minimum is therefore very useful to have. Moreover, the choice of the value for $\gamma$ is very important because it defines how big will be the steps at each iteration. There exist some formulation for convex cost functions but in this case, nothing assures that the cost function is convex. To get an algorithm which is both fast enough and accurate, it is proposed to adapt the value of the $\gamma$ between each iteration. If the sign of the gradient is the same as the previous iteration, the value of $\gamma$ is slightly increased so the algorithm will reach quicker the local minimum because the steps will tend to be larger. However, if the sign of the gradient is not the same as in the previous iteration, the value of gamma is slightly decreased because it means that the step of the previous iteration was too big and that there is a local minimum between the point of the previous iteration and the current one.

These harmonization angles cannot be neglected for high accuracy navigation performances. As it can be seen on these following plots, it really improves the performances, both in attitude and position.
5.3. ADDITION OF THE SIMULATED VISION MEASUREMENTS

Before using the real images took during the helicopter flight, it is interesting to simulate the vision measurements and hybridize those with the inertia measurements. It is useful because it helps understanding how large can be the errors of the vision measurements so that the hybridisation is actually improving the navigation, compared to using only IMU measurements. Moreover, it gives an idea of how much the vision can improve the navigation.

The position and attitude errors are plotted below: the longitude is greatly improved. The 3 sigma envelop (means that the error has only a 0.3% probability to go beyond this envelop) is greatly reduced.

Regarding the attitude errors, adding the vision measurements does not seem to have a huge impact, except reducing the 3 sigma envelop.
6. VISION ALGORITHM

For this thesis, vision algorithms are used to provide observations for the EKF: rotations and translations of the vehicle. The algorithms work on images flows, and estimate the movement of the camera by studying how the environment seen on the images is moving. Many vision algorithms are open-source and can be found on the internet. The main work of this thesis being on the hybridisation, the algorithms are not detailed here.

6.1. PARAMETERS OF CALIBRATION

To get good levels of precision, vision algorithm cannot be applied to images without knowing perfectly the characteristics of the camera. The parameters that are required for visual odometry algorithms are presented below:

- Image size in terms of pixels (height and width)
- Calibration matrix (called the intrinsic parameters)

\[
\begin{bmatrix}
    f_x & 0 & x_0 \\
    0 & f_y & y_0 \\
    0 & 0 & 1
\end{bmatrix}
\]

are relating to the shape of the camera lens

- Distortion coefficients defined by the Brown model, introduced in the paper [21]. Some elements in the camera may not be perfectly aligned, which causes a distortion of the images: straight lines in the scene does not remain perfectly straight in the image. These distortions can be modeled by coefficients of radial distortion and tangential distortion.

- Extrinsic matrix: It is the 3x4 matrix that takes a point in the body/vehicle homogenous coordinates

\[
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix}
\]

and transforms it into the coordinate system \(\begin{bmatrix}
    x' \\
    y' \\
    z'
\end{bmatrix}\) of the camera coordinate system. The expression is \((R|T)\) with \(R\) the rotation from the body system to the camera system and \(T\) the translation expressed in the body frame.

6.2. APPLICATION ON THE IMAGES OF THE HELICOPTER

6.2.1. The data sets

The optic system used during the flight of the helicopter consists of numerous cameras located on the nose of the helicopter. There are 3 different sorts of cameras, each sort having different angles of view (and therefore different
intrinsic characteristics and poses). For the sake of simplicity, only one camera has been used, but it would be interesting to implement the images of the other ones. For the sake of simplicity, the chosen camera is the one having no significant misalignment with the body of the vehicle, i.e. the optic axe is parallel to the forward axis of the helicopter.

6.2.1.1. Parameters

The parameters are currently unknown. It is a problem as it affects a lot the performances of the VSLAM. However, it is possible to estimate some of the parameters, given some assumptions:

Focal lengths: it is assumed the focal lengths are equal in every directions. In practice, if there are not equal, the values are very close. By considering the diagonal distance of the image (assumed not rectified) and the range angle in diagonal, the focal lengths can be calculated.

Another way to compute the focal length is by using the angular resolution of the camera. The data collected give the angular resolution at the center of the sensor which is called the Instantaneous Field of View (IFOV). It means that 1 pixel at the center of the detector corresponds to an angle of field of view whose value is the IFOV of the camera. By using the relation above, we can compute the focal length using the IFOV.

\[
IFOV = 2 \tan^{-1} \left( \frac{d}{2R} \right)
\]
\[
FOV = 2 \tan^{-1} \left( \frac{D}{2R} \right)
\]

6.3. OPEN-SOURCE DIRECT SPARSE ODOMETRY (DSO) VSLAM

The data have been tested with an opensource VSLAM algorithm. The main difference with Bundle Adjustment is that its primal goal is to estimate the position of the environment around the vehicle. But by doing so, it estimates the pose of the vehicle too.

The code is open-source and can be found on https://vision.in.tum.de/research/vslam/dso

The "processed" output of the open VSLAM is presented on the plots below: the attitude with the yaw, pitch, roll and the positions: horizontal and the altitude. It is compared with the reference navigation and the simulated inertia-only stimulation.
The plots of positions above have been adjusted by a scale factor that has been estimated. The details of the process are described in the next part.

### 6.4. PROCESSING OF THE VISION OUTPUTS

The outputs of the two vision algorithms were not directly formatted as yaw, pitch, roll and position in the geographic frame. The output is formatted as a 3x4 matrix (R|T). The output data need to be transformed so the performance of the algorithms could be evaluated and compared to the reference navigation. Moreover, the scale factor is an issue for the comparison of the performance of position.

#### 6.4.1. Frames

For this software, the translations $T_x$ corresponds to the position of the camera at time $k$ in the frame of the camera at the initial time. The transformation is done as previously, except the translations are already in the initial camera frame.
The attitude are expressed as quaternions in the output. It gives the attitude of the camera at time \( k \): \( C_k \) in the \([C_1]\) frame. The quaternions are changed into the cosine rotation matrix \( R_{[c_i]-[c_k]} \) and transformed into \( R_{[g]-[b_k]} \) in a similar way.

### 6.4.2. Processing steps

To get the relative position of the camera at time \( k \) in the frame \([c_i]\), \( C_iC_k \) is projected on the frame \([C_i]\). The following equation is used to visualize the trajectory in the "static" frame of the camera at the initial time:

\[
\begin{bmatrix}
  x_k \\
  y_k \\
  z_k_{[ci]}
\end{bmatrix} = T_{ci/ck} \cdot \begin{bmatrix}
  x_k \\
  y_k \\
  z_k_{[ck]}
\end{bmatrix}
\]

Attitude

\[
T_{bl/bk} = T_{bc} \cdot T_{ci/ck} \cdot T_{cb}^T \\
T_{g/bk} = T_{g/bl} \cdot T_{bi/bk}
\]

Then the Euler angles (yaw, pitch and roll) are extracted from the \( T_{g/bk} \) matrix and compared to the ones given by the IRS.

### 6.4.3. Scale Factor

Because the translations obtained by the vision algorithms are only proportional to the true translations, the final trajectory in the frame \([g]\) cannot be compared to the true one. Two solutions are proposed that can decide how accurate the position estimation by vision is.

#### 6.4.3.1. Estimation of the scale factor

The first approach has been to try to estimate the scale factor using the reference navigation. Indeed, by having the true navigation of the vehicle, it is possible to estimate the displacement length. A gradient descent method has been used. The objective function to minimize is the horizontal position error between the trajectory obtained by vision and the trajectory of reference, the argument being the scale factor that multiply both the north and east coordinates of the vehicle. Even if the scale factor is related to the translations between the different camera positions, as the transformations to get in frame \([g]\) are affine, the scale factor can be introduced on the \([g]\) coordinates.

Taking the example of the KITTI vision-only navigation, the scale factor has been estimated up to 18.34 after 15 iterations. The following figures illustrate the vision-only estimated trajectory with and without the scale factor.
The trajectory with the scale factor presents obvious errors. Even if an error during the turn of the helicopter can be suspected to come from the vision algorithm estimation, a bad estimation of the scale factor can be another source of errors. Indeed, by doing this optimization, the hypothesis is made that the scale factor stays constant. But it is unlikely. It actually seems that the scale factor changes when the vehicle is changing directions.

The estimation of the scale factor is only for the evaluation of the vision performance and is not used during the hybridisation as it is a simulation of real-time estimation.

### 6.4.3.2. Normalize the translations

The estimated translations between camera positions being only proportional to the true estimation, the second approach is to only compare the normalized translations.

By applying this approach to the 240 first seconds of the helicopter data; with using the open-source VSLAM algorithm for the vision, it is possible to get the performance of the inertia-only simulation and vision-only algorithm compared to the reference normalized translations. The results are plotted on the following figures.

The vision only translations are not holding up the performance of the inertia only simulation, as the translations normalized coordinates are diverging. It is therefore useless to implement it into the EKF as it would only worsen the navigation, compared to the inertia only navigation. There are many possible reasons:

- The calibration parameters of the camera may not be accurate enough for using computer vision algorithms. Indeed, the results of vision algorithm are very sensitive to these parameters: the algorithm has been executed with many slightly different values for these parameters and the results were always noticeably different.
- The inertia-only navigation is simply too good to be improved. Above all for such a small amount of time, the drift of the IMU cannot be visible.
• The open-source algorithm is not accurate enough for this application. The open-source software is typically more aimed to robotic or UAV applications. Moreover, it is more used to reconstruct an environment than to estimate one vehicle pose.

In the next part of this thesis will be presented other data that have been implemented to experiment the hybridisation.
7. HYBRIDISATION

In this part is tested the hybridisation of the IMU and the camera observations. The IMU is at the core of a localization system, one of the reason being the high frequency of the measurements and so the high frequency of the estimations. However, using the IMU observations alone do not give accurate performance, because of sensor errors, such as biases of the accelerometers and gyroscopes, that add up and create a drift. That’s why one or several sensors observations need to be fused with the IMU observations: it updates the IMU estimations (via EKF) and so improves the performance and avoid the typical drift of the inertia-only navigation. This is an hybridisation between the IMU and another sensor, in our case a camera. The camera, or the GNSS, -only navigation are expected to be more accurate than the IMU only navigation. However, these two sensors (camera and GNSS) give observations at a lower frequency than the IMU and the availability of their observations depends on the environment outside the vehicle, contrary to the IMU that is only using the inertia of the vehicle.

7.1. THE DATA SET

To do that, we use a new data: a dataset from the Kitti odometry benchmark training set (the 2011_09_26_drive_0009 set that can be found on http://www.cvlibs.net/datasets/kitti/raw_data.php). The data consists of color stereo sequence of the images recorded during the test. These are the input of the vision algorithm. With these images are also the camera calibration data, the data of the GNSS-coupled navigation and the IMU measurements.

Using this dataset presents many advantages:
- It is open-source and so it does not hold any confidential elements. Moreover, the results obtained could be compared with other results.
- The calibration parameters are precisely given and so it should not bring any problems regarding the results.
- The IMU used during the KITTI test is not very accurate, which is actually a rather good thing for this thesis. Indeed, the worse the inertia-only navigation is, the bigger the correction by vision will be.
- The duration of the test is way shorter than with the helicopter, which means that simulations and algorithms computational time will be shorter.

To evaluate the performance of the algorithm, we use the output data of the onboard IRS coupled with GNSS that are available with the sequence of images as the reference navigation. It contains the latitude, longitude, altitude, yaw pitch and roll and other inertial information that are not needed for the validation. The GNSS-coupled navigation is not the real trajectory of the car but the vision-only estimated trajectory is expectedly a lot less accurate. Moreover this part aims to demonstrate that vision can improve inertia-only navigation.

The benchmark test is a car driving for 42 seconds: it goes straight for the 20 first seconds, then turning 90° left at low-speed and then driving straight 10 more seconds and then it stops. The following figure is the trajectory of the car, expressed in the geographic frame [g].
7.2. STIMULATION OF THE INERTIA-ONLY NAVIGATION

The IMU increments have been implemented in the simulator to compute the inertia-only navigation. The data used are "rectified" and gave the IMU increments at the same frequency of the images, which is 10Hz. The frequency of IMU increments is usually 100 Hz but as explained earlier, it is not a problem if the inertia-only navigation is not very accurate.

7.3. VISION ALGORITHM

7.3.1. Direct-Sparse Odometry (DSO) VSLAM

The DSO VSLAM algorithm has been used on the KITTI dataset. The figure is a screenshot of the visualization window of the VSLAM. In the bottom right corner of the figure are the feature points used by the algorithm. It is a sparse algorithm, as the feature points are well distributed in the whole image. The algorithm succeeds in estimating the relative depths of the feature points: the smaller the estimated depth, the warmer is the color of the point. It is not very visible here, but the algorithm can reconstruct the whole environment: the white structures along the trajectory, always to within a scale factor.

The images used are undistorted, which means that the distortions on the images have been corrected.

The processed outputs are plotted on the figures below:
The results of the DSO VSLAM algorithm are not evenly spaced in time, that's mean that the time step between two estimations is not constant. This is an issue, or at least a challenge, for the implementation later in the EKF. The solution proposed is to interpolate the value on a time scale of 10 Hz, so that for one image, there is one estimation of vision and one set of IMU increments.

### 7.3.2. Bundle Adjustment with inertial constraint

An algorithm of Bundle Adjustment has been used, with an inertial constraint. The description of this algorithm is attached as an appendix and the inertial constraint description is in in the literature review.

The plots below present the normalized translations estimated by the GNSS + IMU (reference), the stimulation of the inertia-only navigation and finally the results obtained by the Bundle Adjustment algorithm.
As it can be observed on the plots above, the translations from vision are better than the one calculated using only the inertia sensors. This means that the hybridisation is possible in translation and should improve the navigation.

It is also notable that this approach of comparing the normalized translations is not accurate when the vehicle is static as seen at the very end of these plots. At the end of trajectory, the car stops and so there are no translations anymore, what's left is just some normalized noise.

However, the performances in attitude are not as good. An explanation could be that, as explained previously, the reference navigation is not the true navigation, even if it is supposed to be close enough. Another fact is that GNSS does not measure the attitude of the vehicle, so the GNSS coupled navigation cannot really be trusted regarding attitude.

Besides, the vision attitude estimate is drifting after 35 seconds which corresponds to the time the car begins to stop.

Even if this result is considered as observations of vision, the use of the inertial constraint means that there has already been a hybridisation between inertia measures and vision observations.

### 7.4. RESULT OF HYBRIDISATION

The vision measurements (rotations and translations between frames) have been implemented in the simulator as measures in the Kalman Filter that updates the navigation estimates.

#### 7.4.1. Results with Vision from DSO VSLAM
7.4.2. Results with vision from Bundle Adjustment

7.5. ANALYSIS OF THE RESULTS
The inertial navigation frequency is 10 Hz. First, the Kalman frequency is set to 0.25 Hz, i.e. the Kalman Filter updates the navigation every 4 seconds as it is usually done. This results in the green trajectory on the plot below. It is obviously not good enough. The problem is noticeable above all during the turn, when there are big dynamics. There can be several reasons for this: for example it may come from a bad setting of the parameters, in particular the noises of the measures R and the noises of the state Q. Another reason is that, despite an extra care attached to this issue, it may come from a bad synchronisation between the measures and the increments in the simulator. It means that the translations and rotations implemented in the Kalman Filter do not correspond exactly to the predicted rotations and translations in terms of time. Maybe one of the two has a delay compared to the other.

To counter this issue, the time step of the Kalman cycle has been halved to 2 seconds (0.5 Hz). This results in the pink plot on the figure above. The plots are results from the hybridisation with vision observations coming from the Bundle Adjustment algorithm, which is why it was possible to plot the Vision-only trajectory as the scale factor has been estimated using the inertial constraint. This is a big improvement compared to the green plot. More importantly, it does improve the inertial-only navigation and is better than the vision-only navigation (with inertial constraint), except at the end of the trajectory. The inertia-only navigation drifts a lot at the end of the trajectory and it seems to impact a lot the vision-coupled trajectory, more than the navigation from vision with inertia constraint. The inertia being implemented differently, it can come from an imperfect implementation of the IMU increments in the simulator.

The addition of the vision measurements within the EKF may not seem to bring some huge improvements to the pure-inertia navigation. One first explanation is some possible not accurate enough vision estimations. Another reason is that the trajectory is quite short in time (40 seconds). The vision measurements may be a lot more useful for longer trajectories because it could solve the drifts of positions and attitude caused by the IMU sensor errors. For our simulations, even if the integration of the IMU increments gives a quite poor performance, the impact of the IMU biases is not very noticeable. Finally, one should keep in mind that the navigation of reference is not the real
trajectory, it is an estimation using GNSS observations. Especially, the attitude estimated by the GNSS coupled navigation can be quite different from the real attitude.
8. CONCLUSION AND FURTHER WORK

The work conducted during this master thesis is an important step of the development of navigation systems using camera as a sensor for localisation. It can be split in three main stages: stimulation of the inertia (integration of the IMU increments in the simulator), extracting the measures of Vision from the images and finally fusion of both the IMU increments and the vision measures via an EKF.

The interest of fusing visual odometry with the IMU observations have been demonstrated using only matlab simulations, for both the trajectory and the visual odometry observations. This thesis aimed at demonstrating its efficiency using data coming from field test. It presented several challenges, as the data were not as perfect as the simulated ones. One of the challenges was the calibration between the different sensors and the body frame of the vehicle: the attached frames were found to not be exactly aligned which worsens the performance. By using a gradient descent method, some values of the calibration have been estimated. Moreover, the visual odometry results were not as good as expected for the helicopter data, the main reason being some unknown parameters of the camera which the visual algorithms are very sensitive to. This led to change the dataset to one of the KITTI dataset to test the hybridisation. Finally two different visual algorithmes have been tested so to compare their impact on the localization performance.

The results obtained lead to several conclusions:

- Loosely-coupled visual odometry improves the performance of the Inertial Reference System, but it does not reach the performance of the GNSS. This means that for systems that require accurate localisation such as autonomous cars, visual odometry can only be used either as a punctual substitute of the GNSS or as an additional sensor for redundancy.

- Algorithms of vision need high accuracy of the parameters of calibration of the camera to get good estimations of rotations and translations.

- Fusion using EKF has many advantages compared to VSLAM with inertial constraint. In the results shown previously, the VSLAM with inertial constraint estimations were improved when implemented in the EKF. Besides, it gives a higher frequency of estimation availability, which is required for systems with big dynamics.

The simulations done for this master thesis are not "optimal" for several reasons, and the results can surely be improved. Indeed, when the vision measurements were simulated and not estimated using real images, it did improve the localisation estimation. The simulator has been adapted so different dataset can be tested. It would be very interesting to test it on dataset with long trajectories, perfectly known calibration of the cameras and of the IMU. Also, a next step would be to test the tight-coupled vision hybridisation on real dataset and compare the performance to the loosely-coupled hybridisation.

Using vision so a vehicle can self-locate is a vast field with many different possibilities for improvement, such as the use of maps of the environment or the use of object recognition like road sign. This master thesis is only a humble and tiny step.
9. REFERENCES


APPENDIX 1. THE KALMAN FILTER

This appendix describes very briefly the main principles and basic concept of the Kalman Filter. The Kalman filter is a recursive

A1.1. DEFINITION

A1.1.1. Model of the system

Let be a simple system whose dynamic is defined by this model of state propagation:

\[ \dot{X} = F \cdot X + W, \]

with:

- \( X \) being the state vector, i.e. the vector of all the variables that describes the state of the system.
- Matrix \( F \) being the state-transition model
- \( W \) the vector of white noise of the model

For the Kalman Filter, the model must be expressed in discrete-time. The system is set to evolve at a regular time step. The state propagation model for step time \( k \) becomes: \( X_{k+1} = F_k \cdot X_k + W_k \)

Besides, measurements of the system are made according the following model:

\[ Y_k = H_k \cdot X_k + V_k, \]

with:

- \( Y \) being the vector of measurements
- \( H \) the observation matrix that connects the measurements to the state vector
- \( V \) the vector of observation white noise

There can be several type of measurements, and for each type corresponds one specific line of the observation matrix and one element of the noise vector.

A1.1.2. Concept of the Kalman filter

The main hypothesis of Kalman Filtering is that both the states and the measurements are Gaussian-distributed variables. As such, the state vector is associated with a covariance matrix \( P \) that is estimated by the Kalman Filter too. The covariance matrix is useful as it is an indicator of how accurate the state vector estimates are.

The Kalman filter is a recursive estimator, i.e. it is used to estimate the state vector \( X \) using only the estimate \( X \) of the previous time step and the current measurements. At each time step, the state vector typically undergoes two phases: prediction and update. Prediction is the propagation of the previous state estimate using the state propagation model. The first estimate resulting from that phase is the "a priori" state estimate. The covariance matrix is also propagated.

The update phase uses the measurements to update the a priori state estimate. A gain (the Kalman gain) is calculated that evaluates how much impact should the measurements have on the estimation. For a linear system, this gain is optimal.

The equations are as follow for the time step \( k \):
Prediction

Predicted (a priori) state estimate: \( \hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} \)

Predicted (a priori) error covariance: \( P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \)

Update:

Optimal Kalman gain: \( K_k = P_{k|k-1} H_k^T (R_k + H_k P_{k|k-1} H_k^T)^{-1} \)

Updated (a posteriori) state estimate: \( \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - H_k \hat{x}_{k|k-1}) \)

Updated (a posteriori) estimate covariance: \( P_{k|k} = (I - K_k H_k) P_{k|k-1} \) (when the gain is optimal)

The index \( k|k \) means that the estimate (state or covariance matrix) is the one made at time \( k \) using observations up to and including at time \( k \). \( R_k \) is the covariance matrix of the vector of measurements.

A1.2. THE EXTENDED KALMAN FILTER

Most of the time, a system cannot be defined by linear equations. This is the case for navigation systems. The system is then described as follow:

\[
\begin{align*}
\dot{X} &= f(X(t)) + W(t) \\
Y(t) &= h(X(t)) + V(t)
\end{align*}
\]

The Extended Kalman Filter is an extension of the Kalman Filter for non-linear systems. The main difference is that the state transition function \( f \) and the observation function \( h \) needs to be linearized at the point of state estimate \( \hat{x} \) to get the matrices \( F \) and \( H \) as in the regular Kalman Filter.

The steps of the algorithms are as follow:

Prediction

Predicted (a priori) state estimate: \( \hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}) \)

Predicted (a priori) error covariance: \( P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \)

Update:

Optimal Kalman gain: \( K_k = P_{k|k-1} H_k^T (R_k + H_k P_{k|k-1} H_k^T)^{-1} \)

Updated (a posteriori) state estimate: \( \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - H_k \hat{x}_{k|k-1}) \)

Updated (a posteriori) estimate covariance: \( P_{k|k} = (I - K_k H_k) P_{k|k-1} \) (when the gain is optimal)

where the state transition and observation matrices are the Jacobians of the functions \( f \) and \( h \) at the last state estimate available:

\[
F_k = \frac{\partial f}{\partial x} |_{\hat{x}_{k-1|k-1}} \\
H_k = \frac{\partial h}{\partial x} |_{\hat{x}_{k|k-1}}
\]
APPENDIX 2: FRAMES DEFINITIONS

A.2.1. INERTIAL FRAME [I]

[i] is a direct orthonormal frame where the angular velocity is zero with respect to the absolute space (static with respect to the stars).

Its characteristics are:

- Origin at the center of the ellipsoid WGS84
- Axis Zi directed towards the celestial north pole
- Axis Xi et Yi in the equatorial plane
- At t=0, [i] and [t] coincide

A.2.2. EARTH CENTERED EARTH FIXED FRAME [T]

[t] is a direct orthonormal frame attached to the Earth. Its characteristics are:

- Origin at the center of the ellipsoid WGS84
- Axis Xt coincides with axis Zi
- Axis Yt directed by the intersection of the equatorial plane and the reference meridian (longitude=0)
- Axis Zt in the equatorial plane, completing the frame.

The transformation matrix from frame [i] to frame [t] is:
Where $\Omega$ is the Earth rotation vector.

### A.2.3. LOCAL GEOGRAPHIC FRAME [G]

[g] is a direct orthonormal frame attached to the vehicle. Ses caractéristiques sont :

- Origin : Point of navigation
- axis $X_g$ in the tangential plane of the ellipsoid, directed towards the (true) North,
- axis $Y_g$ in the tangential plane of the ellipsoid, directed towards the West,
- axis $Z_g$ directed towards the Zenith, perpendicular to the plane ($X_g, Y_g$).

The transformation matrix from frame $[t]$ to frame $[g]$ is:

$$ T_{gt} = \begin{pmatrix} \cos(L) & \sin(L) \cdot \sin(G) & -\sin(L) \cdot \cos(G) \\ 0 & \cos(G) & \sin(G) \\ \sin(L) & -\cos(L) \cdot \sin(G) & \cos(L) \cdot \cos(G) \end{pmatrix} $$
with:

- L being the latitude angle of the vehicle navigation origin
- G being the longitude angle of the vehicle navigation origin

When the vehicle is moving on Earth, if moving to the east or west and not ion the equator, the geographic frame is rotating so that Xb is always pointing to the North. It rotates upon the z-axis of a specific angle, called the wander angle $\alpha$, which defines a new coordinate frame: the geographic frame without the rotation of the wander angle.

### A.2.4. BODY FRAME [B]

[b] is a direct orthonormal frame attached to the body of the vehicle. Its characteristics are:

- Origin : Point of navigation
- axis Xb in the forward direction of the body (of the vehicle)
- axis Yb is pointing to the right of the body
- axis Zb is pointing to the bottom of the body

The axis (Xb,Yb,Zb) corresponds respectively to the roll (R), pitch (T) and yaw (C) axis of the vehicle.

The transformation matrix from the frame [g] to frame [b] is:

$$ T_{bg} = \begin{pmatrix}
\cos(T) \cdot \cos(C) & -\cos(T) \cdot \sin(C) & \sin(T) \\
-\sin(C) \cdot \cos(R) + \sin(T) \cdot \cos(C) \cdot \sin(R) & \sin(R) \cdot \sin(T) \cdot \sin(C) - \cos(R) \cdot \cos(C) \cdot \sin(R) & \sin(R) \cdot \cos(T) \\
\sin(C) \cdot \sin(R) + \cos(R) \cdot \sin(T) \cdot \cos(C) & \sin(R) \cdot \cos(C) - \cos(R) \cdot \sin(T) \cdot \sin(C) & -\cos(R) \cdot \cos(T)
\end{pmatrix} $$

### A.2.5. CAMERA FRAME [C]

[c] is the direct orthonormal frame attached to the detector of the camera. It is assumed that the camera and its detector do not have any misalignments.

- Origin : optical center
- axis Xc is directed to the right-side of the camera,
- axis Yc is directed towards the bottom of the camera
- axis Zc coincides with the optical axis (positive forward)
The transformation matrix from \([b]\) to \([c]\) if the camera is looking forward to the vehicles and the harmonization angles are neglected: 
\[
T_{cb} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}
\]

### A.2.6. IMU FRAME [M]

This frame is the direct orthonormal frame of the measured increments of the IMU. There are one gyrometer and one accelerometer on each axis. Usually, this frame coincides with the body frame \([b]\) but there can be some small rotations between the two frames, as it has been estimated in this thesis.

### A.2.7 OTHER CONVENTION

The data of the KITTI benchmark uses different conventions for the angles and the frames, which required extra care when comparing data using different conventions.

Indeed, in the KITTI data, the yaw angle is equal to zero when the vehicle front is pointing towards the east, whereas in the convention used in the simulator, it is zero when pointing to the north.

Moreover, the definition of yaw pitch and roll is also different: in the convention of the simulator to go from frame \([g]\) to \([b]\), the frame \([g]\) is first rotation upon the x-axis of \(\pi\) radians. And then the rotation of yaw, pitch, angle are executed.

This absence of the \(\pi\) rotation is the reason why the body frame is slightly different from the conventional body frame at Safran which is \(y\) to the right and \(z\) down, whereas in the KITTI convention, it is \(y\) to the left and \(z\) up.

The rotation matrices from \([g]\) to \([b]\) are as follow (only the elements of matrices that are interesting are written) depending on the conventions used:

- With the KITTI convention:

  \[
  T_{bg} = \text{Rot}_x(\text{roll}) \ast \text{Rot}_y(\text{pitch}) \ast \text{Rot}_z(\text{yaw}) = \begin{bmatrix}
  \cos(\text{pitch}) \ast \cos(\text{yaw}') & \cos(\text{pitch}) \ast \sin(\text{yaw}') & -\sin(\text{pitch}) \\
  X & X & \cos(\text{pitch}) \ast \sin(\text{roll}) \\
  X & X & \cos(\text{pitch}) \ast \cos(\text{roll})
  \end{bmatrix}
  \]

  with \(\text{yaw}'\) being \(\text{yaw} - \pi/2\).

- With Safran convention:

  \[
  T_{bg} = \text{Rot}_x(\text{roll}) \ast \text{Rot}_y(\text{pitch}) \ast \text{Rot}_z(\text{yaw}) \ast \text{Rot}_z(\pi) = \begin{bmatrix}
  \cos(\text{pitch}) \ast \cos(\text{yaw}) & -\cos(\text{pitch}) \ast \sin(\text{yaw}) & +\sin(\text{pitch}) \\
  X & X & -\cos(\text{pitch}) \ast \sin(\text{roll}) \\
  X & X & -\cos(\text{pitch}) \ast \cos(\text{roll})
  \end{bmatrix}
  \]
APPENDIX 3. BUNDLE ADJUSTMENT ALGORITHM

Bundle adjustment is an algorithm for visual odometry that use feature points. Briefly, the algorithm estimate the poses of the camera (positions and orientations) using a set of successive images with the same feature points.

If we illustrate the algorithm with the following figures: the blue line is the true trajectory of the camera (what needs to be estimated), going from A to D. The round points are the feature points previously extracted that appear on the images at positions A, B, C and D. For each images and feature points, there exists a bundle.

The algorithm does an initial estimation of the trajectory (here the red right) and estimate the 3D position of the feature points (the triangles of the figure) using the bundles. These 3D estimated positions are then projected on the 2D images, using the camera projection model.

Finally, the algorithm aim to minimize the 2D distance between the true feature points on the images and the 2D projection of the estimated positions. It is done by adjusting the positions and orientations of the camera in order to “adjust” the bundles. Several iterations are required to do so and it should retrieve the true positions and orientations of the camera at different times.
Page left blank intentionally.