Hybrid Visual-Inertial/Magnetic 3D Pose Estimation for Tracking Poorly-Textured / Textureless Symmetrical Objects

By

Sapto Wibowo

Thesis submitted to the University of Sheffield for the degree of

Doctor of Philosophy

Department of Automatic Control and System Engineering
The University of Sheffield
Mappin Street, Sheffield S1 3JD
United Kingdom

February 2019

Abstract

The focus of this research is mainly to develop a visual 3D pose estimation that can be used for many purposes including but not limited to autonomous visual inspection support system. The work overcomes the fundamental problem of region-based pose estimation in tracking poorly-textured/textureless symmetrical objects due to non-unique projection shape given numerous different poses. The work improved the existing state-of-the-art region-based pose estimation, known as Pixel-Wise Posterior 3D Pose estimation (PWP3D), by incorporating with inertial/magnetic orientation estimate. For this purpose, an inertial/magnetic orientation estimate expressed as a full optimisation problem is proposed beforehand. The proposed method, referred to NAG-AHRS, aims to deal better with the non-Gaussian noise and the non-linear model. The NAG-AHRS is then analysed by comparing its output to the motion capture system, as well as benchmarked to five state-of-the-art inertial/magnetic orientation estimates. The experiments show NAG-AHRS outperformed other benchmarking. Furthermore, NAG-AHRS facilitates the integration to visual-only pose estimation and to develop hybrid visual-inertial/magnetic pose estimation. In contrast with common visual-inertial integration method that has been dominated by Kalman filtering framework, the proposed method integrates visual and inertial/magnetic as a single optimisation problem. The selected optimisation method is Nesterov's Accelerated Gradient (NAG) descent, hence the proposed method is referred to as PWP3Di-NAG. The developed PWP3Di-NAG algorithm is then validated by comparing its output to the reference pose provided by Aruco marker and at the same time, it is also benchmarked to the original PWP3D algorithm. The validation demonstrated some significant performances improvements. Moreover, integrating visual-inertial as a single optimisation problem requires to transform inertial/magnetic measurements into the object reference frame. The required transformation induces an initialisation stage to accurately estimate the initial pose of the object. A novel framework for serving this purpose that combines region-based and edge-based pose estimation in a particle filtering framework is also proposed. The validation shows that the proposed framework be able to estimate the pose of an object with low pose estimation errors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Important engineering structures such as bridges, chimneys, storage tanks and aircraft need to be inspected periodically to guarantee safe operation (Stumm, Breitenmoser, Pomerleau, Pradalier, & Siegwart, 2012). With an inspection on a regular basis, flaws such as cracks, corrosion and missing parts can be detected in their early stages before they become major and lead to a catastrophic situation. A devastating accident is not only possible to occur due to the existence of major defects, but a seemingly minor damage could also initiate this situation.

The most popular method for periodic inspection is Non-Destructive Test (NDT) as this method does not cause damage to the object being inspected (Kumar & Mahto, 2013). This method also does not require disassembly prior to inspection, avoiding further complexity and risk of damage during the inspection procedure (Cawley, 2001). NDT can be done in many forms and visual inspection is the most widely chosen for periodic inspections due to its effectiveness (Ozaslan et al., 2017; Wenner, Spencer, & Drury, 2003).

Visual inspection is usually conducted by experienced human inspectors performing visual observation at every part of the object (Hellier, 2012). To gain access to all sections of the object, some supporting equipment such as scaffolding, ladders, mobile elevated unit or hanging platform are usually needed. The requirement of additional support systems has significant impact on the inspection time due to: 1. The extensive preparation required before and after the inspection; and 2. Performing inspection from the support system reduces the inspector's mobility. Time is a significant factor in visual inspection, for example in the aerospace sector, EasyJet reports the average loss is more than £20,000 for every hour of flight downtime, even before the inspection is carried out ("EasyJet's Aircraft Maintenance Lifts Off with Vicon," 2015).

Visual inspection is therefore still facing some challenges, despite it being considered as the most economical and the most chosen NDT method. Research that addressed the future of inspection by Cawley (2001) concluded that research is still required to increase inspection quality, to reduce the required preparation time and to develop methods to inspect without shutting down operation. Cawley also suggested an aim to develop inspection method that can be applied with minimum preparation (Cawley, 2001).

These visual inspection challenges can potentially be addressed by robots capable of collecting images from every part of the object and transmitting them to a ground station. As generally robots are used to perform dull tasks, in dirty and dangerous environments, D^3 (Takayama, Ju, & Nass, 2008), utilizing robots for supporting visual inspection offer an opportunity to improve the inspection efficiency and safety, and to reduce overall outage time (Stumm et al., 2012).

As aforementioned, robots for supporting visual inspection has high potential hence it has been intensively researched in both universities and industries. University based research on inspection robots include: pole inspection (Sa, Hrabar, & Corke, 2015), chimney (Nikolic et al., 2013), bridge (La, Gucunski, Dana, & Kee, 2017; Pham & La, 2016), tank (Schempf, Chemel, & Everett, 1995) and penstock in power plant (Ozaslan et al., 2017). In the industrial sector, EasyJet and Airbus are examples of companies actively researching service robotics ("EasyJet's Aircraft Maintenance Lifts Off with Vicon," 2015).

1.1.1 Autonomous Robot Inspection

The main objective for service robots that support visual inspection is to navigate around the object and collect visual information (images). Navigation around the object can be done manually by a pilot team or can be done autonomously without the need of a pilot team. Manual control method facilitates fast deployment of the robot for supporting visual inspection since many commercially available robots can be used for this task. However, this approach requires a highly skilled pilot team that is able to operate the robot near a structure without endangering the object itself. The disadvantages of this approach is that highly skilled pilot might not always be available and hiring pilot team also incurs an additional cost. A robot with an autonomous navigation capability has a better advantage since it does not require an experienced pilot team. With this type of robot, an inspection can be carried on anywhere and can be done more frequently.

This autonomous system is possible if the robot has the capability to know its own position relative to the object being inspected. Knowing the robot position can be done by a number of approaches, for instance by using a GPS system. However, generally the classical GPS system is not suitable for inspection since its accuracy varies with weather, depends on ionosphere condition that induces propagation delays. In some conditions, the accuracy can be more than 7.8m (Renfro, Terry, & Boeker, 2013). This accuracy is not good for operating a robot close to an object.

Another type of GPS system, known as RTK-GPS, can perform correction of the propagation delays so it has a much better precision, up to centimetres in outdoor environment with clear sky view (Chen, Zhao, & Farrell, 2016). However, the precision of RTK-GPS degrades when it operates close to large object due to multipath fading effects. Multipath fading effects occurs due to the GPS signal bounces off a building and incurs additional delay than cannot be modelled and anticipated easily (Bajaj, Ranaweera, & Agrawal, 2002). Utilising RTK-GPS for robot localisation is a good option where the robots operate sufficiently far from large object, but in case of autonomous inspection that normally operates close to large object, utilising RTK-GPS is not a best solution.

Another alternative for knowing robot’s location is using a motion capture system that can reach up to 2mm precision (Merriaux, Dupuis, Boutteau, Vasseur, & Savatier, 2017). This approach has been used by EasyJet for their visual inspection support system (“EasyJet’s Aircraft Maintenance Lifts Off with Vicon,” 2015). However, this approach is not portable, it can only be implemented in a few special depots because the motion capture system is expensive and requires an extensive time for setting up. Motion capture systems are also very limited in workspace size and not suitable for outdoor usage (Neunert, Blösch, & Buchli, 2015).

Recent research has demonstrated a good precision in robot localisation based on camera observations (Bloesch, Omari, Hutter, & Siegwart, 2015; Engel, Koltun, & Cremers, 2018; Engel, Schöps, & Cremers, 2014; Leutenegger, Lynen, Bosse, Siegwart, & Furgale, 2014; Mur-Artal & Tardos, 2017; Wang, Schworer, & Cremers, 2017). As the camera is an ego-motion sensor, generally this method does not need to alter the environment and offers an infrastructure free approach to facilitate autonomous navigation around the object. This approach is also highly portable and does not require a significant preparation time.

1.1.2 Visual 3D Pose Estimation

The widely popular methods for camera based robot localisation are: Visual Odometry (VO), visual SLAM (Simultaneous Localization and Mapping) and 3D pose tracker/estimator. For autonomous visual inspection support system, VO and SLAM are not a straightforward solution as the map obtained from these approaches is usually defined in the global world frame (usually with the initial robot pose as the zero reference point), not in object reference frame as desirable. To obtain map relative to the object, an additional transformation from world frame to object frame is needed. This computation requires an extra registration algorithm that induces an extra complexity and adding the source of uncertainty.

Another method to localize a robot that more suitable for autonomous inspection is visual 3D pose estimation. Given a known model, pose estimation aims to precisely estimate position and orientation of camera, relatively to object(s), based on the visual clues. Utilising 3D pose estimation for autonomous inspection offers a main advantage since the pose estimate is already defined in the object reference frame, therefore, it does not require any extra transformation.

Three dimensional pose estimation can be built based on salient points, edge information and region-based (Kelsey, Byrne, Cosgrove, Seereeram, & Mehra, 2006). Salient point-based pose estimation is popular and can be used for estimating the pose of highly-textured objects such as demonstrated by (Crivellaro et al., 2015; Wagner, Reitmayr, Mulloni, Drummond, & Schmalstieg, 2008).

However, salient point based pose estimation is not the best solution for visual inspection support system as many objects that need to be inspected can be poorly-textured or textureless such as chimneys, overhead tanks, aircraft and poles. In this kind of objects, salient points can still be detected but it is generally not stable, reducing the robustness of the estimated pose (Klein & Murray, 2006). A better approach in dealing with poorly-textured or textureless objects is edge-based pose estimation (Choi & Christensen, 2012). However, since this approach relies on edge information, it suffers from blurry images input. Blurry images may occur due to fast camera motion or camera defocus. Edge-based pose estimation also suffers from visually cluttered background as edges are primitive features that are hard to find their correct correspondence (Koller, Daniilidis, and Nagel 1993).

Another alternative for visual pose estimation works on region information of the input image such as colour information (Victor A. Prisacariu & Reid, 2012; Tjaden, Schwanecke, & Schömer, 2016). As demonstrated by one state-of-the-art algorithm within this category, Pixel-Wise Posterior 3D Pose Estimation (PWP3D) (Victor A. Prisacariu & Reid, 2012), region based method can estimate textured or textureless objects and also work for blurry images in a cluttered backgrounds (Victor A. Prisacariu & Reid, 2012; Tjaden et al., 2016). This approach also computationally efficient, since it works directly on pixel information and it does not requires any feature detection algorithm that mostly require a demanding computational load such as SIFT (Lowe, 2004) and SURF (Bay, Ess, Tuytelaars, & Van Gool, 2008). Even though there is feature detection algorithm that require low computational demand such as Harris corner, the descriptor of Harris corner is not as strong as other method (SIFT, SURF, BRISK, etc) and less stable (Cheng & Tang, 2009). Considering the advantages and drawbacks, region-based method is more suitable for autonomous inspection purpose.

1.2 Research Challenges

As aforementioned, algorithms belong to region-based 3D pose estimation category such as PWP3D is potential for autonomous visual inspection support system. However, this algorithm has a fundamental limitation that needs to be addressed before it is adopted for this purpose. PWP3D performs pose estimation by matching the shape of the projected model to the shape of colour-segmented-region by iteratively adjust the estimated pose. This approach experiences difficulty in estimating the pose of symmetrical objects, since it may generate identical projection shapes given a number of different poses. In this case, the projection is not unique and the pose cannot be fully retrieved. This problem, known as multimodal projection problem, may implies multiple or infinite solutions for the algorithm. This limitation becomes the main challenge in implementing PWP3D for autonomous visual inspection support system.

The non-unique projection shapes may occur from some motion scenarios such as:

- Stand still camera, moving object
- Moving camera, stand still object
- Moving camera and moving object

The multimodal projection problem caused by the first scenario that is stand still camera and moving/rotating object, is still an open challenge. Tracking symmetrical objects in this motion scenario is very hard, as an illustration, a fixed camera pointing toward a textureless cylinder-shaped object will always get a same projection shape for any rotation angle around the cylinder axis. In this case, the full pose cannot be retrieved. The third motion scenario, which is moving camera and moving object, implies more complexity. Both motion scenarios will not be addressed in this research.

This research only focusses in solving multimodal projection problem due to moving camera tracking a static object. This motion scenario matches with the requirement for visual inspection support system. Visual inspection usually inspects still objects by moving inspectors. The multimodal projection problem due to moving camera and stand still object is the main challenge that is addressed in this research.

Retrieving a full object pose from a poorly-textured/textureless symmetrical object requires an additional modality. In this proposed method, the additional modality is obtained from inertial/magnetic sensors measurements. While combining visual clues with inertial/magnetic measurements is common in VO (Gui, Gu, Wang, & Hu, 2015; Li & Mourikis, 2013; Sirtkaya, Seymen, & Alatan, 2013; von Stumberg, Usenko, & Cremers, 2018) and in visual SLAM (Hesch, Kottas, Bowman, & Roumeliotis, 2014; Mur-Artal & Tardos, 2016; Newcombe et al., 2011; Ozaslan et al., 2017; Teixeira, Alzugaray, & Chli, 2018), only a few research have combined visual-inertial/magnetic for pose estimation (Ligorio & Sabatini, 2013; V A Prisacariu, Kähler, Murray, & Reid, 2015). Moreover, as far as author's knowledge, there is no research in combining region-based pose estimation with inertial/magnetic information.

Combining visual and inertial/magnetic information is usually done using Kalman filtering framework. Kalman filtering method has been proved to be optimal for solving a system with linear models and when the noise has Gaussian distribution. However, it is also known when the models are non-linear and the noise is non Gaussian, Kalman filter is not optimal. The fact that the noise in low cost inertial/magnetic sensors is non Gaussian as demonstrated by (El-Sheimy, Hou, & Niu, 2008; Roberts, Corke, & Buskey, 2003) and updating/observing an orientation requires non-linear models (Sabatini, 2006), the common approach to combine visual-inertial using Kalman filtering framework is not optimal and still can be improved. This left another challenge that is how to combine visual-inertial that can handle non-linear model and non-Gaussian noise better.

1.3 Aim and Objectives

The aim of this research is to develop hybrid visual-inertial 3D pose estimation that can be used for autonomous visual inspection support system. The developed hybrid visual-inertial estimation should also be applicable for more general purposes such as: automatic manipulation, automatic grasping, autonomous docking and virtual reality. The developed method should be able to track poorly-textured/textureless symmetrical objects. The objectives then can be described as follows:

- Develop an inertial/magnetic orientation estimation algorithm as the base of hybrid visual-inertial/magnetic pose estimation that better handles non-linear model, non-Gaussian noise and achieves good performances. The inertial/magnetic orientation algorithm should also be highly adaptable and capable to integrate to any optimisation-based visual pose estimation easily.
- Develop a hybrid visual-inertial pose estimation that is capable to track poorly-textured/textureless symmetrical objects. The hybrid visual-inertial/magnetic pose estimation method should be capable to overcome the multimodal projection problem and robust to the presence of blurry image inputs.

- Develop a visual pose estimation framework that is capable to combine region-based tracker and edge-based tracker, to achieve a good accuracy of pose estimate. The framework should be able to accurately retrieve any arbitrary pose of the object given a very minimum manual input.

1.4 Contributions

The aim and objectives led to the following contributions:

- Estimating the attitude given the inertial/magnetic measurements requires a data fusion method. The currently available approaches for computing the orientation can be classified into three categories: 1. Single-frame deterministic method; 2. Stochastic method; and 3. Complementary Filter method. Single-frame method has no single mechanism in dealing with noise, hence the output is very noisy. Stochastic method is dominated by Kalman filtering framework, but due to non-Gaussian inertial/magnetic noise and non-linear kinematic equation for the rotation body, this method is not optimal. Complementary filter has demonstrated competitive performances but still experiences some errors that need to be improved.

  A novel method in inertial/magnetic orientation estimate, referred to NAG-AHRS, is presented in Chapter 3 aims to achieve better performance. The proposed method addresses inertial/magnetic orientation estimate as a full optimisation problem that performs better in the presence of high non-linearity.
- The existing state-of-the-art region-based object tracker, PWP3D, works well in tracking general objects. However, PWP3D experiences difficulty in tracking symmetrical objects as it suffers from multimodal projection. An extension of PWP3D tracker, referred to PWP3Di-NAG, is presented in Chapter 4. The proposed PWP3Di-NAG algorithm overcomes the multimodal projection problem by combining visual and inertial information to retrieve full pose. PWP3Di-NAG takes the advantage of the NAG-AHRS algorithm (proposed earlier) to be able to improve the original PWP3D algorithm. As results, the proposed PWP3Di-NAG does not suffer from multimodal projection problem and be able to track poorly-textured/textureless objects with better accuracy than the original algorithm.

- The proposed PWP3Di-NAG overcomes the multimodal projection problem by incorporating visual and inertial/magnetic information. The incorporation is done by addressing visual-inertial pose estimation as a single optimisation problem. Addressing visual-inertial as a single optimisation problem requires all measurements to be defined in a single common reference system. Due to this requirement, the inertial/magnetic measurements need to be transformed to the object frame and to be able to do this transformation, the initial relative pose between inertial/magnetic sensor to the object must be known. This prerequisite needs an initialisation framework that is capable to retrieve any arbitrary pose of the object given a minimum information of object's position. A framework that is capable to achieve a good accuracy with only a minimum information about position of object is proposed in Chapter 5. The proposed framework facilitates the initialisation stage as required by PWP3Di-NAG, and it achieves a good accuracy by combining region-based tracker and edge-based tracker in a particle filtering framework.

1.5 Thesis Overview

Since this research can be seen as a combination of visual 3D pose estimation and inertial/magnetic orientation estimate, a detailed literature review exploring the methods of both topics is presented in Chapter 2. In this chapter, the state-of-the-art in object tracking methods and inertial/magnetic orientation estimation are presented and critiqued. The discussion led to a finding of research gap that have not been addressed and this finding become the base of this research.

Chapter 3 presents the first contribution of this thesis which is a development of inertial/magnetic orientation estimation, referred to NAG-AHRS, that aims to achieve a better performance than the benchmarking methods. The background section describes the motivation behind the proposed approach is presented before the derivation of the method. Validation of the method by some experiments is then presented.

Chapter 4 covers a novel method in integrating visual localisation with inertial/magnetic orientation estimate. Taking the developed inertial/orientation estimation presented in Chapter 3, a region-based vision-only pose estimate is improved. The proposed method is referred to PWP3Di-NAG that aims to deal with the multimodal projection.

The hybrid visual and inertial/magnetic tracker developed in Chapter 4 requires an initialisation stage to transform inertial/magnetic reference frame to the object frame. Chapter 5 addresses this problem by proposing an initialisation framework based on a Particle Filter approach. Some suggestions are presented and then validated using some experiments.

Last chapter summarises the research, emphasising the contributions and also present some ideas for future work.

Chapter 2

Literature Review

2.1 Introduction

The aim of this research is to develop novel hybrid visual-inertial/magnetic 3D pose estimation that is capable to track symmetrical objects. The key challenges in the visual-estimation part are the algorithm must be able to handle multimodal projection problem and can also track poorly-textured / textureless objects. In the term of inertial/magnetic orientation estimate, the key challenge is how to deal with non-Gaussian noise of the low cost inertial/magnetic sensor as well as the non-linearity of the process/observation model.

Since the proposed method is a combination of vision-only pose tracking and inertial/magnetic orientation estimation, the literature review addresses the existing approaches in both areas. The common approaches to vision-based localisation are presented in Section 2.2 and then Section 2.3 addresses in more detail the chosen vision-based localisation method which is model-based object trackers. As the proposed method combines model-based tracking with inertial/magnetic orientation estimation, Section 2.4 covers the inertial/magnetic orientation estimate. Each section covers the recent existing methods, the state-of-the-art approaches, including their strengths and shortcomings. To highlight the gap of the existing methods, Section 2.5 presents concluding remarks.

2.2 Visual-Base Localisation: Visual Odometry, SLAM and Pose Estimation

Autonomous navigation around the inspected object requires the ability to continuously estimate the robot's location and orientation (pose) within the object's reference frame (Sa, Hrabar, & Corke, 2015). Estimating a robot's pose is known as the localisation problem and along with the ability to create a map of the surrounding environment, it has been referred to as the core of autonomous systems (Durrant-Whyte & Bailey, 2006).

As presented in Section 1.1, localization can be done by processing observation data from various sensors such as: Global Position System (GPS), inertial/magnetic sensors and camera. However, each of the sensors has its own advantages and disadvantages. Recall from Section 1.1, that GPS is not suitable for indoor localisation since it's low-power high frequency radio signal (around 1.5 GHz) cannot penetrate solid objects, GPS positioning accuracy also degrades with the presence of multipath fading due to nearby large structures (Renfro, Terry, & Boeker, 2013). Localisation can be done using camera observation that has been demonstrated achieve better precision up to cm accuracy (Crivellaro et al., 2017; Engel, Koltun, & Cremers, 2018; Prisacariu & Reid, 2012; Tjaden, Schwanecke, & Schömer, 2016).

The popular methods for performing localisation based on visual observation can be classified into three approaches:

- Visual Odometry (VO)
- Simultaneous Localisation and Mapping (SLAM)
- Pose estimation/object tracking

Visual odometry (VO) is a technique to incrementally estimate the pose of a robot from sequentially observed images (Scaramuzza & Fraundorfer, 2011). Incremental pose updates are achieved by integrating the previous pose estimate with the recent motion estimate induced from the change of the observed images. VO is a potential approach as it can achieve small relative position error in static environments with sufficient illumination, enough textures and scene overlap in the image observations between frames (Engel et al., 2018; Forster, Pizzoli, &

Scaramuzza, 2014; Scaramuzza & Fraundorfer, 2011; R. Wang, Schworer, & Cremers, 2017).

VO is only concerned with local consistency since it only optimises the pose over the last n frames or over last n poses. This is different from Simultaneous Localisation and Mapping (SLAM) technique that aims for global map consistency. To achieve a global consistency, the whole robot's path and environment map are needed to be refined comprehensively. Global refining stage is usually executed when the robot visits the pre-visited location, known as loop closing (Davison, Reid, Molton, & Stasse, 2007). This loop closing phase makes the SLAM method differs significantly from VO approach. As a result, SLAM requires more computational resources since it needs to keep track of the global map. The SLAM framework is also more complex than VO since it has to implement a method for detecting and performing loop closing (Scaramuzza & Fraundorfer, 2011).

The third method for vision-based robot localisation is 3D pose estimation. This approach requires to define the object of interest and its initial approximated pose in the first frame, and then afterwards, by having this information, the pose of the object can be estimated in the rest of the frames (Lebeda, Matas, & Bowden, 2012). The object that will be tracked and its initial approximated pose can be selected manually with human intervention (Lebeda et al., 2012) or also can be obtained as a result from object recognition phase that needs to be executed before the object tracking is carried out (Rad & Lepetit, 2017).

These three methods: Visual Odometry, SLAM and pose estimation/object tracking have their own characteristics. In terms of a priori knowledge, VO and SLAM belong to the same class as these approaches do not require any prior knowledge of the environment. This characteristic makes VO and SLAM popular for autonomous cars, search-and-rescue robots, or any other purposes that aim for an autonomous navigation in unknown/unprepared environment. In contrast with VO/SLAM, pose estimation requires a priori knowledge, hence object tracking is popular in robotic manipulators (Munoz, Konishi, Murino, & Del Bue, 2016), autonomous docking (Kelsey, Byrne, Cosgrove, Seereeram, & Mehra, 2006), augmented/virtual reality (Yan & Hu, 2017) and engineering inspection (Ozaslan et al., 2017).

For the purpose of engineering inspection, object-tracking is a better approach instead of VO or SLAM for some reasons. Autonomous navigation around the inspected object can be done by specifying waypoints around the object, so the robot can move autonomously following these predefined points. These predefined waypoints are defined in the object reference frame as the main focus.

The waypoints that are defined in object reference frame adds extra complexity for VO and SLAM approaches since these methods usually yield robot pose estimates in the global coordinate frame, with the initial robot pose as its origin (Klein & Murray, 2007; Leonard & Durrant-Whyte, 1991). In this case, a transformation is needed to transform from the object coordinate frame to the global coordinate frame, to be able to make use of the pre-defined waypoints. This transformation requires to implement an extra method for registering the global map to the object which is non-trivial. This registration has been demonstrated to not be an easy problem and it induces an extra complexity and uncertainty (Segal, Haehnel, & Thrun, 2009; Yang, Li, Campbell, & Jia, 2016).

In contrast with VO/SLAM, object tracking is aimed to estimate the camera's pose relative to the observed object(s) (Han & Zhao, 2015; Valinetti, Fusiello, & Murino, 2001). This means the 3D pose estimate output is already defined in the object coordinate frame. Therefore, the robot can directly take the benefit of the predefined waypoints and carry out an autonomous navigation from the very beginning. In this case, the complexity is minimal and no additional uncertainty comes from any additional algorithms. Another extra benefit of using object-tracking for autonomous inspection is it accounts for the prior knowledge. Having this prior information has a great potential to improve the performance of the autonomous system (Seo & Wuest, 2016).

Based on the review of common approaches of vision-only localisation, this research focussed on model-based object-tracking, also known as pose estimation, to build the proposed approach. Therefore, a detail literature review presenting the existing model-based tracking methods, their advantages along with their shortcomings is required and this literature review is presented in Section 2.3.

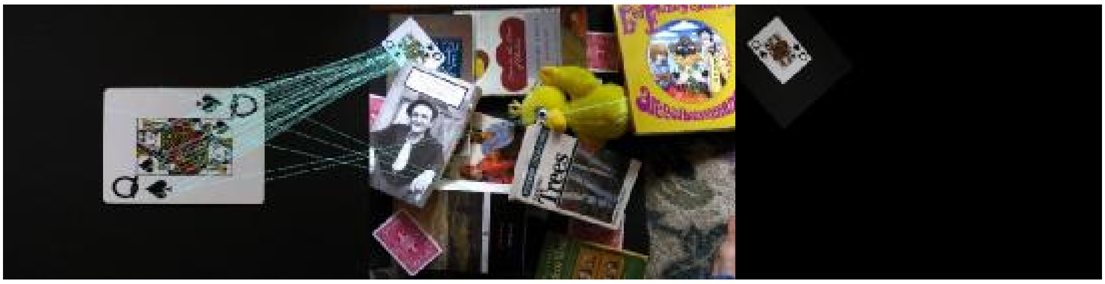
2.3 Model-Based Object Tracking

Model-based object tracking has many potential applications, including engineering inspection (Ozaslan et al., 2017), robotic manipulation (Munoz et al., 2016), autonomous docking (Kelsey et al., 2006) and augmented/virtual reality (Yan & Hu, 2017). This approach requires prior knowledge of the object being tracked and this a priori knowledge can be in the form of set of image descriptors or 3D CAD model. Despite recent research in object-tracking have demonstrated good performance as shown by (Choi & Christensen, 2012a; Gennery, 1992; Kim & Sim, 2010; Munoz et al., 2016; Prisacariu & Reid, 2012), model-based object tracking is still considered as a crucial issue in computer vision due to a number of various objects, illuminations and backgrounds (Seo & Wuest, 2016). Model-based visual tracking track the objects based on natural visual clues such as: salient points, edge or region-based information (Kim & Sim, 2010).

2.3.1 Salient Point Based Pose Estimation

The pose of textured-object can be estimated from the detected salient points around its surface (Seo, Park, Park, & Park, 2013). In this approach, a model is defined as a set of image feature and by knowing the correct correspondence, the transformation can be calculated so the pose can be retrieved. Figure 2.1. illustrates this approach. The salient points are usually detected and described according to well-known image feature detectors/descriptors such as SIFT (Lowe, 2004), SURF (Bay, Ess, Tuytelaars, & Van Gool, 2008), ORB (Rublee, Rabaud, Konolige, & Bradski, 2011) and FAST (Rosten & Drummond, 2006).

This approach is usually robust as salient points provide a plenty information that easy to find their correct correspondence between the consecutive frames. This method usually achieves a good performance especially for tracking highly-textured objects as demonstrated by (Crivellaro et al., 2015; Wagner, Reitmayr, Mulloni, Drummond, & Schmalstieg, 2008). However, salient point-based pose estimation typically requires heavy computing demands since most of feature point extraction, i.e. SIFT, SURF, ORB are still computationally expensive (Rosten & Drummond, 2006). While there is lightweight feature extraction such as Harris corner, but it provides less information and make it harder to find the correct correspondence between consecutive frames (Le, Woo, & Jo, 2011). As consequences, the utilisation of Harris corner usually requires a stronger robustification stage to remove false correspondence (Gauglitz, Höllerer, & Turk, 2011) which induces an additional computational load. Therefore, in general, feature based pose estimation require intensive computational demand, cannot achieve real-time tracking in low cost embedded system and only well adapted for highly textured objects.



(a) (b) (c)

Figure 2.1. Texture-based object tracking. The model is described as a set of salient points (a). The detected salient points are extracted from the query image (b) and then find the correspondence to the model as shown by blue lines. After an robustification and optimisation, the pose of object estimated is shown in image (c). Images are reproduced from: https://robwhess.github.io/opensift/

2.3.2 Edge Based Pose Estimation

A model-based tracker that demands lower computation power is built on edge information. Extracting edges requires lower computational demand than extracting salient points (Kim & Sim, 2010). In general, this approach basically tries to align the projected edges of an object to the extracted edges from the query image.

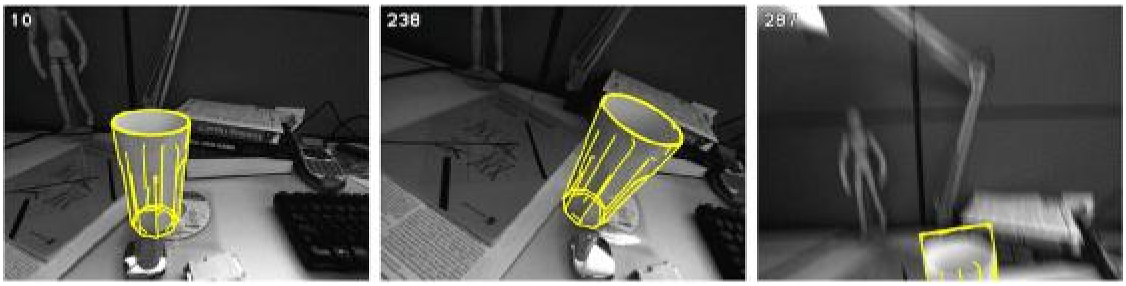


Figure 2.2. An example of edge-base object tracker given its 3D CAD model. The tracker aligns the projected model to the detected edges of the query image. In this case, a Particle Filter was employed. Images are reproduced from (Choi & Christensen, 2012b)

The disparity between projected model's edge to the observed edge is then minimised using some optimisation algorithm (Gennery, 1992; Harris & Stennett, 1990; Worrall, Marslin, Sullivan, & Baker, 1991). This approach works well for textureless objects and is being used in industry (Comport, Marchand, Pressigout, & Chaumette, 2006; Drummond & Cipolla, 2002). Figure 2.2 illustrates this approach.

One of the state-of-the-art edge tracking proposed by Harris is the RAPID algorithm (Harris & Stennett, 1990) that has become a building block for many model-based tracking algorithms (Seo, Park, Park, Hinterstoisser, & Ilic, 2014; Vacchetti, Lepetit, & Fua, 2004; G. Wang, Wang, Zhong, Qin, & Chen, 2015). Given a model, an initial approximation of model pose and calibrated camera parameters, the first step of the RAPID algorithm is to project the model into a 2D image plane. Afterwards, the fitness of wireframe projections are measured against the detected edges on the observed image where, in a good pose estimate, the projected lines should align to the image edges. When the projection lines are not aligned to the image edge due to incorrectness of pose estimate, the distance between projected line and edge are measured and the sum of squared distance is minimized by updating the pose estimate.

RAPID is a very efficient algorithm as it achieves real time operation by only processing a sparse set of control points along the edge, not the whole edge as proposed by an earlier method (A. J. Bray, 1990). The assumption that the points on the model's edge correspond to the nearest of image edge in normal direction works well when the pose different between frames is small. However, when the observed image consists of highly cluttered background, or the pose difference between frames is large due to fast motion, this simple correspondence leads to a large number of mismatches. In this condition, the solution might not converge, or converges to a local optima.

Worrall *et. al.* (Worrall et al., 1991) improved the tracker performance for a special case. The proposed method was designed for tracking objects that move on the ground such as cars. In this case, instead of searching for the best fit pose in 6DOF, the proposed method only optimises on 3DOF pose parameters: 2 translation and 1 rotation. The optimisation was done using 3 separate, one-dimensional gradient descent. The trajectory is then smoothed using a Kalman filter assuming a constant velocity for its motion model.

The RAPID method and its improvements as proposed by Worrall *et. al.* (Worrall et al., 1991) still suffer from high cluttered environments. Koller *et al.* (Koller, Daniilidis, & Nagel, 1993) addressed this problem by treating edges as parameterized segmented lines and the correspondences are achieved by matching the line parameters. This approach achieved a better tracking in cluttered background. However, the requirement to extract line segments and compute their parameters, along with the requirements to do optimization for finding best line segment correspondence make this approach very slow and far from real time performance. The proposed method is also not easy to adapt to any new model as it requires to parametrise the edges of the model.

To achieve real time operation (Drummond & Cipolla, 2002) proposed a model-based approach that can be easily adopted to track any object given its CAD model. The proposed method takes advantage of Graphical Processor Unit (GPU) for performing rendering. After the visible edges have been located, similar to the RAPID method, some control points are selected and then minimized.

Some other methods improved the point correspondence stage by adding more information rather than just using line as a primitive features. Pupilli and Calway (Pupilli & Calway, 2006) introduced junctions, which are a group of line segments (branches) that share common corners and Lebeda *et al* (Lebeda et al., 2012) introduced virtual corners. Virtual corners are obtained by extending line segments along their tangent directions until they intersect with other extended line segments. This approaches provide more information so it is easier to find their correspondence instead of edges as primitive features. These approaches demonstrated effective and achieved real-time operation.

Some other methods proposed to combine edge with texture information (Vacchetti et al., 2004) and with observed background geometry (Seo et al., 2013). These approaches demonstrated the potential to improve the tracking. However, the performance is highly depend on texture information and the background.

2.3.3 Region Based Pose Estimation

Edges based tracking has potential to yield good tracking. However, it still struggles in tracking an object in visually cluttered environment and also suffers from the presence of blurry images. In a blurry image, edges cannot be located precisely or events cannot be detected. Image feature quality also degrades significantly as a consequence of blurry textures. In terms of handling blurry images due to fast motion or other reasons, region-based trackers usually outperform edge-based and texture-based trackers. Region based trackers such as (Prisacariu & Reid, 2012; Tjaden et al., 2016; Tjaden, Schwanecke, & Schömer, 2017) utilise color information about the model and the observed image. In general, region based methods try to maximise the segmentation between the object and its background based on selected information i.e. color histogram, by adjusting the pose.

One state-of-the-art region-based object tracker was proposed by Prisacariu (known as Pixel-Wise Posterior 3D Pose estimation / PWP3D) that works by maximizing the pixel-wise color posterior probability of the observed image (Prisacariu & Reid, 2012). This tracker works by building a colour histogram model of the object and its background, then maximizing the segmentation posterior probability by adjusting the object pose. The contour that splits between the foreground and background is specified implicitly by using a level set and the proposed approach maximizes the posterior of colour histogram by deriving the level set evolution with respect to the pose estimate. This approach can then be considered as simultaneous segmentation and 3D pose estimation. PWP3D has been demonstrated to track objects in blurry images (Figure 2.3) which is possible since it employs pixel-wise color information not edges.

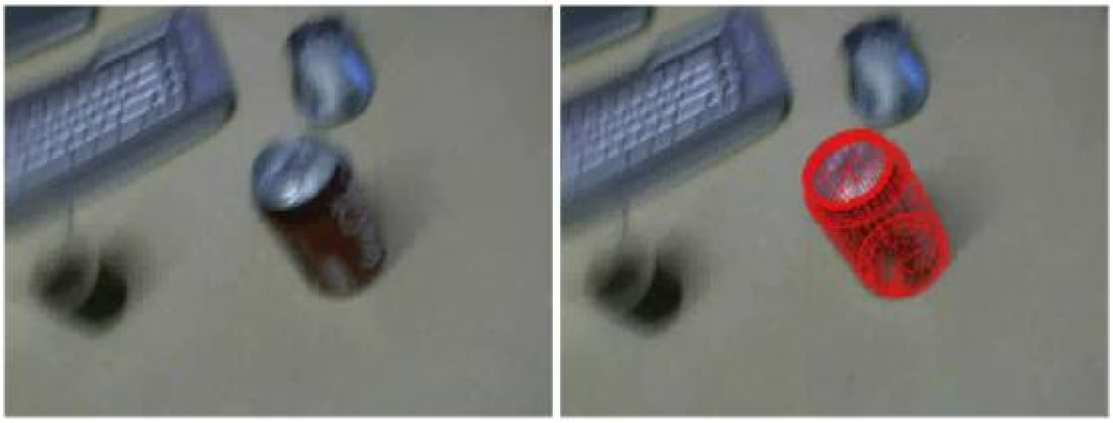


Figure 2.3. Region-based object tracker works on region information such as the color histogram. The pictures were taken from PWP3D algorithm that maximises the posterior color probability that best segment foreground and background area. Region-based approaches do not require to extract image feature such as edges and salient point, hence they can cope with blurry images. Images are reproduced from (Prisacariu & Reid, 2012)

However, as PWP3D basically maximizes the segmentation posterior probability between the region inside the projected model silhouette and the region beyond the silhouette, it suffers from the multimodal projection problem. Different poses can have the same silhouette, so for this case the pose estimate cannot achieve a unique solution. Another problem arises as the algorithm relies on the colour histogram. It, therefore suffers from colour perception problem which might arise due to the presence of shadows and due to different illumination.

The proposed method in this thesis aims to extend this state-of-the-art method by elaborating orientation estimate from inertial/magnetic sensor. While the common method for visual inertial fusion is conducted by implementing the Kalman Filter (Fang, Zheng, & Deng, 2016; Jiang & Yin, 2017; Ligorio & Sabatini, 2013; Sirtkaya, Seymen, & Alatan, 2013; Tian, Li, Li, & Cheng, 2017), the proposed approach includes some additional constraints directly into the existing PWP3D's non-linear system of equation. The additional constraint was taken from inertial/magnetic orientation estimate which is expressed in a full optimisation framework. By this approach, the visual and inertial tracking are integrated into a single optimization problem that can be solved simultaneously in an effective manner. To achieve this objective, a literature study in existing inertial/magnetic sensor orientation estimate is needed and is presented next.

2.4 Inertial/Magnetic Orientation Estimate

As this research combines visual model-based pose tracking with inertial/magnetic orientation estimate as aforementioned in Chapter 1, this section is focussed on the inertial/magnetic orientation estimate. This section addresses the existing state-of-the-art method for estimating the orientation from inertial/magnetic sensor.

Estimating orientation is part of robot localisation as the full pose consists of position and orientation information. Robot orientation can be estimated from external reference system but it can also be recovered from ego-motion sensor that works in any environments. The popular ego-motion sensors that have been used for many applications such as UAV (Roberts, Corke, & Buskey, 2003), body tracking (Angelo Maria Sabatini & Maria, 2011) are accelerometer, gyroscope (inertial measurement unit) and magnetometer. Each of the sensors has its own characteristic. Accelerometer that measures static gravity field provides a non-drifted measurement. However, the accelerometer reading is noisy and the gravity measurement is easily influenced by linear motions. Magnetometer that measures static magnetic field also provides a non-drifted measurement, but its reading is prone to local magnetic disturbances from ferrous objects, electrical appliances or other magnetic field sources. The update rate of magnetometer is also slow. Gyroscope that measures angular rate performs well in high dynamic motions, but it suffers from bias and noises.

A fusion method then has to be implemented to combine the measurement from these sensors to recover full orientation estimate. Extensive research has been conducted in this area and recently the existing method can be classified into 3 categories (Bleser & Hendeby, 2010; Filippeschi et al., 2017; Valenti, Dryanovski, & Xiao, 2016):

1. Deterministic Single-Frame Algorithm
2. Stochastic Estimation Algorithm
3. Complementary Filter

2.4.1 Deterministic Single-Frame Approach

Deterministic single-frame algorithms provides algebraic and geometrically approaches in combining 2 or more measurements for estimating the orientation. These approaches are very efficient and only require low computation load. The state-of-the-art methods in this category are TRIAD and QUEST proposed by Black and Shuster respectively (D. Black, 1964; Lefferts, Markley, & Shuster, 1982) that have been cited in more than 1240 publications. These two methods have became parts of many modern algorithms such as (Valenti et al., 2016) and still widely chosen as standard benchmark for new algorithms. TRIAD and QUEST are deterministic and perform the orientation estimate from a single-frame and it does not retain any past information in any form (Yun, Bachmann, & McGhee, 2008).

The TRIAD method proposed by (D. Black, 1964) requires 2 pairs of vectors: measurement vectors and reference vectors. TRIAD method then uses these two pair of vectors to build an orientation matrix. The TRIAD algorithm is a general framework that can be used for determining orientation based on any two observation vectors such as: sun direction, star direction or other observations. In the inertial/magnetic sensor case, the observation vector is usually taken from the accelerometer and magnetometer. In this case, the fixed reference vectors are: ${}^{E}v_1$ is gravity vector that has only component in z axis so ${}^{E}v_1 = [0 \quad 0 \quad 1]$, ${}^{E}v_2$ is the earth magnetic field that usually symbolised as ${}^{E}v_2 = [h_x \quad h_y \quad h_z]$. This earth magnetic vector depends on the location. Given these pair of static reference vectors ${}^{E}v_1, {}^{E}v_2$ (gravity and magnetic field in this case) and its observation ${}^{S}w_1, {}^{S}w_2$ (magnetometer and accelerometer measurement – normalised into unit vector) TRIAD method then creates a two set of triads of orthonormal unit vectors (r_1, r_2, r_3) and (s_1, s_2, s_3). Furthermore, each set of triads is combined to create measurement matrix $M_{obs} = [s_1 \quad s_2 \quad s_3]$ and reference matrix $M_{ref} = [r_1 \quad r_2 \quad r_3]$. Each of these matrices has a dimension of 3×3. The orientation matrix A is then calculated by a multiplication of measurement matrix and transpose of reference matrix $A = M_{obs}M_{ref}^{T}$. Reader interested in detail method and the algebraic prove can refer to (D. Black, 1964).

TRIAD method offers a deterministic and simple method that can be easily implemented in low computational embedded system. The main shortcoming of this method is the measurement noise/error level affects directly the estimation accuracy. The absence of any bias correction method is another drawbacks. The cross-product operation for computing set of triads $(r_i, s_i)_{i=1\ldots3}$ also eliminates any contribution of the magnetic measurement relative to the vertical axis (Yun et al., 2008). In this case, the accelerometer measurement is the only one that regulates pitch and roll angle estimate. The cross-product operation also makes the algorithm sensitive to the processing order, as any contribution of ${}^{E}v_2$ and ${}^{S}w_2$ relative to the vertical axis are discarded (Angelo Maria Sabatini & Maria, 2011). Considering this, sensor observation with higher accuracy should be defined as ${}^{S}w_1$.

While TRIAD method is only capable to integrate 2 measurements, another state-of-the-art method proposed by Shuster (Shuster & Oh, 1981) can account for more than 2 observations. The orientation is parameterised using quaternion and the proposed method is well known as QUEST (QUaternion ESTimation). The input of this algorithm is similar to the TRIAD method: fixed reference vectors ${}^{E}v_i$ and sensor observations ${}^{S}w_i$. However, QUEST method accommodates up to n observations instead of 2, so for this case $i = 1 \ldots n$. QUEST also implements a weighting constant a_i to accommodate different accuracy of the measurements.

Given reference vectors ${}^{E}v_i$, observation vector ${}^{S}w_i$ and its correspondence weighting coefficient a_i, the QUEST method then builds a matrix K with dimension of 4×4. Shuster then proved that optimal quaternion q corresponds to the eigenvector with largest eigenvalues of matrix K. Therefore, estimating orientation in this method is a problem in finding eigenvalues and eigenvectors of 4×4 matrix. However, QUEST method does not need to perform eigenvalues/eigenvector decomposition. By deriving the equation further for this specific case (orientation estimation) and for avoiding to compute complete solution eigenvalues /eigenvectors, QUEST method only requires to approximate the square roots of the derived equations. The square root approximation is then searched using Newton-Rapson method which mostly only requires 1 or 2 iterations (Shuster & Oh, 1981). This method demonstrated low computation demand and performs better than TRIAD method.

The chosen orientation parameter in QUEST algorithm is quaternion that is well known does not suffer from gimbal lock problem. However, as in QUEST algorithm the formulation was derived from Gibbs vector, QUEST suffers from singularity for rotation around π (Angelo Maria Sabatini & Maria, 2011). To avoid this problem, a sequential rotation method is implemented in QUEST with the consequence of increasing the computation time (Angelo Maria Sabatini & Maria, 2011). The assumption that the reference vectors are fixed in QUEST method also influences the estimation accuracy. Fixed reference vector assumption works well for gravity especially when object does not experience large linear motion. However, this assumption does not work well for earth magnetic field. Practically, earth magnetic field is easily disturbed by distortion whether from soft-iron or hard-iron distortions (Madgwick, Harrison, & Vaidyanathan, 2011). With the highly-coupled nature of the algorithms, the presence of these disturbances not only affects the yaw orientation angle estimate, but it also influences the pitch and roll estimation.

With the intention to suppress the influence of magnetic field noise affecting the roll and pitch estimate, Yun *et. al*. proposed a method that is well known as Factored Quaternion Algorithm (FQA) (Yun et al., 2008). FQA algorithm decouples accelerometer and magnetometer measurements so mainly the pitch and roll are estimated from the accelerometer, whereas yaw is computed from the magnetometer. The formulation given by FQA is effective as demonstrated in the experimental result. Similar to QUEST algorithm, FQA also suffers from singularity when the pitch angle approaches $\pm\frac{\pi}{2}$ since FQA was derived based on half-angle formulas. Similar with QUEST, FQA also implemented sequential rotation approach for dealing with this singularity. However, overall, the FQA method demonstrated a significant computation performance improvement than QUEST algorithm when at the same time it produces an orientation estimate with identical accuracy performance. The FQA formulations that avoid any trigonometry function and implemented half-angle formulas demonstrated more efficient than QUEST by about 25% (Yun et al., 2008).

The single-frame deterministic approaches as represented by the three state-of-the-art methods: TRIAD, QUEST and FQA share some common characteristics as this class of algorithm does not have a good noise anticipation in their formulation. This means, an error in one sensor degrades the orientation estimate significantly, i.e. error from large linear acceleration due to object motion that affecting accelerometer measurements. This approach is good for slow motion objects in a magnetically clean environment (Angelo Maria Sabatini & Maria, 2011). These algorithms also do not facilitate any bias corrections, since the output is only computed based on the most recent frame, and ignores any historical information.

Maintaining previous information improves the orientation estimate significantly as demonstrated by stochastic approach. In general, stochastic approach such as (Yun & Bachmann, 2006)(Yun et al., 2008) outperformed the point-to-point approaches especially when the non-linearity is low.

2.4.2 Stochastic Estimation Algorithms

Stochastic approach enables integrating previous information into the orientation estimate that cannot be found in the deterministic frame-to-frame approach. By accommodating previous information, stochastic approach achieves superior accuracy in the orientation estimate when the non-linearity is not high as shown in (Choukroun, Bar-Itzhack, & Oshman, 2006; Farrell, 1970; Marins, Yun, Bachmann, McGhee, & Zyda, 2001; Psiaki, Martel, & Pal, 1990; Yun & Bachmann, 2006). The most common framework is that suitable for this purpose is Kalman filter (KF) (Kalman, 1960) as KF is special case of recursive Bayesian state estimation under linear and Gaussian circumstances (Chen, 2003).

Kalman Filter is a general framework that can be implemented for many purposes. The filter design is related to the selection of the state, process model and observation model. These important parts of Kalman filtering can be specified freely according to the necessity. In the case of inertial/magnetic orientation estimate, this freedom has resulted in many approaches for inertial/magnetic orientation. Therefore, while Kalman Filter framework itself has been considered as a standard in fusing inertial/magnetic measurements (A M Sabatini, 2006), there is no clear, fixed, single standard on how to implement it. The selection of orientation parameterisation such as Euler angle (Farrell, 1970) or quaternion (Lefferts et al., 1982; Marins et al., 2001; Psiaki et al., 1990), has significant influences on the estimation performance. The decision not to include bias in the state (Marins et al., 2001; Psiaki et al., 1990; L. Wang, Zhang, & Sun, 2015; Yun & Bachmann, 2006) or to include bias (Lefferts et al., 1982; A M Sabatini, 2006) requires different process model design. Inertial/magnetic orientation estimate can be implemented within a pure Kaman Filter framework (Farrell, 1970; Lefferts et al., 1982; Psiaki et al., 1990; A M Sabatini, 2006; Trawny & Roumeliotis, 2005), but it is also possible to introduce additional, pre/post processing stage beyond Kalman filtering frameworks (Liu, Inoue, & Shibata, 2011; Marins et al., 2001; Valenti et al., 2016; L. Wang et al., 2015; Yun & Bachmann, 2006; Zhang, Meng, & Wu, 2012). These different approaches influence the efficiency, accuracy of the estimate and the demand of the computation resources.

Lefferts *et. al.* (Lefferts et al., 1982) implemented pure standard Kalman Filter for fusing three-axis gyroscope and line-of-sight attitude sensor, without any additional pre/post processing. In contrast with the research done by Farrell (Farrell, 1970) that implemented Euler angle, Lefferts *et. al.* chose quaternion as the orientation parameter. In general, the significant advantage of using quaternion instead of Euler angle is to avoid singularity. However, the highly coupled of the four elements in the quaternion render difficulty in standard development of Kalman Filtering framework, since Kalman Filtering framework does not provide a method to preserve any constraints (Markley, 2003).

Aiming to deal with the non-linearity of the observation matrix, Marins *et. al.* proposed an approach that introduced a pre-processing stage (Marins et al., 2001). In the proposed method, the accelerometer and magnetometer are integrated outside the Extended Kalman framework to compute a quaternion that later serves as observation orientation. Therefore, since the accelerometer and magnetometer integration already yield the orientation in quaternion parameter, the measurement equation becomes linear. However, since the process model is not linear, it still requires to implement Extended Kalman Filter. Even though only the observation model can be modelled as a linear system, this approach significantly reduced the computation time and achieved a real-time operation.

Another Kalman-filter based orientation estimate was developed by Sabatini (A M Sabatini, 2006). The filter concerned in dealing with noises in accelerometer and magnetometer by performing in-line procedure for estimating accelerometer and magnetometer bias. Accelerometer and magnetometer bias became part of the state being estimated. In the experiment, Sabatini investigated the effect of in-line calibration procedure as well as the effect of adaptive weighting measurement by activating/de-activating these features on the same dataset. It demonstrated that the in-line bias calibration and adaptive weighting managed to supress the temporal noises.

A similar two layer approach was also proposed by Yun and Bachmann (Yun & Bachmann, 2006), with a difference in the pre-processing algorithm. The proposed method implemented state-of-the-art QUEST algorithm. The quaternion output of QUEST algorithm that combined accelerometer and magnetometer then also forwarded to the Extended Kalman Filter framework.

The pre-processing stage that leads to linear measurement equations as proposed by Marins *et. al.* (Marins et al., 2001) as well as Yun and Bachmann (Yun & Bachmann, 2006) is effective, however the implementation still requires Extended Kalman Filter as the process model is still non-linear. Zhang *et. al.* (Zhang et al., 2012) proposed an approach that enables to integrate inertial/magnetic sensor in the linear Kalman Filter framework. This is achieved by using gyroscope measurement to dynamically update some designed coefficients inside the process model, while the process model itself has been derived as a linear model. The linear process model transforms gyroscope measurement into the quaternion angular velocity. Having this quaternion angular velocity, the orientation state is updated by integrating this quaternion angular velocity with the previous orientation state.

Stochastic method for fusing inertial/magnetic sensor that mostly builds upon Kalman Filter framework yields a good orientation estimate and the performance of Kalman filter-based algorithm is considered as standard method for inertial/magnetic fusion (A M Sabatini, 2006). However, this approach assumes linearity and Gaussian statistic that are not always well-suited especially for low-cost sensors (Jensen, Coopmans, & Chen, 2013; Robert Mahony, Hamel, & Pflimlin, 2005). The complexity of the method, that involves many matrix operations, executing expensive trigonometric functions or performing iterative optimisation, within a limited time constraint due to fast update rate (up to 8kHz for gyroscope and up to 1kHz for accelerometer for MPU6050) also burdens the implementation in a low cost embedded system.

Another approach that has been developed to account for the computation constraints is known as complementary filter. This approach does not require an assumption related with the linearity and the statistical model. Complementary filter method investigates the characteristic of the sensors, and removes unwanted signal before combining the filtered outputs that complement each other. Since the outputs of the filter should complement each other, hence the idiom complementary filter is known.

2.4.3 Complementary Filter

In its early stage, the complementary filter was designed in frequency domain and known as Frequency Domain Complementary Filter / FDCF (Jensen et al., 2013). Accelerometer has a slow update rate, suffers from high frequency noise but not drifting. In contrast, gyroscope is reliable in providing high frequency data, but suffers from drift. These characteristics complement each other, so the accelerometer can be filtered using low-pass filter to suppress the high frequency noise and gyroscope output can be filtered using high-pass filter to remove low frequency or DC component bias. The general FDCF block diagram can be seen in Figure 2.4.

Let x be the state that defines the true orientation, $H_1(s)$,$H_2(s)$ are sensor's transfer function; x_{m1}, x_{m2} observation output from the two sensors that are subject to noise or model uncertainty, the estimated state $\hat{x}$ is obtained from the combination of the output of both sensors after being filtered by $G_1(s)$ and $G_2(s)$. The true state can be recovered, $\hat{x} = x$, if $H_1(s)G_1(s) + H_2(s)G_2(s) = 1, \ \forall s$.

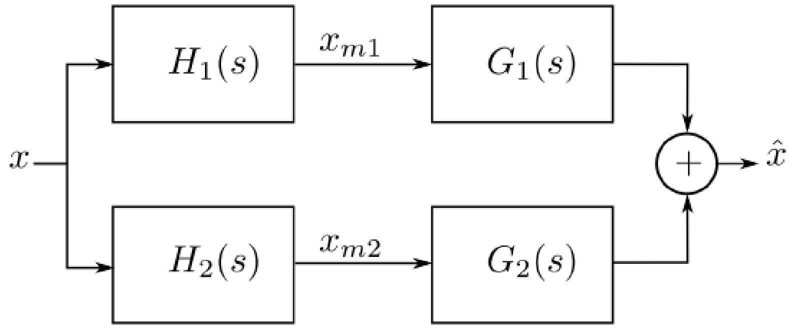


Figure 2.4. Block diagram of Frequency Domain Complementary Filter (FDCF). The method requires Low-Pass Filter $G_1(s)$ for suppressing high frequency noise of accelerometer measurements modelled by transfer function $H_1(s)$ and a High-Pass Filter $G_2(s)$ for removing low-frequency drift from gyroscope measurement modelled by transfer function $H_2(s)$. The output of both filters is then integrated so the filtered information complement each other. Image is reproduced from (Jensen et al., 2013)

However, since the sensor's true transfer function is not known, or varies from one to another, the common implementation usually assumes that the transfer function maps the input to output perfectly so $H_1(s) = H_2(s) = 1$.

Giving this assumption, therefore the filter $G_1(s), G_2(s)$ can be designed freely and just limited by a single constraint $G_1(s) + G_2(s) = 1, \forall s$. That means any pair of low-pass filter and high pass filter that has complementary frequency response as can be seen in Figure 2.5 suits this purpose.

This method can also be adapted to discrete systems by using digital filtering technique. Depending on the filter design, the computation demand can be low and can be implemented in low cost embedded system.

Measurement vector of accelerometer, magnetometer or gyroscope consist of 3 elements, one element for each 3D axis. The frequency domain complementary filter treats each element independently by performing filtering for each separate stream of the measurement signal. This approach ignores the fact that these measurements are highly coupled. The frequency filtering method also introduces some significant latency that might not be accepted in some applications. The design of cut off frequency also has to consider the dynamics of the system.

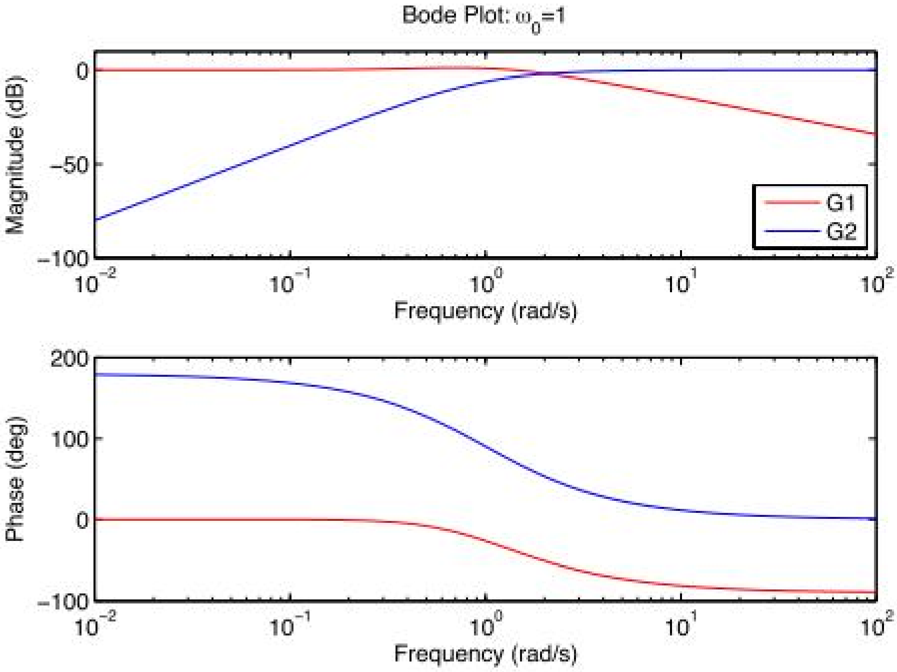


Figure 2.5. The frequency response of both filters that comply with the complementary filter requirement as it follows the constraints $G_1(s) + G_2(s) = 1, \forall s$. Image is reproduced from (Jensen et al., 2013)

A further development of complementary filter interpreted the problem in state space form to enable multiple input-multiple output system and it can take the benefit of classical control theory. Interpreting in state space form also ease the realization in digital system (Jensen et al., 2013). Roberts *et. al.* (Roberts et al., 2003) implemented complementary filter for estimating the orientation of an autonomous helicopter. The block diagram of the demonstrated method can be seen in Figure 2.6.

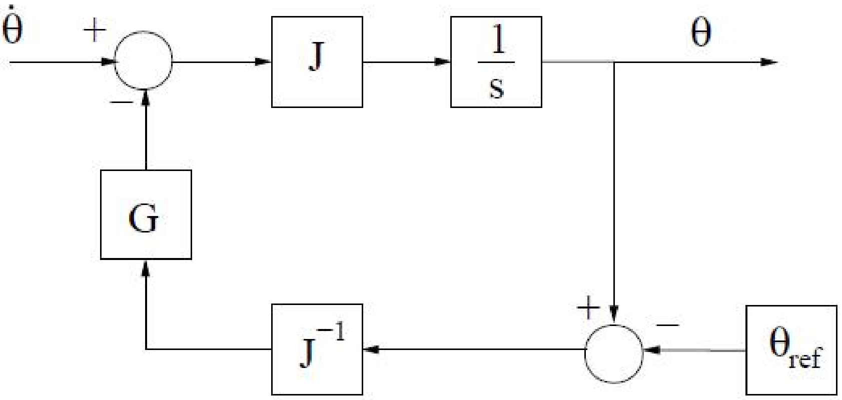


Figure 2.6. Block diagram of basic State Space Complementary Filter (SSCF) that accepts two measurements: angular rate from gyroscope $\dot{\theta}$ and angle from accelerometer θ_{ref}. Given these two measurements the orientation θ is then estimated. Image is reproduced from (Roberts et al., 2003).

The method is to integrate gyroscope measurement, $\dot{\theta}$, that has been transformed from sensor body frame to global earth body frame by J to obtain orientation estimate θ. Before the integration, the gyroscope measurement $\dot{\theta}$ is corrected by the difference between estimated angle θ and measured angle given by accelerometer measurement θ_{ref}. However, since the error is in global earth frame, it is required to transform into sensor body frame by J^{-1}. To account for noise, this transformed error is scaled by a free-tuned constant coefficient G. Roberts *et. al.* only demonstrated that this method can be used to retrieve roll and pitch only.

A more complex implementation of State Space Complementary filter in quaternion descriptor was proposed by Bachmann *et. al.* (Bachmann et al., 1999). The block diagram of this approach can be seen in Figure 2.7.

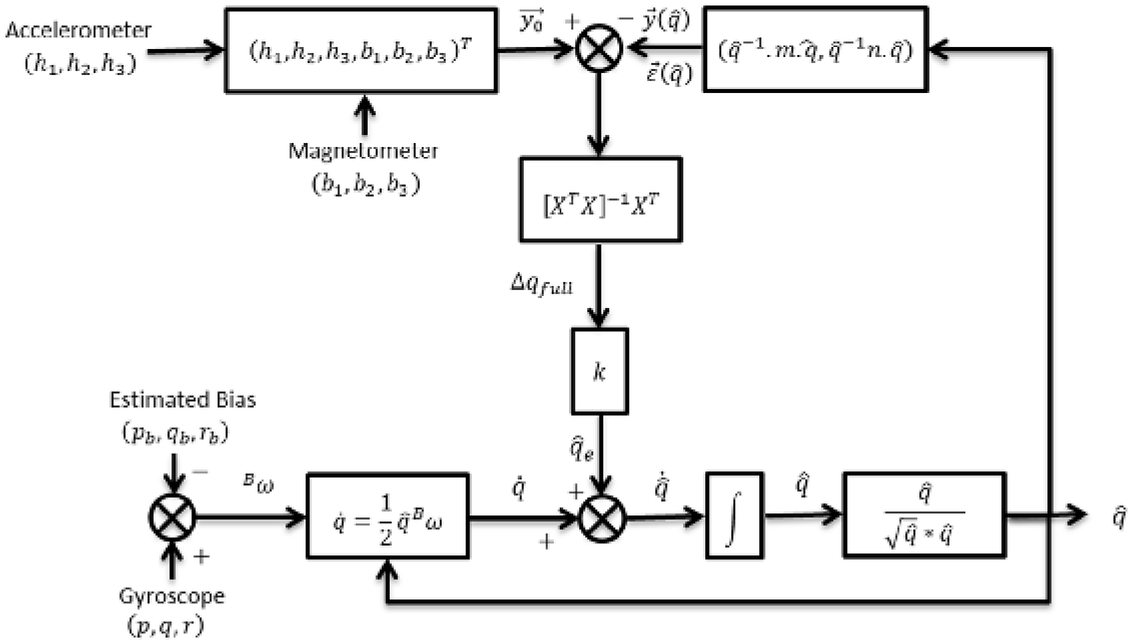


Figure 2.7. Block diagram of quaternion-based complementary filter. This approach combines three sensors: accelerometer, magnetometer and gyroscope. Image is reproduced from (Bachmann et al., 1999).

Given an angular rate defined in sensor body frame $^B\omega$ the quaternion rate $\dot{q}$ can be calculated from estimated orientation $\hat{q}$ by using this formula

$$\dot{q} = \frac{1}{2}\hat{q}\,^B\omega$$

The quaternion rate $\dot{q}$ then needs to be integrated to attain the estimated orientation $\hat{q}$. To deal with bias, this angular rate is corrected by $\Delta q_{full}.k$ before integration. The k coefficient is introduced to account for noise (Bachmann et al., 1999). The Δq_{full} is the difference between the estimated orientation $\hat{q}$ and observed orientation given from accelerometer and magnetometer measurement. To calculate this orientation difference Δq_{full}, Bachmann *et. al.* implemented Gauss-Newton approach that requires Jacobian function X of the function $\bar{y}(\hat{q})$ that transforms fixed reference vectors (gravity and earth magnetic field) into sensor body frame. The Gauss-Newton also requires residual $\varepsilon(\hat{q})$ that is retrieved by subtraction of accelerometer/magnetometer reading and transformed gravity field/magnetic earth field. The proposed approach demonstrated a good result but it requires to compute inverse 6×6 matrix that is associated with an additional computation load.

Hamel and Mahony proposed a method known as explicit complementary filter (Robert Mahony et al., 2005). The original method was developed in orientation matrix, however Hamel and Mahony suggested the implementation in quaternion that demand lower computation load (R Mahony, Hamel, & Pflimlin, 2008). The block diagram can be seen in Figure 2.8.

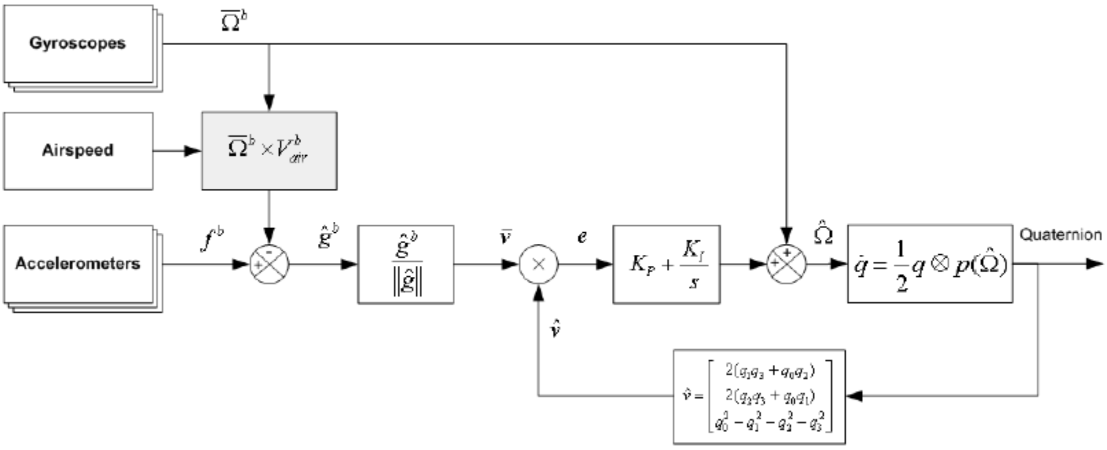

Gyroscopes
Airspeed
Accelerometers
$\overline{\Omega}^b$
$\overline{\Omega}^b \times V^b_{air}$
f^b
$\hat{g}^b$
$\frac{\hat{g}^b}{\|\hat{g}\|}$
$\bar{v}$
e
$K_P + \frac{K_I}{s}$
$\hat{\Omega}$
$\dot{q} = \frac{1}{2} q \otimes p(\hat{\Omega})$
Quaternion
$\hat{v}$
$\hat{v} = \begin{bmatrix} 2(q_1q_3 + q_0q_2) \\ 2(q_2q_3 + q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 - q_3^2 \end{bmatrix}$

Figure 2.8. Complementary filter proposed by (Robert Mahony et al., 2005) that integrates gyroscope and accelerometer measurement. The method considers the recent error and the integrated error to correct the gyroscope bias. The recent error and the integrated error are scaled by some coefficients K_p and K_I. Image is reproduced from (Robert Mahony et al., 2005).

The proposed approach performs correction to the gyroscope measurement ${}^B\omega$ before it is integrated to obtain the estimated angle $\hat{q}$. The correction is calculated from error e, that specifies difference between normalised accelerometer measurement ${}^Bv$ to the estimated gravity direction ${}^B\hat{v}$. In contrast with Bachman et. al. (Bachmann et al., 1999), the error e calculation does not requires a complex matrix inversion. The error e is computed by cross product between normalised accelerometer measurement ${}^Bv$ and the estimated gravity direction ${}^B\hat{v}$. Furthermore, Mahony treat this recent error e as well as the integrated error e_{int} and introduced two scaling coefficient K_p and K_I to account both errors.

The explicit complementary filter is fast and provides two user-adjustable coefficients to eliminate drift. However, Mahony et. al. only fused accelerometer and gyroscope and ignore magnetometer. Madgwick et. al. (Madgwick et al., 2011) developed inertial/magnetic that also accommodates magnetometer measurement into its framework. The algorithm integrates accelerometer and magnetometer using Gradient Descent as can be found in Marin's method (Marins et al., 2001). The block diagram of this method is shown in Figure 2.9.

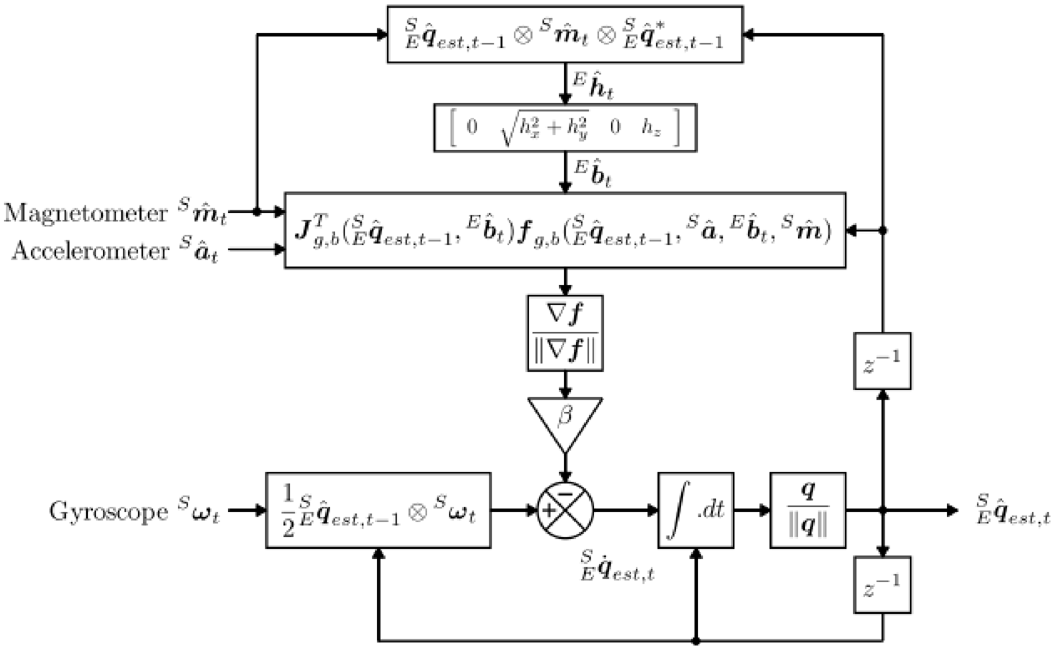

$^S_E\hat{q}_{est,t-1} \otimes {}^S\hat{m}_t \otimes {}^S_E\hat{q}^*_{est,t-1}$
$^E\hat{h}_t$
$\begin{bmatrix} 0 & \sqrt{h_x^2+h_y^2} & 0 & h_z \end{bmatrix}$
$^E\hat{b}_t$
Magnetometer $^S\hat{m}_t$
Accelerometer $^S\hat{a}_t$
$J^T_{g,b}({}^S_E\hat{q}_{est,t-1}, {}^E\hat{b}_t) f_{g,b}({}^S_E\hat{q}_{est,t-1}, {}^S\hat{a}, {}^E\hat{b}_t, {}^S\hat{m})$
$\frac{\nabla f}{\|\nabla f\|}$
β
z^{-1}
Gyroscope $^S\omega_t$
$\frac{1}{2}{}^S_E\hat{q}_{est,t-1} \otimes {}^S\omega_t$
−
+
$\int .dt$
$\frac{q}{\|q\|}$
$^S_E\hat{q}_{est,t}$
$^S_E\dot{q}_{est,t}$
z^{-1}

Figure 2.9. Block diagram of complementary filter that integrates three sensor. The approach adapted Gradient Descent method for combining magnetometer and accelerometer measurements. Image is reproduced from (Madgwick et al., 2011).

However, the computed quaternion output does not integrate with gyroscope using Kalman filter as proposed in Marins's, instead using complementary filter method as presented by Bachmann (Bachmann et al., 1999). This method has a better efficiency than Bachmann's approach since it precludes the expensive 6×6 matrix inversion as required in Bachmann's approach while keeping the advantage of quaternion complementary filter method. Furthermore, Madgwick et. al. also introduced a different bias correction method that demonstrated a better performance. The performance of algorithm was benchmarked to Kalman-based algorithm and achieved smaller Root Mean Square error with lower computation requirement.

2.5 Concluding Remarks

Retrieving robot location using a camera as the main sensor can be done by Visual Odometry, SLAM and pose estimation algorithm. Among of these approaches, pose estimation is more suitable for autonomous inspection since the estimated pose is already in the object reference frame. This reduces the complexity and minimises the uncertainty source and it potentially achieves a good accuracy.

Visual three dimensional pose estimate works on visual clues, such as: edges, salient point or statistical appearance model. Among of these categories, the statistical appearance based pose estimation has demonstrated a good performance in tracking poorly-textured objects, robust to the presence of blurry image inputs and less affected by visual clutters.

Among the statistical appearance based pose estimation, Pixel-Wise Posterior 3D Pose Estimation (PWP3D) proposed by (Prisacariu & Reid, 2012) is one of the state-of-the-art that has demonstrated good performance. However, since PWP3D works on the projection shape of object, it suffers from multimodal projection problem. This multimodal projection problem of PWP3D has not been addressed and becomes the gap that needs to be improved.

The proposed method aims to overcome this problem by integrating visual estimate with inertial/magnetic orientation estimate as a single optimisation problem. For this purpose, an inertial/magnetic orientation estimate that is expressed as a pure optimisation problem is needed. Addressing inertial/magnetic as a pure optimisation problem has never been done before. Therefore, an inertial/magnetic orientation estimate as a pure optimisation problem is proposed before developing the hybrid visual-inertial/magnetic pose estimate. This approach offers a main benefit as it does not require a linear/linearized model neither Gaussian noise assumptions. The development of the proposed method is presented in the next chapter.

Chapter 3

Orientation Estimate in a Full Optimisation Framework

3.1 Background

In Section 2.4 the importance of an accurate orientation from inertial/magnetic sensor measurements is presented. An accurate inertial/magnetic orientation estimate has many purposes in avionics, robotics and object tracking. There are many algorithms that have been developed for estimating orientation from inertial/magnetic sensors and generally, these algorithms gain a good performance in orientation tracking as shown by (Braud & Ouarti, 2016; Cavallo et al., 2014; Filippeschi et al., 2017; Nowicki, Wietrzykowski, & Skrzypczynski, 2015). However, some level of error is still observed that needs to be addressed to gain a better accuracy.

According to (Roberts, Corke, & Buskey, 2003), the main problem in estimating the orientation of objects using inertial/magnetic sensor comes from the non-linear model and the presence of non-Gaussian noise. This is confirmed by (Chang, Xue, Qin, Yuan, & Yuan, 2008; El-Sheimy, Hou, & Niu, 2008; Petkov & Slavov, 2010; Senyurek, Baspinar, & S Varol, 2014) that demonstrated the low-cost gyroscope noise is non-Gaussian. Along with the kinematic of rotation body that require non-linear models, the orientation estimate becomes challenging (Roberts et al., 2003).

This chapter proposes an algorithm that aims to improve the accuracy by better dealing with the non-linearity problem. The proposed method addresses the inertial/magnetic orientation estimate as a pure optimisation approach as optimisation has been used for solving non-linear problems and yields good results (Botev, Lever, & Barber, 2017; Qu & Li, 2017; Su et al., 2017).

This chapter is structured as follows: The proposed approach is presented in Section 3.5, but before that, a brief literature review of some methods that have been developed for inertial/magnetic orientation estimation is presented in Section 3.2 and problem definition is presented in Section 3.3. As the proposed method can be seen as an extension of Madgwick algorithm (Madgwick, Harrison, & Vaidyanathan, 2011) and the proposed method implements Nesterov Accelerated Gradient descent (Nesterov, 1983) Section 3.4 presents these methods along with an underlying theory about quaternion. Some experiments for validating the proposed framework along with results and discussion is then presented in Section 3.6 and finally, a concluding remarks is are covered in Section 3.7.

3.2 Literature Review

Recently, the existing methods for estimating an orientation given inertial/magnetic measurements can be classified into three categories, which are: stochastic method, single frame deterministic method and complementary filter (Filippeschi et al., 2017). The stochastic method generally implements Kalman filtering frameworks (Lefferts, Markley, & Shuster, 1982; Li & Mourikis, 2013; Liu, Inoue, & Shibata, 2011; Angelo Maria Sabatini & Maria, 2011; Yun & Bachmann, 2006). However, since inertial/magnetic measurements are subject to a non-Gaussian noise as demonstrated by (El-Sheimy et al., 2008; Petkov & Slavov, 2010; Senyurek et al., 2014; Shiau, Huang, & Chang, 2012), and the kinematic of rotation body is non-linear (A M Sabatini, 2006), this non-linear model and non-Gaussian noise breaches the optimality property requirement for Kalman filtering. This causes the Kalman filtering approach to be non-optimal and not guaranteed to converge (Roberts et al., 2003).

The single-frame deterministic mode only process the recent measurements, hence the presence of noise degrades the estimation accuracy of the deterministic single-frame method. The performance degrades since this deterministic method does not provide any explicit mechanism to cope with noise. Among two main algorithms in this category (TRIAD and QUEST), QUEST is slightly better than TRIAD in handling inaccuracies since it implements a simple static weighting constant to address different sensor accuracies (Shuster & Oh, 1981). However, the presence of significant noise still affects the overall performance directly (Yun & Bachmann, 2006).

The complementary filter (CF) category addresses the orientation estimate in frequency domain by performing high pass filtering to remove accelerometer and magnetometer noise, and performing low pass filtering to remove gyroscope's bias. The output is then combined to obtain final estimate (Jensen, Coopmans, & Chen, 2013; Roberts et al., 2003). More recent CF method addressed the orientation estimate in state space form as shown by (Euston et al., 2008; Madgwick et al., 2011). CF approach does not require a linear assumption or Gaussian noise model. Hence, in general it can cope better with the non-linearity problem (Jensen et al., 2013). Two main algorithms with a competitive result in this category were proposed by Mahony (Euston et al., 2008) and Madgwick (Madgwick et al., 2011).

Recall the non-Gaussian property of the noises as well as the non-linear process/measurement model, this research is proposing to address attitude estimate as a pure optimisation problem. This motivation came since optimisation has been proven can produce a good output for solving multivariate, complex and non-linear functions with hundreds/thousand variables (Qu & Li, 2017; Su et al., 2017). As far as the author's knowledge, addressing inertial/magnetic attitude estimate as a pure optimisation problem has not been addressed before. The known closest approach to the proposed method is Madgwick method (Madgwick et al., 2011) but it only fuses the accelerometer and magnetometer data within the optimisation framework, while the gyroscope is integrated outside the optimisation framework. Therefore, Madgwick method does not fully benefit from the optimisation framework.

The existing optimisation algorithms can be classified as local optimisation and global optimisation, where the local optimisation algorithms are mostly gradient-based and the global optimisation is dominated by evolutionary algorithms such as: Genetic Algorithm, Simulated Annealing, Differential Evolution, Particle Swarm Optimisation, Ant Colony (Venter, 2010). Global optimization algorithms are designed to provide a better chance of finding the global optimum than the local optimisation algorithms, however it still cannot guarantee to convergence on a global optimum (Liberti, Di Milano, Zza, & Da Vinci, 2006). Another main disadvantage of global optimisation is high computational cost (Venter, 2010). With the inertial/magnetic measurement rate that can reach up to 1000 Hz (MPU-6050 datasheet), the optimisation algorithm has to converge within a few milliseconds. In this case, since the gradient of the objective function can be solved analytically and efficiently, local optimisation is more suitable for this purpose. The guarantees to converge to the minimum of the basin become another significant benefit to avoid attitude tracking loss, especially with the high update rate that keeps the attitude change close to the previous state. Among the algorithms within local optimisation are: Newton method, Gradient Descent, Momentum and Nesterov's Accelerated Gradient descent.

This chapter also proposes to implement Nesterov Accelerated Gradient (NAG) descent that has better convergence characteristics than the classical Gradient Descent as demonstrated in (Sutskever, Martens, Dahl, & Hinton, 2013). The proposed method that implements NAG can potentially provide a better performance than the Madgwick method since Madgwick method implements the classical Gradient Descent (Madgwick et al., 2011).

The proposed approach also opens the possibility to integrate inertial/magnetic orientation estimate to any purely optimisation based algorithm easily. This is possible since the proposed approach is developed as a pure optimisation problem, integration can be done by just adding extra constraints to their system of non-linear equations. For instance, any recently available visual object tracking that tries to track an object by optimising a function that derived only from vision information can take the benefit of the proposed method and build a hybrid visual-inertial tracking easily.

Therefore, the benefit of the proposed approach can be summarized as follows:

- Better performance of inertial/magnetic orientation estimate
- Easily integrate inertial/magnetic into any other optimisation-based orientation estimate as an additional constraint to their existing system of non-linear equations

3.3 Problem Definition

Given observation vectors from accelerometer ${}^{S}\hat{a}$, magnetometer ${}^{S}\widehat{m}$, gyroscope ${}^{S}\widehat{\omega}$ that all defined in sensor's local frame S, and also given fixed reference vectors: gravity field ${}^{E}\hat{g}$ and earth magnetic field ${}^{E}\hat{h}$ defined in earth frame E, the objective is to estimate the quaternion orientation ${}^{S}_{E}\hat{q}$ that defines the orientation of the earth reference frame E with respect to the sensor frame S.

More specifically, this research deals with the problem of how to find the orientation quaternion ${}^{S}_{E}\hat{q}$ using a pure optimisation framework. As the selected optimisation method is based on Nesterov Accelerated Gradient descent which is an extension of classical Gradient Descent, the problem being addressed is how to build the residual function $f\left({}^{S}_{E}\hat{q},\ {}^{S}\hat{a},\ {}^{S}\widehat{m},\ {}^{S}\widehat{\omega},\ {}^{E}\hat{g},\ {}^{E}\hat{h}\right)$ given observation vectors $\left({}^{S}\hat{a},\ {}^{S}\widehat{m},\ {}^{S}\widehat{\omega}\right)$ and reference vectors $\left({}^{E}\hat{g},\ {}^{E}\hat{h}\right)$, and how to minimise this residual function in terms of the orientation quaternion ${}^{S}_{E}\hat{q}$ to find the best estimate using Nesterov Accelerated Gradient optimisation:

$$\min_{{}^{S}_{E}\hat{q} \in \mathbb{R}^4} f\left({}^{S}_{E}\hat{q},\ {}^{S}\hat{a},\ {}^{S}\widehat{m},\ {}^{S}\widehat{\omega},\ {}^{E}\hat{g},\ {}^{E}\hat{h}\right)$$

3.4 Underlying Theories

Regarding the objective of the proposed method which is to develop an inertial/magnetic orientation estimate in full optimisation framework, therefore this proposed method has some level of similarities with the existing state-of-the-art algorithm by Madgwick (Madgwick et al., 2011) since Madgwick method already implemented some level of optimisation in its algorithm. Madgwick method only combines accelerometer and magnetometer within optimisation framework and still requires other computation for fusing gyroscope. In contrast, the proposed method combine all of the sensor measurement within an optimisation framework.

The proposed algorithm also improves the optimisation by implemented Nesterov Accelerated Gradient (NAG) descent instead of classical Gradient Descent that has been implemented in the Madgwick method. To clarify the difference between the proposed method to the state-of-the-art Madgwick method, flow diagrams of both methods are presented in Figure 3.1 and 3.2.

Since the proposed method based is built on state-of-the-art AHRS proposed by Madgwick (Madgwick et al., 2011) and it implements the optimisation proposed by Nesterov (Nesterov, 1983) these algorithms are presented. The orientation representation being used is quaternion, therefore basic theorem of quaternion and its operation is also presented.

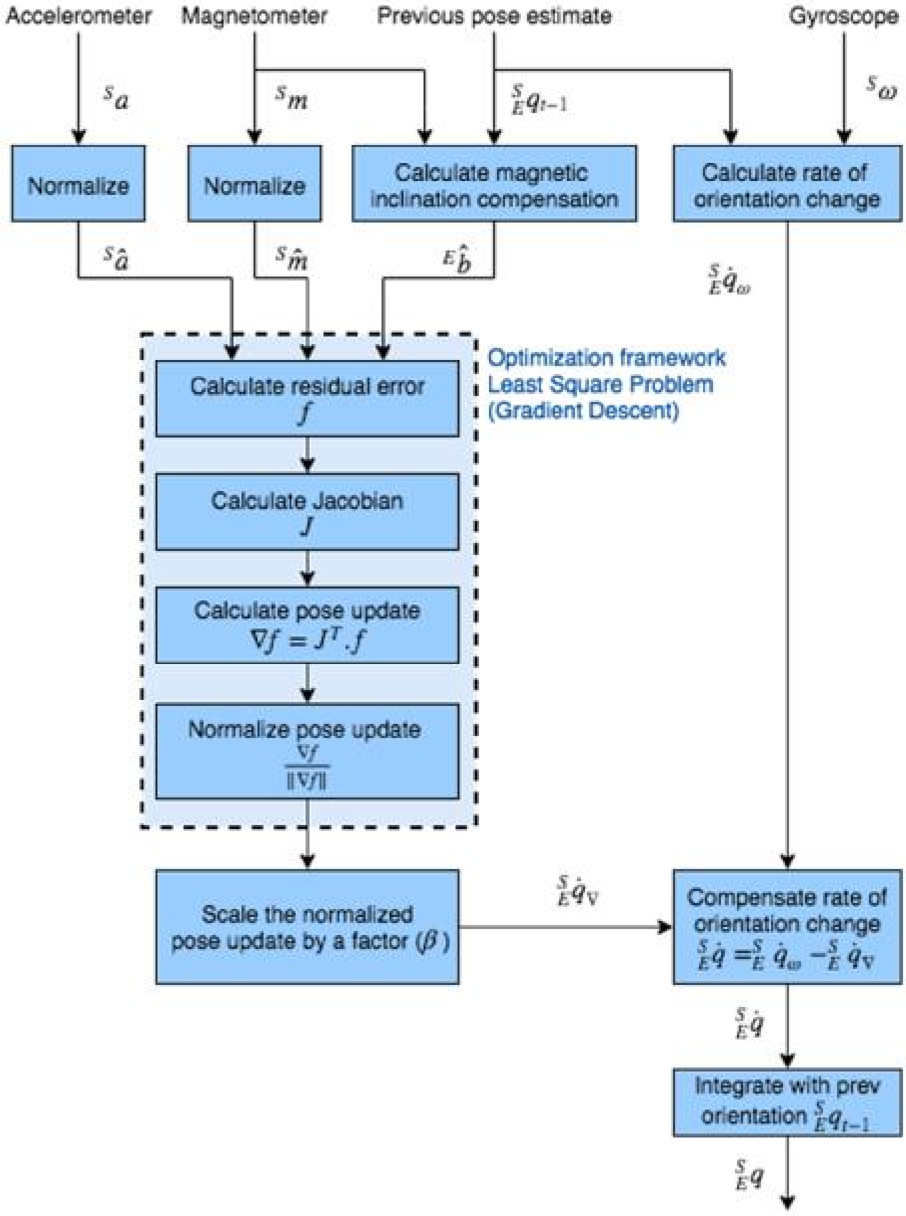

Accelerometer
Magnetometer
Previous pose estimate
Gyroscope
^{S}a
^{S}m
$^{S}_{E}q_{t-1}$
$^{S}\omega$
Normalize
Normalize
Calculate magnetic inclination compensation
Calculate rate of orientation change
$^{S}\hat{a}$
$^{S}\hat{m}$
$^{E}\hat{b}$
$^{S}_{E}\dot{q}_{\omega}$
Optimization framework
Least Square Problem
(Gradient Descent)
Calculate residual error f
Calculate Jacobian J
Calculate pose update $\nabla f = J^{T}.f$
Normalize pose update $\frac{\nabla f}{\|\nabla f\|}$
Scale the normalized pose update by a factor (β)
$^{S}_{E}\dot{q}_{\nabla}$
Compensate rate of orientation change $^{S}_{E}\dot{q} = ^{S}_{E}\dot{q}_{\omega} - ^{S}_{E}\dot{q}_{\nabla}$
$^{S}_{E}\dot{q}$
Integrate with prev orientation $^{S}_{E}q_{t-1}$
$^{S}_{E}q$

Figure 3.1. In the Madgwick method the integration of gyroscope information is achieved outside the Gradient Descent framework and the integration ratio between accelerometer/magnetometer to the gyroscope is specified by a weighting parameter.

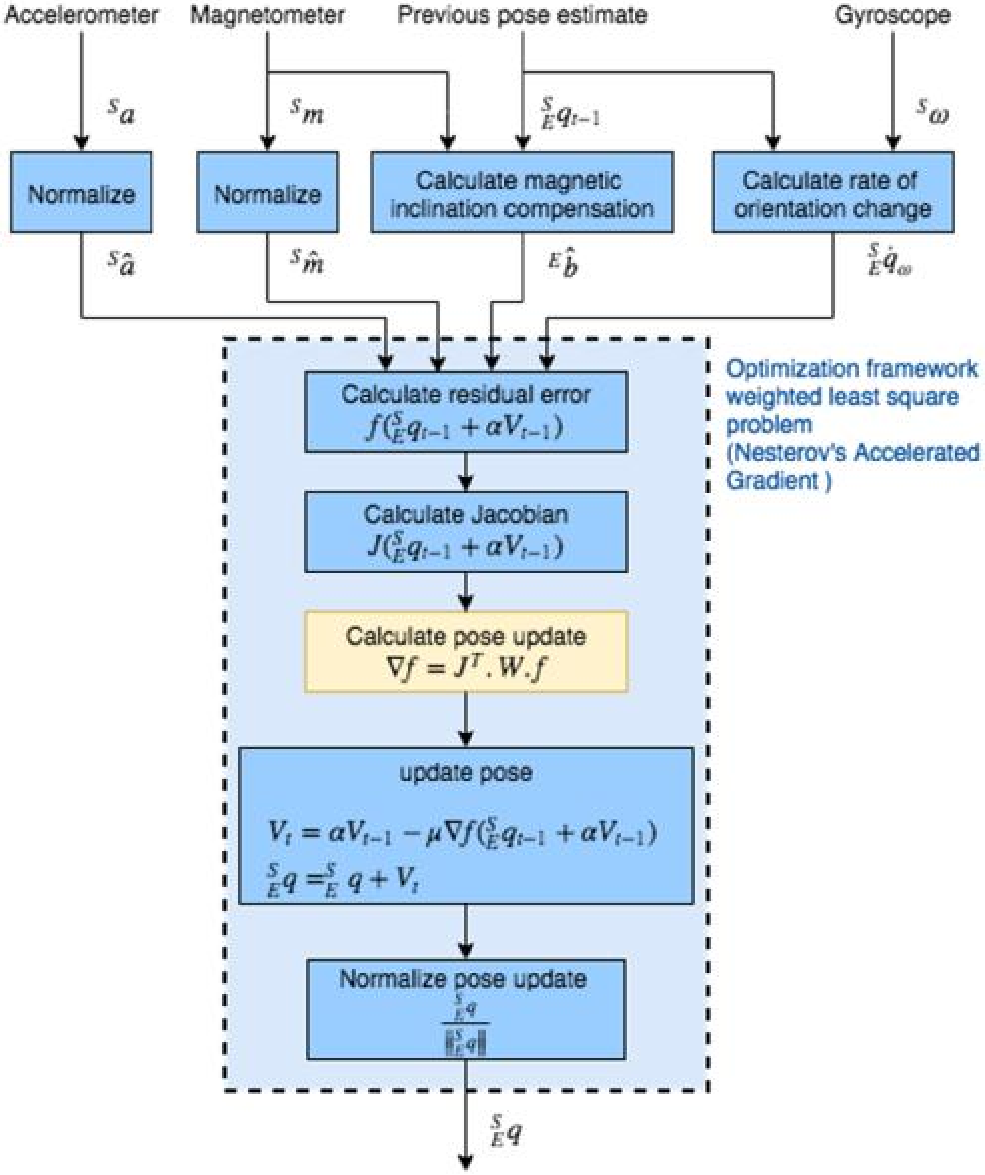

Accelerometer
Magnetometer
Previous pose estimate
Gyroscope
^{S}a
^{S}m
$^{S}_{E}q_{t-1}$
$^{S}\omega$
Normalize
Normalize
Calculate magnetic inclination compensation
Calculate rate of orientation change
$^{S}\hat{a}$
$^{S}\hat{m}$
$^{E}\hat{b}$
$^{S}_{E}\dot{q}_{\omega}$
Optimization framework weighted least square problem (Nesterov's Accelerated Gradient)
Calculate residual error $f(^{S}_{E}q_{t-1}+\alpha V_{t-1})$
Calculate Jacobian $J(^{S}_{E}q_{t-1}+\alpha V_{t-1})$
Calculate pose update $\nabla f = J^{T}.W.f$
update pose
$V_t = \alpha V_{t-1} - \mu \nabla f(^{S}_{E}q_{t-1}+\alpha V_{t-1})$
$^{S}_{E}q =^{S}_{E} q + V_t$
Normalize pose update $\frac{^{S}_{E}q}{\|^{S}_{E}q\|}$
$^{S}_{E}q$

Figure 3.2. The proposed method integrates all measurements including the gyroscope measurement within the optimization scheme. The integration ratio is achieved using a weighting matrix. In this case, the proposed approach can be seen as pure weighted least square optimization problem that is solved by using Nesterov Accelerated Gradient descent.

3.4.1 Quaternion Representation

Orientation of an object in 3D Cartesian space can be represented using a four-dimensional complex number known as a quaternion. Suppose the object's own coordinate system is B and the reference global frame is defined by A, An arbitrary orientation of the object relative to frame A can be achieved through a rotation around an axis $^{A}\hat{r}$ in the amount of an angle θ. The object orientation can then be described using the quaternion $^{A}_{B}\hat{q}$

$$ {}^{A}_{B}\hat{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ -r_x \sin\frac{\theta}{2} \\ -r_y \sin\frac{\theta}{2} \\ -r_z \sin\frac{\theta}{2} \end{bmatrix} $$

A unit quaternion is a quaternion of norm one, such that

$$ \|\hat{q}\|_2 = 1 $$

While ${}^{A}_{B}\hat{q}$ defines the orientation of frame B with respect to frame A, the inverse quaternion that specifies the orientation of frame A with respect to frame B is symbolised as ${}^{B}_{A}\hat{q}$. In this case, ${}^{B}_{A}\hat{q}$ is the conjugate of ${}^{A}_{B}\hat{q}$ or symbolized as ${}^{A}_{B}\hat{q}^*$ and the relation of is defined by

$$ {}^{A}_{B}\hat{q}^* = {}^{B}_{A}\hat{q} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} $$

Given more than 2 systems of coordinate frames, a quaternion product can be used for defining a compound orientation. For example, given two orientations ${}^{A}_{B}\hat{q}$ and ${}^{B}_{C}\hat{q}$ the compound orientation ${}^{A}_{C}\hat{q}$ is given by

$$ {}^{A}_{C}\hat{q} = {}^{B}_{C}\hat{q} \otimes {}^{A}_{B}\hat{q} $$

where the quaternion product, denoted by $\otimes$ is defined by

$$ p \otimes q = [p_0 \quad p_1 \quad p_2 \quad p_3] \otimes [q_0 \quad q_1 \quad q_2 \quad q_3] = \begin{bmatrix} p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 \\ p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2 \\ p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1 \\ p_0q_3 + p_1q_2 - p_2q_1 + p_3q_0 \end{bmatrix} $$

Any vector in 3D Cartesian space can be rotated using a quaternion by slightly modifying the 3D vector into a 4 elements vector by inserting 0 as its first element

$$ v_q = \begin{bmatrix} 0 \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ v_x \\ v_y \\ v_z \end{bmatrix} $$

subsequently, the rotation is given by

$$ {}^{B}v_q = {}^{A}_{B}\hat{q} \otimes {}^{A}v_q \otimes {}^{A}_{B}\hat{q}^* $$

The inverse rotation is achieved by using the conjugate of the rotation quaternion

$$ {}^{A}v_q = {}^{A}_{B}\hat{q}^* \otimes {}^{B}v_q \otimes {}^{A}_{B}\hat{q} = {}^{B}_{A}\hat{q} \otimes {}^{B}v_q \otimes {}^{B}_{A}\hat{q}^* \qquad (3.1) $$

Given a quaternion, a rotation of a 3D vector also can be done by converting the quaternion into a rotation matrix R using

$$ R\left({}^{A}_{B}\hat{q}\right) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} $$

The inverse rotation can then be calculated directly from the quaternion $R\left({}^{A}_{B}\hat{q}\right)$ or by transposing the rotation matrix produced from initial quaternion

$$ {}^{A}v_q = R\left({}^{A}_{B}\hat{q}\right) {}^{B}v = R^T\left({}^{B}_{A}\hat{q}\right) {}^{B}v $$

3.4.2 Madgwick AHRS

Inertial/magnetic sensors consist of accelerometer, magnetometer and gyroscope. Accelerometer measures the gravity and the linear acceleration, magnetometer measures the earth/local magnetic field and gyroscope provides angular rate observation. Each sensor consists of three individual sensors placed in three mutually orthogonal axes so it is capable to produce 3D vector measurement $\hat{s}$. This measurement vector is defined in its own local sensor frame S so the measurement vector is symbolised as ${}^{S}\hat{s}$.

The field being measured ${}^{E}\hat{d}$ can be gravity or earth magnetic field depending on the sensor being used. These fields are assumed to be static and defined using a global earth reference frame following North East Down (NED) convention.

The relation between the observation vector (defined in sensor body frame) and reference field (defined in earth reference frame) as formulated according to Equation 3.1 is then given by

$$ {}^{S}\hat{s} = {}^{E}_{S}\hat{q} \otimes {}^{E}\hat{d} \otimes {}^{E}_{S}\hat{q}^* = {}^{S}_{E}\hat{q}^* \otimes {}^{E}\hat{d} \otimes {}^{S}_{E}\hat{q} \qquad (3.2) $$

where

${}^{S}_{E}\hat{q}$: Quaternion of earth frame with respect to the sensor coordinate system. This quaternion presents full orientation information.

${}^{E}\hat{d}$: Predefined (static) reference field vector in the earth frame

${}^{S}\hat{s}$: Observation of the reference field in the sensor reference frame

Equation 3.2 shows that the rotated field vector ${}^{E}\hat{d}$ according to the correct quaternion ${}^{S}_{E}\hat{q}$ should align to the observation vector in the sensor frame ${}^{S}\hat{s}$. The unknown quaternion may be found by solving the minimisation

$$\min_{{}^{S}_{E}\hat{q} \in \mathbb{R}^4} f\left({}^{S}_{E}\hat{q},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right) \tag{3.3}$$

where

$$f\left({}^{S}_{E}\hat{q},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right) = {}^{S}_{E}\hat{q}^{*} \otimes {}^{E}\hat{d} \otimes {}^{S}_{E}\hat{q} - {}^{S}\hat{s} \tag{3.4}$$

$${}^{S}_{E}\hat{q} = [q_w \quad q_x \quad q_y \quad q_z]$$

$${}^{E}\hat{d} = [0 \quad d_x \quad d_y \quad d_z]$$

$${}^{S}\hat{s} = [0 \quad s_x \quad s_y \quad s_z]$$

For solving the optimization problem, Madgwick implemented a gradient descent which has common form:

$${}^{S}_{E}\hat{q}_{k+1} = {}^{S}_{E}\hat{q}_{k} - \mu \frac{\nabla f\left({}^{S}_{E}\hat{q},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right)}{\left\|\nabla f\left({}^{S}_{E}\hat{q},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right)\right\|} \tag{3.5}$$

where

$$\nabla f\left({}^{S}_{E}\hat{q},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right) = J^{T}\left({}^{S}_{E}\hat{q}_{k},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right)\, f\left({}^{S}_{E}\hat{q}_{k},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right) \tag{3.6}$$

and μ is the step size of the gradient descent. Equation 3.6 shows that Gradient Descent requires residual function $f\left({}^{S}_{E}\hat{q}_{k},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right)$ and the Jacobian of the residual function $J^{T}\left({}^{S}_{E}\hat{q}_{k},\, {}^{E}\hat{d},\, {}^{S}\hat{s}\right)$.

As both field measurements should refer to the same quaternion orientation, a system of equations can be derived for the accelerometer and magnetometer as follows.

Equation System Derived from Accelerometer

For the accelerometer part, the predefined direction field ${}^{E}\hat{d}$ is gravity, expressed in North East Down (NED) earth frame, so the gravity field vector is given by ${}^{E}\hat{g} = [0 \quad 0 \quad 0 \quad 1]$. Symbolizing the normalized accelerometer measurement as ${}^{S}\hat{a} = [0 \quad a_x \quad a_y \quad a_z]$ then substituting into Equation 3.4 to get the residual function from accelerometer part ($\boldsymbol{f}_{acc}$).

$$\boldsymbol{f}_{acc}\left({}^{S}_{E}\hat{\boldsymbol{q}},\ {}^{S}\hat{\boldsymbol{a}}\right) = \begin{bmatrix} 2\left(q_x q_z - q_w q_y\right) - a_x \\ 2\left(q_w q_x + q_y q_z\right) - a_y \\ 2\left(\frac{1}{2} - q_x^2 - q_y^2\right) - a_z \end{bmatrix} \tag{3.7}$$

Partially differentiating Equation 3.7. to obtain the Jacobian of accelerometer's residual function ($\boldsymbol{J}_{acc}$)

$$\boldsymbol{J}_{acc}\left({}^{S}_{E}\hat{\boldsymbol{q}}\right) = \begin{bmatrix} -2q_y & 2q_z & -2q_w & 2q_x \\ 2q_x & 2q_w & 2q_z & 2q_y \\ 0 & -4q_x & -4q_y & 0 \end{bmatrix} \tag{3.8}$$

Equation System Derived from Magnetometer

For the magnetometer part, the predefined direction is magnetic field. Due to the inclination of the magnetic field to the horizontal, the vector only has components in the horizontal axis (x) and vertical axis (z) so ${}^{E}\hat{b} = [0 \quad b_x \quad 0 \quad b_z]$ (Madgwick et al., 2011). The compensation for magnetic field inclination error is computed by rotating the normalized magnetometer measurement by the previously estimated orientation as follows.

$${}^{E}h_t = [0 \quad h_x \quad h_y \quad h_z] = {}^{S}_{E}\hat{q}_{t-1} \otimes \ {}^{S}m_t \otimes {}^{S}_{E}\hat{q}^{*}_{t-1}$$

$${}^{E}b_t = \left[0 \quad \sqrt{h_x^2 + h_y^2} \quad 0 \quad h_z\right]$$

Along with the normalized magnetometer reading ${}^{S}\hat{\boldsymbol{m}} = [0 \quad m_x \quad m_y \quad m_z]$ the residual function ($\boldsymbol{f}_{mag}$) and the Jacobian ($\boldsymbol{J}_{mag}$) of the magnetometer can be computed by substituting into Equation 3.4

$$\boldsymbol{f}_{mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}},\ {}^S\hat{\boldsymbol{m}}\right) = \begin{bmatrix} 2b_x\left(\frac{1}{2} - q_y^2 - q_z^2\right) + 2b_z\left(q_xq_z - q_wq_y\right) - m_x \\ 2b_x\left(q_xq_y - q_wq_z\right) + 2b_z\left(q_wq_x + q_yq_z\right) - m_y \\ 2b_x\left(q_wq_y + q_xq_z\right) + 2b_z\left(\frac{1}{2} - q_x^2 - q_y^2\right) - m_z \end{bmatrix} \tag{3.9}$$

$$\begin{aligned} &\boldsymbol{J}_{mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}}\right) \\ &= \begin{bmatrix} -2b_zq_y & 2b_zq_z & -4b_xq_y - 2b_zq_w & -4b_xq_z + 2b_zq_x \\ -2b_xq_z + 2b_zq_x & 2b_xq_y + 2b_zq_w & 2b_xq_x + 2b_zq_z & -2b_xq_w + 2b_zq_y \\ 2b_xq_y & 2b_xq_z - 4b_zq_x & 2b_xq_w - 4b_zq_y & 2b_xq_x \end{bmatrix} \end{aligned} \tag{3.10}$$

Combining Accelerometer and Magnetometer into a Single System of Equations

Having the residual function for accelerometer and magnetometer as presented in Equation 3.7 and Equation 3.9 respectively, and the Jacobian of accelerometer and magnetometer (Equation 3.8 and Equation 3.10), the orientation can be computed by combining these equations into a system of non-linear equations that can then be solved for the orientation ${}^S_E\hat{q} = [q_w \quad q_x \quad q_y \quad q_z]$

$$f_{acc,mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^S\hat{\boldsymbol{a}},\ {}^E\hat{\boldsymbol{b}},\ {}^S\hat{\boldsymbol{m}}\right) = \begin{bmatrix} \boldsymbol{f}_{acc}\left({}^S_E\hat{\boldsymbol{q}},\ {}^S\hat{\boldsymbol{a}}\right) \\ \boldsymbol{f}_{mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}},\ {}^S\hat{\boldsymbol{m}}\right) \end{bmatrix} \tag{3.11}$$

$$J_{acc,mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}}\ \right) = \begin{bmatrix} \boldsymbol{J}_{acc}\left({}^S_E\hat{\boldsymbol{q}}\right) \\ \boldsymbol{J}_{mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}}\right) \end{bmatrix}^T \tag{3.12}$$

Recalling the general form of Gradient Descent as presented in Equation 3.5 and Equation 3.6 the update rate is defined by

$$\nabla f\ = J^T_{acc,mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}}\ \right).f_{acc,mag}\left({}^S_E\hat{\boldsymbol{q}},\ {}^S\hat{\boldsymbol{a}},\ {}^E\hat{\boldsymbol{b}},\ {}^S\hat{\boldsymbol{m}}\right) \tag{3.13}$$

The orientation estimate from accelerometer and magnetometer is then calculated as follows

$${}^S_E\hat{\boldsymbol{q}}_\nabla = {}^S_E\hat{\boldsymbol{q}}_{t-1} - \mu_t\frac{\nabla f}{\|\nabla f\|} \tag{3.14}$$

The parameter μ_t specifies the step size for refining the estimate, where in the physical meaning, it relates to the angular rate measured by the gyroscope. Therefore, the Madgwick method calculates the step size from: 1. the absolute value of gyroscope measurement; 2. the sample periods; and 3. a free-to-tune parameter α to account for measurement uncertainty (Madgwick et al., 2011). The step size is then computed according to

$$\mu_t = \alpha \left\| {}^S_E\dot{q}_{\omega,t} \right\| \Delta t \tag{3.15}$$

Referring to Equation 3.14, ${}^S_E\hat{\boldsymbol{q}}_\nabla$ refers to the orientation estimate gained from the gradient descent method that just considers accelerometer and magnetometer.

The gyroscope measurement is then fused outside the optimization framework, first by calculating gyroscope's orientation estimate according to:

$$ {}^S_E q_{\omega,t} = {}^S_E\hat{q}_{\omega,t-1} + \left(\frac{1}{2} \ {}^S_E\hat{q}_{\omega,t-1} \otimes {}^S\omega_t \right) . \Delta t \tag{3.16}$$

After both estimates from the accelerometer/magnetometer ${}^S_E\hat{q}_\nabla$ and gyroscope ${}^S_E q_{\omega,t}$ have been obtained, they are combined by

$$ {}^S_E\hat{q}_{est,t} = \gamma_t {}^S_E\hat{q}_\nabla + (1-\gamma_t) {}^S_E q_{\omega,t} \tag{3.17}$$

Madgwick method then performs a further derivation for γ_t by defining some assumptions, which is not covered in this paper as it does not directly relate to the proposed algorithm. Interested reader can refer to (Madgwick et al., 2011) for detailed explanation.

3.4.3 Nesterov Accelerated Gradient Descent (NAG)

Classical Gradient Descent (GD) is an extremely popular optimization method because of its simplicity and it only requires the first derivative of the objective function. In general, GD iteratively updates the estimate in the opposite direction of the gradient, factored by a step size. The selection of step size is hard since a small step size can avoid jumping over optima but at the same time increases the convergence rate. In contrast, a large step size can provoke an unnecessary oscillation and at some point can lead to divergence. The convergence rate also depends on the objective function characteristic and the distance between initial guess and the solution (Sutskever et al., 2013).

An improved GD version with a better convergence rate is achieved by accumulating previous gradients in a decaying manner. This approach is known as Momentum method (Sutskever et al., 2013). This Momentum approach maintains progress along the direction of previous update and typically reaches the solution in a shorter time. However, since it accumulates the gradients, the momentum can be very high and possibly surpass the optimum. Nesterov proposed a better method by computing the gradient correction velocity at the predicted position ahead $\nabla f(\theta_t + \alpha v_t)$ instead of the gradient at the current location $\nabla f(\theta_t)$. Nesterov's Accelerated Gradient (NAG) is also able to smooth oscillations by slowing the update in the unnecessary direction when the gradient oscillates. The Nesterov Accelerated Gradient is given by

$$v_{t+1} = \alpha v_t - \mu \nabla f(\theta_t + \alpha v_t)$$

$$\theta_{t+1} = \theta_t + v_{t+1}$$

Nesterov's accelerated gradient has a convergence rate $O(1/k^2)$ which is better than classical Gradient Descent that has $O(1/k)$ convergence rate. Where k is a constant proportional to the derivative and the squared Euclidean distance to the solution (Sutskever et al., 2013). The implementation of the NAG has a big potential to improve the original Madgwick algorithm that implemented classical Gradient Descent.

3.5 Proposed Method: NAG-AHRS

The proposed method (referred to Nesterov Accelerated Gradient descent-AHRS / NAG-AHRS) algorithm can be seen as an extension of the Madgwick algorithm. To give a clear idea on how each algorithm differs see Figure 3.1. From Figure 3.1 it is clear that whilst Madgwick method only combines accelerometer and magnetometer into optimization framework, the proposed approach combines all sensors into optimization framework.

Recall Equation 3.16 which calculates the gyroscope orientation update. It can then be rearranged as a residual function f_{gyro} as

$$f_{gyro}\left({}^S_E\hat{q}_t, {}^S_E q_{t-1}, {}^S\omega_t\right) = {}^S_E\hat{q}_{\omega,t-1} - {}^S_E q_{\omega,t} + \left(\frac{1}{2}\,{}^S_E\hat{q}_{\omega,t-1} \otimes {}^S\omega_t\right).\Delta t$$

As the orientation was expressed in quaternion form, f_{gyro} consists of the set of equations

$$f_{gyro}\left({}^{S}_{E}\hat{q}_t, {}^{S}_{E}q_{t-1}, {}^{S}\omega_t\right)$$
$$= \begin{bmatrix} q_{w,t-1} - q_{w,t} + \frac{\Delta t}{2}\left(-q_{x,t-1}.\omega_x - q_{y,t-1}.\omega_y - q_{z,t-1}.\omega_z\right) \\ q_{x,t-1} - q_x + \frac{\Delta t}{2}\left(q_{w,t-1}.\omega_x - q_{z,t-1}.\omega_y + q_{y,t-1}.\omega_z\right) \\ q_{y,t-1} - q_y + \frac{\Delta t}{2}\left(q_{z,t-1}.\omega_x + q_{w,t-1}.\omega_y - q_{x,t-1}.\omega_z\right) \\ q_{z,t-1} - q_z + \frac{\Delta t}{2}\left(q_{x,t-1}.\omega_y - q_{y,t-1}.\omega_x + q_{w,t-1}.\omega_z\right) \end{bmatrix} \quad (3.18)$$

The Jacobian of the gyroscope residual function with respect to the quaternion q_w, q_x, q_y, q_z is

$$J_{gyro}\left({}^{S}_{E}\hat{q}_t, {}^{S}_{E}q_{t-1}, {}^{S}\omega_t\right) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (3.19)$$

Having f_{gyro} and its Jacobian J_{gyro}, the proposed NAG-AHRS incorporates Equations 3.18 and 3.19 to the whole system of non-linear equations:

$$\boldsymbol{f}\left({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{S}\hat{\boldsymbol{a}}, {}^{E}\hat{\boldsymbol{b}}, {}^{S}\hat{\boldsymbol{m}}\right) = \begin{bmatrix} \boldsymbol{f}_{acc}\left({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{S}\hat{\boldsymbol{a}}\right) \\ \boldsymbol{f}_{mag}\left({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}, {}^{S}\hat{\boldsymbol{m}}\right) \\ \boldsymbol{f}_{gyro}\left({}^{S}_{E}\hat{q}_t, {}^{S}_{E}q_{t-1}, {}^{S}\omega_t\right) \end{bmatrix} \quad (3.20)$$

where the Jacobian is given by

$$\boldsymbol{J}\left({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}\right) = \begin{bmatrix} \boldsymbol{J}_{acc}\left({}^{S}_{E}\hat{\boldsymbol{q}}\right) \\ \boldsymbol{J}_{mag}\left({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}\right) \\ J_{gyro}\left({}^{S}_{E}\hat{\boldsymbol{q}}_t, {}^{S}_{E}\hat{\boldsymbol{q}}_{t-1}, {}^{S}\boldsymbol{\omega}_t\right) \end{bmatrix} \quad (3.21)$$

In the detailed form, the residual function and the Jacobian are:

$$\boldsymbol{J}\left({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}\right)$$
$$= \begin{bmatrix} -q_y & q_z & -q_w & q_x \\ q_x & q_w & q_z & q_y \\ 0 & -2q_x & -2q_y & 0 \\ -b_z q_y & b_z q_z & -2b_x q_y - b_z q_w & -2b_x q_z + b_z q_x \\ -b_x q_z + b_z q_x & b_x q_y + b_z q_w & b_x q_x + b_z q_z & -b_x q_w + b_z q_y \\ b_x q_y & b_x q_z - 2b_z q_x & b_x q_w - 2b_z q_y & b_x q_x \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (3.22)$$

$$f\left({}_{E}^{S}\hat{\boldsymbol{q}},\ {}^{S}\hat{\boldsymbol{a}},\ {}^{E}\hat{\boldsymbol{b}},\ {}^{S}\hat{\boldsymbol{m}}\right) = \begin{bmatrix} q_x q_z - q_w q_y - \frac{1}{2} a_x \\ q_w q_x + q_y q_z - \frac{1}{2} a_y \\ \frac{1}{2} - q_x^2 - q_y^2 - \frac{1}{2} a_z \\ \frac{1}{2} b_x - b_x q_y^2 - b_x q_z^2 + b_z q_x q_z - b_z q_w q_y - \frac{1}{2} m_x \\ b_x q_x q_y - b_x q_w q_z + b_z q_w q_x + b_z q_y q_z - \frac{1}{2} m_y \\ b_x q_w q_y + b_x q_x q_z + \frac{1}{2} b_z - b_z q_x^2 - b_z q_y^2 - \frac{1}{2} m_z \\ q_{p,w} - q_w + \frac{\Delta t}{2} \left(-q_{p,x}.\omega_x - q_{p,y}.\omega_y - q_{p,z}.\omega_z\right) \\ q_{p,x} - q_x + \frac{\Delta t}{2} \left(q_{p,w}.\omega_x - q_{p,z}.\omega_y + q_{p,y}.\omega_z\right) \\ q_{p,y} - q_y + \frac{\Delta t}{2} \left(q_{p,z}.\omega_x + q_{p,w}.\omega_y - q_{p,x}.\omega_z\right) \\ q_{p,z} - q_z + \frac{\Delta t}{2} \left(q_{p,x}.\omega_y - q_{p,y}.\omega_x + q_{p,w}.\omega_z\right) \end{bmatrix} \tag{3.23}$$

The orientation estimate is obtained by solving these systems of non-linear equations.

Madgwick method implements a fusion parameter γ to facilitate a different weight factor between accelerometer and magnetometer as shown in Equation 3.17. The proposed method accounts this γ parameter as weighting matrix W. Given this weighting matrix, the update becomes:

$$\lambda = \lambda - \mu J^T.W.f \tag{3.24}$$

where W is a diagonal matrix that specifies the weights and the f is a generic objective function. Following the Madgwick method for assigning the weight between accelerometer/magnetometer and gyroscope as presented in Equation 3.17, the weight matrix is used to express γ as

$$W = diag\left(\gamma, \gamma, \gamma, \gamma, \gamma, \gamma, (1-\gamma), (1-\gamma), (1-\gamma), (1-\gamma)\right) \tag{3.25}$$

The first 6 diagonal elements are related with accelerometer and magnetometer system of equations, hence the assigned weight is γ. The last 4 elements are linked to the gyroscope so the weight is $(1-\gamma)$.

While the Madgwick method solves the optimisation problem using standard Gradient Descent, the proposed method implements Nesterov Accelerated

Gradient (NAG) method that has a faster convergence time. The orientation estimate can then be solved by using NAG as follows:

$$v_t = \alpha . v_{t-1} - \mu J\left({}^S_E\hat{q}_{t-1} + \alpha . v_{t-1}\right)^T . W . f\left({}^S_E\hat{q}_{t-1} + \alpha . v_{t-1}\right) \tag{3.26}$$

$${}^S_E\hat{q}_t = {}^S_E\hat{q}_{t-1} + v_t \tag{3.27}$$

where μ is given by Equation 3.15 and α is the momentum parameter of NAG. Figure 3.3 expresses this method in block diagram.

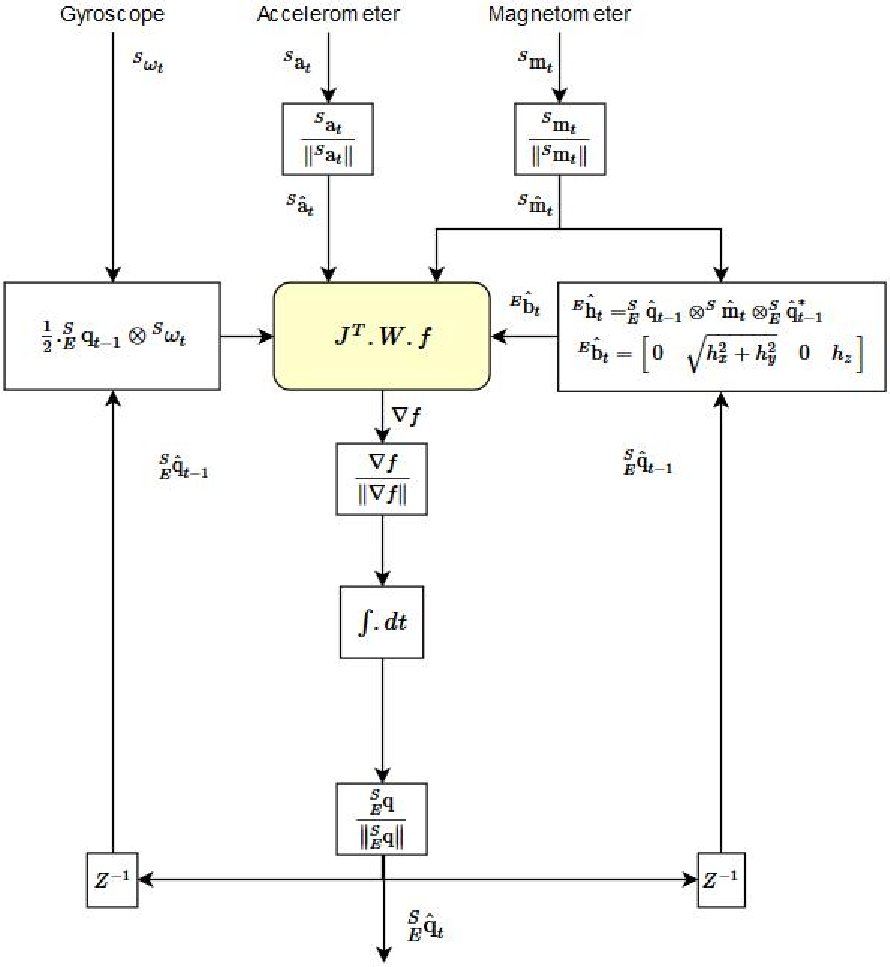

Gyroscope
Accelerometer
Magnetometer
$^S\omega_t$
$^S\mathrm{a}_t$
$^S\mathrm{m}_t$
$\frac{^S\mathrm{a}_t}{\|^S\mathrm{a}_t\|}$
$\frac{^S\mathrm{m}_t}{\|^S\mathrm{m}_t\|}$
$^S\hat{\mathrm{a}}_t$
$^S\hat{\mathrm{m}}_t$
$\frac{1}{2} \cdot {}^S_E\mathrm{q}_{t-1} \otimes {}^S\omega_t$
$J^T . W . f$
$^E\hat{\mathrm{b}}_t$
$^E\hat{\mathrm{h}}_t = {}^S_E\hat{\mathrm{q}}_{t-1} \otimes {}^S\hat{\mathrm{m}}_t \otimes {}^S_E\hat{\mathrm{q}}^*_{t-1}$
$^E\hat{\mathrm{b}}_t = \left[0 \quad \sqrt{h_x^2 + h_y^2} \quad 0 \quad h_z\right]$
∇f
$\frac{\nabla f}{\|\nabla f\|}$
${}^S_E\hat{\mathrm{q}}_{t-1}$
${}^S_E\hat{\mathrm{q}}_{t-1}$
$\int . dt$
$\frac{{}^S_E\mathrm{q}}{\|{}^S_E\mathrm{q}\|}$
Z^{-1}
Z^{-1}
${}^S_E\hat{\mathrm{q}}_t$

Figure 3.3. NAG-AHRS algorithm that addressed inertial/magnetic orientation estimate as weighted least square minimization problem and solves it using Nesterov Accelerated Gradient. The residual function of the proposed NAG-AHRS method refers to Equation 3.23, the Jacobian refers to Equation 3.22 and the weighting matrix W refers to Equation 3.25.

3.6 Experiments and Results

This section covers the evaluation of the proposed method. Sub-section 3.6.1. presents the selected dataset for the experiment. This sub-section covers the description of the dataset, the structure and the motion scenarios of the dataset. To gain more comprehensive performance assessment, the proposed method is also benchmarked to other state-of-the-art algorithms which are presented in this sub-section. After the dataset and the benchmarking algorithms have been presented, sub-section 3.6.2 describes the parameters setup for all algorithms. The experimental results then are presented in sub-section 3.6.3.

3.6.1 Dataset and Benchmarking Algorithms

The proposed method was validated using an inertial/magnetic dataset provided by Silesian University of Technology (Szczęsna et al., 2016). This dataset was selected as it provides recorded inertial/magnetic sensor observation along with accurate reference obtained from a motion capture system. The inertial/magnetic sensor was Xsens type MTi-G-28 A53 G35 and the recording has been synchronised to Vicon Nexus observations. The dataset also provides a comprehensive motion scenario such as slow and fast rotation, structured and random rotations as well as structured translation and freehand motion. The datasets used for validation can be seen in Table 3.1

Table 3.1. Datasets for validating the proposed method (NAG-AHRS) and also for benchmarking with other state-of-the-art algorithms such as Extended Kalman Filter-AHRS (EKF-AHRS), QUEST, TRIAD, Mahony filter and Madgwick Filter.

Dataset	sampling (Hz)	duration (sec)	note
Dataset 1	100	89.92	Slow constant rotation (< 20 deg/sec) about X axis
Dataset 2	100	89.62	Slow constant rotation (< 20 deg/sec) about Y axis
Dataset 3	100	88.42	Slow constant rotation (< 20 deg/sec) about Z axis
Dataset 4	100	88.11	Fast constant rotation (> 75 deg/sec) about X axis
Dataset 5	100	87.31	Fast constant rotation (> 75 deg/sec) about Y axis
Dataset 6	100	89.99	Fast constant rotation (> 75 deg/sec) about Z axis
Dataset 7	100	89.94	Freehand slow rotation about three axes (non-linear rotation)
Dataset 8	166	57.8	Pendulum motion
Dataset 9	100	89.99	Translation along X axis
Dataset 10	100	89.99	Translation along Y axis
Dataset 11	100	89.99	Translation along Z axis

These motion scenarios were chosen to investigate the performance of the proposed algorithm as well as the performance of the benchmarking algorithms. Datasets 1-6 serve to exploit the effect of motion speed to the estimation accuracy, while Datasets 7 and 8 were chosen mainly to investigate the presence of highly non-linear motion on the estimate. In Datasets 1-6, the sensor was steered carefully in a constrained motion to achieve a constant rotation. The difference between these datasets was only the speed and the rotation axes. Datasets 7 and 8 contain highly non-linear motion since they consist of random freehand motion recording (Dataset 7) and pendulum swing motion (Dataset 8).

The last three datasets (Dataset 9-11) complement the motion scenario by providing disturbance due to linear motion. The linear motion will be picked up by accelerometer and affect the gravity observation and at the end it may disrupt the orientation estimate.

To obtain more comprehensive performance validation, the proposed algorithm was not only compared to reference obtained from motion 0capture system, but it was also benchmarked to the other inertial/magnetic AHRS algorithms. The list of other algorithm that had been selected can be seen in Table 3.2

Table 3.2. List of AHRS algorithms that have been used for benchmarking the proposed algorithm. These algorithms were selected as they are the best performing in each of their class.

Algorithms	category
EKF-AHRS (Angelo Maria Sabatini & Maria, 2011)	stochastic method
TRIAD (D. Black, 1964)	deterministic method
QUEST (Shuster & Oh, 1981)	deterministic method
Mahony Filter (Euston et al., 2008)	complementary filter
Madgwick Filter (Madgwick et al., 2011)	complementary filter

Kalman Filter for estimating orientation inertial magnetic estimate have been developed using various methods. For benchmarking purpose, an EKF-AHRS method proposed by (Angelo Maria Sabatini & Maria, 2011) that highly cited and has been widely used for benchmarking was selected.

3.6.2 Tuning the Algorithm's Parameters

The proposed algorithm and other five benchmarking algorithms have some free-to-tune parameters that influence their performance significantly. To get a fair comparison between algorithms, these parameters have to be selected carefully, and this was done as follows:

- The EKF-AHRS requires the measurement of the noise variance of the three sensor. Since the dataset was obtained from the Silesian University of Technology the noise variance is already measured and this research adapted these values which are σ_a^2 = 0.001 (accelerometer), σ_g^2 = 0.0001 (gyroscope) and σ_m^2 = 0.000001 (magnetometer)(Szczęsna & Pruszowski, 2016). Since in the selected EKF-AHRS algorithm the gyroscope is used for computing the state transition matrix, the process covariance is calculated as follows (Angelo Maria Sabatini & Maria, 2011):

$$Q = \left(\frac{\Delta t}{2}\right)^2 \Xi\left(\sigma_g^2\, I_{4\times 4}\right)\Xi^T \quad \text{where } \Xi = \begin{bmatrix} q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \\ -q_x & -q_y & -q_z \end{bmatrix}$$

- TRIAD algorithm has no free parameters to tune
- QUEST has parameters w_1 and w_2 for facilitating different weighting for accelerometer and magnetometer measurements. The magnetometer is related to w_1 and the accelerometer measurement weight is w_2. In this experiment, a value of w_1 = 62.5% and w_2 = 37.5% were used adopting the research done by (Kuga & Carrara, 2013) that demonstrated a good result.
- Madgwick method has one free-to-tune parameter β and in the experiment this parameter was set to β = 0.033 as suggested in the original paper (Madgwick et al., 2011)
- Mahony has 2 free-to-tune parameters K_p and K_i and these parameters were K_p = 0.04 and K_i = 0 as suggested in the original paper (Euston et al., 2008)
- The proposed method - NAG-AHRS only has 1 free-to-tune parameter γ as this parameter was derived from Madgwick $\gamma = 0.033.\frac{\Delta t}{\mu_t}$ was selected.

3.6.3 Experiment Results

The experiment was done using a computer with Intel Core i5-4590 CPU @3.30GHz, 4G RAM and NVIDIA Quadro K620 graphic card. The dataset that consists of accelerometer, gyroscope and magnetometer readings was loaded into memory and then processed by the algorithms to yield outputs. The algorithms were developed using Matlab. The attitude output was then converted into Euler angle and compared to the Vicon reading that also available from the dataset. The error between the estimate and the reference was then used to compute the RMSE and MAE.

All eleven datasets were investigated and then the RMSE and MAE of each dataset result for each algorithms were then recorded. The output of all RMSE and MAE is analysed to gain the performance of the algorithms. Some of the orientation results are presented also as representative of all dataset. The selected presented dataset outputs are:

- Dataset 1 as the representation of slow constant pure rotation
- Dataset 6 as the representation of fast constant pure rotation
- Dataset 8 as the representation of highly non-linear motion
- Dataset 11 as the representation of the presence of linear motion

Dataset 1 – slow constant rotation about x axis

The first output presented here is Dataset 1 which consists of two full rotations (720°) about the x axis. Figure 3.4 shows the orientation estimate of each algorithm. This dataset can be considered as the easiest motion scenario as the rotation was slow, hence, from the output, it shows that all algorithms generally managed to track the rotation about x axis. However, from visual observation, the orientation estimate of roll angle ω_x from EKF-AHRS is slightly misaligned to the reference while other algorithms had better fit.

An easier observation can be seen from Figure 3.5 that presents an absolute error. The absolute error is obtained by comparing the output to the reference given by the motion capture system. Figure 3.5 shows the EKF-AHRS ω_x error was higher than the other algorithms and it is confirmed by the orientation output in Figure 3.4 that shows a large displacement with respect to the reference.

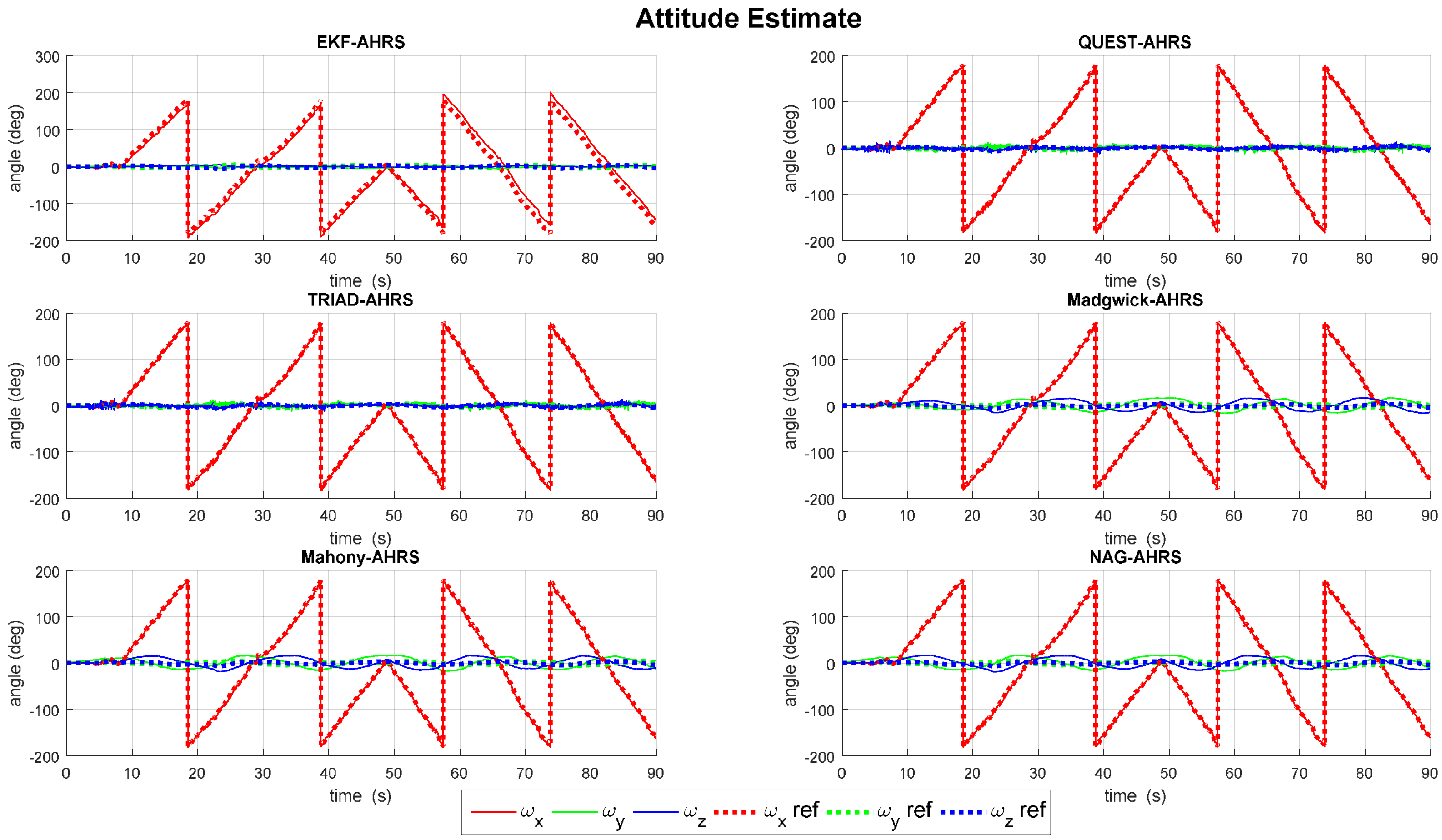


Figure 3.4. The orientation estimate given Dataset 1 as input. The dataset consists of a slow constant angular velocity around x axis. Other axes (y and z axes) were not experienced any rotation. It general, all algorithms managed to track the orientation. Some delayed responses were observed in the EKF-AHRS and some noises were observed in ω_y and ω_z of TRIAD and QUEST algorithms. The other three algorithms: Madgwick-AHRS, Mahony-AHRS and NAG-AHRS had similar results that suffered from oscillation in in ω_y and ω_z estimates.

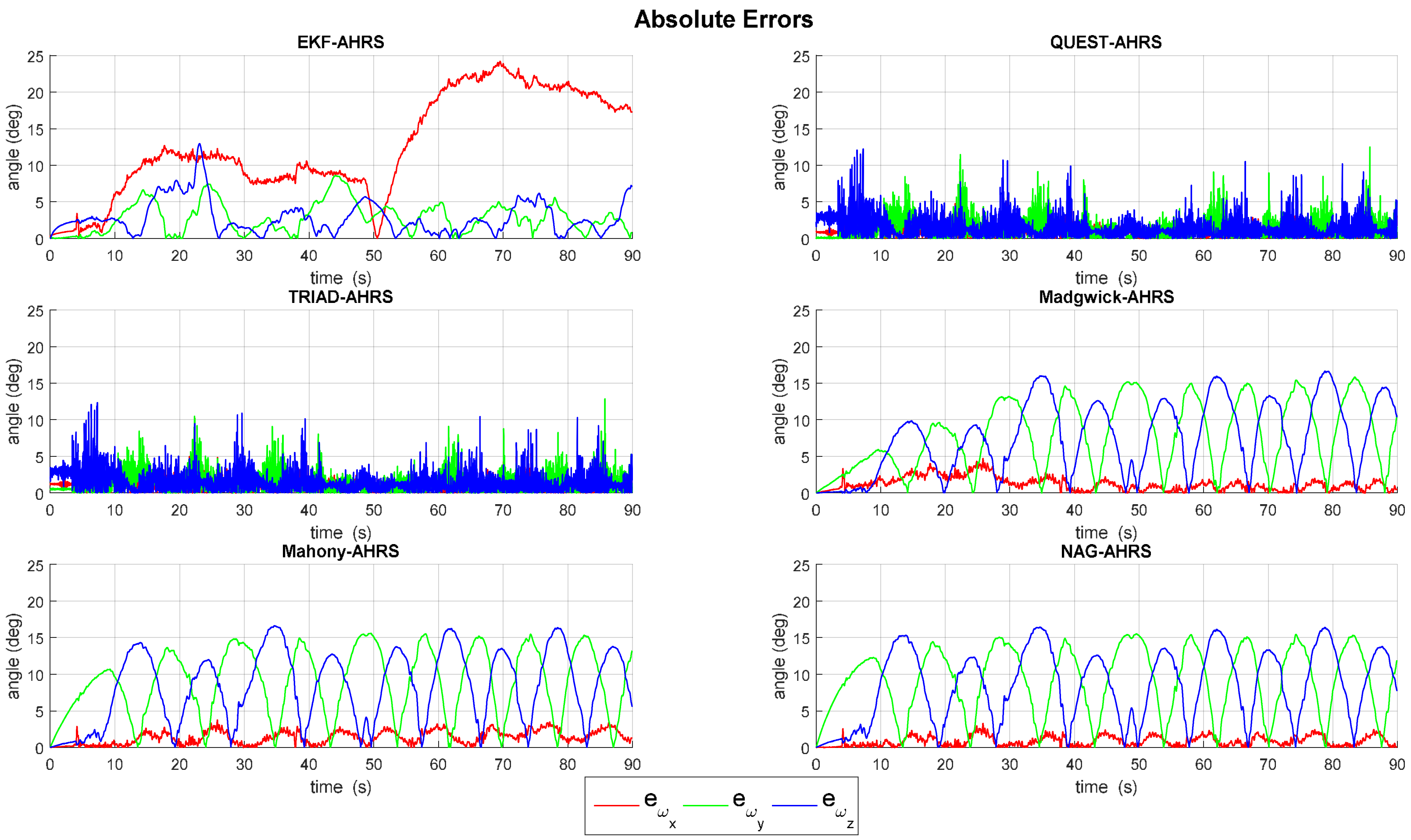


Figure 3.5. The absolute error given Dataset 1. EKF-AHRS shows a different the orientation error before and after 50 seconds. The different errors were observed due to a different angular rotation speed. After 50 seconds, the angular velocity was a slightly higher and then it the error become higher. QUEST and TRIAD achieved a similar result while the other three algorithms: Mahony, Madgwick and NAG-AHRS also yielded similar results.

Figure 3.5 also shows that EKF-AHRS error is influenced by the rotation speed. During the first 50 seconds, the rotation speed was slightly slower. This different speed yields a difference error level with about 10 ° in the slow rotation period (first 50 seconds) and 20 ° for faster rotation period (after 50 seconds). The observed high level of error is interesting since the input motion can be considered linear as shown by Figure 3.4 and the EKF-AHRS was expected to achieve a small error. Even though the motion was linear, the models within EKF-AHRS algorithm were a linearization of – actually – non-linear model hence it breach the optimality criteria of Kalman filter and contributes to the presence of large error.

A lower error level was observed for QUEST and TRIAD algorithms with maximal observed error less than 14° at any time. However as these algorithms estimate the orientation from a single-recent measurement, the outputs were very responsive and very noisy. This confirms that the deterministic single-frame cannot suppress noise.

The last three algorithms behave similarly in this motion scenario. The three algorithm can track the moving angle ω_x precisely but the other angles (ω_y and ω_z) suffer from large oscillations. The selected K_p in Mahony, β in Madgwick matrix contributes to this behaviour. The setting up of these parameters were intended to be able to track a fast motion changes and the setting become too big for the static angle (Euston et al., 2008; Madgwick et al., 2011). The NAG-AHRS was built from Madgwick algorithm hence it behave similarly. An RMSE and MAE observation is presented in Table 3.3

Table 3.3 The RMSE and MAE measurements of Dataset 1 that consist of slow linear rotation about axis x. The bold text indicate the lowest RMSE/MAE score

Algorithms	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
EKF-AHRS	14.28	3.504	3.618	12.526	2.823	2.828
QUEST-AHRS	1.461	**1.802**	**2.054**	1.234	**1.297**	**1.547**
TRIAD-AHRS	1.583	1.846	2.093	1.335	1.378	1.592
Madgwick-AHRS	1.534	9.249	9.095	1.234	7.987	7.659
Mahony-AHRS	1.642	10.036	9.705	1.375	8.953	8.361
NAG-AHRS	**1.07**	10.225	9.809	**0.836**	9.174	8.495

Dataset 6 – Fast constant rotation about z axis

The next presented output was from Dataset 6 that consists of fast linear rotation scenario. The result can be seen in Figure 3.6 and 3.7. As the motion speed was faster, the output from EKF-AHRS suffers from large error due to its inability to catch up the right attitude within a right time. The single-point deterministic methods (QUEST and TRIAD) managed to track the attitude in this higher dynamics motion as shown in Figure 3.6 but it again failed to supress the noise. This noisy output can be seen easily from the absolute error as presented in Figure 3.7. The best performance for this motion scenario was shown from the complementary filter method (Mahony and Madgwick filter) as well as the proposed method (NAG-AHRS). The RMSE/MAE table for this fast rotation dataset is provided in Table 3.4. The Mean Absolute Error of complementary filter as well as the NAG-AHRS were less than 4° and the outputs were also smooth.

Table 3.4 The RMSE and MAE measurements of Dataset 6 that consist of fast linear rotation about axis z. The bold text indicate the lowest RMSE/MAE score

Algorithms	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
EKF-AHRS	73.63	37.042	92.493	54.876	28.43	74.232
QUEST-AHRS	12.551	14.384	37.651	8.425	8.728	23.469
TRIAD-AHRS	13.684	17.046	37.719	9.384	11.235	23.392
Madgwick-AHRS	2.825	**2.1**	4.007	2.223	**1.69**	3.091
Mahony-AHRS	3.947	2.712	**2.467**	3.193	2.158	**1.828**
NAG-AHRS	**2.779**	2.143	3.777	**2.191**	1.723	2.854

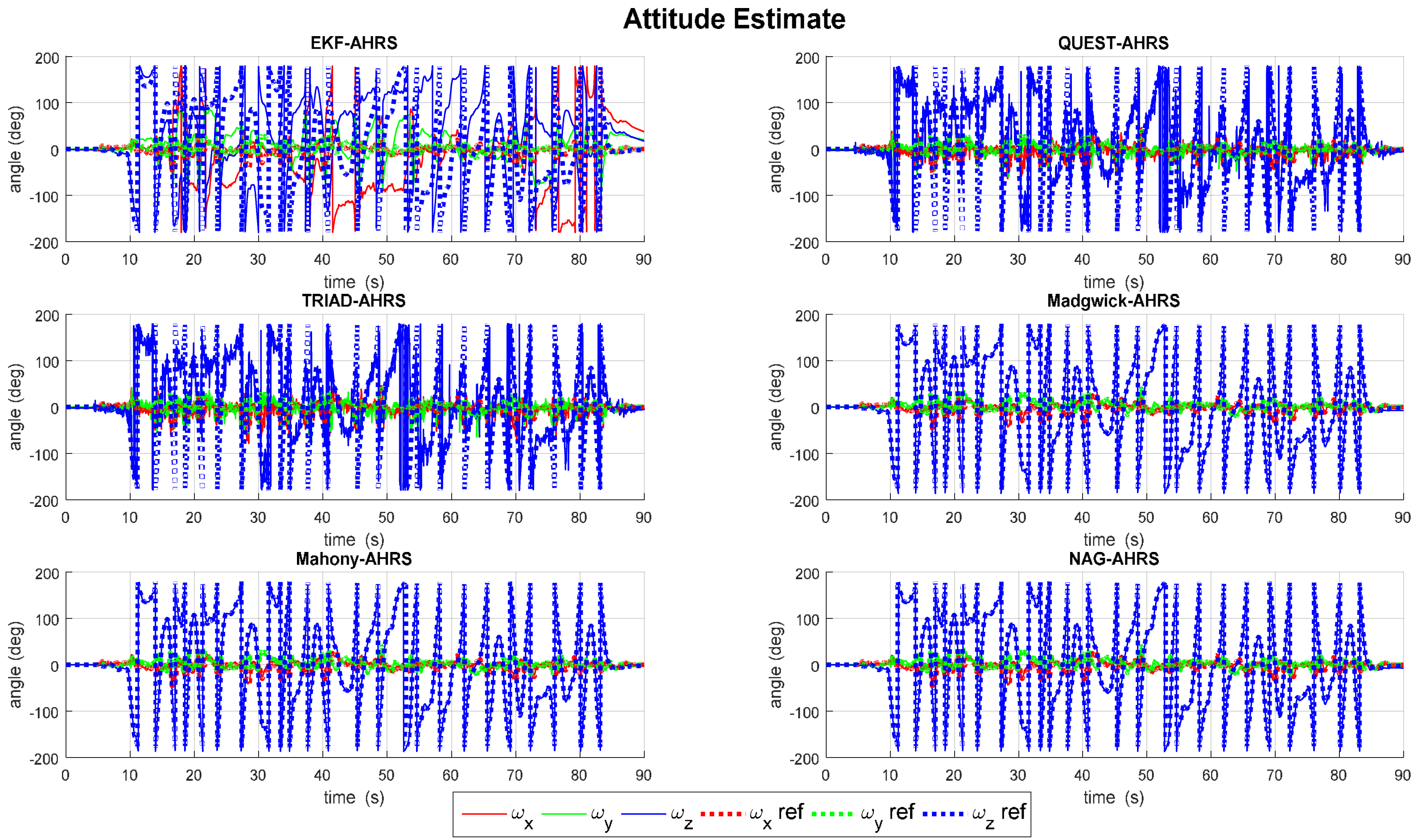


Figure 3.6. Orientation estimate of fast rotation about z axis (Dataset 6). The EKF-AHRS performance that is highly affected by the motion speed was not able to track accurately. Better tracking was observed from TRIAD and QUEST but these algorithms were not supressing any noise hence the noisy output was achieved. The Madgwick, Mahony and NAG-AHRS performed similarly and the difference is observed by the plot of absolute error as shown in Figure 3.7

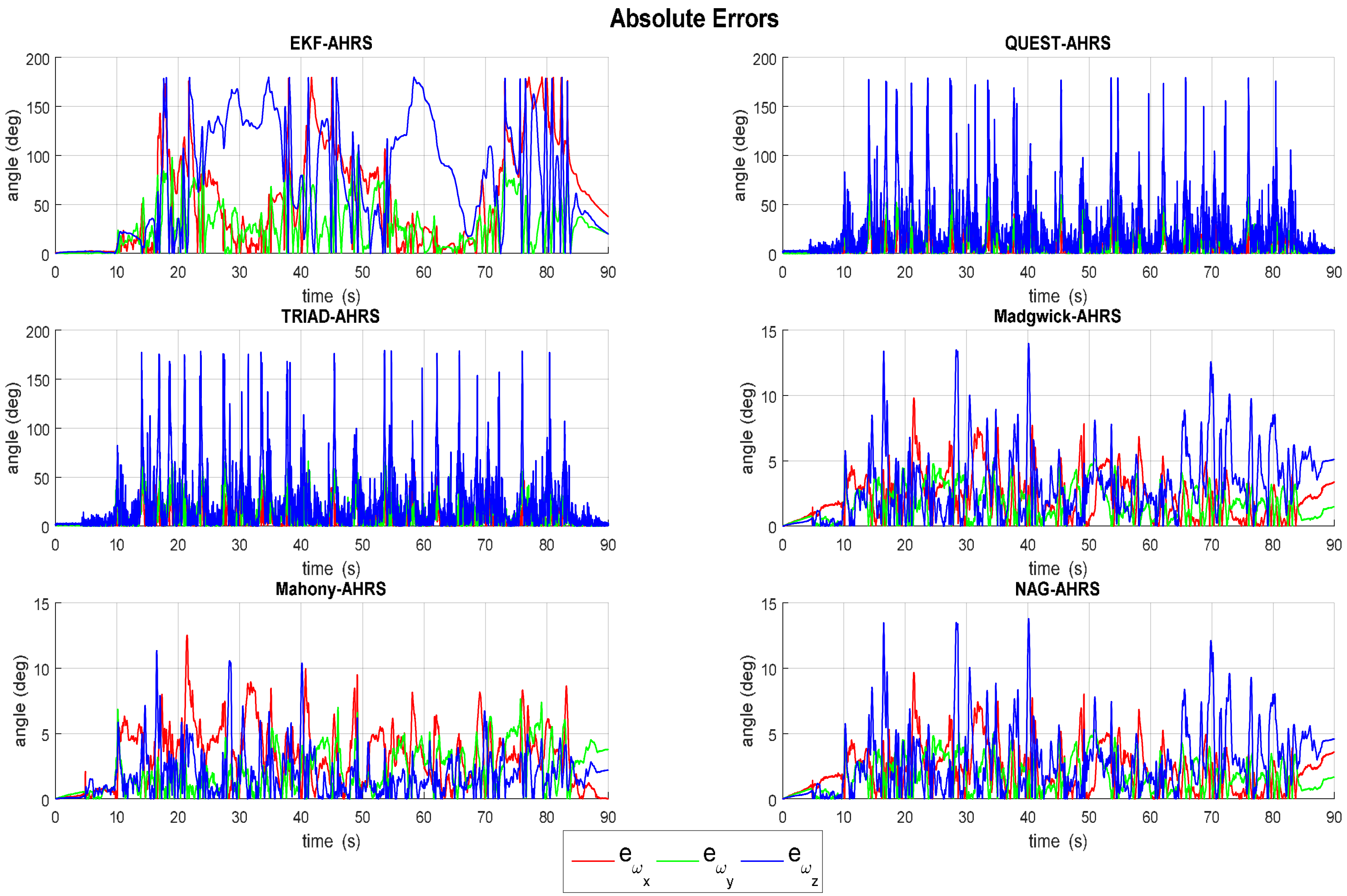


Figure 3.7. Absolute orientation error given fast constant rotation about z axis (Dataset 6). This high motion speed triggered large error in EKF-AHRS. The fast response but noisy outputs were observed at QUEST and TRIAD algorithms. Madgwick, Mahony and NAG-AHRS had a similar result with Mahony that performed best in this motion scenario

Dataset 8 – Pendulum motion

Dataset 8 is more challenging since it consists of a highly non-linear motion. Figure 3.8 shows the EKF-AHRS can track the attitude but from error plot in Figure 3.9 it shows EKF-AHRS suffered from large error. Interestingly TRIAD and QUEST that were expected to have a rapid response also suffered from large error. Even though TRIAD and QUEST only need to process the very recent measurement, the accelerometer and magnetometer measurements have a slow update rate, hence the outputs were not fast. Better performance was shown by the complementary filter category. However, in this motion scenario, the proposed algorithm NAG-AHRS achieved the best performance with the lowest RMSE and MAE.

The optimisation approach can handle non-linearity better and at the same time can take benefit of integrating gyroscope to its framework to achieve lower response delay.

Table 3.5 The RMSE and MAE measurements of Dataset 8 that consist of non-linear motion obtained from pendulum swing about z axis. The bold text indicate the lowest RMSE/MAE score

Algorithms	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
EKF-AHRS	10.815	5.708	4.658	6.265	3.945	2.667
QUEST-AHRS	11.316	2.548	5.464	5.795	0.859	3.102
TRIAD-AHRS	18.163	5.765	16.994	4.195	3.148	4.89
Madgwick-AHRS	1.325	3.443	3.087	0.79	2.491	2.074
Mahony-AHRS	1.372	1.481	2.862	0.81	1.042	1.837
NAG-AHRS	**0.772**	**0.667**	**1.327**	**0.485**	**0.437**	**0.881**

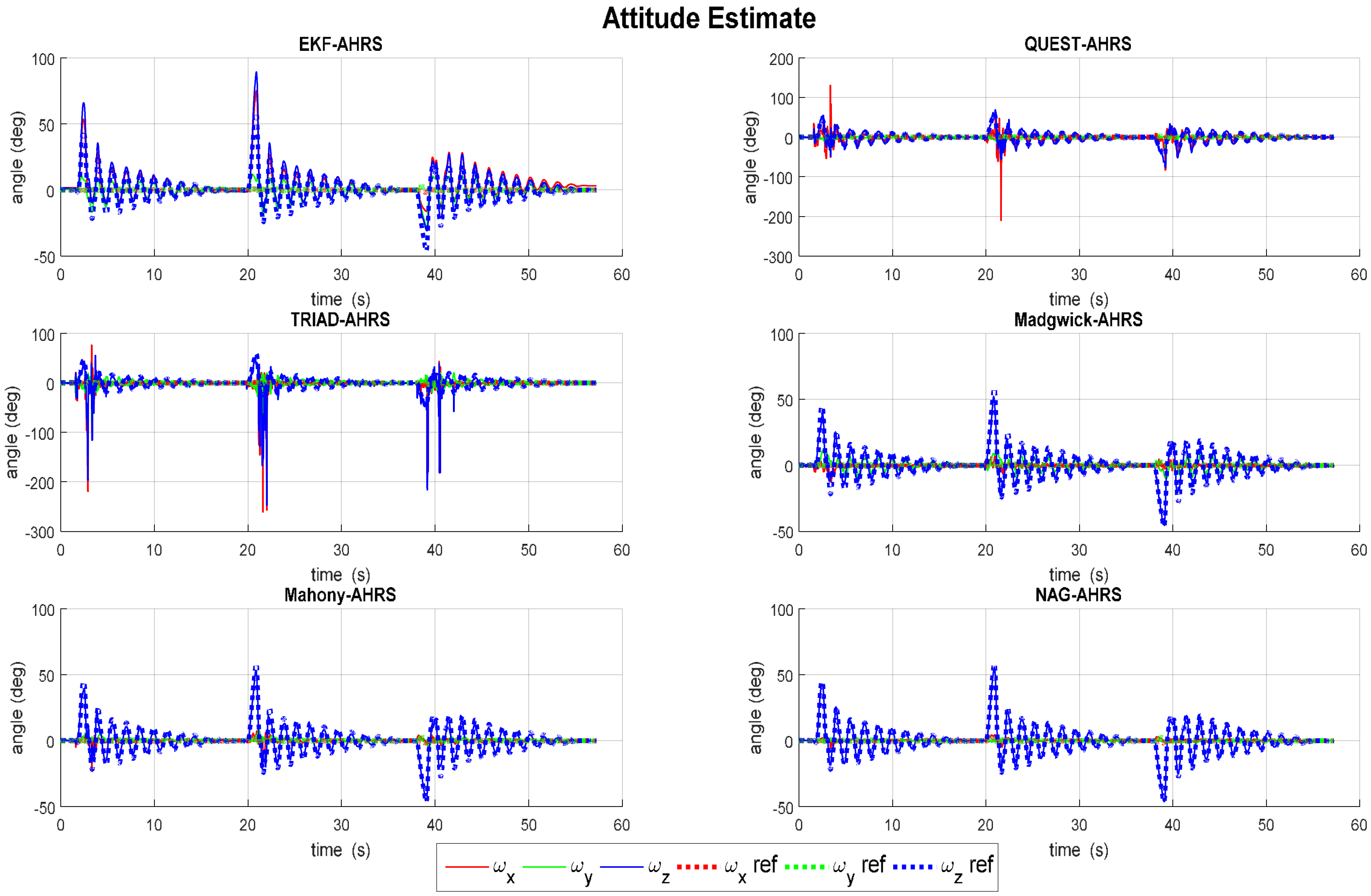


Figure 3.8 Orientation estimate output for pendulum swing motion (Dataset 8). This dataset is chosen to investigate how a highly non-linear motion affects the orientation estimate. It shows all algorithms can track the attitude in a sufficiently good performance, except for two algorithms: TRIAD and QUEST. TRIAD and QUEST attitude estimate experienced some spikes so the accuracy was lower than other algorithms. A better observation can be seen from the absolute error plot that is presented in Figure 3.9.

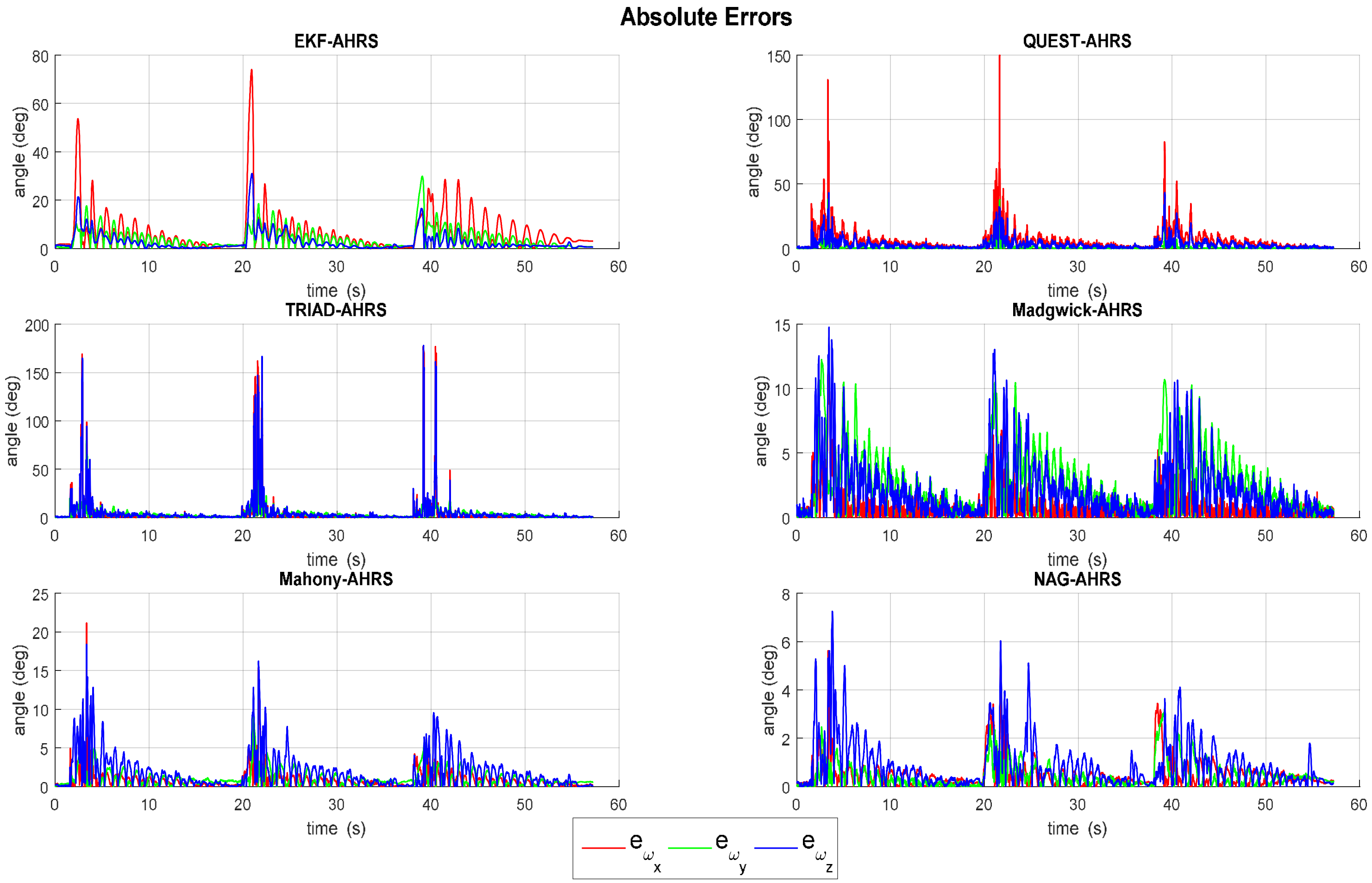


Figure 3.9. Absolute orientation error of pendulum swing motion. The worst performance is observed for TRIAD and QUEST. EKH-AHRS was second worst and it demonstrated the Kalman based orientation was not good in handling highly non-linear motion. Complementary Filter category: Madgwick-AHRS and Mahony-AHRS achieved a significantly better performance with lower error. However, the proposed pure optimisation attitude estimate EKF-AHRS outperformed all of other algorithms.

Dataset 11 – translation along z axis

The next experiment investigated the effect of disturbance due to linear-motion as simulated by Dataset 11. As can be seen from Figure 3.10 it shows the yaw estimate ω_z from Kalman Filter suffered from large error, while the roll and pitch had a better output. The QUEST and TRIAD algorithms suffer from linear motion noise and yielded a large error for all three angles (ω_x, ω_y and ω_z). The best performance was obtained from the proposed NAG-AHRS method. Whereas the yaw angle estimate ω_z of NAG-AHRS was bigger than Madgwick method, the output of NAG-AHRS was more consistent. In Madgwick method, the yaw angle estimate was the best and achieved a RMSE of only 2.147, but other two angles estimate roll and pitch (ω_x, ω_y) were high. A better consistency was achieved by the Mahony filter but still less accurate than NAG-AHRS.

Table 3.6. The RMSE and MAE measurements of Dataset 11 that consist of translation motion along z axis. The bold text indicates the lowest RMSE/MAE score

Algorithms	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
EKF-AHRS	5.669	3.701	27.486	4.563	2.754	24.548
QUEST-AHRS	28.593	19.884	32.674	10.841	8.924	17.018
TRIAD-AHRS	46.096	15.299	30.636	17.128	6.695	15.939
Madgwick-AHRS	24.784	8.018	**2.147**	20.81	6.258	**1.721**
Mahony-AHRS	7.8	3.233	5.242	5.695	2.646	4.228
NAG-AHRS	**5.117**	**2.545**	4.393	**4.137**	**1.804**	3.723

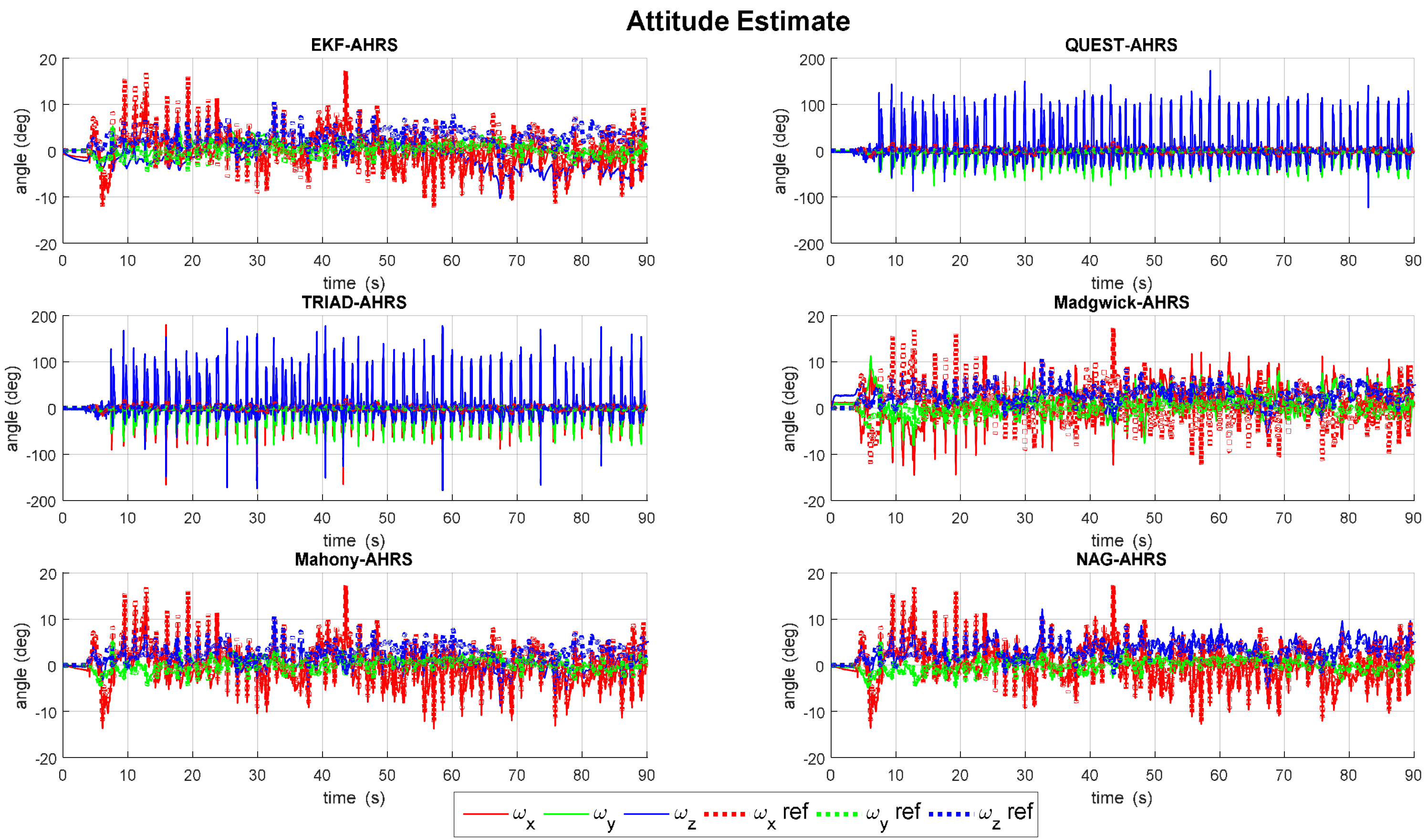


Figure 3.10. Orientation estimate output of 6 algorithms given Dataset 11 which consists of translation motion along z axis. This motion scenario was selected to investigate the effect of linear motion to the attitude estimate output as the presence of linear motion influences the accelerometer measurement. This noise had a significant effect in degrading the accuracy of ω_z in EKF-AHRS algorithm (a) as it can be seen the ω_z drifted away from the ground truth. TRIAD and QUEST that only perform estimate from current measurement suffered heavily with this noise and yielded a very large error. Mahony-AHRS, Madgwick-AHRS and NAG-AHRS had a similar output that is difficult to observe from this chart visually. A clearer observation can be seen from the absolute error as presented in Figure 3.11.

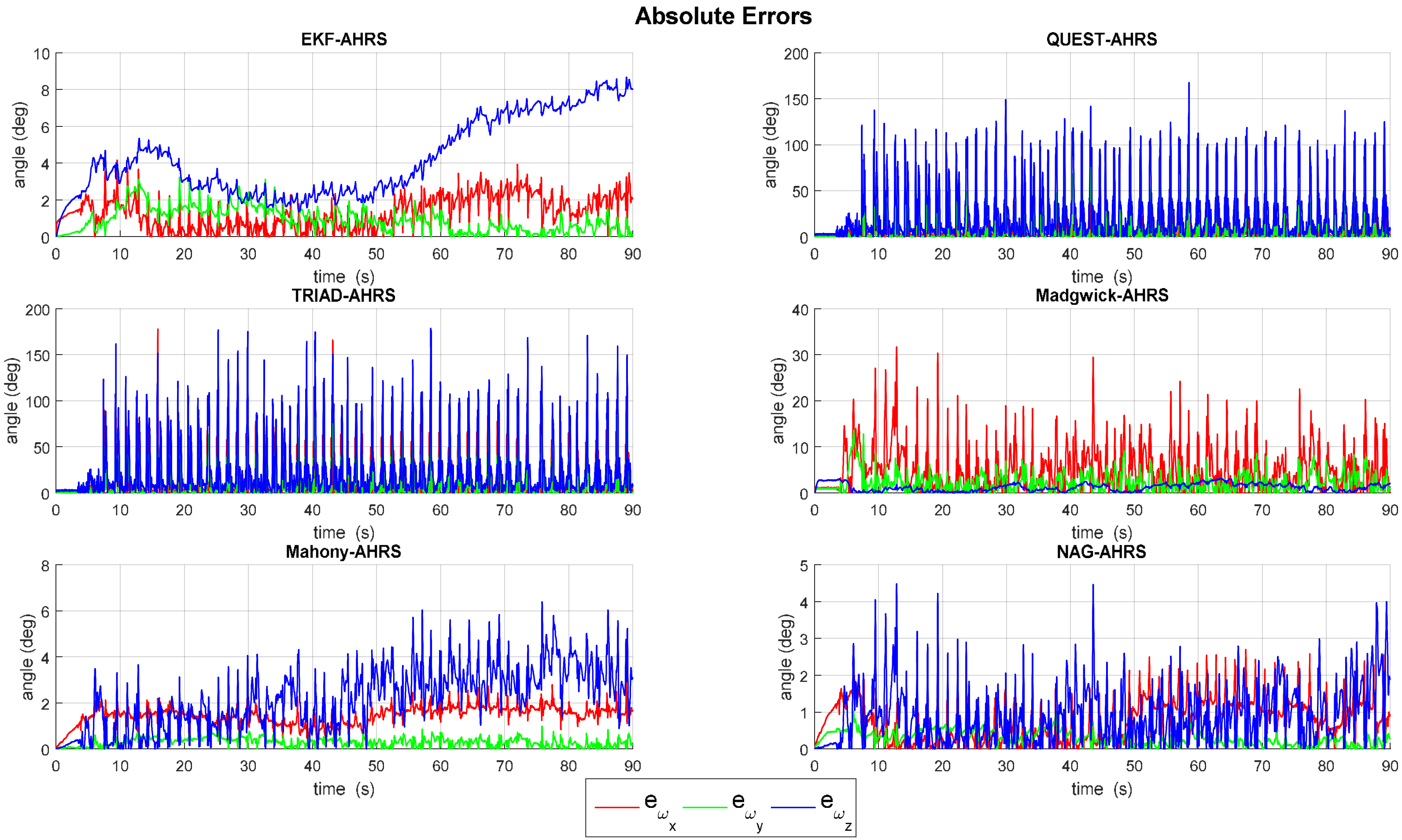


Figure 3.11. Absolute error output of 6 algorithms given Dataset 11. EKF-AHRS has not converged and yielded large error. TRIAD and QUEST that do not have any method for suppressing noise also suffered from large absolute error. While Madgwick and Mahony method performed better than TRIAD/QUEST, the smallest absolute error was achieved by NAG-AHRS algorithm

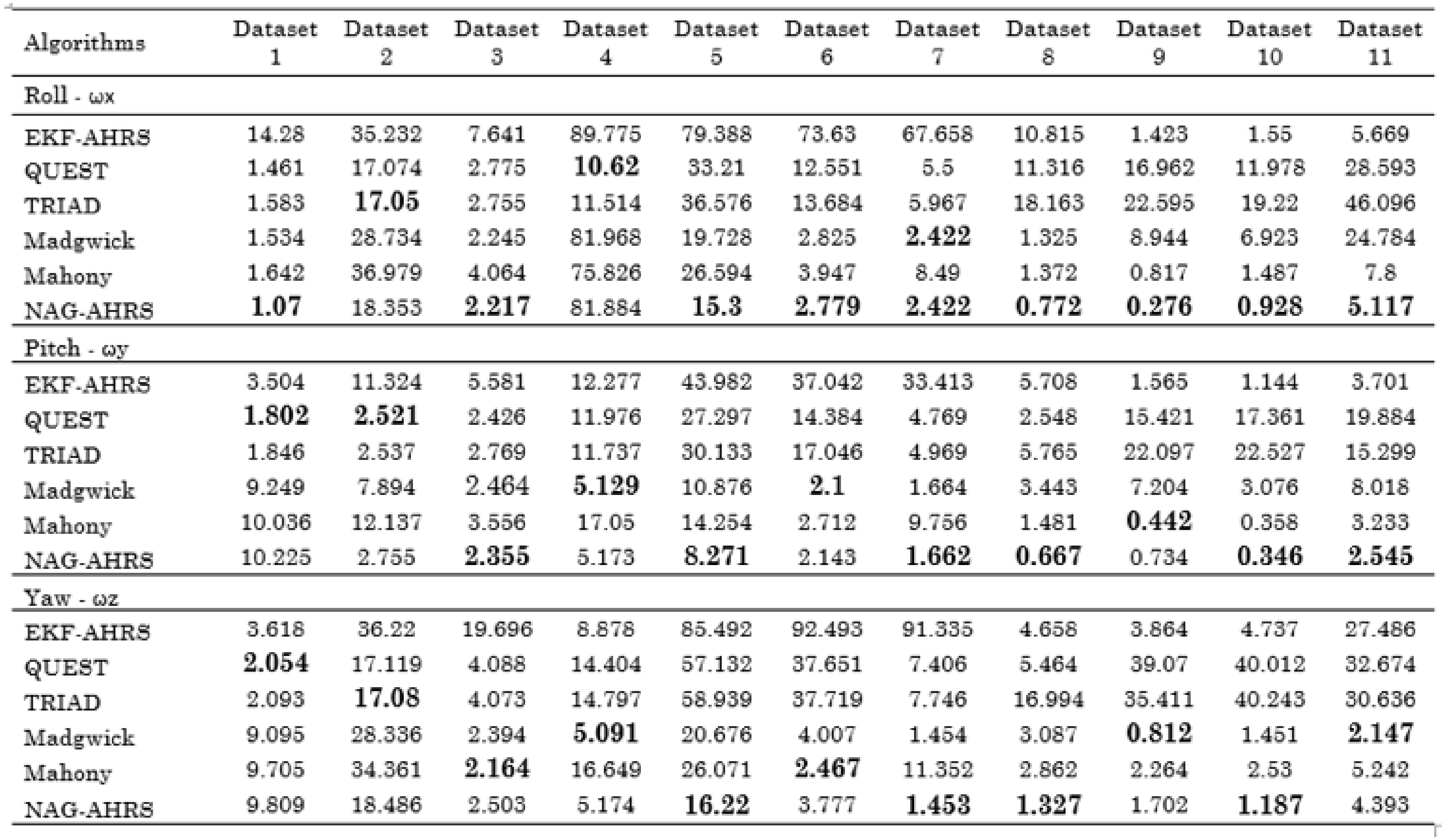
The summary of RMSE and MAE is presented in Table 3.7 and 3.8. Both tables show that the performance of EKF-AHRS was better in slow motion (Dataset 1 - 3) and in low non-linear motion as shown in pendulum swing motion (Dataset 8). The performance of EKF-AHRS degraded significantly in fast motion (Dataset 4-6) and in highly non-linear motion (Dataset 7). This behaviour was expected as EKF-AHRS is better in dealing with systems with low non-linarites. In terms of dealing with noise, EKF-AHRS managed to suppress the disturbance better than single-frame deterministic method (Dataset 9 – 11). The bad orientation estimate from Kalman-based algorithm on pendulum motion scenario (Dataset 8) also observed by (Szczęsna & Pruszowski, 2016). In their research the average observed error achieved more than 13°.

Single-frame deterministic methods TRIAD and QUEST demonstrated a fast response hence its performance in low noise dataset (Dataset 1-11) were considerably good. However, the performance degrades significantly in the presence of noise (Dataset 5-6 and Dataset 9 -10). The performance of TRIAD/QUEST cannot be accessed only from RMSE/MAE due to their noisy output. A better observation can be done from the plot of orientation output and absolute error. From the orientation output plot, it clearly shows that the output of TRIAD and QUEST were noisy.

A better and smoother performance was obtained from complementary filter category (Mahony and Madgwick filter). This category of filter in general can suppress the noise while maintaining good accuracy of orientation estimate. This performance achieved as the complementary filter does not require a linearity assumption for both motion model, neither observation model. In contrast to the TRIAD algorithm, the complementary filter also take benefit of fusing all three sensors to get a better performance.

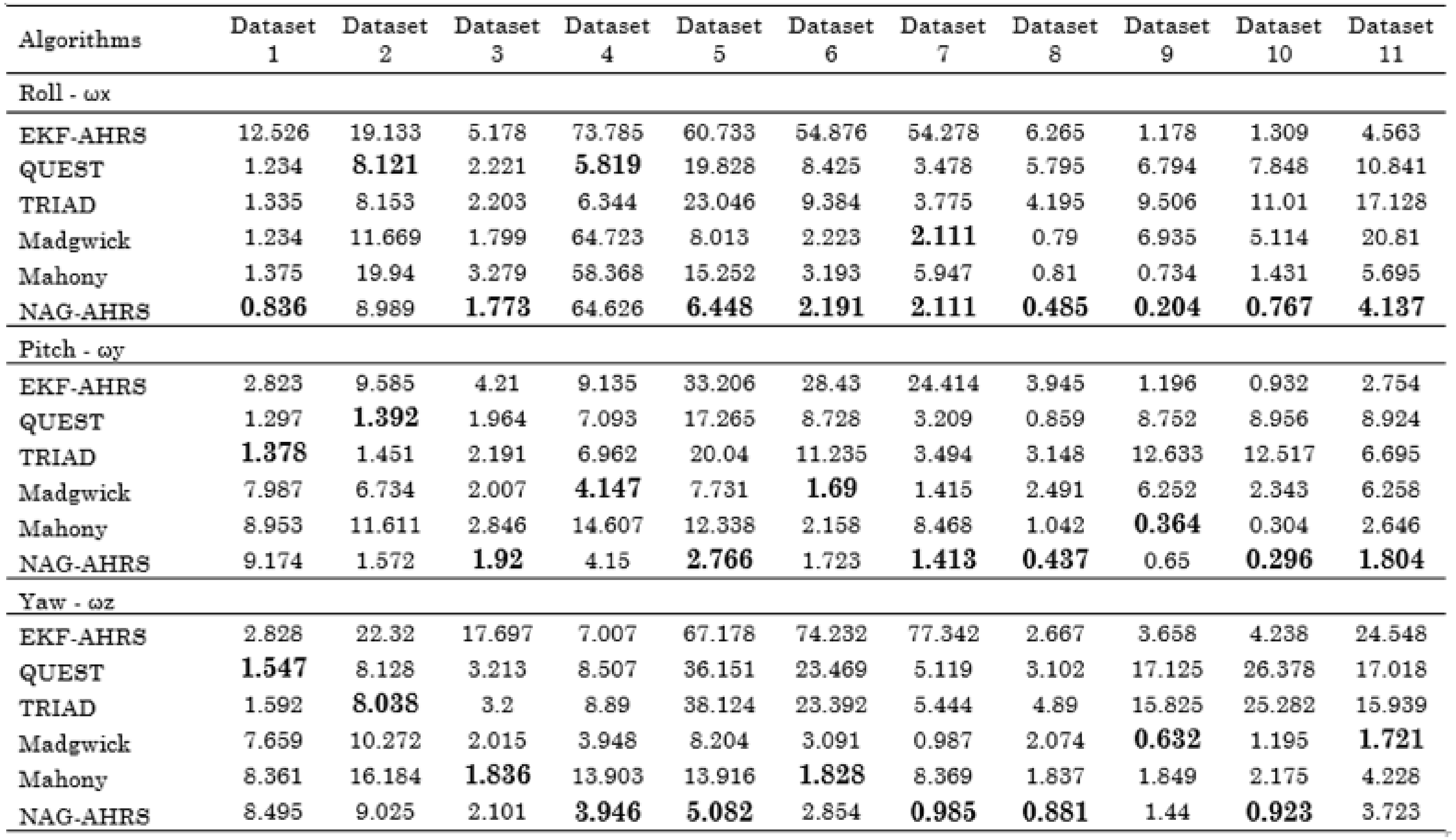
The proposed method, referred to NAG-AHRS, that was built as a pure optimisation framework demonstrated competitive results. NAG-AHRS achieved lowest error in more motion scenarios than other benchmarking algorithms as follows summarised in Table 3.9. This achievement of NAG-AHRS outperformed other benchmarking algorithms.

Table 3.7. The RMSE result of 11 datasets and 6 algorithms. The bold text shows the smallest RMSE. From the table it shows that the proposed algorithm yielded lowest error in more datasets than other algorithms. Note: Dataset 1-3 slow linear rotation, Dataset 4-6 fast linear rotation, Dataset 7 freehand non-liner motion, Dataset 8 pendulum swing non-linear motion, Dataset 9-11 translation motions



Algorithms	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9	Dataset 10	Dataset 11
Roll - ωx											
EKF-AHRS	14.28	35.232	7.641	89.775	79.388	73.63	67.658	10.815	1.423	1.55	5.669
QUEST	1.461	17.074	2.775	**10.62**	33.21	12.551	5.5	11.316	16.962	11.978	28.593
TRIAD	1.583	**17.05**	2.755	11.514	36.576	13.684	5.967	18.163	22.595	19.22	46.096
Madgwick	1.534	28.734	2.245	81.968	19.728	2.825	**2.422**	1.325	8.944	6.923	24.784
Mahony	1.642	36.979	4.064	75.826	26.594	3.947	8.49	1.372	0.817	1.487	7.8
NAG-AHRS	**1.07**	18.353	**2.217**	81.884	**15.3**	**2.779**	**2.422**	**0.772**	**0.276**	**0.928**	**5.117**
Pitch - ωy											
EKF-AHRS	3.504	11.324	5.581	12.277	43.982	37.042	33.413	5.708	1.565	1.144	3.701
QUEST	**1.802**	**2.521**	2.426	11.976	27.297	14.384	4.769	2.548	15.421	17.361	19.884
TRIAD	1.846	2.537	2.769	11.737	30.133	17.046	4.969	5.765	22.097	22.527	15.299
Madgwick	9.249	7.894	2.464	**5.129**	10.876	**2.1**	1.664	3.443	7.204	3.076	8.018
Mahony	10.036	12.137	3.556	17.05	14.254	2.712	9.756	1.481	**0.442**	0.358	3.233
NAG-AHRS	10.225	2.755	**2.355**	5.173	**8.271**	2.143	**1.662**	**0.667**	0.734	**0.346**	**2.545**
Yaw - ωz											
EKF-AHRS	3.618	36.22	19.696	8.878	85.492	92.493	91.335	4.658	3.864	4.737	27.486
QUEST	**2.054**	17.119	4.088	14.404	57.132	37.651	7.406	5.464	39.07	40.012	32.674
TRIAD	2.093	**17.08**	4.073	14.797	58.939	37.719	7.746	16.994	35.411	40.243	30.636
Madgwick	9.095	28.336	2.394	**5.091**	20.676	4.007	1.454	3.087	**0.812**	1.451	**2.147**
Mahony	9.705	34.361	**2.164**	16.649	26.071	**2.467**	11.352	2.862	2.264	2.53	5.242
NAG-AHRS	9.809	18.486	2.503	5.174	**16.22**	3.777	**1.453**	**1.327**	1.702	**1.187**	4.393

Table 3.8. The MAE result of 11 datasets and 6 algorithms. The bold text shows the smallest MAE. From the table it shows that the proposed algorithm yielded lowest error in more datasets than other algorithms.



Algorithms	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9	Dataset 10	Dataset 11
Roll - ωx											
EKF-AHRS	12.526	19.133	5.178	73.785	60.733	54.876	54.278	6.265	1.178	1.309	4.563
QUEST	1.234	**8.121**	2.221	**5.819**	19.828	8.425	3.478	5.795	6.794	7.848	10.841
TRIAD	1.335	8.153	2.203	6.344	23.046	9.384	3.775	4.195	9.506	11.01	17.128
Madgwick	1.234	11.669	1.799	64.723	8.013	2.223	**2.111**	0.79	6.935	5.114	20.81
Mahony	1.375	19.94	3.279	58.368	15.252	3.193	5.947	0.81	0.734	1.431	5.695
NAG-AHRS	**0.836**	8.989	**1.773**	64.626	**6.448**	**2.191**	**2.111**	**0.485**	**0.204**	**0.767**	**4.137**
Pitch - ωy											
EKF-AHRS	2.823	9.585	4.21	9.135	33.206	28.43	24.414	3.945	1.196	0.932	2.754
QUEST	1.297	**1.392**	1.964	7.093	17.265	8.728	3.209	0.859	8.752	8.956	8.924
TRIAD	**1.378**	1.451	2.191	6.962	20.04	11.235	3.494	3.148	12.633	12.517	6.695
Madgwick	7.987	6.734	2.007	**4.147**	7.731	**1.69**	1.415	2.491	6.252	2.343	6.258
Mahony	8.953	11.611	2.846	14.607	12.338	2.158	8.468	1.042	**0.364**	0.304	2.646
NAG-AHRS	9.174	1.572	**1.92**	4.15	**2.766**	1.723	**1.413**	**0.437**	0.65	**0.296**	**1.804**
Yaw - ωz											
EKF-AHRS	2.828	22.32	17.697	7.007	67.178	74.232	77.342	2.667	3.658	4.238	24.548
QUEST	**1.547**	8.128	3.213	8.507	36.151	23.469	5.119	3.102	17.125	26.378	17.018
TRIAD	1.592	**8.038**	3.2	8.89	38.124	23.392	5.444	4.89	15.825	25.282	15.939
Madgwick	7.659	10.272	2.015	3.948	8.204	3.091	0.987	2.074	**0.632**	1.195	**1.721**
Mahony	8.361	16.184	**1.836**	13.903	13.916	**1.828**	8.369	1.837	1.849	2.175	4.228
NAG-AHRS	8.495	9.025	2.101	**3.946**	**5.082**	2.854	**0.985**	**0.881**	1.44	**0.923**	3.723

Table 3.9. The achievement of smallest RMSE and MAE in all motion scenarios by six algorithms

Algorithm	Number of motion scenarios with smallest					
	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
EKF-AHRS	-	-	-	-	-	-
QUEST	1	2	1	2	1	1
TRIAD	1	-	1	-	1	1
Madgwick-AHRS	1	2	3	1	2	2
Mahony-AHRS	-	1	2	-	1	2
NAG-AHRS	8	6	4	8	6	5

3.7 Concluding Remarks

The experiment demonstrated that single-frame deterministic methods have a reasonably good performance in estimating orientation. However, since TRIAD and QUEST do not retain any past information in any form, the measurement noise/error level affects directly the estimation accuracy. The experiment demonstrated that in the presence of linear motion that disturbing only one sensor (accelerometer) the orientation estimate degraded significantly. TRIAD and QUEST are good for estimating the orientation in slow motion scenarios.

Despite Kalman filtering has been considered as standard method for inertial/magnetic orientation estimate, the experiment demonstrated the bad performance of EKF-AHRS given these motion scenarios. These results shows that the linearity and Gaussian assumptions were severely violated, hence large errors were observed. The complementary filter performed better in this experiment. Madgwick-AHRS performed better than Mahony-AHRS since Mahony-AHRS only fuses accelerometer and gyroscope and ignore magnetometer.

However, the proposed NAG-AHRS outperformed all of the benchmarking algorithms. NAG-AHRS was built on optimisation framework and does not require any assumption related with the linearity property of motion and observation model. Therefore, any motion scenarios basically do not violate any assumptions. This is validated by this experiment as the algorithm achieved the smallest errors in many motion scenarios. The NAG-AHRS is then potential to be combined to visual-only pose estimation with aims to overcome their constraint as presented in the next chapter.

Chapter 4

Hybrid Visual-Inertial/Magnetic 3D Pose Estimation

4.1 Background

In the previous chapter, a novel method for determining attitude and heading reference system (AHRS) is proposed to achieve a better handling on the noise and the model nonlinearities. The method addressed AHRS as a pure optimisation problem by deriving a system of non-linear equations from the accelerometer, magnetometer and gyroscope. The developed optimisation problem is then solved by using Nesterov's Accelerated Gradient descent (NAG). The proposed algorithm NAG-AHRS, has comprehensively validated using a motion capture system as the ground truth, and has benchmarked to other well-known state-of-the-art AHRS algorithms (Extended Kalman Filter, QUEST, TRIAD, Mahony-AHRS and Madgwick-AHRS). The conducted experiments demonstrated competitive results to these widely-known AHRS algorithms, hence the proposed approach offered an alternative method for solving the attitude estimation problem.

This chapter proposes an extension of the developed method NAG-AHRS to solve one of the fundamental problems in region-based vision-only 3D pose estimation. The fundamental problem being addressed is the multimodal projection problem. Region-based vision-only 3D pose estimation algorithms such as PWP3D (Prisacariu & Reid, 2012) estimate the object's pose based on the shape of the segmented region (its silhouette). This becomes problematic for tracking objects that have some level of symmetry since these kind of objects have a similar projection shape for multiple different poses. For instance, any ball-shaped objects (e.g. overhead water/oil storage tank) have the same projection shape regardless of orientation. In this case, vision-only region-based algorithms clearly cannot recover the orientation. Another example is cylinder-like shapes (e.g. chimney, tower, pole, soft-drink can). While different orientations related to roll or pitch angle can have a different projection silhouette, for any yaw-angle orientation the projections are the same. This chapter proposes an extension of the state-of-the-art region-based 3D pose estimation (PWP3D) by combining visual and inertial information to overcome the multimodal projection problem.

This chapter is organized as follows. A brief study of related works is presented in Section 4.2 and then to highlight the problem that will be addressed, a problem definition is presented in Section 4.3. As the proposed method can be seen as an extension of PWP3D algorithm, Section 4.4 covers this basic algorithm in detail. After the base algorithms are presented, the proposed method is presented in Section 4.5. The performance of the proposed method is then analysed and discussed in Section 4.6 and Section 4.7 provides a summary of the chapter.

4.2 Literature Review

An important task in machine vision is three-dimensional pose estimation (Prisacariu & Reid, 2012). The goal of 3D pose estimation is to precisely estimate position and orientation of object(s), relatively to a camera. Knowing the pose of objects has numerous practical purposes: such as autonomous inspection (Sa, Hrabar, & Corke, 2015), automatic object grasping and manipulation (Yang & Cao, 2012), automatic docking, visual servoing and autonomous tracking (Kelsey, Byrne, Cosgrove, Seereeram, & Mehra, 2006).

Considering the importance of 3D pose estimation, many algorithms have been developed to estimate the object's pose such as (Wei Fang, Zheng, Deng, & Zhang, 2017; Koller, Daniilidis, & Nagel ', 1993; Lebeda, Matas, & Bowden, 2012; Seo & Wuest, 2016; Vacchetti, Lepetit, & Fua, 2004; Worrall, Marslin, Sullivan, & Baker, 1991; Wuthrich, Pastor, Kalakrishnan, Bohg, & Schaal, 2013). Some of these algorithms do not require a 3D CAD model (Lebeda et al., 2012; Seo & Wuest, 2016) and some others require a CAD model (Koller et al., 1993; Pupilli & Calway, 2006; Tjaden, Schwanecke, & Schömer, 2017; Worrall et al., 1991; Wuthrich et al., 2013). The option to employ a priori known CAD model into the algorithms improves the accuracy as shown in (Koller et al., 1993; Pupilli & Calway, 2006; Tjaden et al., 2017). The requirement to provide 3D model is also highly practical, as in many applications the objects that will be tracked are known a priori. For instance, in industrial inspection, the object that will be inspected is known beforehand or the products being produced are known in advance.

Many existing state-of-the-art model-base tracking algorithms were developed based on edge information such as the algorithms proposed by (A. J. Bray, 1990; Armstrong & Zisserman, 2000; Drummond & Cipolla, 2002; Harris & Stennett, 1990; Klein & Murray, 2006; Koller et al., 1993; Pupilli & Calway, 2006). Edge-based trackers can estimate object's pose but due to edges are primitive features, it is hard to find their correct correspondences. Relying only on edge information leads to wrong correspondences and the solution traps on local optima.

To improve the estimation accuracy, some methods were developed by combining edge information with other features such as texture information (Vacchetti et al., 2004) or background geometry (Seo, Park, Park, & Park, 2013). Combining edge information with other information demonstrated to yield a better tracking, however, it still suffers in the presence of blurry images. In blurry images, edges cannot be located precisely or cannot be detected. Image feature quality also degrades significantly as a consequences of blurry textures.

To address this problem, Prisacariu (Prisacariu & Reid, 2012) proposed a model-based 3D tracker known as Pixel-Wise Posterior 3D Pose estimation (PWP3D) by maximizing the pixel-wise color posterior probability. This state-of-the-art tracker works by building a colour histogram model of the object and its background, then maximizing the segmentation posterior probability between these two regions by adjusting the object pose. This approach can then be considered as simultaneous segmentation and 3D pose estimation. PWP3D demonstrated it can handle blurry images in real time as it does not depend on the edges. However, as PWP3D algorithm basically maximizes the segmentation posterior probability based on the projection silhouettes, it suffers from the multimodal projection problem. Different poses that have the same silhouettes lead to multiple solutions.

Another weakness of PWP3D that has been demonstrated by (Tjaden, Schwanecke, & Schömer, 2016) is it easily loses track in tracking object with large variation in appearance. The large variation of appearance usually occurs when the camera comes too close to the object then moves away too far from the object. The large appearance variation can also occur when the camera moves sideways too fast resulting in large projection displacement between two consecutive frames. Depending on the object size and the initial position, a small position difference can have a large appearance difference. Hence PWP3D needs some improvement in tracking object with wider dynamic of camera motion.

The presented method proposes to overcome the fundamental PWP3D limitation (multimodal projection problem) and improve its performance in tracking objects by two means: first, incorporate camera orientation estimate from inertial/magnetic sensor (attached to the camera) to avoid multiple solutions; and secondly, improve the optimization method from classic gradient descent into a Nesterov's Accelerated Gradient descent (Nesterov, 1983) that has been proved has a better performance (Botev, Lever, & Barber, 2017; Sutskever, Martens, Dahl, & Hinton, 2013; Timothy Dozat, 2015). The proposed method is applicable for tracking a static object by a moving the camera that is generally required for autonomous engineering inspection, autonomous docking, etc.

The proposed method combines the inertial orientation and visual pose estimate into a single system of non-linear equations that is solved as a single, pure optimization problem. In contrast with other fusion methods that usually implement the Kalman filter for visual-inertial tracking (W Fang, Zheng, & Deng, 2016; Jiang & Yin, 2017; Ligorio & Sabatini, 2013; Sirtkaya, Seymen, & Alatan, 2013; Tian, Li, Li, & Cheng, 2017) and need a significant system redesign, the proposed approach only requires to add some additional constraints to the system.

Therefore, the key contributions of this chapter are:

- Improve the vision-only 3D pose estimator (PWP3D) into hybrid visual-inertial 3D pose estimation (PWP3Di) to deal with the multimodal projection problem.
- Improve the PWP3D to be able to handle larger appearance difference with the implementation of a better optimisation algorithm. The proposed method

implements Nesterov Accelerated Gradient (NAG) descent that has been proved has a better performance than classical Gradient Descent.

4.3 Problem Definition

Given a query image Q, an object's statistical appearance model M_f, a background's statistical model M_b and the CAD model of an object of interest M, the objective is to find a closed-curve boundary, as a function of object's pose $\lambda = \left(t_x, t_y, t_z, q_w, q_x, q_y, q_z\right)$, that best segment object and its background in the query image. The closed-curve boundary is expressed implicitly as zero-level-set of signed distance function ϕ.

The best segmentation is computed by minimising log of the posterior probability of the boundary given the image query $P(\Phi|\, Q)$. The problem then can be defined on how to develop an energy function that serves this purpose and how to express it as an optimisation problem. Since the minimization is proposed using NAG, therefore the minimisation problem should be expressed in the general form of NAG:

$$v_{t+1} = \alpha.v_t - \mu.\nabla f(\lambda_t + \alpha.v_t)$$

$$\lambda_{t+1} = \lambda_t + v_{t+1}$$

Moreover, since the method proposes to develop hybrid visual-inertial/magnetic pose estimation, another main problem is how to combine the system of non-linear equations developed in NAG-AHRS into the visual optimisation problem that then will be solved simultaneously.

4.4 Underlying Theory: Pixel-Wise 3D Pose Estimation (PWP3D)

The proposed method can be seen as an extension of the existing state-of-the-art algorithm PWP3D by combining this vision-only estimate with inertial and magnetic orientation estimate. The inertial/magnetic orientation estimate in this chapter is the NAG-AHRS algorithm that was proposed in Chapter 3. The visual estimation is built on PWP3D algorithm, therefore, knowing the detail of PWP3D algorithm is required before improving it further. This section addresses the detail of the PWP3D algorithm.

PWP3D is a 3D pose tracker based on color histogram. PWP3D was developed based on an energy function introduced by (Bibby & Reid, 2008). By assuming pixel-wise independence, the probability of an embedded function Φ given an observation query image Q is given by:

$$P(\Phi|\, Q) = \prod_{x\,\in\,\Omega}\big(H_e(\Phi)P_f + \big(1 - H_e(\Phi)\big)P_b\big) \tag{4.1}$$

where Φ is an embedding function which implicitly specifies the contour and $H_e(\Phi)$ is the Heaviside of this embedding function. The contour C is defined where $\Phi = 0$ or notated as $C = \{(x, y) \in \mathbb{R}^2 \mid \Phi(x, y) = 0\}$. The embedding function Φ is in the form of a signed-distance function where the area inside the contour has negative distance and the area beyond the contour has positive values as expressed by

$$\Phi(x) = -d(x), \forall x \in \Omega_f$$

$$\Phi(x) = d(x), \forall x \in \Omega_b$$

Maximizing this posterior probability function (4.1) can be done by minimizing its negative log and this leads to the energy function

$$E(\Phi) = -\sum_{x\in\Omega} \log\big(H_e(\Phi)P_f + \big(1 - H_e(\Phi)\big)P_b\big)$$

For solving the segmentation problem and pose tracking simultaneously, this energy function is then differentiated with respect to the pose parameters $\lambda_i = \big(t_x, t_y, t_z, q_w, q_x, q_y, q_z\big)$ yielding

$$\frac{\partial E}{\partial \lambda_i} = -\sum_{x \in \Omega} \frac{P_f - P_b}{H_e(\Phi)P_f + \big(1 - H_e(\Phi)\big)P_b} \delta_e(\Phi) \begin{bmatrix} \frac{\partial \Phi}{\partial x} & \frac{\partial \Phi}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \lambda_i} \\ \frac{\partial x}{\partial \lambda_i} \end{bmatrix} \tag{4.2}$$

where:

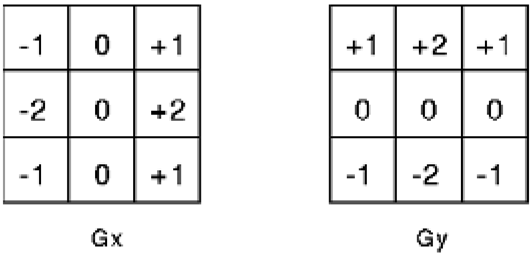
$$P_f = \frac{P(y|M_f)}{\eta_f P(y|M_f) + \eta_b P(y|M_b)}$$

$$P_b = \frac{P(y|M_b)}{\eta_f P(y|M_f) + \eta_b P(y|M_b)}$$

$$\eta_f = \sum_{x \in \Omega} H_e\big(\Phi(\mathrm{x})\big)$$

$$\eta_b = \sum_{x \in \Omega} \Big(1 - H_e\big(\Phi(\mathrm{x})\big)\Big)$$

The embedded function Φ specifies the contour or object's outer boundaries given a certain pose. In implementation, given a particular pose estimate, the 3D model is rendered and projected to obtain its 2D silhouette. The contour of the silhouette is then extracted and performs a signed distance function to the contour to achieve the Φ. The derivative of this function $\begin{bmatrix} \frac{\partial \Phi}{\partial x} & \frac{\partial \Phi}{\partial y} \end{bmatrix}$ which is also required by Equation 4.2 is obtained by performing a convolution in the x and y directions with a derivation-approximation kernel given by Sobel operator (Sobel, 2014).



-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

The relation between a 2D point in image plane (x, y) to the corresponding 3D point (X, Y, Z) is:

$$\begin{bmatrix} x \\ y \end{bmatrix}^T = \begin{bmatrix} f_u \dfrac{X}{Z} + u_0 \\ \dfrac{Y}{Z} + v_0 \end{bmatrix}^T$$

where f_u and f_v are the horizontal and vertical focal length, u_0 and v_0 are the horizontal and vertical focal point / centre. The derivative of the 2D pixel location with respect to the 3D pose parameters can be calculated as:

$$\frac{\partial x}{\partial \lambda_i} = f_u \frac{1}{Z^2}\left(Z\frac{\partial X}{\partial \lambda_i} - X\frac{\partial Z}{\partial \lambda_i}\right)$$

$$\frac{\partial y}{\partial \lambda_i} = f_v \frac{1}{Z^2}\left(Z\frac{\partial Y}{\partial \lambda_i} - Y\frac{\partial Z}{\partial \lambda_i}\right)$$

The 3D point in camera reference system has a corresponding 3D point in object coordinate system according to the relation:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = R \begin{bmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix}$$

where

$$R = \begin{bmatrix} 1 - 2q_y^2 - 2q_z^2 & 2q_xq_y - 2q_zq_w & 2q_xq_z + 2q_yq_w & 0 \\ 2q_xq_y + 2q_zq_w & 1 - 2q_x^2 - 2q_z^2 & 2q_yq_z - 2q_xq_w & 0 \\ 2q_xq_z - 2q_yq_w & 2q_yq_z + 2q_xq_w & 1 - 2q_x^2 - 2q_y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally, the partial derivative of the 3D points in camera reference system to the pose parameters are shown in Table 4.1.

Table 4.1. Partial derivative of 3D point in camera reference system to the pose parameters

λ_i	$\frac{\partial X}{\partial \lambda_i}$	$\frac{\partial Y}{\partial \lambda_i}$	$\frac{\partial Z}{\partial \lambda_i}$
t_x	1	0	0
t_y	0	1	0
t_z	0	0	1
q_w	$-2q_zY_O + 2q_yZ_O$	$2q_zX_O - 2q_xZ_O$	$-2q_yX_O + 2q_xY_O$
q_x	$2q_yY_O + 2q_zZ_O$	$2q_yX_O - 4q_xY_O - 2q_wZ_O$	$2q_zX_O + 2q_wY_O - 4q_xZ_O$
q_y	$-4q_yX_O + 2q_xY_O + 2q_wZ_O$	$2q_xX_O + 2q_zZ_O$	$-2q_wX_O + 2q_zY_O - 4q_yZ_O$
q_z	$-4q_zX_O - 2q_wY_O + 2q_xZ_O$	$2q_wX_O - 4q_zY_O + 2q_yZ_O$	$2q_xX_O + 2q_yY_O$

The PWP3D method then can be summarized as the block diagram in Figure 4.1.

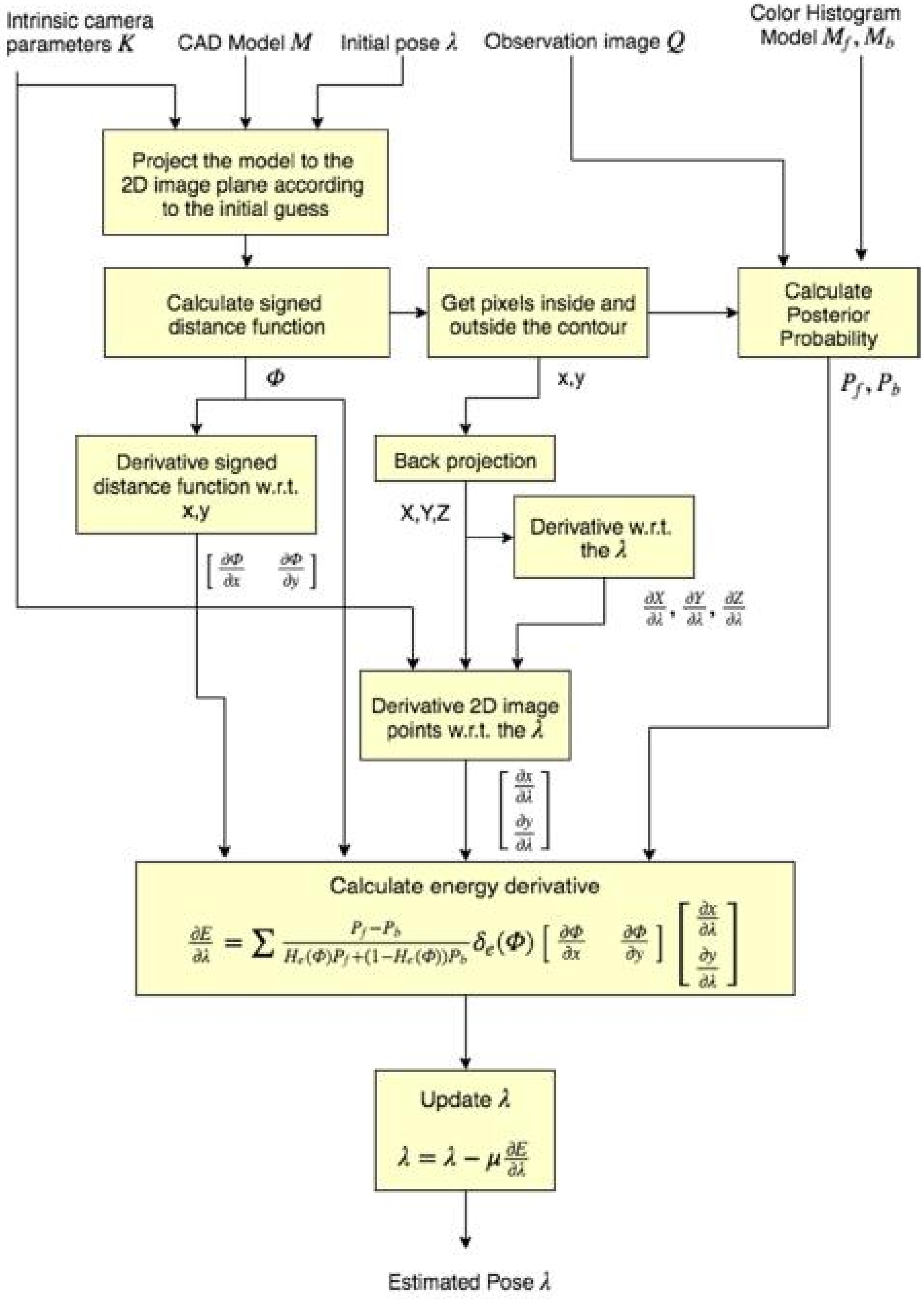

Intrinsic camera parameters K
CAD Model M
Initial pose λ
Observation image Q
Color Histogram Model M_f, M_b
Project the model to the 2D image plane according to the initial guess
Calculate signed distance function
Get pixels inside and outside the contour
Calculate Posterior Probability
Φ
x,y
P_f, P_b
Derivative signed distance function w.r.t. x,y
Back projection
X,Y,Z
Derivative w.r.t. the λ
$\left[\frac{\partial\Phi}{\partial x} \quad \frac{\partial\Phi}{\partial y}\right]$
$\frac{\partial X}{\partial\lambda}, \frac{\partial Y}{\partial\lambda}, \frac{\partial Z}{\partial\lambda}$
Derivative 2D image points w.r.t. the λ
$\begin{bmatrix}\frac{\partial x}{\partial\lambda} \\ \frac{\partial y}{\partial\lambda}\end{bmatrix}$
Calculate energy derivative
$\frac{\partial E}{\partial\lambda} = \sum \frac{P_f - P_b}{H_e(\Phi)P_f + (1-H_e(\Phi))P_b}\delta_e(\Phi)\left[\frac{\partial\Phi}{\partial x} \quad \frac{\partial\Phi}{\partial y}\right]\begin{bmatrix}\frac{\partial x}{\partial\lambda} \\ \frac{\partial y}{\partial\lambda}\end{bmatrix}$
Update λ
$\lambda = \lambda - \mu\frac{\partial E}{\partial\lambda}$
Estimated Pose λ

Figure 4.1. Model-based simultaneous segmentation and pose estimation by maximizing posterior probability that segments foreground and background regions based on their color distribution model.

4.5 Proposed Method: Pixel-Wise Posterior and Inertial 3D Pose Estimation (PWP3Di-NAG)

The proposed method aims to improve the PWP3D algorithm by combining this visual tracking method with an inertial orientation estimate. As the proposed approach addresses the visual-inertial integration as a single optimization problem, the first step is expressing the PWP3D approach in the general form of Gradient Descent. Recall the original PWP3D expressed in general form of Gradient Descent:

$$\lambda_{new} = \lambda_{old} - \mu . J^T f \tag{4.3}$$

where the estimated pose λ_{new} is iteratively improved from the previous pose λ_{old}, refined by the product of the Jacobian function J and the residual function f, scaled by a factor as specified by the step size μ. The original PWP3D method assumes each pixel has an equal importance, therefore, f is a column matrix with all elements equal to 1.

Given the PWP3D expressed in general form of Gradient Descent, adding inertial/magnetic constraint into the system is achieved by adding this extra constraint into the Jacobian and residual functions. The residual functions of hybrid visual-inertial 3D pose estimation then become:

$$\boldsymbol{f}\left({}^S_E\hat{\boldsymbol{q}},\ {}^S\hat{\boldsymbol{a}},\ {}^E\hat{\boldsymbol{b}},\ {}^S\hat{\boldsymbol{m}}\right) = \begin{bmatrix} \boldsymbol{f}_{accelero}\left({}^S_E\hat{\boldsymbol{q}},\ {}^S\hat{\boldsymbol{a}}\right) \\ \boldsymbol{f}_{magneto}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}},\ {}^S\hat{\boldsymbol{m}}\right) \\ \boldsymbol{f}_{gyro}\left({}^S_E\hat{q}_t, {}^S_E q_{t-1},\ {}^S\omega_t\right) \\ \boldsymbol{f}_{PWP3D}(1)_{\Omega\,\times 1} \end{bmatrix} \tag{4.4}$$

and the Jacobian is

$$\boldsymbol{J}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}}\right) = \begin{bmatrix} \boldsymbol{J}_{accelero}\left({}^S_E\hat{\boldsymbol{q}}\right) \\ \boldsymbol{J}_{magneto}\left({}^S_E\hat{\boldsymbol{q}},\ {}^E\hat{\boldsymbol{b}}\right) \\ \boldsymbol{J}_{gyro}\left({}^S_E\hat{\boldsymbol{q}}_t, {}^S_E\hat{\boldsymbol{q}}_{t-1},\ {}^S\boldsymbol{\omega}_t\right) \\ \boldsymbol{J}_{PWP3D}\ \left({}^S_E\hat{\boldsymbol{q}}, \lambda_{t-1}\right) \end{bmatrix} \tag{4.5}$$

The optimal solution for this system of non-linear equation is the solution provided by both visual and inertial/magnetic constraints.

To facilitate an adjustment on how significant the visual part and the inertial/magnetic part affects or contributes to the final solution, a weighting scheme is used. This weighting scheme provides a free to adjust parameter that can be adjusted depending on the accuracy of the visual part and the inertial/magnetic part. For instance, when the object being tracked has a very clear, distinguishable appearance with its background, the visual-based estimation can be very accurate so the weight in the visual part can be tuned to a high value. In contrast, when the object appearance model and the background appearance model are less discriminative, the weight of the visual part should be lower. This adjustment should also consider to the accuracy of the inertial/magnetic sensor.

Due to the weighting scheme, the weighting matrix is introduced and the refining become:

$$\lambda_{new} = \lambda_{old} - \mu.J^T.W.f$$

where W is a diagonal matrix consisting weight for each constraint. Addressing this by using Nesterov Accelerated Gradient descent (Nesterov, 1983) yields

$$v_t = \alpha.v_{t-1} - \mu.J(\lambda_{t-1} + \alpha.v_{t-1})^T.W.f(\lambda_{t-1} + \alpha.v_{t-1}) \qquad (4.6)$$

$$\lambda_t = \lambda_{t-1} + v_t \qquad (4.7)$$

where v_t is the recent correction factor that is computed from the previous correction factor v_{t-1} by a scale specified by the momentum parameter α and the recent update $\mu.J(\lambda_{t-1} + \alpha.v_{t-1})^T.W.f(\lambda_{t-1} + \alpha.v_{t-1})$.

Following the original PWP3D algorithm, the update rate μ is selected manually. The weighting matrix W which specifies the relative weight for the visual and inertial parts also can be chosen manually. Figure 4.2 shows the PWP3Di-NAG in a block diagram.

Integrating visual and inertial sensors as a single optimisation problem requires the inertial/magnetic estimate to also be expressed in the same reference system, hence, before the tracking is conducted, the raw inertial and magnetic observation must be transformed into the object reference system. This operation requires to know the initial transformation between inertial/magnetic frames to object reference frame, hence it induces an extra step for initialisation. Once the initialisation is done, and the initial transformation is known, the tracking can then be carried out.

This chapter is focussed on validating the tracking performance, this is done by assuming that the inertial/magnetic reading has been transformed / normalised to the object reference frame. Finding the initial transformation between inertial/magnetic coordinate system to object reference will be discussed in detail in Chapter 5.

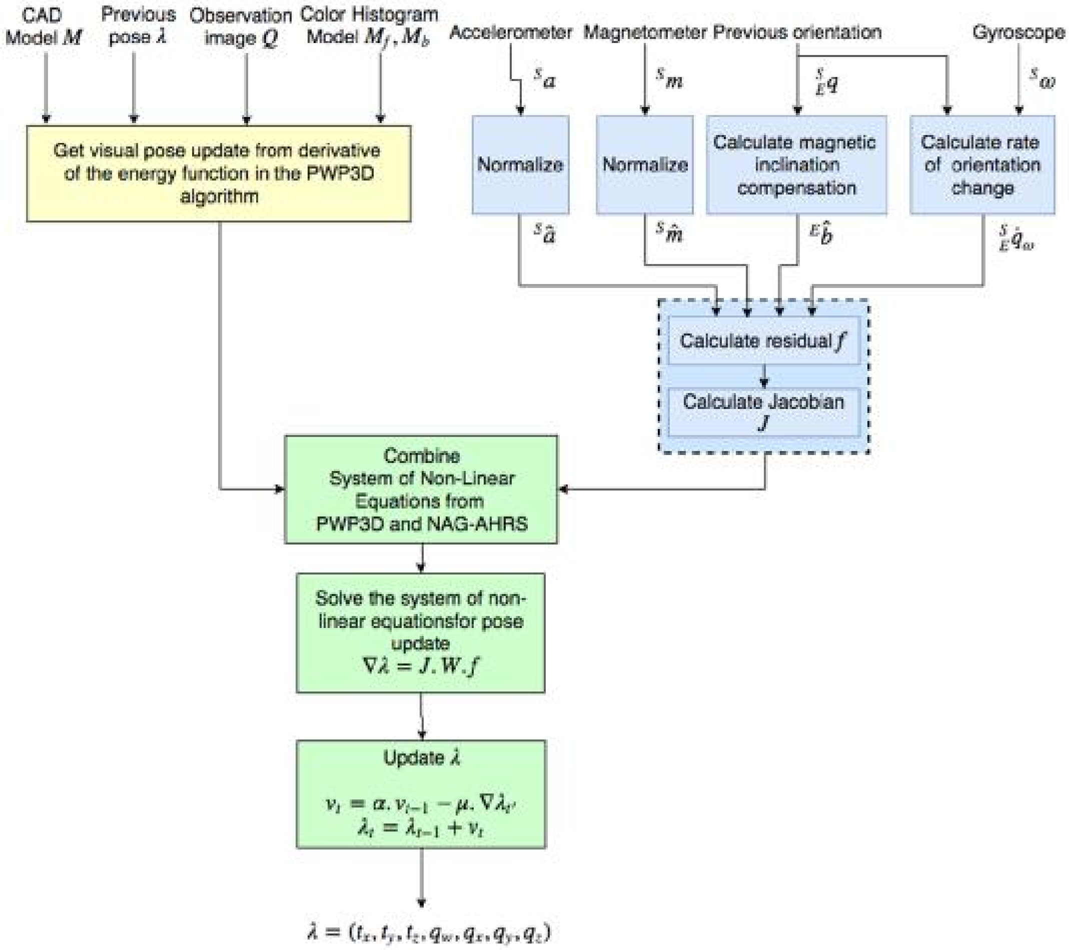

CAD Model M
Previous pose λ
Observation image Q
Color Histogram Model M_f, M_b
Accelerometer
Magnetometer
Previous orientation
Gyroscope
${}^{S}a$
${}^{S}m$
${}^{S}_{E}q$
${}^{S}\omega$
Get visual pose update from derivative of the energy function in the PWP3D algorithm
Normalize
Normalize
Calculate magnetic inclination compensation
Calculate rate of orientation change
${}^{S}\hat{a}$
${}^{S}\hat{m}$
${}^{E}\hat{b}$
${}^{S}_{E}\dot{q}_{\omega}$
Calculate residual f
Calculate Jacobian J
Combine System of Non-Linear Equations from PWP3D and NAG-AHRS
Solve the system of non-linear equationsfor pose update
$\nabla\lambda = J.W.f$
Update λ
$v_t = \alpha . v_{t-1} - \mu . \nabla\lambda_{t'}$
$\lambda_t = \lambda_{t-1} + v_t$
$\lambda = (t_x, t_y, t_z, q_w, q_x, q_y, q_z)$

Figure 4.2. The block diagram of the PWP3Di-NAG algorithm that combines visual pose estimate with inertial orientation estimate within a single optimization framework. The yellow blocks are the visual pose estimate part and the detailed explanation can be found in Section 3.2. The inertial/magnetic orientation estimate refers to Section 4.4. Combination of both methods as weighted least square problem is then solved using Nesterov's Accelerated Gradient.

4.6 Experiments and Results

This section covers the evaluation of the proposed method. The experiments were designed to investigate the performance of the method in dealing with multimodal projection problem as well as in tracking objects with a large appearance difference. The setup and the dataset is presented in the Section 4.6.1.

4.6.1 Experiment Setup and Datasets

The experiments were done using a computer with Intel Core i5-4590 CPU @3.30GHz, 4G RAM and NVIDIA Quadro K620 graphic card. The implementation was done using C++ run in Ubuntu 16.04LTS operating system. Data was recorded synchronously in rosbag format using ROS Indigo version. As this chapter is focussed on tracking performance, the initialisation stage is not discussed. The inertial/magnetic reading has been normalised to the object reference frame.

The proposed method PWP3Di-NAG is benchmarked to the original PWP3D algorithm as well as compared to the pose output obtained from a Aruco marker. Aruco was selected as it has been widely used as the benchmark for pose estimation and it has a good precision (Garrido-Jurado, Muñoz-Salinas, Madrid-Cuevas, & Marín-Jiménez, 2014).

The proposed algorithm was evaluated in various scenarios. In terms of the motion characteristics, two categories of datasets are created:

- Static datasets
- Dynamic datasets

Static Datasets

The datasets are recorded while both the object and the camera were in a fixed pose. In this case, the output of pose estimation should remain the same, however due to illumination changes (i.e. different lighting scheme or the presence of shadows), the pose output could suffer from instabilities. For symmetric objects that suffer from multimodal projection problem, a drift might also occur. Therefore, the purpose of these datasets is to investigate:

- Convergence rate and the capability in reaching local optima with small errors

Given any initial pose, the algorithm requires some iterations to reach the solution. With a static object and a static camera pose, the convergence rate can be easily investigated and compared.

When the initial pose is far from the global optima, the final estimate can be trapped and converge to local optima. These static datasets are also designed to check the performance of algorithms in dealing with this problem.

- The influence of noise

  Since the object being tracked and the camera are in a static position, the output of the pose estimate should not experience any changes. However, due to the illumination changes during the recording, the pose estimation that is computed based on color histogram theoretically can be affected. The performance of the pose estimation with the presence of illumination changes is investigated using the static datasets. The robust estimation algorithm should be able to cope with the noise.

- The multimodal projection problem

  A static scene should result in a static pose estimate, however due to the multimodal projection problem, a pose estimate of symmetrical object on a static scene theoretically can change. Therefore this static dataset also serves for investigating this problem.

Dynamic dataset

This dataset are created to evaluate the performance of the algorithm during tracking of an object. The evaluation included how the performance of the algorithm deals with large appearance different (i.e. the camera moves too close or too far from the object in a single dataset) and speed changes (i.e. the camera does not move at all in the beginning, then starts to moves slowly and then moves quickly around the object).

Object Being Tracked

As the shape of the object being tracked has some impact on the tracking quality (i.e. the level of symmetry, the color properties) the experiments are conducted using two different objects:

- Box

  The box object is selected to represent an object that has a low level of symmetry. A box has symmetry in $x - y$, $x - z$ and $y - z$ plane, therefore the projection is similar in four different poses: front-face, back-face, upside-down front-face and upside-down back-face. However, the orientation distance between optima are large and the optimisers should be able to deal with this condition.
- Soft-drink can

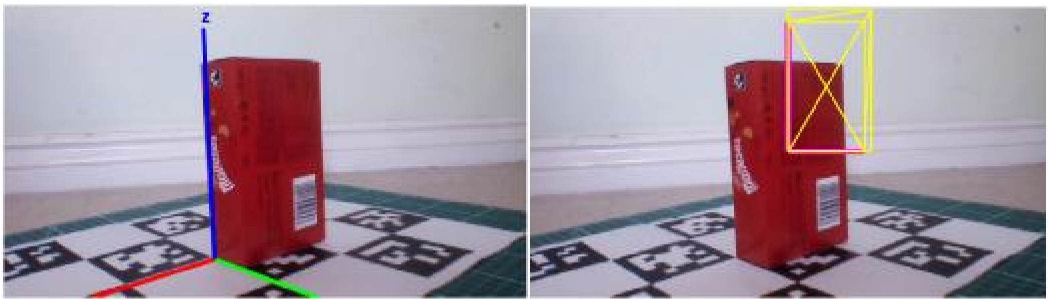
  The soft-drink can object is selected to represent an object that is symmetrical in one axis (z axis). Despite it only being symmetric in one axis, the silhouette of the soft-drink can in any yaw direction will be the same. It will lead to infinite solutions as one axis cannot be retrieved. This property makes this object very good for investigating the multimodal projection problem.

4.6.2 Experimental Results

The first experiment investigated the performance of the algorithms when the initial position was far from the correct pose. The first dataset being explored was static red-box dataset.

Red Box Dataset

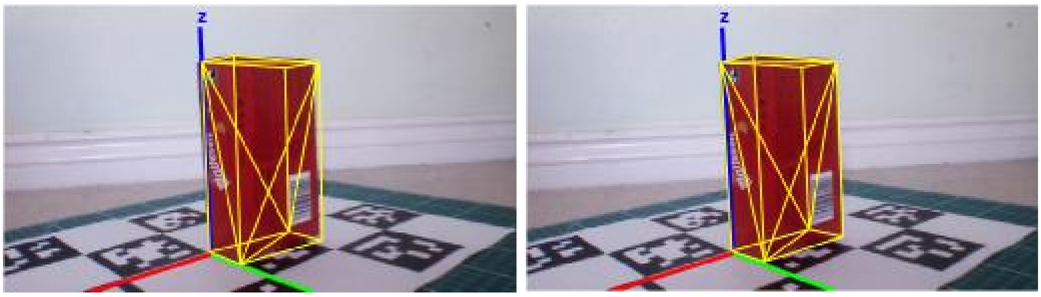
The experiment was done by giving the same initial pose for both algorithms: PWP3D and PWP3Di-NAG. The initial position setup was t_x=0, t_y=0 and t_z = 0.5m and the orientation setup was $q_w = 1, q_x = 0, q_y = 0$ and $q_z = 0$. The reference pose was given by Aruco output. Figure 4.3 shows the Aruco pose and the initial estimate of the object. Both of the algorithms then executed for all 450 static frames.



(a) (b)

Figure 4.3. The query image and the pose estimate from Aruco as the reference (a). The initial object position estimate (plotted as yellow coloured wireframe) was far from the correct pose as shown in (b)

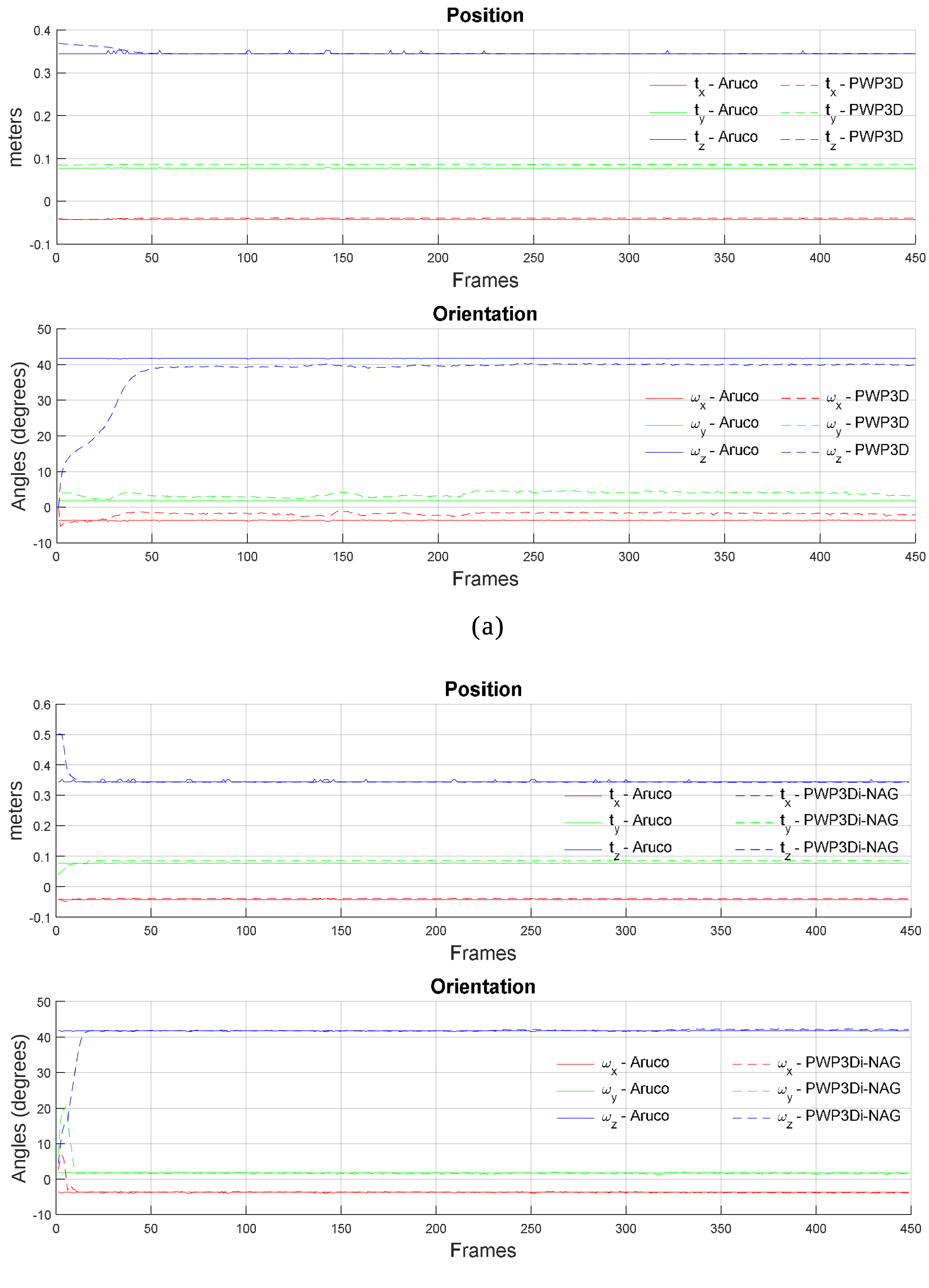
The pose estimate of both algorithms at frame 100 were good as can be seen in Figure 4.4.



(a) (b)

Figure 4.4. Output at frame 100 of PWP3D (a) and PWP3Di-NAG (b). The estimated pose (yellow coloured wireframe) shows that both results were good with only a small difference that cannot easily be observed visually.

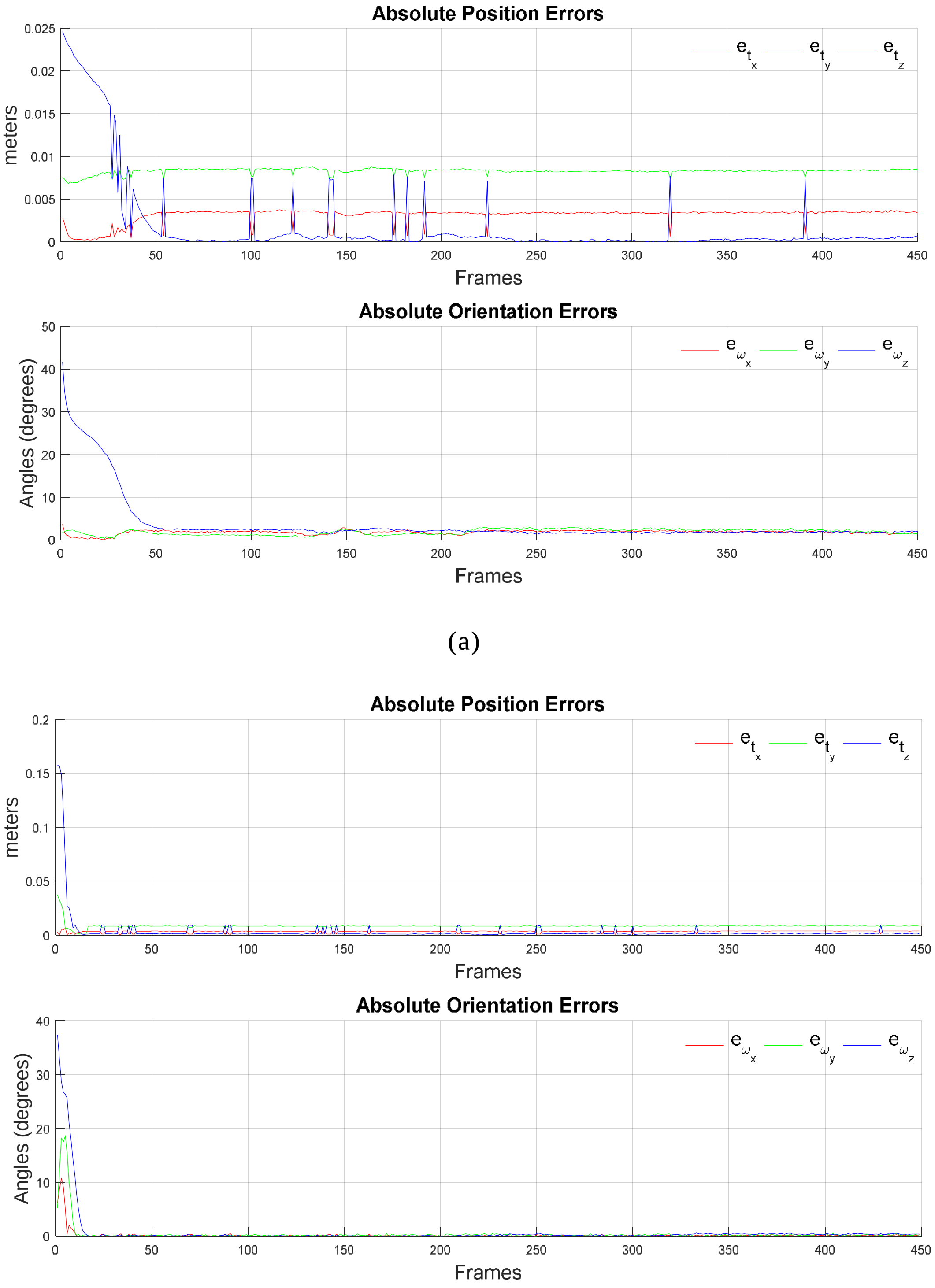
In frame 100 the PWP3Di-NAG had a slightly better accuracy than PWP3D but due to a small difference, it was hard to observe the difference visually. The better accuracy of PWP3Di-NAG compared to PWP3D can be better seen from plot of pose estimate that is presented in Figure 4.5 and the plot of absolute error can be seen in Figure 4.6.



(a)

(b)

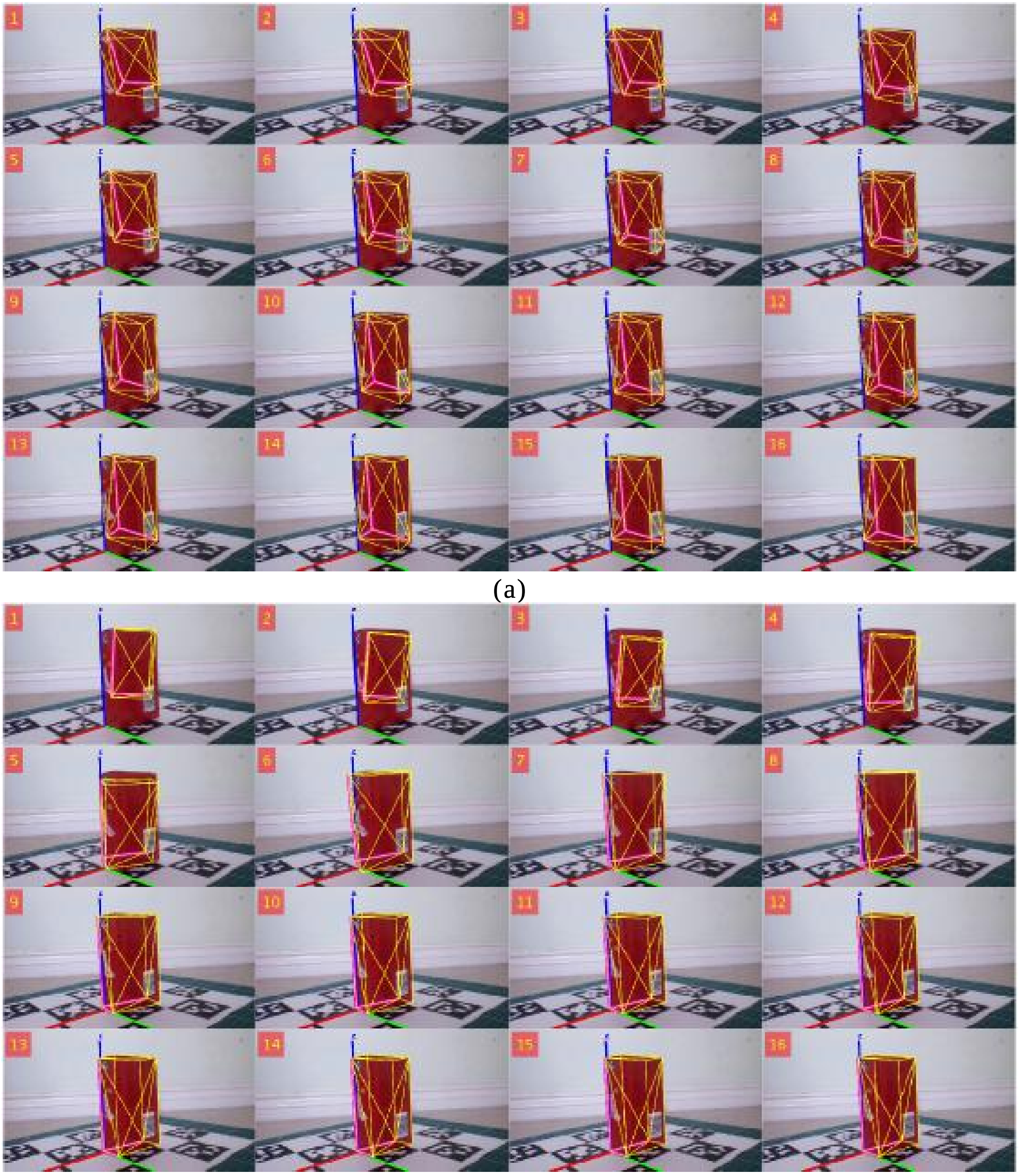
Figure 4.5. The pose estimation output of PWP3D (a) and PWP3Di-NAG (b). It shows that final position errors are similar for both algorithms. However, PWP3D converged slower than PWP3Di-NAG. PWP3D required around 170 iterations while PWP3Di-NAG only required less than 20 iterations. The orientation estimate of PWP3Di-NAG is slightly better than PWP3D and the convergence time is also much faster.



(a)

(b)

Figure 4.6. The absolute position and orientation error of box dataset. The PWP3D algorithm slowly refines the pose orientation and converged in about 170 frames (a), while the PWP3Di-NAG required less than 20 frames.



(a)

(b)

Figure 4.7. The film strip pose estimation output of PWP3D algorithm (a) and PWP3Di-NAG algorithm (b). In the picture (a) it shows the pose correction was very slow and the PWP3D was not converge during first 16 frames. A significantly better performance was observed from PWP3Di-NAG (b) that managed to converge within 16 first frames. Both experiments were done using the same initial pose and the same step size setting.

From the plot it shows a significant difference was observed in the first period of frames, where the algorithms evolve from the initial pose to the steady-state pose. However, after each of the algorithms reached the stable state the different errors between the two algorithms were insignificant. This slow convergence rate can be also observed from the film strip of the first 16 frames of the result as presented in Figure 4.7. Despite both algorithms implemented the same step size μ = 0.001 for refining the position and μ = 0.05 for refining the orientation, the film strip shows that the pose correction of the PWP3D was very slow. The visual-inertial weighting parameter γ setup was 0.5, this value was selected assuming visual and inertial have a similar accuracy. The proposed method that implemented a same setup had a significantly better convergence rate.

For another quantitative comparison, a Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of both algorithms are also presented. The MAE and RMSE computation are given by

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

where n is the number of data, and $y_j - \hat{y}_j$ is the error.

Table 4.2 and Table 4.3. provide this error measurement

Table 4.2. The RMSE and MAE Position output of PWP3D and PWP3Di-NAG algorithm in red-box dataset. The smallest errors are indicated using bold font.

	RMSE			MAE		
	t_x	t_y	t_z	t_x	t_y	t_z
PWP3D	0.0066	0.0090	0.0198	0.0045	0.0084	0.0066
PWP3Di-NAG	**0.0036**	**0.0083**	**0.0093**	**0.0034**	**0.0083**	**0.0026**

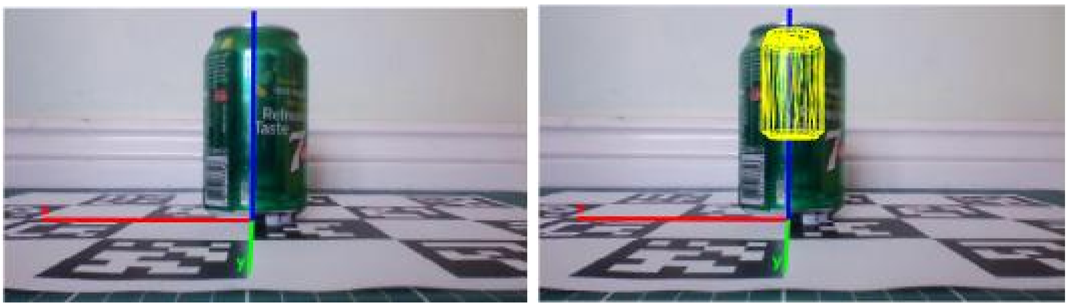
Table 4.3. The RMSE and MAE Orientation output of PWP3D and PWP3Di-NAG algorithm in red-box dataset. The smallest errors are indicated using bold font.

	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
PWP3D	1.9499	4.3611	21.26	1.8398	2.1564	8.6663
PWP3Di-NAG	**0.5576**	**1.1677**	**2.349**	**0.1601**	**0.2988**	**0.4203**

The RMSE and MAE measurements show the proposed PWP3Di-NAG achieved a better performance than PWP3D. PWP3D output was also good as it achieved a pose estimate with mean absolute position error less than 1 cm and the mean absolute orientation error less than 10 degrees. The good pose estimation quality was obtained due to some factors, i.e: the contrast color of the object to its background and the less symmetrical shape. These factors made both algorithms perform well.

Soft-drink Can Dataset

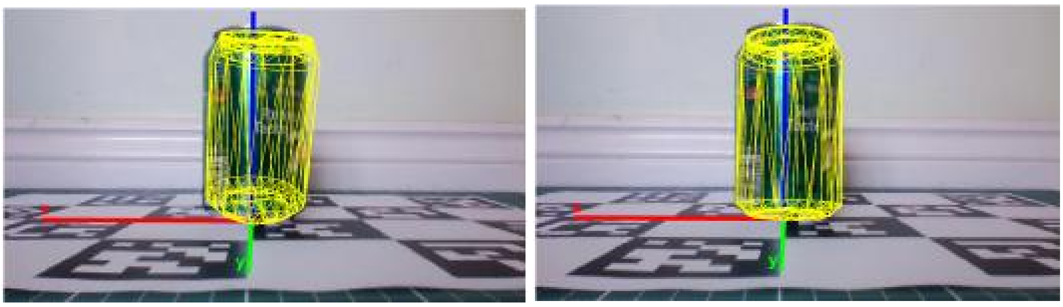
The next experiment was done using different dataset. The object being tracked was a soft-drink can that is symmetrical in the z axis. The query image input along with the Aruco pose is presented in Figure 4.8(a) and the projection of initial pose is presented in Figure 4.8(b).



(a) (b)

Figure 4.8. The query image and the pose estimate from Aruco as the reference (a). The initial object position estimate (yellow-coloured wireframe) was far from the correct pose as shown in (b)

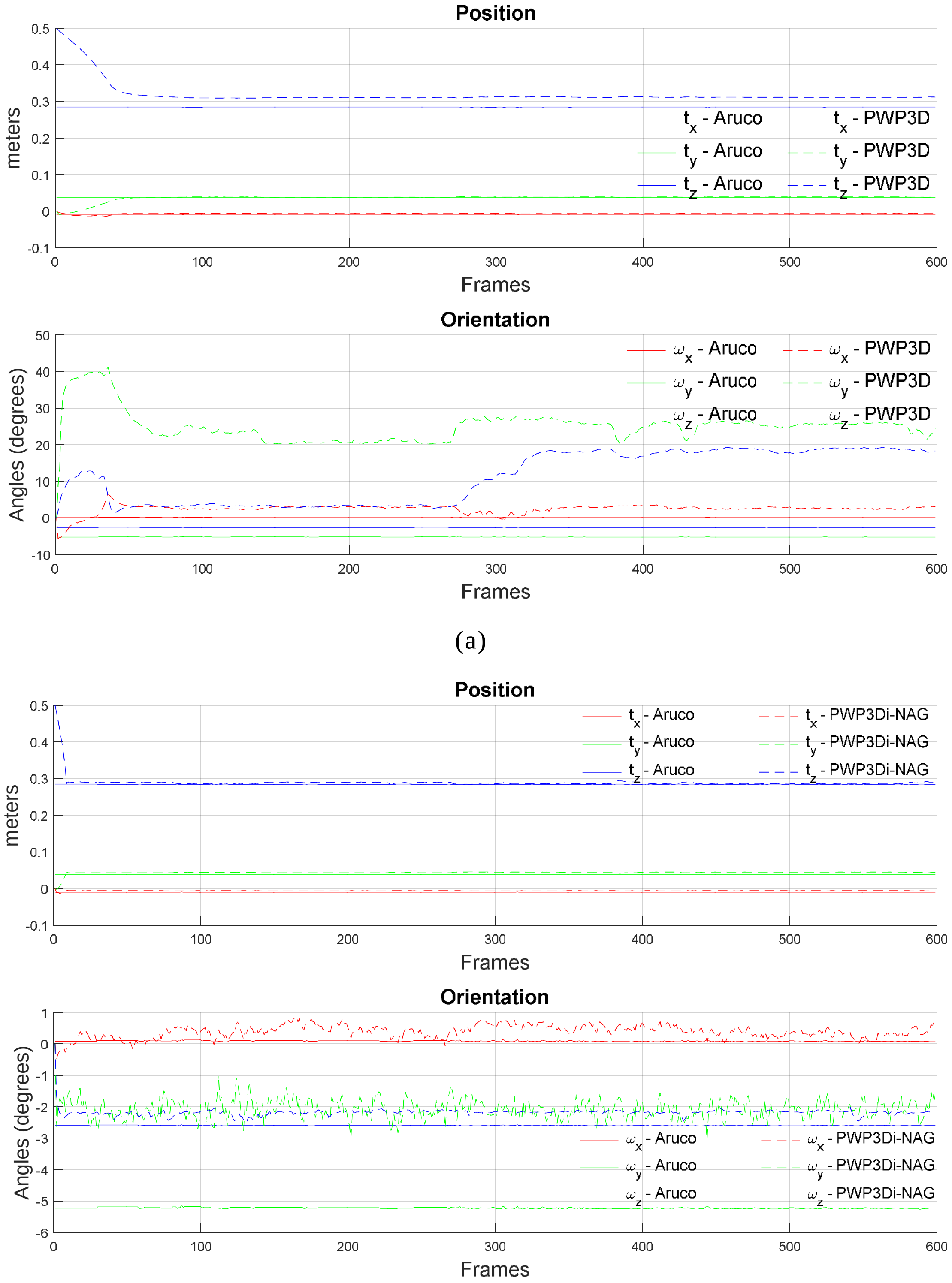
The initial position setup was t_x=0, t_y=0, t_z = 0.5m and the initial orientation was q_w = 1, q_x = 0 , q_y = 0, q_z = 0. After a few frames both algorithms converged to their stable state. The output from frame 200 is presented in Figure 4.9.



(a) (b)

Figure 4.9. Pose estimate output (yellow-coloured wireframe) at frame 200 of PWP3D (a) and PWP3Di-NAG (b). It shows the PWP3D converged to an optima that was less accurate than the estimated pose obtained by PWP3Di-NAG.

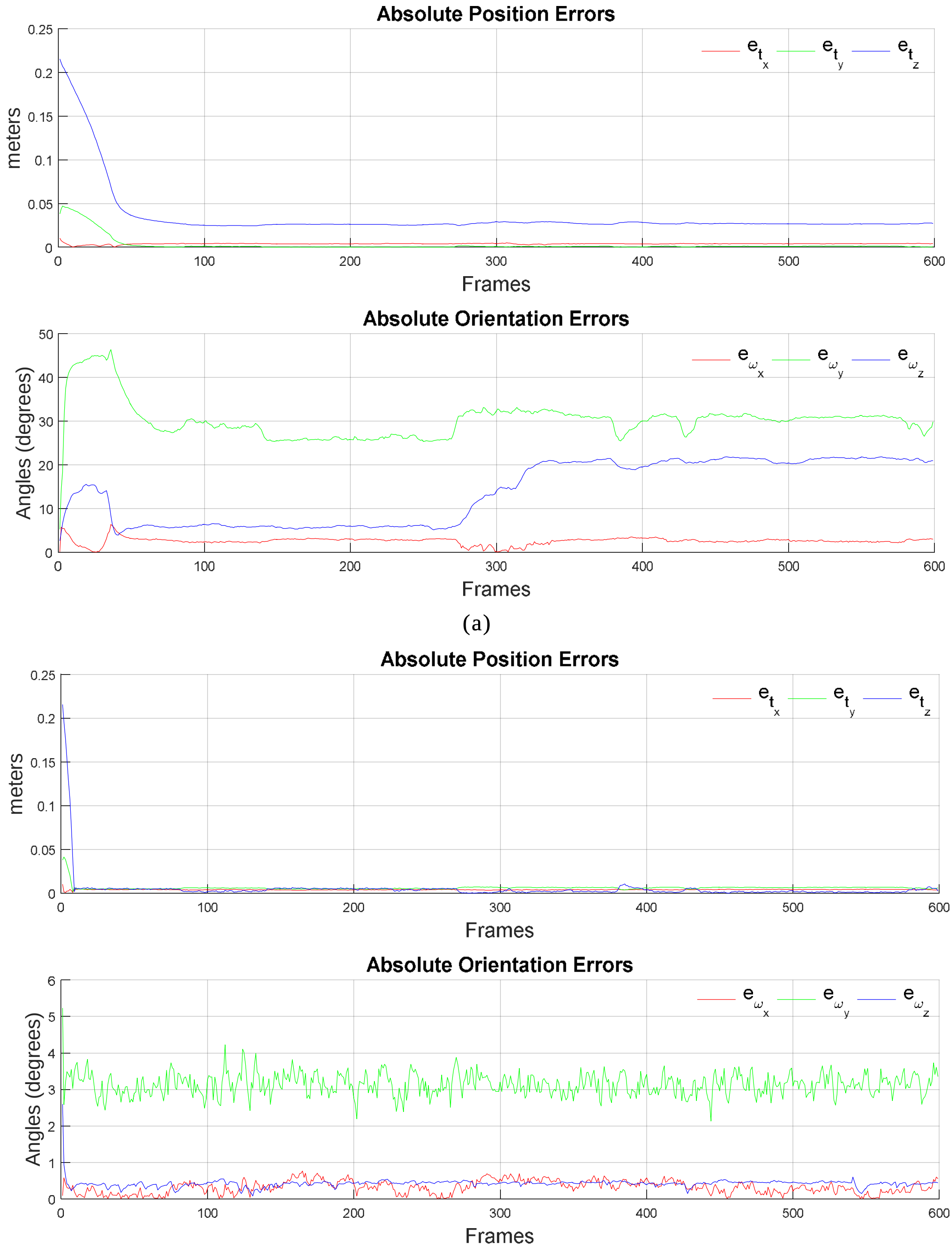
From the visual observation of frame 200 it clearly shows that the output of PWP3D was less accurate than the proposed PWP3Di-NAG. The pose output of PWP3D shows the soft-drink can was too tilted toward the camera direction. This result was also validated by plotting the error with respect to Aruco output as the reference. The complete plot of the pose estimate of both algorithms is presented in Figure 4.10 and the absolute error with respect to the pose provided by Aruco is presented in Figure 4.11. To facilitate a visual observation, a film strip is provided in Figure 4.12.



(a)

(b)

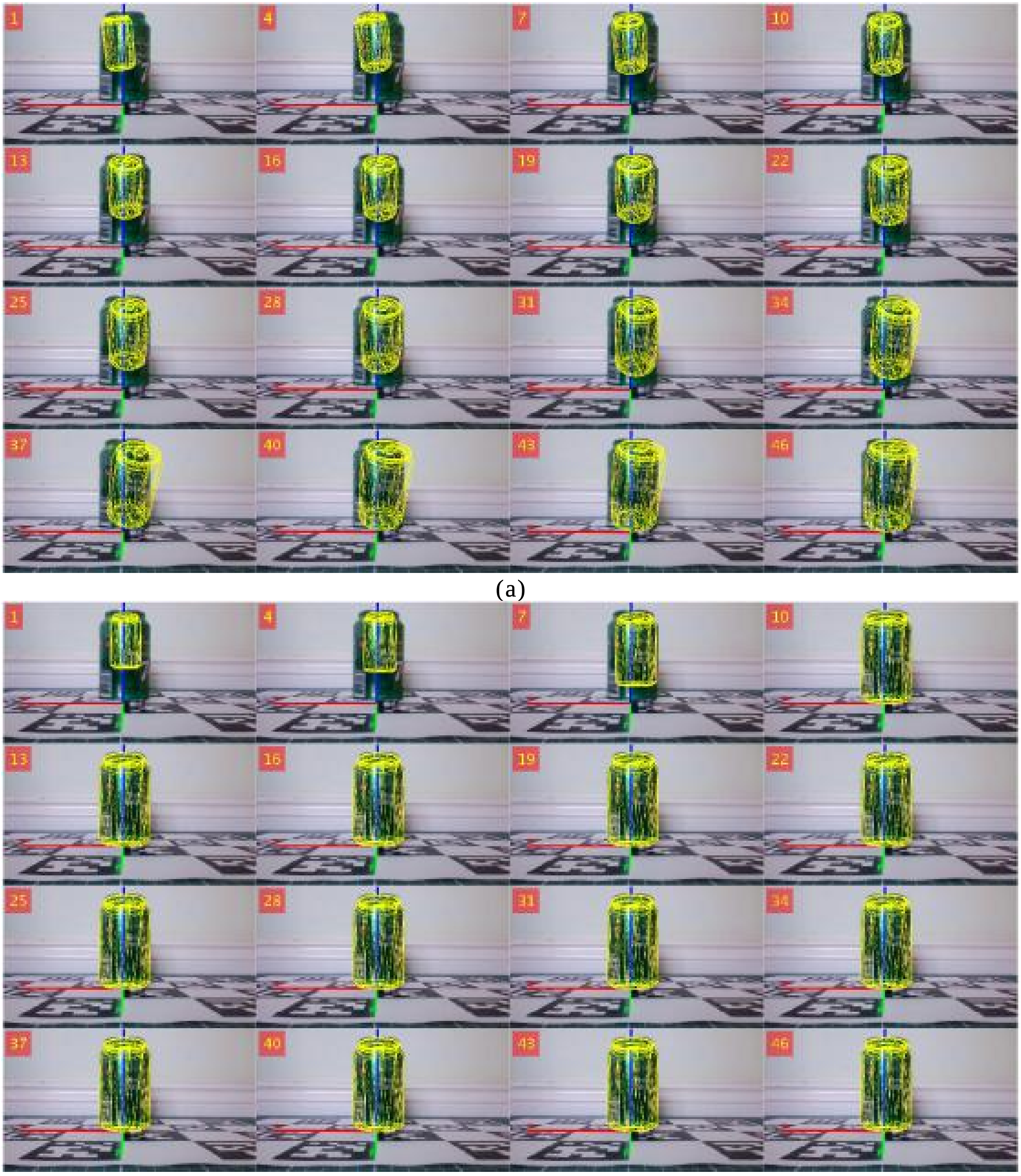
Figure 4.10. The position and orientation estimates from PWP3D (a) and PWP3Di-NAG (b). Both of the outputs were benchmarked to the Aruco output as reference. It shows the position was estimated better by both algorithms than the orientation. The PWP3D orientation estimate was not accurate and not stable. As can be seen yaw angle ω_z changed significantly despite the object was at the same pose from the very beginning till the very end of frames. The PWP3Di-NAG achieved a significantly better orientation estimate and was stable.



(a)

(b)

Figure 4.11. The absolute error of position and orientation compared to Aruco output. It shows that PWP3D converged much slower than PWP3Di-NAG. While the PWP3D can estimate the position better than the orientation, the PWP3Di-NAG output was still superior compared to PWP3D. PWP3Di-NAG achieved lower absolute error and was stable. This behaviour can also be easily observed visually from the film strip given by Figure 4.12.

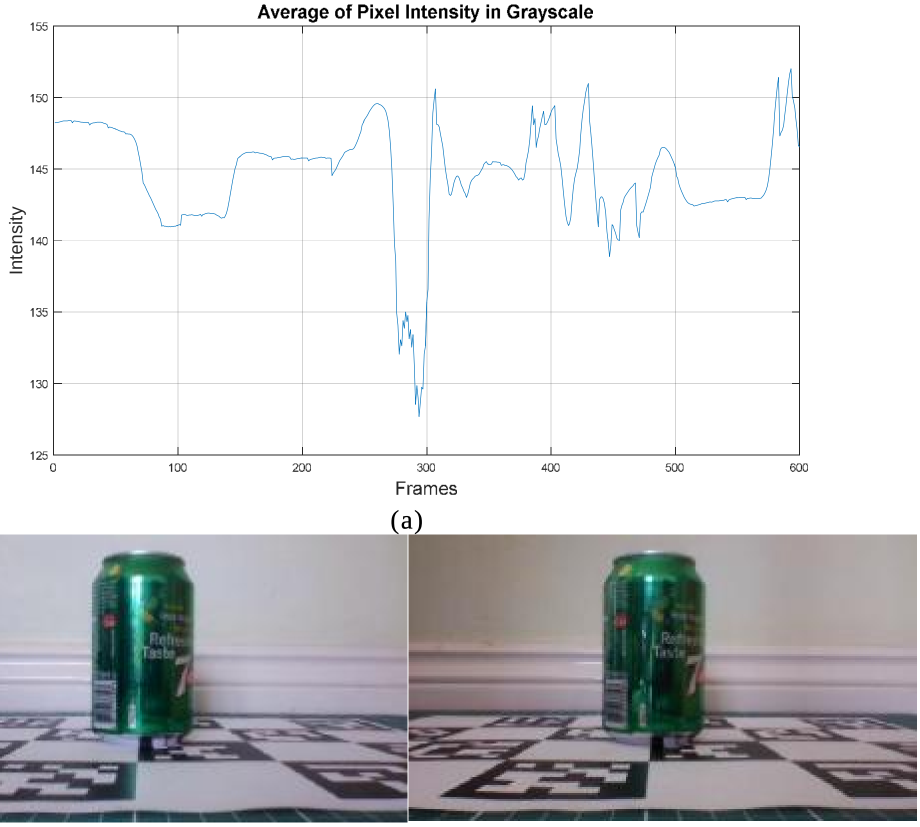


(a)

(b)

Figure 4.12. The film strip of PWP3D output (a) and PWP3Di-NAG (b). It shows the PWP3D refines the pose slowly and converges in about 40 frames. The final pose estimate was not accurate, the yaw angle ω_z error was large as basically the yaw angle cannot be retrieved from this object. The pitch angle ω_y error was also large due to the solution trapped in a bad local optima. The PWP3Di-NAG converged faster and reached a better pose estimate. As can be seen from the film strip, during this initial period, the orientation estimate did not suffer from large error hence it managed to escape from bad local optimum and converged to a better orientation estimate.

Figure 4.10 to Figure 4.12 show that the PWP3D converged to an optima that was far from the Aruco solution. The position was estimated with lower error while the orientation estimate suffered from large error. Figure 4.10 also shows the orientation estimate PWP3D was not stable, as can be seen it started drifting in the yaw angle ω_z from about frame 275. This drifting was caused by illumination changes. Figure 4.13 (a) shows the plot of average grayscale of the pixel intensity. It shows that about frame 275 there was a sharp intensity drop due to the presence of shadow. At Figures 4.13 (b) and (c) present two frames that have a different level of intensity average.



(a)

(b)

(c)

Figure 4.13. The plot of average pixel intensity in grayscale (a) and it shows a significant intensity difference occurred in about frame 275-300. To facilitate a visual observation, two frame are presented in (b, c). The frame 275 is presented in (b) along with frame 290 (c). Despite both of the frames had a different illumination level, the object was at a same pose. Therefore a good pose estimate should yield a same result for both frames.

This behaviour was observed since the soft-drink can is symmetric in the z axis. This symmetrical property means the visual-only pose estimate cannot retrieve the yaw angle and the yaw estimate was not locked to a single solution. This behaviour confirms that PWP3D suffers from multimodal projection problem.

A significantly different output was observed from the PWP3Di-NAG. From Figure 4.10 and Figure 4.11 the PWP3Di-NAG achieved a good accuracy. The PWP3Di-NAG also converged much faster in about 11 frames. The faster convergence rate expected as the Nesterov's Accelerated Gradient descent has been proved in many experiments (Botev et al., 2017; Timothy Dozat, 2015) to have a superior convergence property than classical Gradient Descent. The output of PWP3Di-NAG also demonstrated a better stability, as can be seen the pose output did not suffer from large variation despite the illumination differences. This output was also expected since the proposed method not only relies on color histogram but also the inertial / magnetic observations.

A quantitative measurement that validated the PWP3Di-NAG has a superior output than PWP3D is also given by RMSE and MAE measurement as presented by Table 4.4 and 4.5.

Table 4.4. The RMSE and MAE Position output of PWP3D and PWP3Di-NAG algorithms in soft-drink can dataset. The lowest errors are indicated by bold font style.

	RMSE			MAE		
	t_x	t_y	t_z	t_x	t_y	t_z
PWP3D	0.004	0.0061	0.0379	0.0039	0.0018	0.0306
PWP3Di-NAG	**0.004**	**0.0060**	**0.0125**	**0.0039**	**0.0015**	**0.0043**

Table 4.5. The RMSE and MAE Orientation output of PWP3D and PWP3Di-NAG algorithms in soft-drink can dataset. The lowest errors are indicated by bold font style.

	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
PWP3D	2.751	29.645	18.190	2.6825	29.453	17.145
PWP3Di-NAG	**0.409**	**3.121**	**0.4311**	**0.3720**	**3.1211**	**0.4240**

The next experiment was done for investigating the tracking performance in a moving scene.

Dynamic Dataset

The dynamic dataset was created to evaluate the tracking performance of the algorithms. This dataset also consist of frame with significant appearance difference. Since this experiment only concerns the tracking performance, the initial pose estimate was chosen to be the true pose and it was provided from the Aruco pose.

The first experiment in dynamic dataset was done in tracking the red box. In this dataset, the camera was steered by hand around the red box object. The camera's motion speed was fast to investigate the performance of both algorithms in keeping track of the object in this high dynamic dataset. The film strip output of PWP3D is presented in Figure 4.14. It shows that starting from frame around 385 the algorithm did not manage to keep track of the correct pose. Some missing alignment was observed until the last frame 755.

A different performance was observed from the PWP3Di-NAG as can be seen in Figure 4.15. From visual observation, the estimated poses were aligned well to the query images. The PWP3Di-NAG managed to keep up with the fast motion of the camera and was able to track the object from the very beginning to the end.

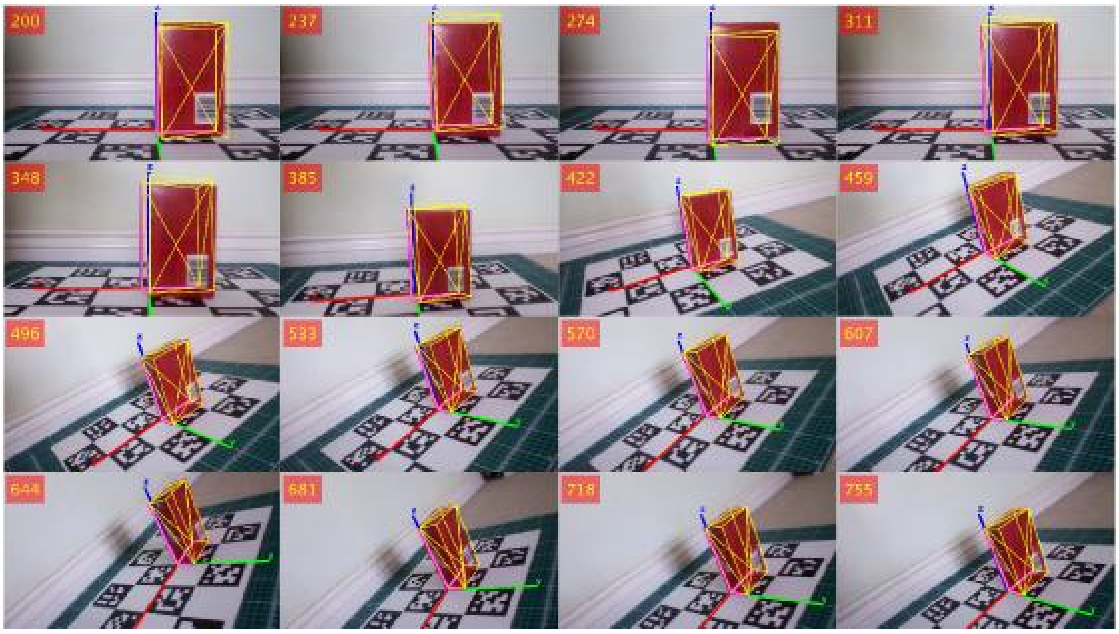


Figure 4.14. Film strip of PWP3D output in tracking red box object in a fast motion. While in the beginning the pose estimates were good, soon after frame 385 a significant missed alignment to the query image is observed. This induced a large error in orientation estimate as shown in the plot of absolute error given by Figure 4.16.

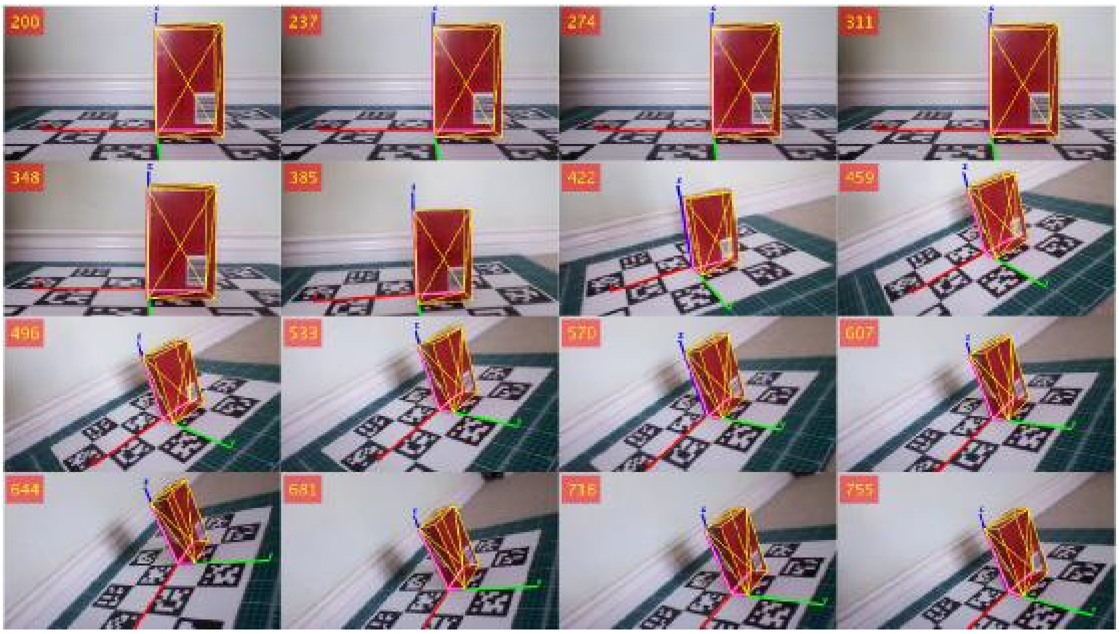
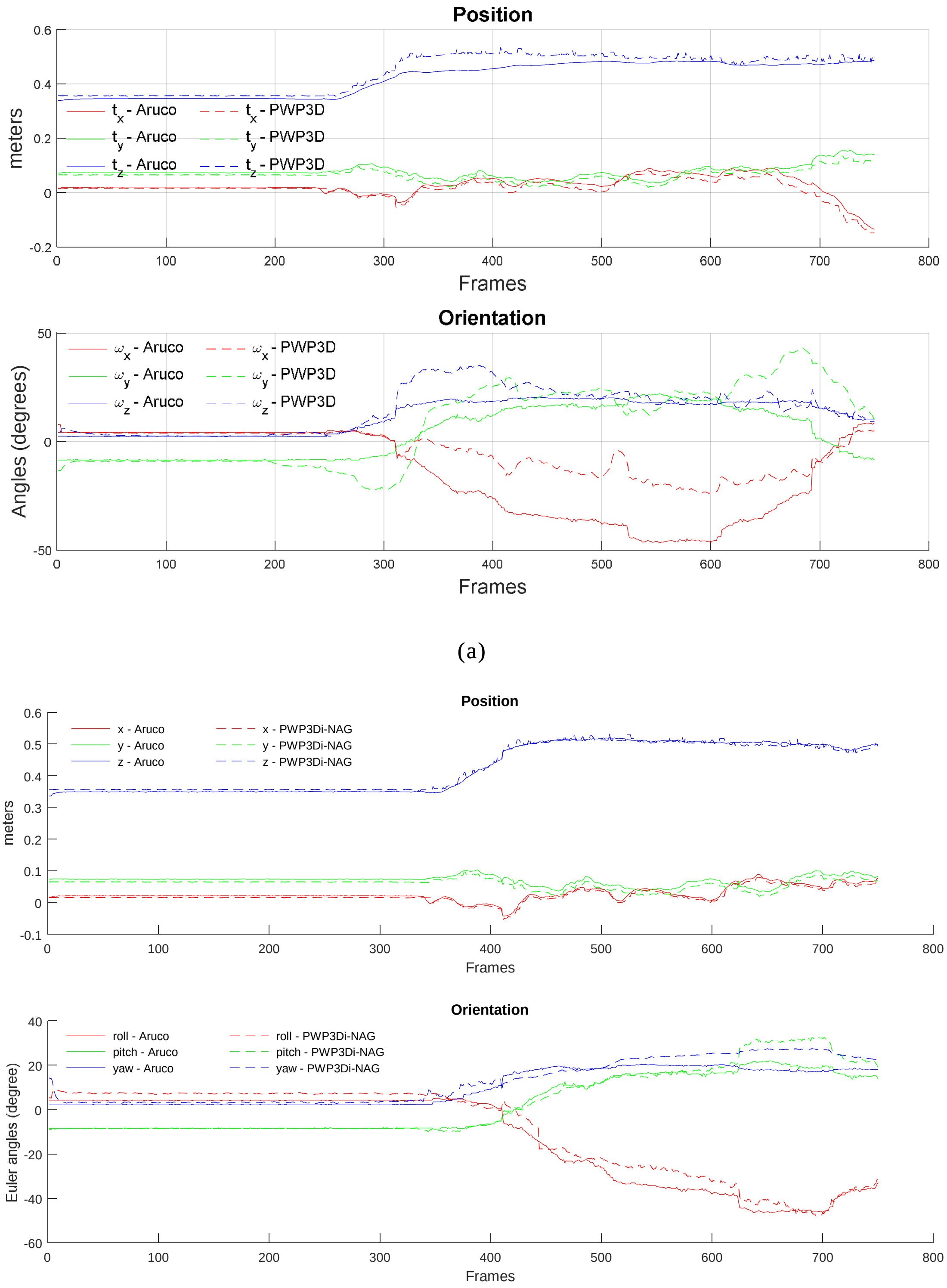


Figure 4.15. Film strip of the proposed algorithm PWP3Di-NAG. The pose estimation were good and from visual observation, the pose estimate was well aligned to the query images. The PWP3Di-NAG managed to track from the very beginning to the very end.

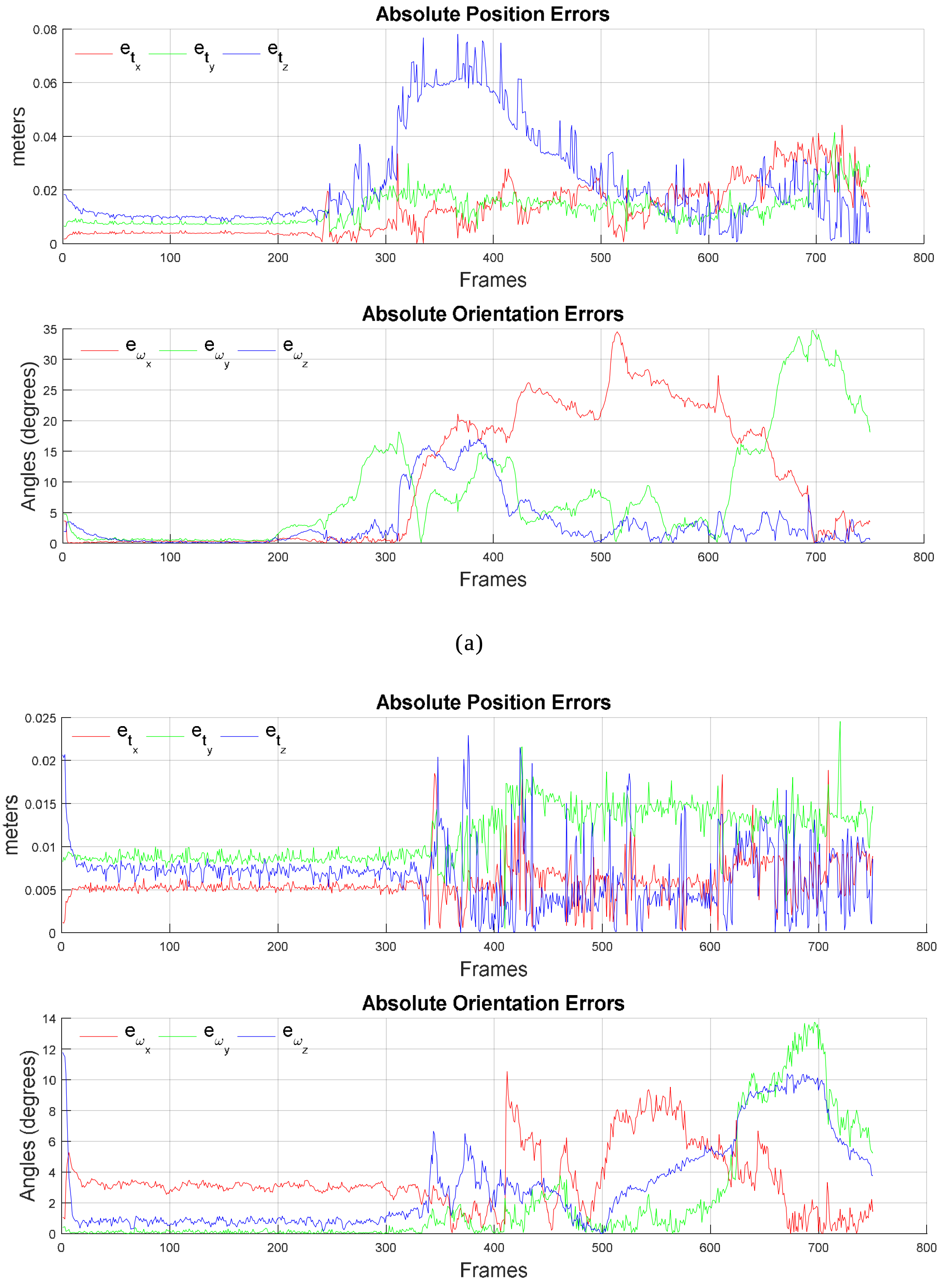
The plot of pose output of both algorithms are presented in Figures 4.16 and 4.17.



(a)

(b)

Figure 4.16. PWP3D pose estimate output of dynamic red box dataset (a) and PWP3Di-NAG pose output (b). While the PWP3D managed to track the position with low error, the orientation estimate was not good. A significantly different output was observed from PWP3Di-NAG that achieved a better accuracy.



(a)

(b)

Figure 4.17. The absolute error plot of PWP3D (a) and PWP3Di-NAG (b) in tracking red box object.

The RMSE and MAE of the position and orientation error of both algorithms are presented in Table 4.6 and Table 4.7.

Table 4.6 The RMSE and MAE Position output of PWP3D and PWP3Di-NAG algorithm in dynamic red-box dataset. The lower errors are indicated by bold font style.

	RMSE			MAE		
	t_x	t_y	t_z	t_x	t_y	t_z
PWP3D	0.0159	0.0141	0.0288	0.0125	0.0129	0.0227
PWP3Di-NAG	**0.0063**	**0.0117**	**0.0076**	**0.0058**	**0.0113**	**0.0067**

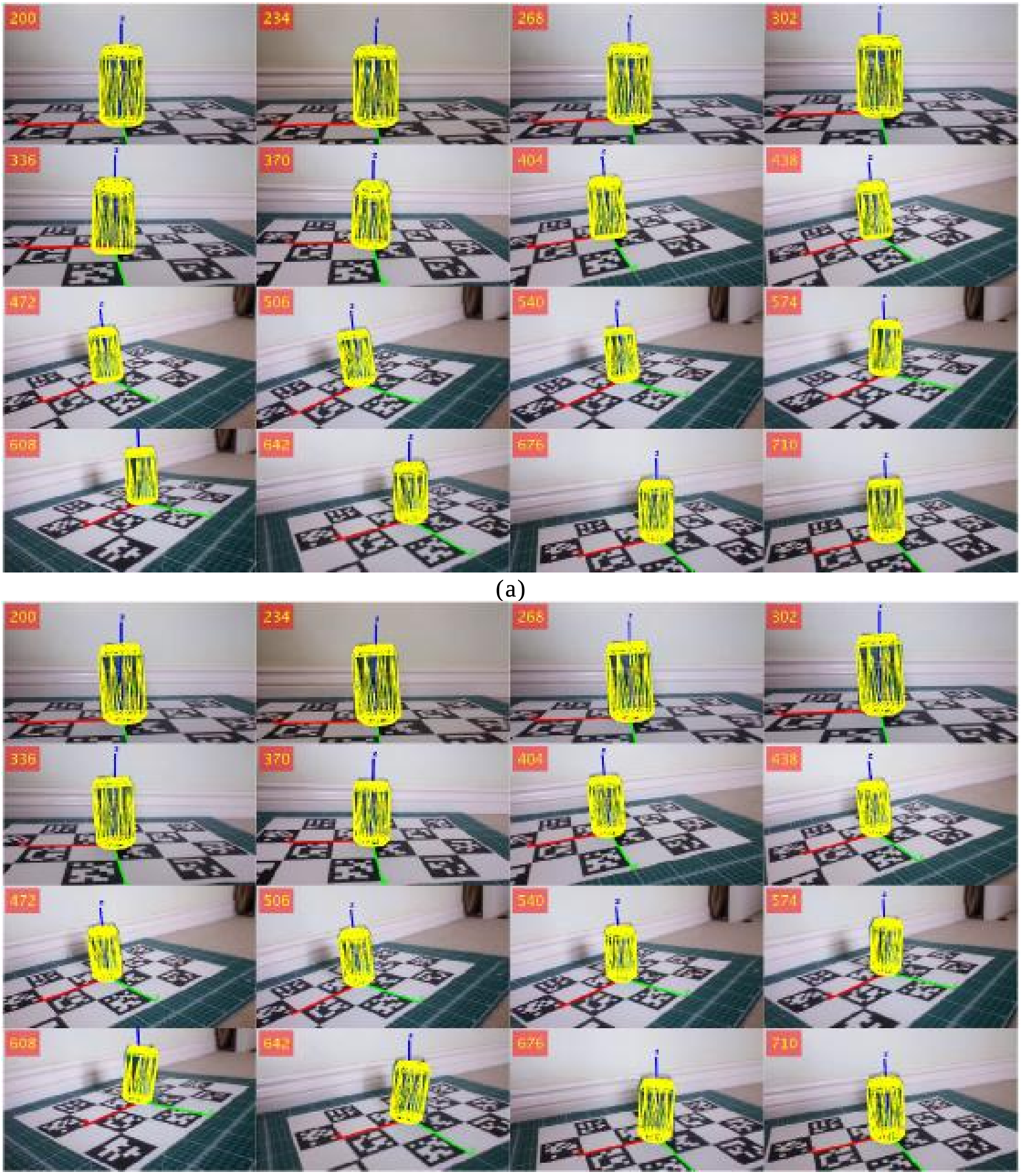
Table 4.7 The RMSE and MAE Orientation output of PWP3D and PWP3Di-NAG algorithm in dynamic red-box dataset. The lower errors are indicated by bold font style.

	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
PWP3D	14.8762	12.1774	5.4210	10.4242	8.2856	3.2434
PWP3Di-NAG	**4.0823**	**4.1550**	**4.2014**	**3.5083**	**2.1351**	**3.0459**

From these tables it shows the proposed PWP3Di-NAG outperformed the original PWP3D algorithm in tracking the red box object.

The last experiment was done on tracking the soft-drink can object. In the beginning the camera was in a steady position and after a few frames the camera start moving around the soft-drink can. The motion around the object was unstructured and it was done by hand. The Aruco pose estimate was extracted to provide the reference for the experiment.

The film strip output of this experiment can be seen in Figure 4.18. A similar result was also observed from the PWP3Di-NAG algorithm as can be seen in Figure 4.19. It shows both algorithms also managed to track the object from the very beginning to the very last frame, but the accuracy was difference.

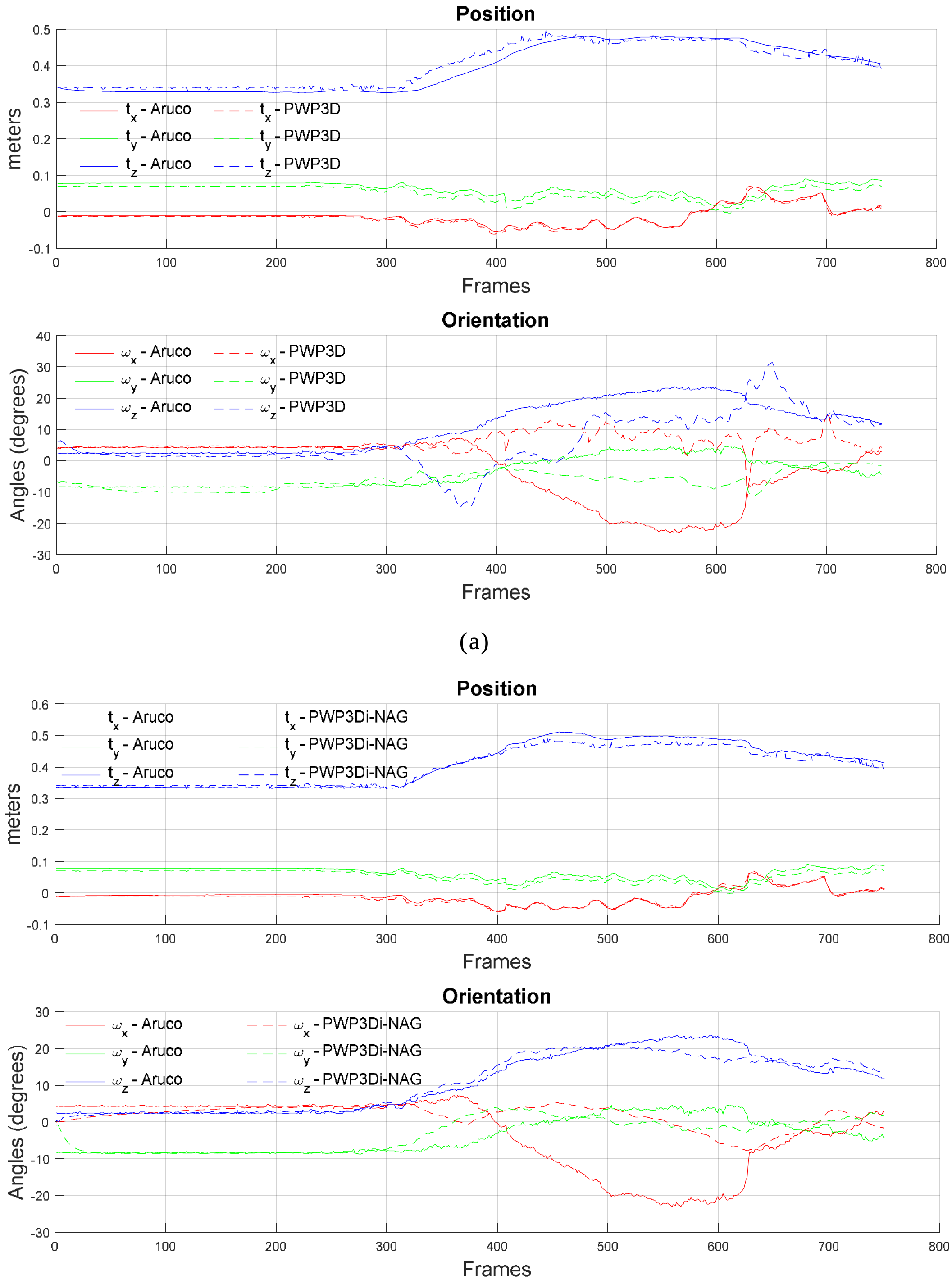


(a)

(b)

Figure 4.18. Output of the PWP3D (a) and PWP3Di-NAG (b) in tracking of soft-drink can object. In general both algorithms managed to track the object with a similar observed result.

While the film strip shows that both algorithms managed to track the object, the level error was not easy to observe visually. For the purpose of evaluating the performance, plots of the pose estimate output along with Aruco output are presented in Figures 4.19 – 4.20.



(a)

(b)

Figure 4.19. The output of PWP3D (a) and PWP3Di-NAG (b) algorithms in tracking soft-drink can object. It shows that the position tracking of both algorithms was accurate with a low level of error. However, the orientation estimate suffered from larger error. The large orientation error resulted as some part of object was less distinguishable from the background, such as the silver coloured top-part of can. The presence of shadow also makes the posterior probability become unclear hence the orientation error was observed. In the PWP3Di-NAG the orientation estimate was slightly better since it has an additional information from inertial/magnetic sensor. However inertial/magnetic orientation estimate also has some level of error so the final pose estimate still suffers from error.

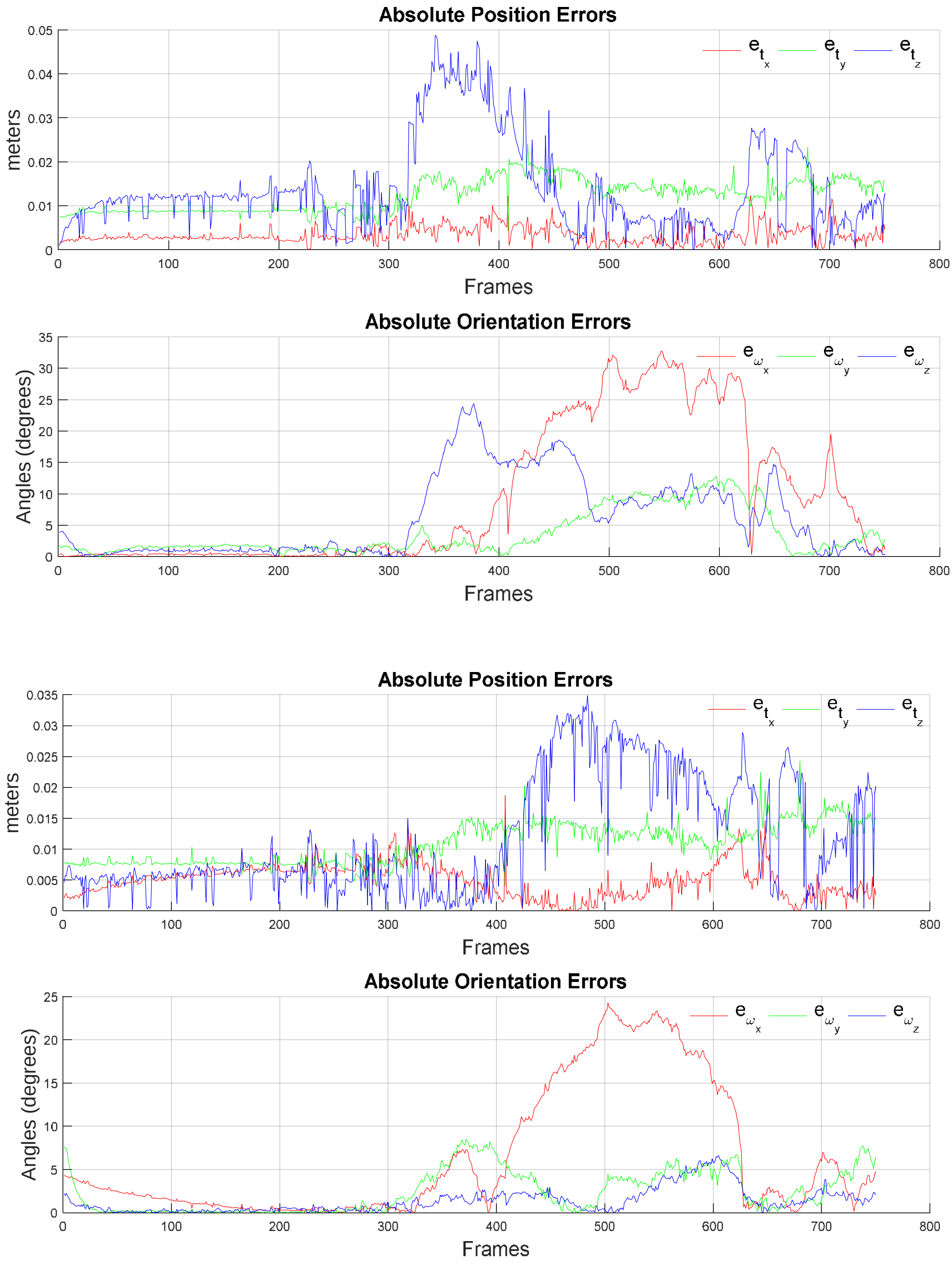


Figure 4.20. The tracking output error of PWP3D (a) and PWP3Di-NAG (b) given a dynamic soft-drink can dataset.

The RMSE and MAE measurement of the dynamic soft-drink can dataset is provided by Table 4.8. and Table 4.9.

Table 4.8. The RMSE and MAE Position output of PWP3D and PWP3Di-NAG algorithm in dynamic soft-drink can dataset. The lower errors are indicated by bold font style.

	RMSE			MAE		
	t_x	t_y	t_z	t_x	t_y	t_z
PWP3D	**0.0039**	0.0129	0.0274	**0.0033**	0.0123	0.0240
PWP3Di-NAG	0.0055	**0.0112**	**0.0251**	0.0049	**0.0108**	**0.0219**

Table 4.9. The RMSE and MAE Orientation output of PWP3D and PWP3Di-NAG algorithm in dynamic soft-drink can dataset. The lower errors are indicated by bold font style.

	RMSE			MAE		
	ω_x	ω_y	ω_z	ω_x	ω_y	ω_z
PWP3D	14.6107	5.0472	8.8485	9.4682	3.6078	6.1544
PWP3Di-NAG	**9.9951**	**3.4352**	**1.9720**	**6.4730**	**2.4038**	**1.3563**

From these tables it shows the proposed PWP3Di-NAG outperformed the original PWP3D algorithm in tracking the red box object

4.7 Concluding Remarks

This chapter presented an algorithm that incorporated visual and inertial tracking into a single and neat system of non-linear equations that was then addressed as a pure optimization problem. The chosen method for solving the problem was Nesterov Accelerated Gradient descent that is widely known has a better performance than classical Gradient Descent. As the proposed algorithm works on pixel-wise color posterior probability combined with inertial sensor, the author refers to the new algorithm as PWP3Di-NAG (Pixel-wise Color Posterior and Inertial/Magnetic 3D pose tracker solved by using Nesterov Accelerated Gradient descent). The proposed PWP3Di-NAG can be seen as an improvement of the existing state-of-the-art algorithm (PWP3D). The PWP3Di-NAG also heavily utilises the inertial/magnetic orientation estimate that has been developed previously and known as NAG-AHRS.

The validation shows that PWP3Di-NAG outperformed the PWP3D algorithm and it managed to overcome multimodal projection problem which is one of the fundamental problems in the original algorithm. The incorporation of inertial/magnetic orientation estimate also demonstrated can avoid local optima better than the original algorithm. Lastly, the implementation of Nesterov Accelerated Gradient descent also improved the capability in handling wider dynamic motion of the object. As a conclusion, the PWP3Di-NAG offered a significant improvement of the original PWP3D algorithm.

PWP3Di-NAG integrates vision and inertial/magnetic information as a single optimisation problem, it requires all measurements, whether visual or inertial, to be defined in a single common reference system. Due to this requirement, the inertial/magnetic measurements need to be transformed to object frame and to be able to do this transformation, the initial relative pose between inertial/magnetic to the object must be estimate. Finding the relative pose between inertial/magnetic to the object requires an extra initialisation stage which is presented in the next chapter.

Chapter 5

Initialisation Framework Based on PWP3D-NAG and Particle Filter

5.1 Background

In Chapter 4 an algorithm that combines visual and inertial information is proposed to deal with a fundamental problem in vision-only 3D pose estimation known as the multimodal projection problem. The proposed method improves the state-of-the-art vision-only pose estimation algorithm known as PWP3D (Victor A. Prisacariu & Reid, 2012) by combining it with a novel inertial/magnetic orientation estimate that has been developed in Chapter 3 (NAG-AHRS). The proposed hybrid visual-inertial algorithm in Chapter 4 (PWP3Di-NAG) has been validated and it demonstrated better results than the original algorithm. However, since the integration between vision and inertial/magnetic information is addressed as a single optimisation problem, the proposed PWP3Di-NAG algorithm requires all measurements, whether visual or inertial, to be defined in a single common reference system. Due to this requirement, the inertial/magnetic measurements need to be transformed to object frame and to be able to do this transformation, the initial relative pose between inertial/magnetic to the object must be known.

This chapter addresses this problem by proposing an initialisation framework for PWP3Di-NAG that is built from a structured multiple PWP3D-NAG initialisation. PWP3D-NAG itself is another improvement of PWP3D algorithm, which is also proposed in this chapter. The multiple hypotheses output of the multiple initialisation PWP3D-NAG algorithm are then combined using a particle filtering framework. The particle filtering pose estimation that works on edge-based information is selected to complements the region-based pose estimation to refines the final pose estimate. The detailed proposed framework is presented in Section 5.3, but before that, a brief literature review is presented in Section 5.2. Experiments for validating the proposed framework along with results and discussion is then presented in Section 5.4 and finally, concluding remarks are covered in Section 5.5.

5.2 Literature Review

Estimating the three dimensional pose of an object is required for autonomous inspection (Tjaden, Schwanecke, Schömer, & Cremers, 2018) and the pose of an object can be estimated by relying on: salient points, edges, or statistical appearance of a model (Bibby & Reid, 2008; Comport et al., 2006; Crivellaro et al., 2015). However, as aforementioned in Chapter 4, the pose of poorly-textured / textureless objects is better addressed by using region-based methods (Bibby & Reid, 2008; Dambreville et al., 2008; Kehl et al., 2017; V A Prisacariu et al., 2015; Victor A. Prisacariu & Reid, 2012; Tjaden et al., 2016, 2017, 2018).

As region-based method basically tries to minimise the discrepancy between boundaries of the segmented image and the boundaries of a silhouette achieved from the projected model, it suffers from the multimodal projection problem when tracking symmetrical-objects. Chapter 4 already dealt with this problem, and a hybrid visual-inertial/magnetic 3D pose estimation, referred to as PWP3Di-NAG, has been developed. PWP3Di-NAG combines visual and inertial/magnetic information as a single optimisation problem, hence it requires all measurements to be done in a same reference system. The chosen reference system is the object's coordinate system since the primary concern in 3D pose estimation is to estimate the camera's pose with respect to the object being tracked.

The visual part of the algorithm that is built on the PWP3D method is already developed in this reference frame, hence it does not require any modification. A different situation applies to the NAG-AHRS inertial/magnetic orientation estimate. NAG-AHRS is developed in a global reference frame, therefore it requires additional pre-processing to transform the raw inertial/magnetic measurements into the object's coordinate system before combining it as a single optimisation problem. This transformation can only be done if the pose of the inertial/magnetic sensors with respect to the object reference frame is known in the beginning. As a consequence of this requirement, an additional step, known as initialisation stage, is necessary. The initialisation stage is aimed to know the pose of the inertial/magnetic sensor with respect to the object in the first frame.

Since the inertial/magnetic sensor is placed rigidly with respect to the camera, the transformation from inertial/magnetic frame to the camera frame is fixed regardless the object pose. The relation between inertial/magnetic, camera and object frame is shown in Figure 5.1.

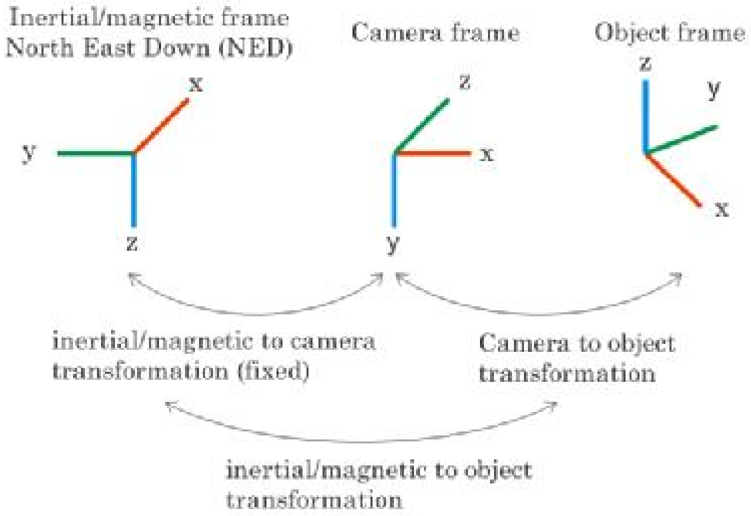

Inertial/magnetic frame
North East Down (NED)
Camera frame
Object frame
x
y
z
z
x
y
z
y
x
inertial/magnetic to camera transformation (fixed)
Camera to object transformation
inertial/magnetic to object transformation

Figure 5.1. Transformation from the inertial/magnetic frame to the object frame can be computed through the camera frame. Since the inertial/magnetic sensor is placed in a fixed position with respect to the camera, the transformation from inertial/magnetic to camera frame is fixed regardless the object frame pose. In this case, there is only one problem left, which is to estimate camera to object frame transformation.

In this case, there is only one unknown transformation left for computing inertial/magnetic to object transformation which is: the camera to object transformation. The initialisation framework addresses this problem by proposing a novel framework for initialisation. The strategy is mainly based on the idea of performing multiple pose estimations using an improved PWP3D algorithm, referred to PWP3D-NAG, with different initial poses. The different initial pose guesses are obtained structurally by taking different viewpoints obtained from different vertices in a subdivided icosahedron geometry or icosphere. Utilising multiple viewpoints from icosphere vertices is well known for generating templates as shown in (Hinterstoisser et al., 2013; Tjaden et al., 2017) but as far as author's knowledge, no research has been done in using this approach for selecting the initial pose, especially for region-based pose estimation.

As the PWP3D-NAG performs pose estimation based on color histogram information, hence when some parts of the object have a similar color to the background, PWP3D-NAG cannot address it correctly. In this case, the pixel posterior probability of being part of the object or background becomes unclear and it reduces the accuracy of the pose estimate. The proposed initialisation framework overcomes this problem by performing a refining stage by running edge-based pose estimation to complement the color-based pose estimation. The chosen edge-based pose estimation is based on a particle filter as it shows a good result (Choi & Christensen, 2011; Kim & Sim, 2010; Klein & Murray, 2006; Pupilli & Calway, 2006).

Therefore, the contributions of this chapter can be summarised as follows:

- An improvement of PWP3D to gain better convergence by replacing classical Gradient Descent with Nesterov's Accelerated Gradient descent (PWP3D-NAG); and
- An initialisation framework mainly based on structured multiple viewpoint initialisation of region-based pose estimation PWP3D-NAG, refined using edge-base pose estimation on particle filtering method. A refining stage is aimed for dealing with color similarity problem and to increase the accuracy.

5.3 The Initialisation Framework

In this section, a structured multiple viewpoint initialisation is introduced as part of the proposed initialisation framework. This framework consists of an improvement of the PWP3D algorithm by implementing a better optimisation method which is Nesterov's Accelerated Gradient. Therefore, this method is discussed first before the framework.

PWP3D-NAG

The proposed framework utilises PWP3D with an improvement in the optimisation method. Instead of using Gradient Descent, a Nesterov's Accelerated Gradient descent is adopted. The classical Gradient Descent which only requires first derivatives of the objective function refines the estimation in the opposite direction of the gradient. The update rate is scaled by a parameter known as step size μ. Setting the step size to a large number can shorten the convergence time, but at the same time it increases the possibility to jump over the optimum and triggers unnecessary oscillations. In contrast, a small step size decreases the convergence rate significantly. In a case when the gradient of the objective function $f(\theta_t)$ is very moderate, the convergence time can be very tedious (Timothy Dozat, 2015). The classical Gradient Descent is given by:

$$\theta_{t+1} = \theta_t - \mu.\nabla f(\theta_t)$$

where θ is the pose, $f(\theta_t)$ is the objective function to be minimized, μ is the step size and $\nabla f(\theta_t)$ is the gradient at θ_t.

A faster convergence time can be achieved by accumulating the previous gradient, known as the momentum method (Botev, Lever, & Barber, 2017). Momentum method maintains progress along the direction of the previous update hence it can reach the solution in a shorter time (Goh, 2017). The momentum gradient descent is given by:

$$v_{t+1} = \alpha.v_t - \mu.\nabla f(\theta_t)$$

$$\theta_{t+1} = \theta_t + v_{t+1}$$

where v_{t+1} is the velocity term at which parameter should be refined and $\alpha > 0$ is a momentum coefficient which determines the accumulation of the previous gradient. However, since it accumulates the gradients from the previous updates, when it is close to the minimum point, the momentum can be very high and possibly surpass the optimum (Goh, 2017).

To deal with this problem, Nesterov's Accelerated Gradient descent (NAG) was proposed by computing the gradient correction velocity in the predicted position ahead $\nabla f(\theta_t + \alpha.v_t)$ instead of the gradient at the current position $\nabla f(\theta_t)$ (Nesterov, 1983). This prediction allows the NAG to refine the update in a better direction, reduce unnecessary update in wrong direction and finally improves the responsiveness. NAG is also able to smooth the oscillations by damping the update in turbulent direction (Hinterstoisser et al., 2013)(Sutskever, Martens, Dahl, & Hinton, 2013). The NAG is given by:

$$v_{t+1} = \alpha . v_t - \mu . \nabla f(\theta_t + \alpha . v_t)$$

$$\theta_{t+1} = \theta_t + v_{t+1}$$

The convergence rate of NAG is $O(1/k^2)$ which is better than classical Gradient Descent that has $O(1/k)$ convergence rate. The k constant is proportional to the squared Euclidean distance to the solution (Sutskever et al., 2013).

Implementing NAG into the PWP3D is done by modifying the original PWP3D update that is given by:

$$\lambda_{t+1} = \lambda_t - \mu . \frac{\partial E(\lambda_t)}{\partial \lambda_t}$$

where λ is the pose parameters given by $(t_x, t_y, t_z, q_w, q_x, q_y, q_z)$, μ is the step size of Gradient Descent and $\frac{\partial E}{\partial \lambda}$ is the partial derivative of the energy function. Recall the partial derivative of the energy function is formulated by:

$$\frac{\partial E}{\partial \lambda} = - \sum_{x \in \Omega} \frac{P_f - P_b}{H_e(\Phi) P_f + \left(1 - H_e(\Phi)\right) P_b} \delta_e(\Phi) \begin{bmatrix} \frac{\partial \Phi}{\partial x} & \frac{\partial \Phi}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \lambda_i} \\ \frac{\partial x}{\partial \lambda_i} \end{bmatrix}$$

In the proposed PWP3D-NAG, the pose update becomes:

$$v_{t+1} = \alpha . v_t - \mu . \frac{\partial E(\lambda_t + \alpha . v_t)}{\partial (\lambda_t + \alpha . v_t)}$$

$$\lambda_{t+1} = \lambda_t + v_{t+1}$$

Multiple Viewpoint Initialisation

The proposed PWP3D-NAG is aimed to improve the convergence time while still maintaining the benefit of classical Gradient Descent in terms of convergence guarantee. However, this method can only converge to the nearest minimum within its convergence basin. To facilitate the solution to converge to the global optimum or at least to a better local optimum, the PWP3D-NAG is executed multiple times with different initial poses. The different initial poses come from different camera locations around the object pointing toward the object. In this case, the camera positions are on a sphere surface pointing to the origin.

To determine the camera location and orientation, a structured ball shape geometry is required. The typical method for generating a sphere in computer graphics is UV sphere and icosphere. UV sphere uses segments and rings while icosphere is built from subdivided icosahedron. Figure 5.2 illustrates both methods.

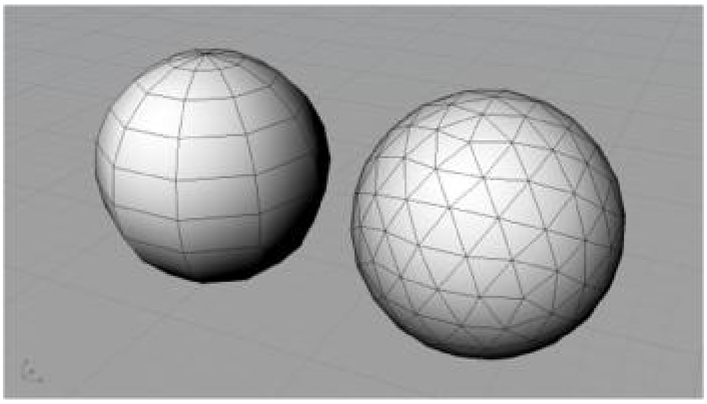


Figure 5.2. A sphere created from the different method: UV sphere (left) and icosphere (right). UV sphere uses segments and rings where icosphere is built from subdivided icosahedron. The UV sphere's faces have different areas that can be easily observed by comparing the area in the equator to the area close to the poles. The icosphere maintains the same face area so the vertices are distributed evenly. Image is reproduced from https://www.food4rhino.com/app/icosphere

Among these geometries, the icosphere is preferred as the vertices locations are distributed evenly across the surface. In contrast to the UV sphere where segments become close between each other in the area near the poles, the vertices are not distributed equally. Another benefit of using icosphere is it can be recursively divided to achieve a different finer viewpoint level while still maintaining equal distribution on the ball surface. Figure 5.3 shows that a different level of fine sphere can be generated depending on the subdivision level.

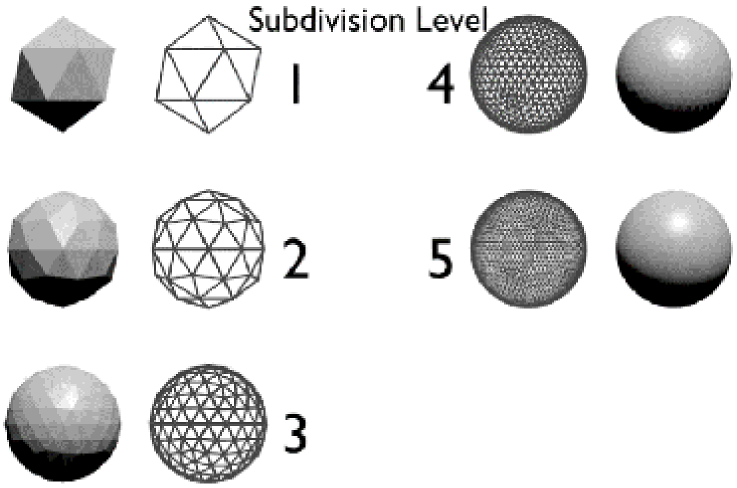


Figure 5.3. A sphere created by subdividing the icosahedron can be very smooth depending on the level of subdivision. The vertices distribution are always even regardless of the subdivision level. This geometry has the benefit for the framework as the set of orientations can be generated in finer resolution, depending on the subdivision level. Image is reproduced from https://en.wikipedia.org/wiki/Icosphere

The utilisation of icosphere is well known in computer vision with the common application to generate templates from multiple viewpoints (Hinterstoisser et al., 2013; Tjaden et al., 2017) as can be seen from Figure 5.4.

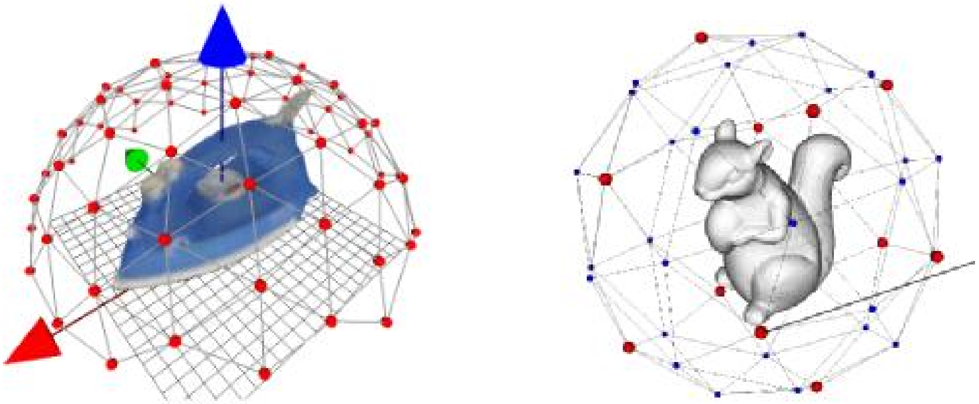


Figure 5.4. Some researches that generated templates by utilising subdivided icosahedron geometry as shown by (Hinterstoisser et al., 2013) and (Tjaden et al., 2017). Images are reproduced from (Hinterstoisser et al., 2013; Tjaden et al., 2017)

The proposed initialisation framework utilises icosphere for choosing the initial poses for the multiple PWP3D-NAG executions

Particle Filter for Edge-Based 3D Pose Estimation

The initialisation strategy to achieve better pose estimate is done by executing PWP3D-NAG multiple times with different initial poses. By this approach, it is expected that given some initial poses, the PWP3D-NAG converges to the global optimum or at least a better local optimum. The output of the multiple PWP3D-NAG executions still has multiple hypotheses that need to be selected. In the proposed framework, the selection is done by particle filtering method. The particle filter takes all of the outputs as the initial state of the particles, then executes it to find the final pose.

The particle filtering stage is also aimed to refine the pose estimate. PWP3D-NAG works on the statistical appearance model, therefore if in some parts the color of the object and background are similar, it becomes a bad input for the algorithm and it affects the accuracy of the pose estimation. By implementing a particle filter that works on edge information it complements the estimation. The excellent performance of edge-based pose estimation has been demonstrated by (Comport et al., 2006; Drummond & Cipolla, 2002; Kim & Sim, 2010; Lebeda et al., 2012; Seo et al., 2014; Wang et al., 2015). The main drawback of edge-based pose estimation which is tend to suffers from visually cluttered background and from blurry images does not apply for this initialisation frame. In this case, since the initial pose is already obtained from PWP3D-NAG and the pose estimation is already close to the optimum point, the visually cluttered background has no significant burden in this case. The second drawback that comes from blurry images also does not apply in this case, as in the initialisation stage the camera and object are in a fixed position that has a very low possibility in getting blurry images.

Particle filter which combines Bayesian filter with Monte Carlo sampling represents the posterior density $p(X_t|Z_{1:t})$ by using a set of particles with associated weights. Each particle stores pose information and their weight. The state of each particle consists of the position and orientation of the object so $\lambda_t = X_t = [t_x \quad t_y \quad t_z \quad \omega_x \quad \omega_y \quad \omega_z]$ is the state at time t. Where t_x, t_y, t_z is the location of the object in three dimensional Cartesian space and $\omega_x, \omega_y, \omega_z$ is the orientation of object represented in Euler angles. In particle filtering framework the posterior density $p(X_t|Z_{1:t})$ is represented as a set of weighted particles symbolized as:

$$S_t^\pi = \{\left(X_t^{(0)}, \pi_t^{(0)}\right), \left(X_t^{(1)}, \pi_t^{(1)}\right), \left(X_t^{(2)}, \pi_t^{(2)}\right) \dots \left(X_t^{(N)}, \pi_t^{(N)}\right)\}$$

where S_t^π is a set of weighted particles with number of particles N, $X_t^{(n)} \in \mathbb{R}^6$ represents samples of the current state X_t, $\pi_t^{(n)}$ is weight of each particle that is normalized so the total of all particle weights is equal to 1 and it is proportional to the likelihood function $p\left(Z_t|X_t^{(n)}\right)$.

The current state can be obtained by weighted particle mean

$$X_t = \sum_{n=1}^{N} \pi_t^{(n)} X_t^{(n)}$$

or from the state of the particles having the best weight

$$X_t = X_t^{(j)}, j = \arg\max_j \pi_t^{(j)}$$

In the ideal case, the correct pose should have the same projection shape as the input image and it also aligns perfectly so the total of Euclidian distance is zero.

In more detail, the weight of a particle n at time t is computed by projecting the CAD model according to its state $X_t^{(n)}$ to obtain projected model $M_t^{(n)}$. The weight of this particle is the average of the distance of each pixel $m \in M_t^{(n)}$ to the nearest edge pixel q in the edge input image Q:

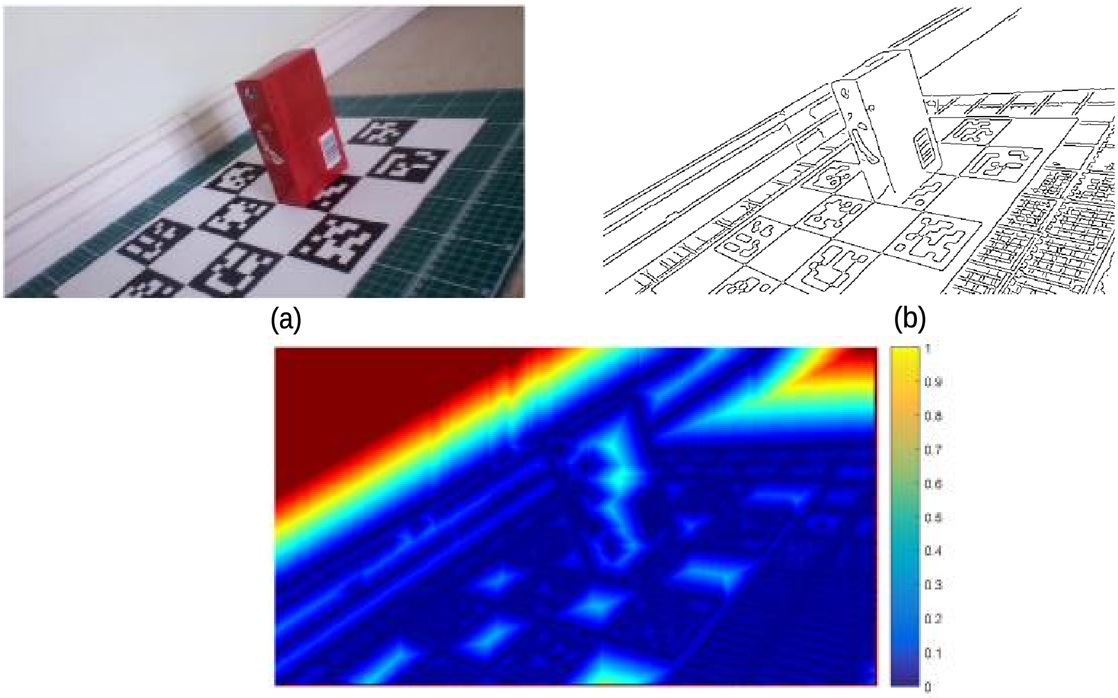
The weight $\pi_t^{(n)}$ can then be calculated by

$$\pi_t^{(n)} = \frac{1}{\left\|M_t^{(n)}\right\|} \sum_{m \in M_t^{(n)}} \min_{q \in Q} \|m - q\|$$

where $\left\|M_t^{(n)}\right\|$ is the number of pixels in the projected model $M_t^{(n)}$. Finding the minimum distance can be done by exhaustively searching the distance one by one (brute force). This naive implementation is clearly not efficient since the complexity to compute each pixel is $O(k \left\|M_t^{(n)}\right\|)$ where k is the number of pixels in edge query image Q.

A more efficient method for computing the distance to the nearest detected edge was introduced by Borgefors (Borgefors, 1986). In the proposed method, a distance transform is performed for building a distance map image. Each pixel in this image stores the distance information of that particular pixel location to the nearest edge. By having this distance map, an exhaustive search is not required. An example result of distance transformation is presented in Figure 5.5

The distance map can be calculated efficiently by sweeping a mask from top left to the bottom right of the edge input image then sweeping a different mask in the opposite direction afterwards (Barrow, Tenenbaum, Bolles, & Wolf, 1977). Therefore the complexity for finding the weight becomes $O(k) + O(\left\|M_t^{(n)}\right\|)$ since the distance map is only needed to compute one time for measuring all of the particle weights.



(a) (b)

(c)

Figure 5.5. The fitness is computed from the distance of the pixel to the extracted edge. The proposed framework implements an efficient distance transform for this purpose. The picture shows the input image (a), the extracted edges (b) and its corresponding distance transform (c).

The Initialisation Framework Summary

The initialisation framework consists of 4 main stages: 1. Coarse position estimate; 2. Set of orientation generators; 3. Pose estimation refining by using multiple PWP3D-NAG; and 4. Edge-base pose estimation using particle filter. The block diagram is shown in Figure 5.6.

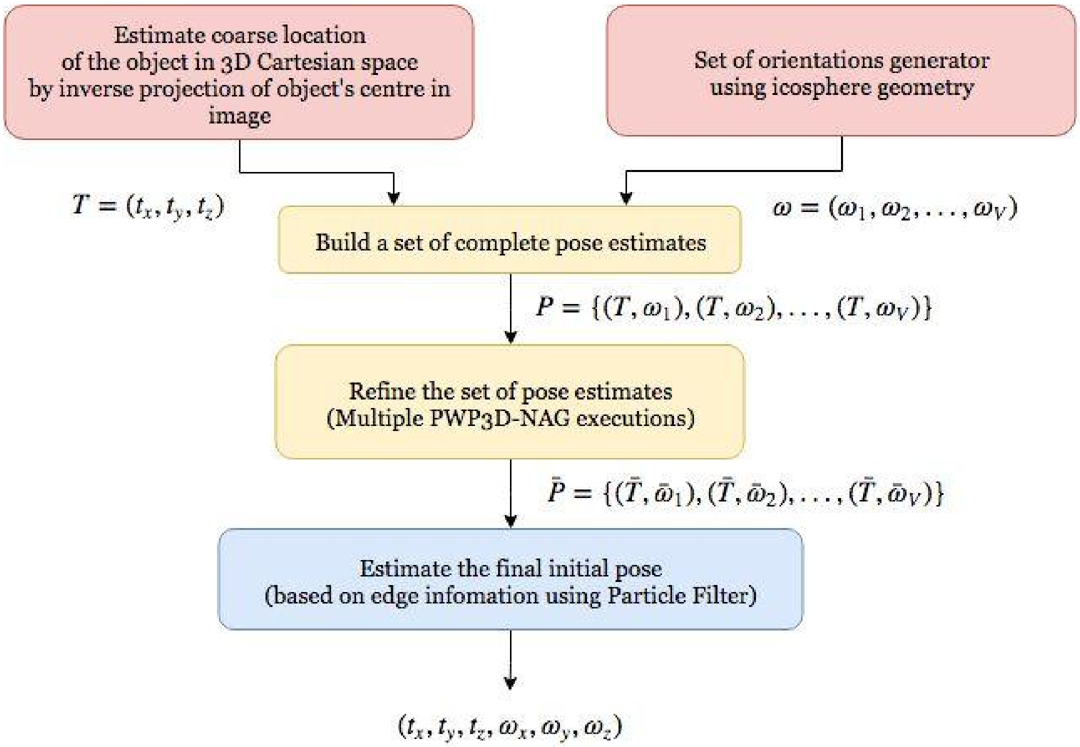

Estimate coarse location of the object in 3D Cartesian space by inverse projection of object's centre in image
Set of orientations generator using icosphere geometry
$T = (t_x, t_y, t_z)$
$\omega = (\omega_1, \omega_2, \ldots, \omega_V)$
Build a set of complete pose estimates
$P = \{(T, \omega_1), (T, \omega_2), \ldots, (T, \omega_V)\}$
Refine the set of pose estimates (Multiple PWP3D-NAG executions)
$\bar{P} = \{(\bar{T}, \bar{\omega}_1), (\bar{T}, \bar{\omega}_2), \ldots, (\bar{T}, \bar{\omega}_V)\}$
Estimate the final initial pose (based on edge infomation using Particle Filter)
$(t_x, t_y, t_z, \omega_x, \omega_y, \omega_z)$

Figure 5.6. The initialisation framework. Given a query image, firstly, a coarse object location is computed by an inverse projection of the centre region of the object in 2D image plane. Secondly, along with multiple viewpoint orientation obtained from icosphere vertices, PWP3D-NAG is executed multiple times to get multiple hypotheses of object pose. The output of this stage is then passed as initial state of particles in PF and then PF is executed to get final initial pose.

As shown in Figure 5.6; the initialisation framework is started by estimating the coarse object position in 3D space by inverse projection of object's centre (u, v) in 2D image plane. Finding object's centre is done by getting each pixel's color (y_i) and then calculating the probability of the color given object's appearance model so $P_f = P(y_i|M_f)$. After all of the pixels probabilities are obtained, the centre is computed from the weighted mean of all pixels. The coarse 3D location (t_x, t_y, t_z) is then computed by inverse projection. The Algorithm 1 shows this stage.

Algorithm 1: Estimate coarse position in 3D space

```
Input: $Q, M_f, d, K$
Output: $t_x, t_y, t_z$
// Build probability map
For each pixel in image input $Q$ do
{
        $y \leftarrow$ get pixel's color
        calculate the $p(y|M_f)$
        Store in probability map $Q_p \leftarrow p(y|M_f)$
}
Normalise the probability map $\bar{Q}_p = \frac{Q_p}{\sum p(y|M_f)}$
// Centre of object computation
Compute the object's centre $(u, v)$ by
{
        Compute weighted mean in $x$ direction $u = \sum_{x=1}^{N\ of\ pixels} \bar{p}(y|M_f).x$
        Compute weighted mean in y direction $v = \sum_{x=1}^{N\ of\ pixels} \bar{p}(y|M_f).y$
}
// Inverse projection to get object position in 3D space
Compute the 3D location of object $\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}.d$
```

The output of this stage can only produce a rough estimate of the position without any information about the object's orientation. To build a full pose estimate that consists of both the position and orientation, the next stage requires to create a set of possible orientations. The set of possible orientations is generated structurally by utilizing icosphere geometry. The orientations are generated by assuming cameras are located on the vertices, pointing toward the centre of the sphere. Having coarse estimates of the position and the set of orientations, a set of full pose estimates can then be built. This set of poses then need to be refined by executing multiple PWP3D-NAG. This process is illustrated in Figure 5.7.

The next step is selecting the best pose estimate among all the refined poses and then improving its accuracy by implementing edge-based pose estimation built on Particle Filtering (PF). As the PF can accept multiple inputs, therefore all of the results from multiple initialisation PWP3D can be directly passed to PF as particles. The PF refines the pose estimation and then the pose with best fitness is selected as the final result of this initialisation stage.

The block diagram of the edge-base pose estimation using particle filter is shown in Figure 5.8

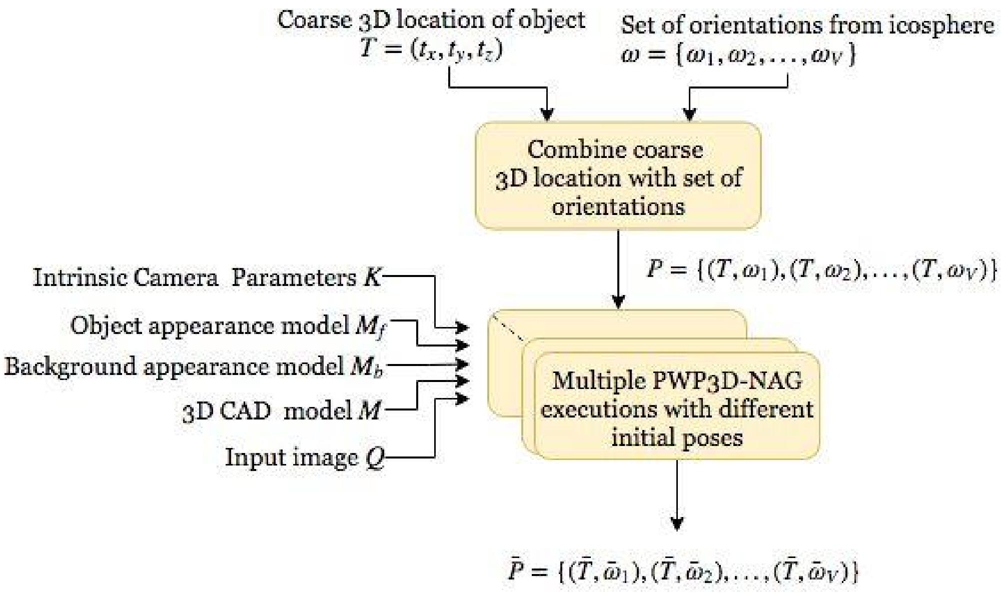

Coarse 3D location of object
$T = (t_x, t_y, t_z)$
Set of orientations from icosphere
$\omega = \{\omega_1, \omega_2, \dots, \omega_V\}$
Combine coarse 3D location with set of orientations
$P = \{(T,\omega_1),(T,\omega_2),\dots,(T,\omega_V)\}$
Intrinsic Camera Parameters K
Object appearance model M_f
Background appearance model M_b
3D CAD model M
Input image Q
Multiple PWP3D-NAG executions with different initial poses
$\bar{P} = \{(\bar{T},\bar{\omega}_1),(\bar{T},\bar{\omega}_2),\dots,(\bar{T},\bar{\omega}_V)\}$

Figure 5.7. Given a coarse position T of object and set of orientation ω a set of poses P is built. Each of the poses within this set then is refined by executing multiple PWP3D-NAG and resulting in a refined set of pose estimates $\bar{P}$.

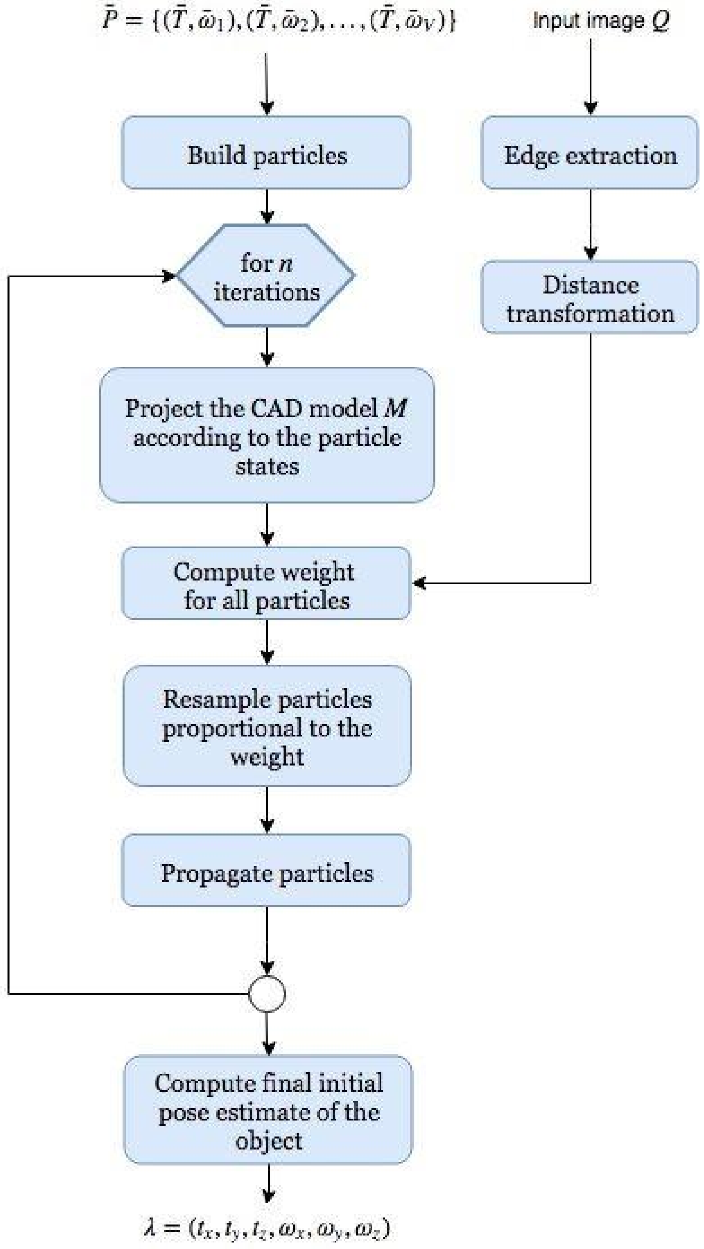

$\bar{P} = \{(\bar{T},\bar{\omega}_1),(\bar{T},\bar{\omega}_2),\ldots,(\bar{T},\bar{\omega}_V)\}$
Input image Q
Build particles
Edge extraction
for n iterations
Distance transformation
Project the CAD model M according to the particle states
Compute weight for all particles
Resample particles proportional to the weight
Propagate particles
Compute final initial pose estimate of the object
$\lambda = (t_x, t_y, t_z, \omega_x, \omega_y, \omega_z)$

Figure 5.8. Block diagram of edge-based pose estimation on particle filtering framework. The result of this stage becomes the final pose of the proposed framework.

5.4 Experiments and Results

Some experiments were carried out with the main objective to investigate the capability of the proposed framework to recover the pose of an object given an input image Q, statistical appearance model M_f, M_b and a CAD model M. The experiments were also aimed to assess the accuracy of the estimation and moreover, the validation was done several times with different objects in different poses to assess the generality of the proposed framework. As the framework consists of a number of stages, the validation was investigated in each of the stages as follows:

- Coarse 3D location estimation

  The main goal in this stage is to provide coarse estimation of the object's position in 3D space by inverse projection of the 2D centre of the object in image plane. Therefore, the validation was done by assessing the estimation accuracy both in 2D image plane as well as in 3D Cartesian space. In image plane, the verification was done by measuring the error between the 2D estimated centres of the object to the true projected centre of the object. The estimated centre of the object was obtained from the weighted mean of the probability map.

  In 3D Cartesian space, the verification was done by measuring the error between the 3D centre of the object achieved from inverse projection to the true position gained from Aruco marker.

- Orientation generator from structural icosahedron geometry

  At this stage, the verification was mainly to assess the strategy of using structured icosahedron geometry for generating multiple orientations that further will be refined by PWP3D-NAG algorithm. In this part of the experiment, some projection of the object on the generated orientation will be presented.

- Multiple PWP3D-NAG

  The purpose of this stage is to refine the pose estimates obtained from previous orientation generator. Therefore, the validation was done by comparing the error before and after PWP3D-NAG executions and the output was also benchmarked to the ground truth.

- Particle Filtering

  The final stage of the framework is Particle filtering with the aim to accept multiple hypotheses input and process it to provide final result. The validation was done by comparing the output to the true pose. The behaviour of the particle filtering in estimating 3D pose of object was also investigated and some performance indicators such as the number of iterations required to converge and some other indicators will be presented.

5.4.1 The Experiment Setup

The selected objects for the experiments were red box and soft-drink can. These two objects were chosen to represent poorly-textured objects and also for representing objects with some level of symmetry. Two datasets were created for the experiment that consist of recorded freehand motion video around the objects. The objects were placed carefully on the top of Aruco marker map sheet so the axis of the objects were aligned to the axis of Aruco marker map. Given this setup, the position and orientation of the object can be retrieved from the Aruco marker map. For each of the recorded videos, two frames that show the extreme different pose of the objects were selected randomly and it became the test input for the initialisation framework. The setup and the selected frames, along with the Aruco marker pose are shown in Figure 5.9

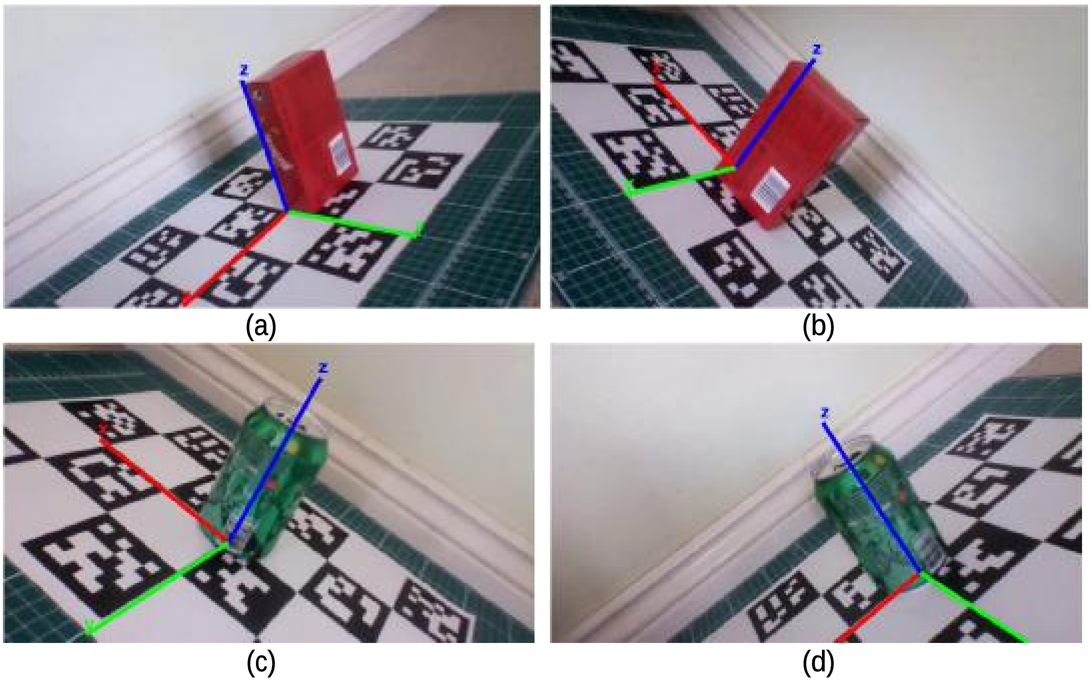


Figure 5.9. The experiment setup where the objects were placed on the top of the Aruco marker map and the axes of objects were aligned to the axes of the Aruco. By having this setup, the pose of the objects can be retrieved from the pose of Aruco marker map. The Aruco pose is shown from the plot of the axes. For each dataset, two frames with extreme different pose of object were randomly selected and became the input for the experiment. The selected inputs for this experiment were frames 579 and 972 of red-box dataset and frame 1120 and 1721 of soft-drink can dataset.

The initial orientations for multiple PWP3D-NAG were obtained from hemisphere vertices locations, pointing toward the centre of the icosphere. The number of vertices was 73. The rotation around z axis ω_z and y axis ω_y were obtained directly from the azimuth and elevation angles respectively given the 3D position of each vertex. The rotation around the x axis, ω_x, was obtained by assuming the camera rolled in angles -45°, 0° and 45°. Therefore the number of initial orientations was 219 (73 x 3). The PWP3D-NAG was executed 219 times to refine each of the initial guesses and generate a set of refined-hypotheses for the next stage (particle filtering stage).

The particle filtering stage utilized 219 particles, with the state $X = \{t_x, t_y, t_z, \omega_x, \omega_y, \omega_z\}$ and the initial state was obtained from multiple PWP3D-NAG outputs. The propagation stage implemented Gaussian random walk and the likelihood measurement was done using normalised sum to the nearest edge, obtained from distance transform map. Better fitness is indicated by lower score of the normalised sum of distance. The number of iterations of each particle filtering execution was 300 and to gather statistical data, the experiment for each input was repeated 20 times.

The experiment was done on a computer with Intel Core i5 running at 3.30GHz with 4Gb RAM and the graphical processor unit was NVIDIA Quadro K620 GPU. The code was written in C++ and executed under Ubuntu 14.04 LTS. The recorded data was further analysed using Matlab.

5.4.2 Coarse 3D Pose Estimation

The first stage of the framework is coarse 3D pose estimation that requires an input image Q, the object's statistical appearance model M_f and a rough distance guess of the object from camera d (in meters). The rough distance guess is needed as depth information cannot be retrieved from a single frame image, whereas in inverse projection, this information is needed. The distance guess does not need to be accurate, as it will be refined later. In this experiment a fixed distance guess at 0.6 m was used for all experiments. Considering the true distance was about 40-50 cm, the 60 cm distance guess is reasonably inaccurate especially when considering the objects were small (red-box dimension is 14 cm x 8 cm x 3.5 cm and soft-drink can dimension is 12cm height with 7 cm diameter). This is selected to investigate how good the final output of the framework is in dealing with inaccurate input.

Given the object's model M_f and input image Q, the first step is to find the 2D centre of the object in the image plane (u, v). The centre of the object is obtained by the weighted mean of the probability map. The probability map Q_P indicates the possibility of each pixel being part of the object by observing its probability of color y given the object's model $p(y|M_f)$. The equation for calculating this probability is Equation 4.1.

Figure 5.10 shows the probability map with red color indicating the pixel has higher $p(y|M_f)$ score and blue color represent low probability score. The estimated geometrical centre of the object computed from weighted mean is also shown along with the true geometrical centre projection of the object. The true geometrical projection was obtained by projecting the point according to the Aruco pose.

From visual observation, Figure 5.10 shows that the mean average (indicated by white + marker) did not lie on the exact centre of object (indicated by yellow ⊗ marker) since the estimation was also affected by the background pixels that falsely classified as object pixels. This is confirmed by the measurement of both centres as given by Table 5.1

Table 5.1 Estimated centre of object projection in image plane

Frame	Estimated		Ground Truth		Error (pixels)	
	u	v	u	v	u	v
Red-box frame 1	351.95	174.06	360.35	151.32	-8.4	22.74
Red-box frame 2	288.66	169.07	283.91	169.60	4.75	-0.53
Soft-drink can frame 1	290.64	171.75	322.96	148.27	-32.32	23.48
Soft-drink can frame 2	384.81	209.05	387.06	193.19	-2.25	15.86

However, the errors between the estimated centre of the object with true object's centre were small with maximal 32.32 pixels in u direction. With the image width of 640 pixels the percentage of error is 5.05%. In v direction with image height 360 pixels, the maximum error at 23.48 pixels is equal to 6.5%. As the aim is to find coarse estimate of the object, this result demonstrated that the strategy for finding centre object from weighted average was successful.

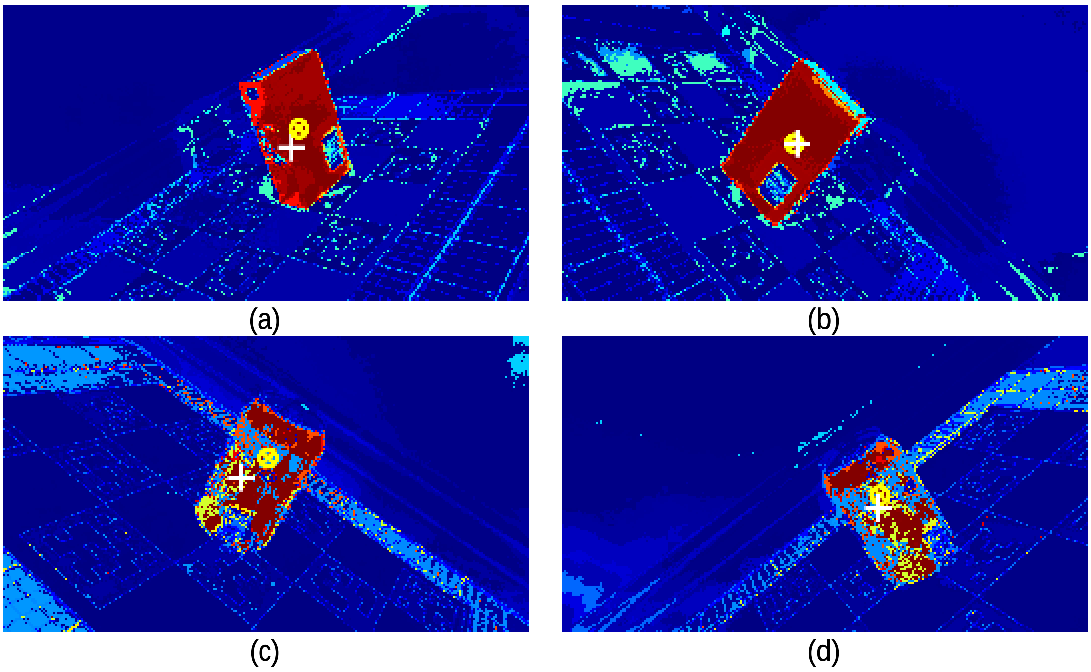


Figure 5.10. The object's probability map of the input frames. The red coloured pixels represent high probability of the pixel being part of the object and blue coloured pixels represent low probability. The estimated object's centre is presented by white + marker while the true projected geometrical centre is presented by ⊗ marker.

The next stage is to compute the 3D object pose given the estimated centre and the distance guess. To be able to calculate the inverse projection, an intrinsic camera parameter is required. The camera that was used for the experiment was the front Parrot ARDrone camera with image resolution of 640 x 360 pixels. The intrinsic camera parameters were obtained using camera calibration procedure using OpenCV and the obtained K matrix was:

$$K = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 557.50 & 0 & 320.26 \\ 0 & 559.71 & 187.61 \\ 0 & 0 & 1 \end{bmatrix}$$

Given the estimated centre of object (u, v), the distance guess $d = 0.6\,m$, the intrinsic camera parameters K, the centre of objects location in 3D Cartesian space was calculated using the inverse projection

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} . d$$

The results of the inverse projection given the estimated object's centre are presented in Table 5.2 along with the corresponding errors.

Table 5.2 The coarse 3D estimated location compared to the true position.

Dataset	Estimated Position			True Position			Error		
	t_x	t_y	t_z	t_x	t_y	t_z	t_x	t_y	t_z
Red-box frame 579	0.0341	-0.0145	0.6	0.012	0.052	0.503	0.0221	-0.0665	0.097
Red-box frame 972	-0.0340	-0.0199	0.6	-0.0922	0.0054	0.5161	0.0582	-0.0253	0.0839
Soft-drink can frame 1120	-0.0319	-0.0170	0.6	-0.0290	0.0291	0.377	-0.0029	-0.0461	0.223
Soft-drink can frame 1721	0.0695	0.0230	0.6	0.0860	0.0605	0.383	-0.0165	-0.0375	0.217

Table 5.2 shows that the position estimate errors are large and need to be refined in the next stage. For qualitative measurement, Figure 5.11 presents the projection of this very first stage in the framework.

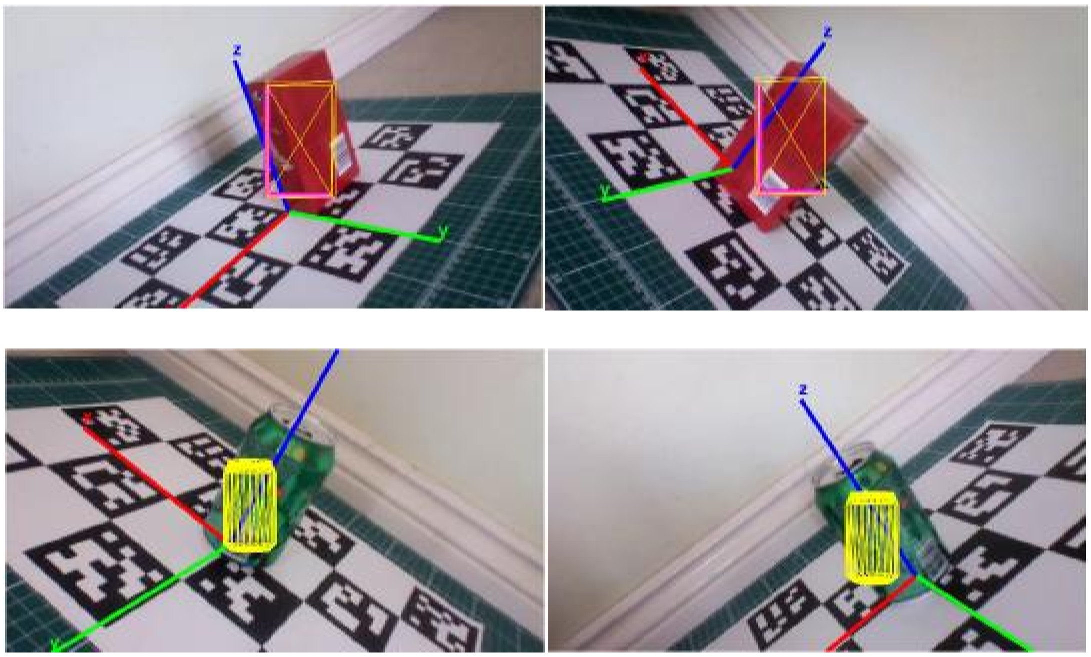


Figure 5.11. Pose estimation output from the first stage. It shows a large error both on position as well as orientation.

5.4.3 Orientation Generator from Structural Icosahedron Geometry

The output from the previous stage is just capable to provide a rough estimate of the position only without any orientation. To obtain a complete pose estimate that consists of both position and orientation, in a better accuracy, a PWP3D-NAG is needed. PWP3D-NAG requires an initial position and orientation as its input. This initial position can be obtained from the previous stage but not for the orientation as the previous stage does not have any information about the orientation. A naive approach can be assuming the orientation as $\omega_x = 0, \omega_y = 0, \omega_z = 0$ or the object does not rotate anywhere. However, PWP3D-NAG algorithm implements Nesterov's Accelerated Gradient descent, this algorithm can only converge to local optimum. Setting to a single particular initial orientation will lead the pose estimate to be trapped in local optimum that might have a large error.

A better strategy proposed in the framework that aims to converge to the global optimum or at least to better optimum, is to execute multiple PWP3D-NAG with different initial poses. To increase the probability of converging to better optimum, the different initial orientations were generated structurally to capture some possible viewpoint angles. These different orientations are obtained structurally from icosphere geometry.

By assuming cameras are located on the vertices $V = (V_1, V_2, V_3, \dots, V_V)$ and pointing to the centre of the icosphere, a set of orientations was generated. Assuming the object is static and always facing up, and camera's orientation never experiences an extreme maneuver such as upside down, only half of ball shape is used. The orientation ω_i was computed by converting vertices location in 3D Cartesian coordinate system into spherical coordinate system. The spherical coordinate system has three parameters: radial distance r, polar angle θ and azimuth angle φ. Since the azimuth angle φ defines the angle around z axis in Cartesian space therefore it serves as yaw angle, so $\omega_z \leftarrow \varphi$. The polar angle θ defines the angle between vertical axis z to the line between vertices to the centre of sphere, hence it can be used to compute the pitch angle. Assuming the pitch axis is y axis, therefore $\omega_y = 90 - \theta$. The roll angle ω_x cannot be recovered from spherical coordinate system as the sphere coordinate system actually is a direction vector not a rotation vector. The strategy to deal with this is by assuming three different roll angles on each of the vertices, that is $-45°, 0$ and $45°$.

The icosphere that was used has 73 vertices above the half plane, and since each vertex generates three different orientations, the total number of orientations is 219. The resulting set of orientations can be visualised by projecting the CAD model according to the orientation set. Some examples that represent this orientation set are shown in Figure 5.12. The projection of the generated orientation shows that the proposed strategy is capable to generate multiple orientations that is needed for the framework.

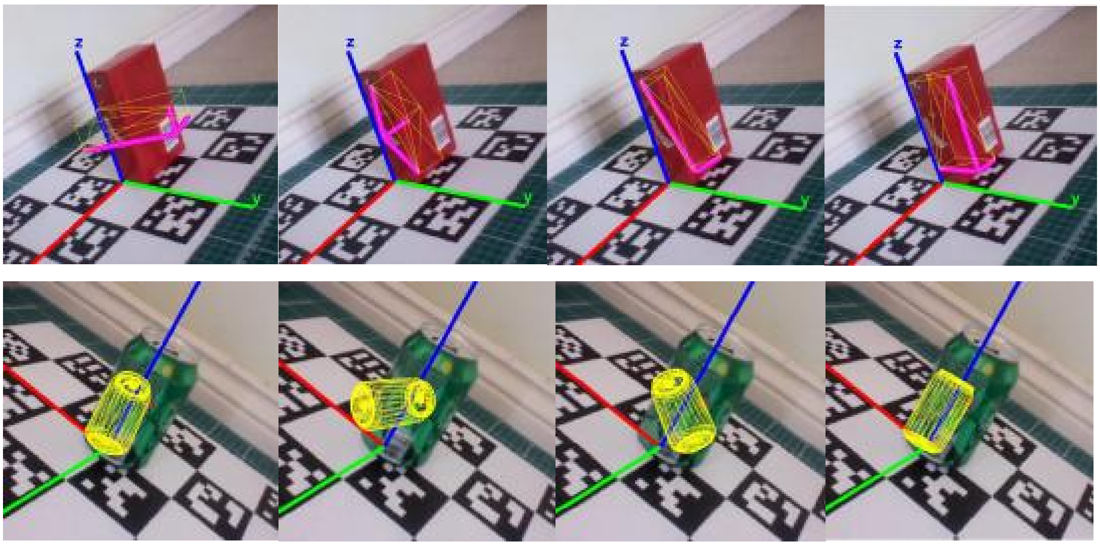
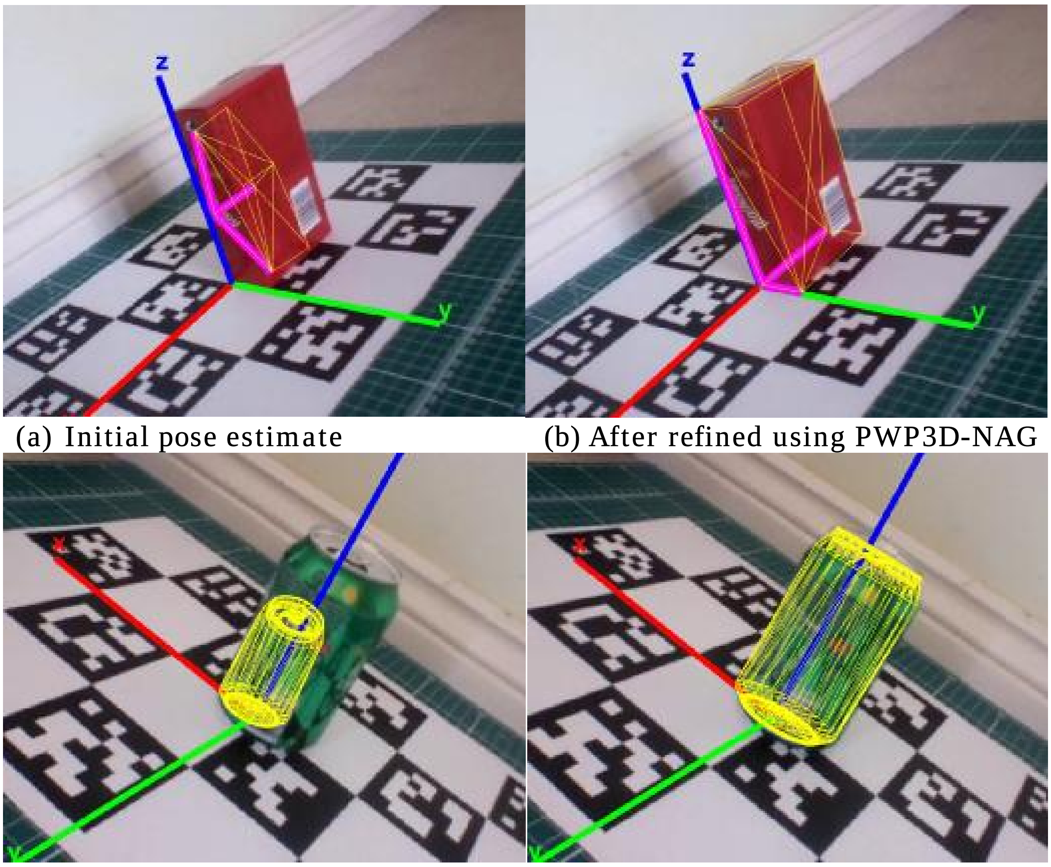


Figure 5.12. Some projection examples of red-box object and soft-drink can given initial position estimate from previous stage and multiple different orientations obtained from vertices location (this stage). These orientations serve as initial attitude that later will be refined.

5.4.4 Multiple PWP3D-NAG Executions

The initial orientation generator yields a set of orientations $R = (\omega_1, \omega_2, \dots, \omega_V)$ where $\omega_i = \left(\omega_x, \omega_y, \omega_z\right)_i$ and along with the coarse 3D position $t = \left(t_x, t_y, t_z\right)$ obtained from the earlier stage a set of 219 initial poses $P = \{(t, \omega_1), (t, \omega_2), \dots, (t, \omega_V)\}$ was built. This initial pose has a large error that needs to be refined using PWP3D-NAG. Given this set of initial poses PWP3D-NAG was executed 219 times with different initial poses. In general, PWP3D-NAG managed to refine the estimation significantly as shown in Figure 5.13.



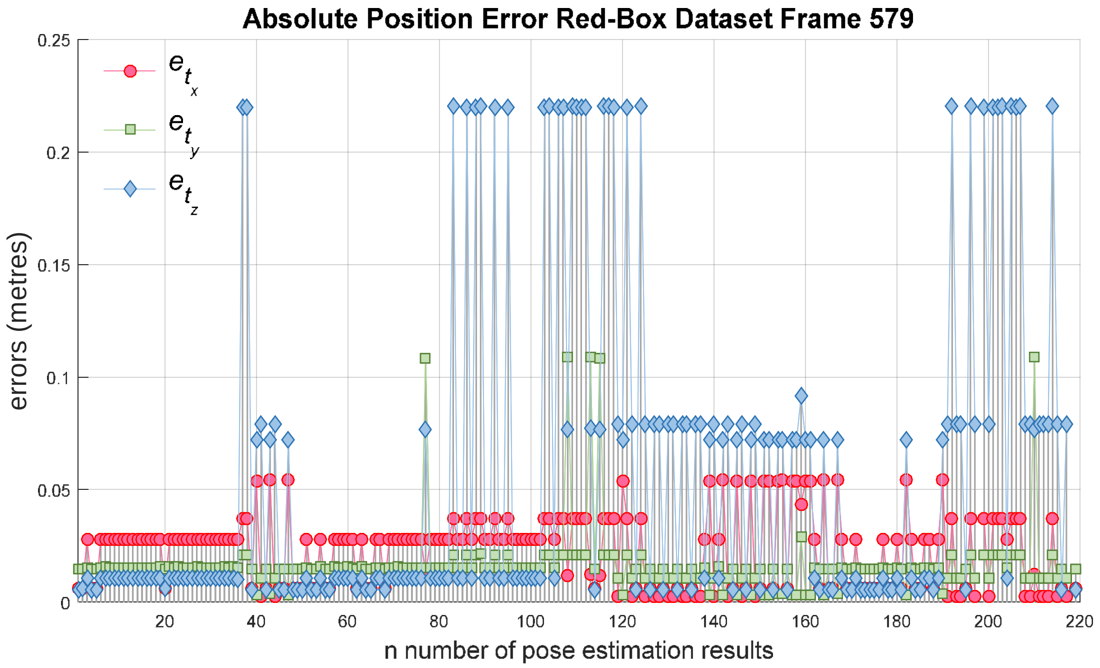
(a) Initial pose estimate (b) After refined using PWP3D-NAG

(c) Initial pose estimate (d) After refined using PWP3D-NAG

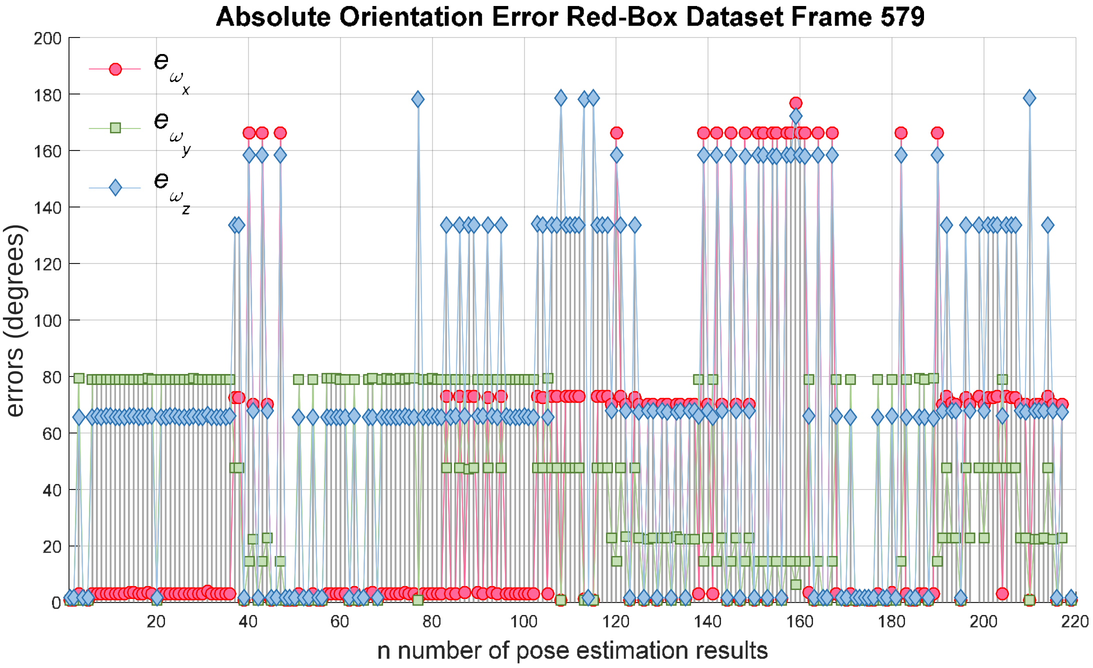
Figure 5.13. Comparing the estimated pose from previous stage (a,c) and after being refined in this stage (b,d). It shows that this stage contributed significantly in reducing errors,

In the experiment, each PWP3D-NAG was executed with 100 iterations. Each of the final estimated poses was then also compared to the true pose provided by Aruco marker. The absolute position and orientation error of red-box dataset is presented in Figure 5.14 for the frame 579 frame and Figure 5.15 for the frame 972.

From the chart it shows that given 219 different initial poses, the red-box dataset converged only to a few optima. Some of them have small errors, whereas other optima suffered from large inaccuracies. The multiple optima are mostly observed due to the symmetrical shape of the red-box, where exactly the same projection shape can be observed when the object is flipped with respect to some planes. For the red-box object, there will be 4 main optima that cannot be distinguished from the shape of the projection silhouette, such object position flipped to the xy plane and yz plane. This behaviour confirmed the fundamental limitation of vision-only pose estimation which is known as multimodal projection problem. This condition can be observed visually in Figure 5.16.

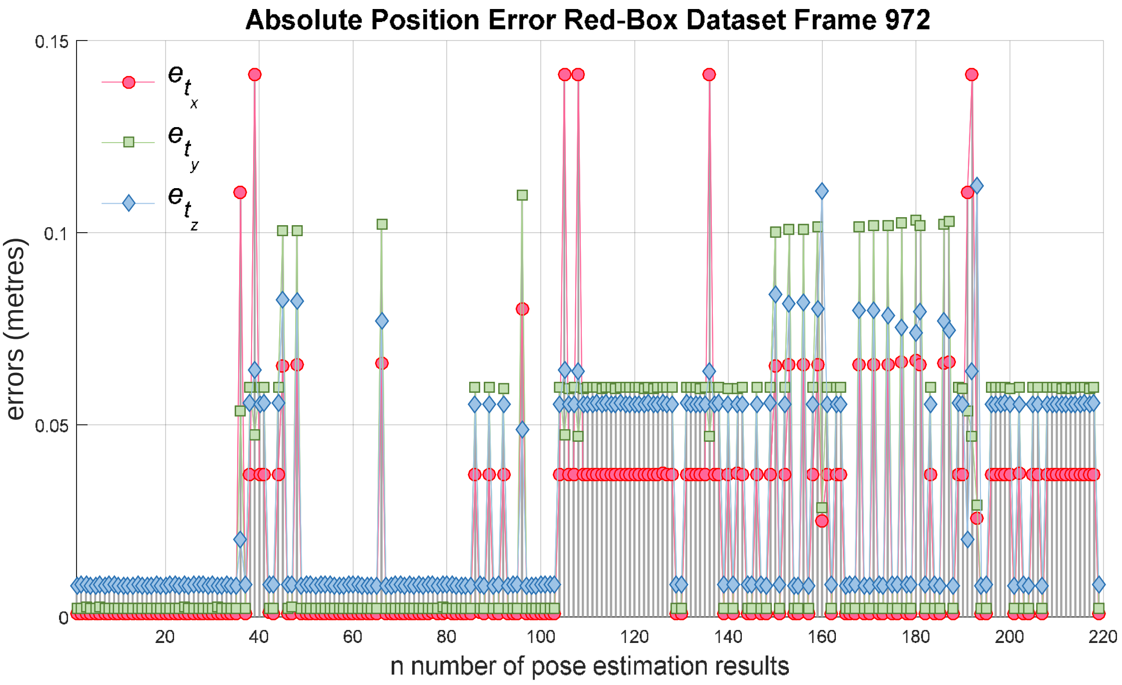


(a)

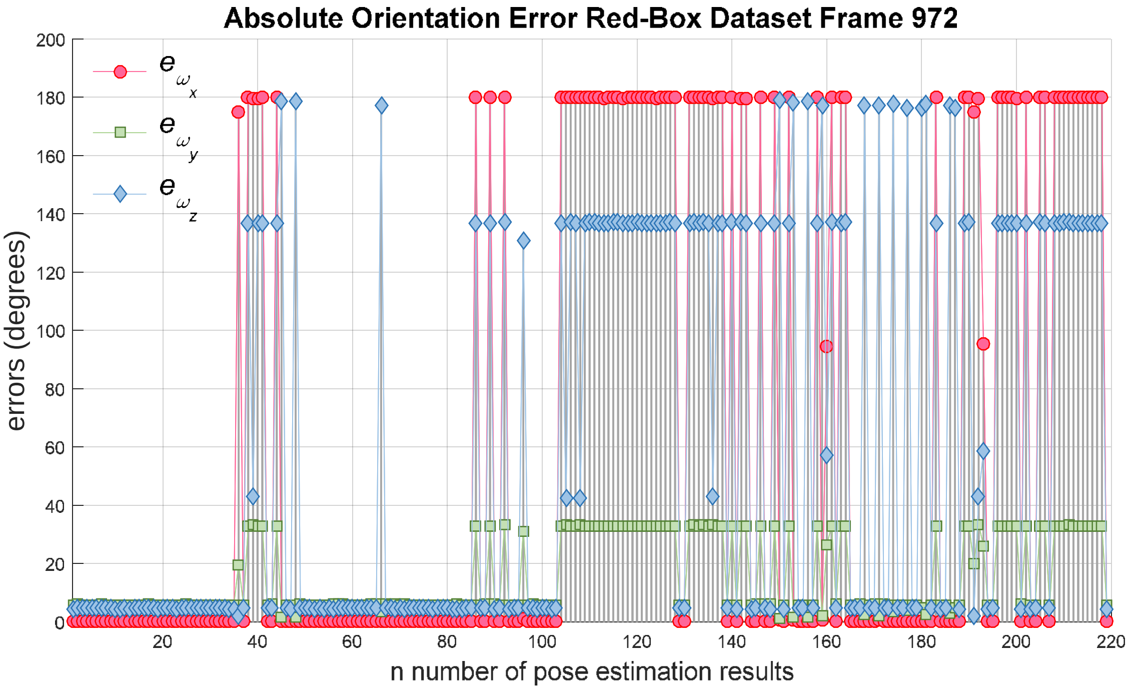


(b)

Figure 5.14. Plot of absolute position error (a) and absolute orientation error (b) of red-box dataset, frame no 579. It shows the output converged in a few poses despite the input was 219 different poses. The convergence to only a few poses was observed due to the red-box being symmetrical in xy and yz plane.

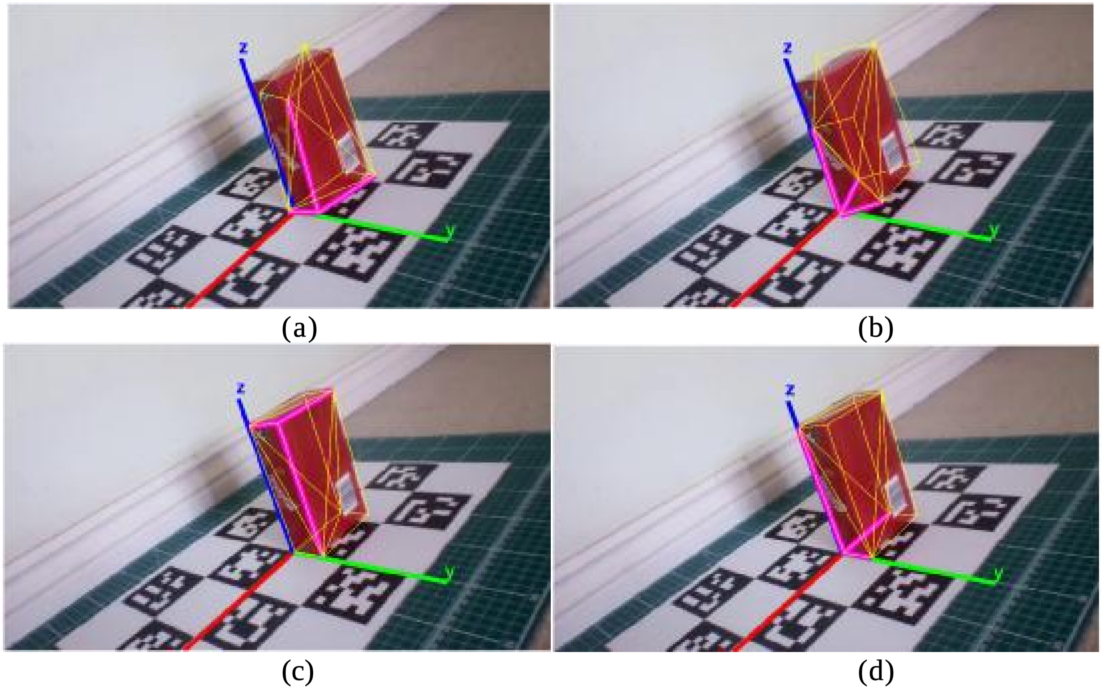


(a)



(b)

Figure 5.15. Plot of absolute position error (a) and absolute orientation error (b) of red-box dataset, frame no 972. Similar to the frame 579, it shows the output converged in a few poses. Both of the frames have the same object hence a similar behaviour was observed.

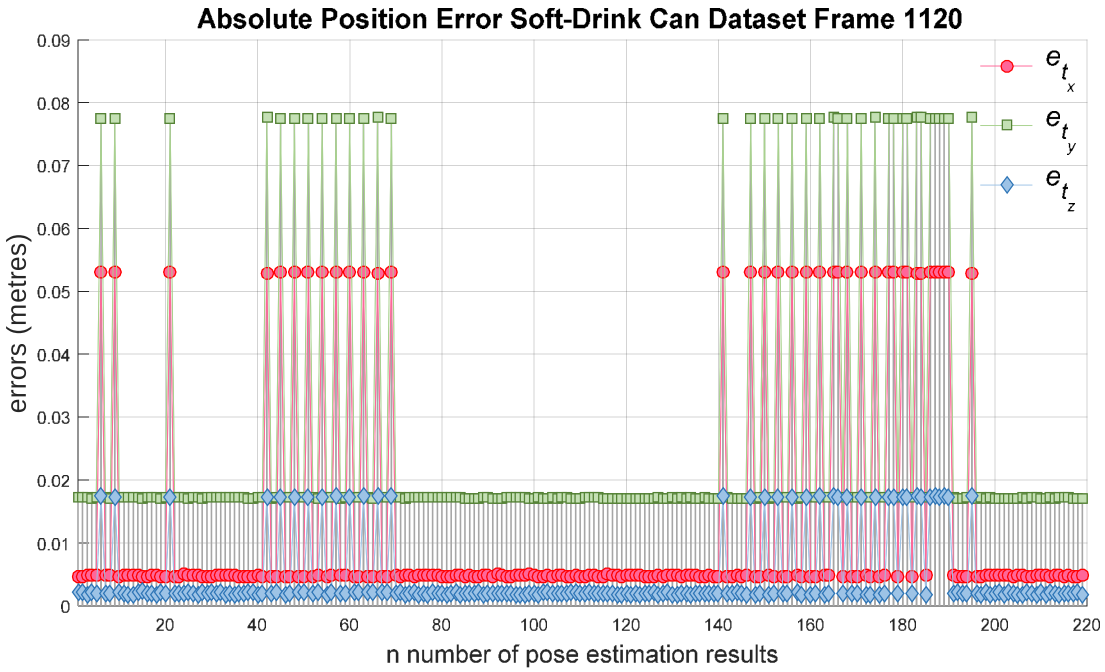


(a) (b)

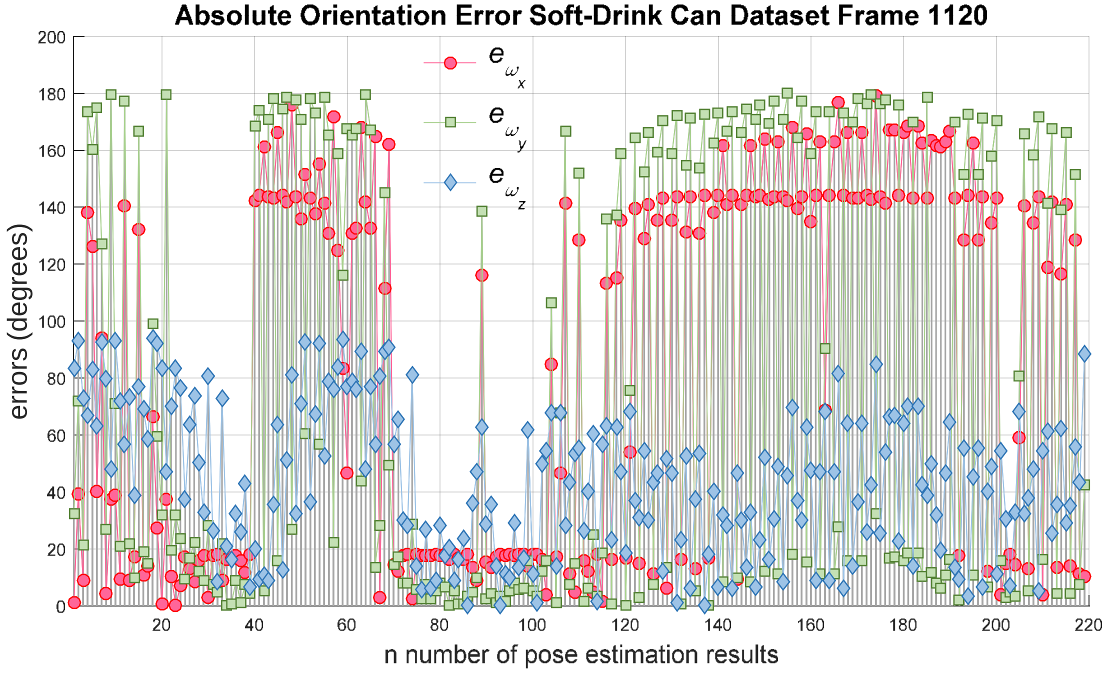
(c) (d)

Figure 5.16. Some results of multiple PWP3D-NAG executions from 31st initial pose (a), 43rd initial pose (b), 77th initial pose (c) and 173rd initial pose (d). They show that some outputs managed to provide good estimate (d) but some other results show large incorrect orientation estimate (a) and (b). An interesting result also shows in (c) where the pose estimate converged to an upside down position. This happened since the object was symmetrical in xy plane and yz plane.

A different result was observed for the soft-drink can output as shown in Figure 5.17 and 5.18. From the position error plot it shows the pose mainly converged to two positions. These can be explained as the soft-drink can converged to the correct standing pose and to upside down pose as shown in Figure 5.19. However, while the position error shows a more regular pattern, the plot of the orientation error shows a different behaviour. The orientation error shows a wide variety of results. This happened as the soft-drink can is symmetrical in the z axis, hence any rotation around the z axis gives the same projection shape so it has infinite solutions. In this case, PWP3D-NAG converged to more variety optima and cannot recover the ω_z angle.

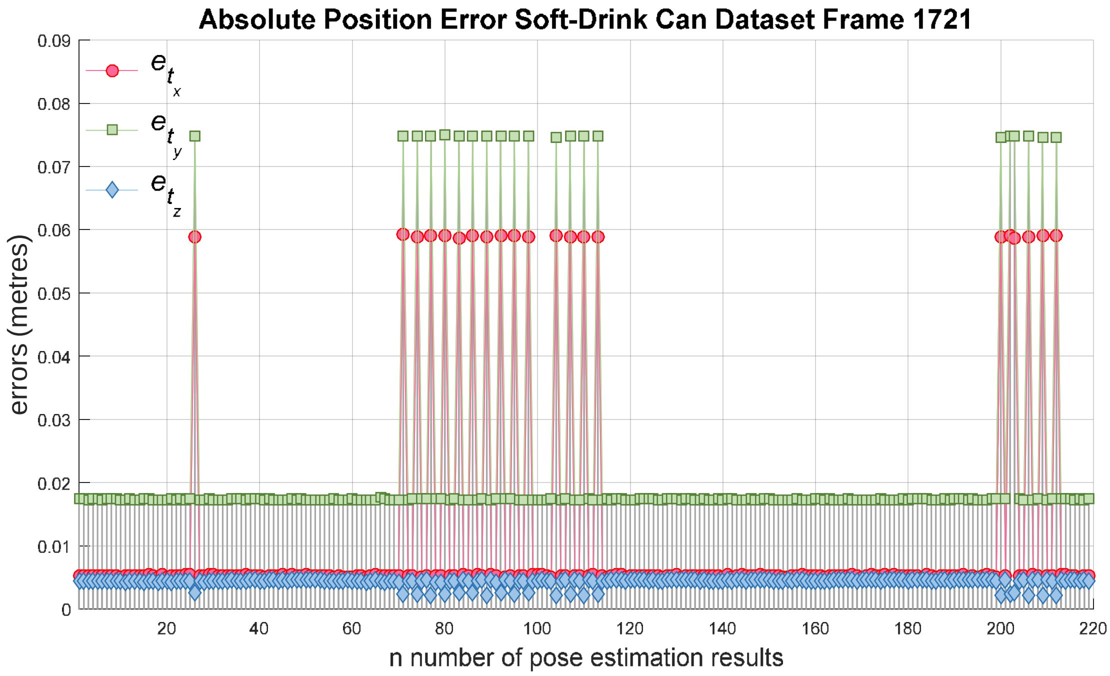


(a)

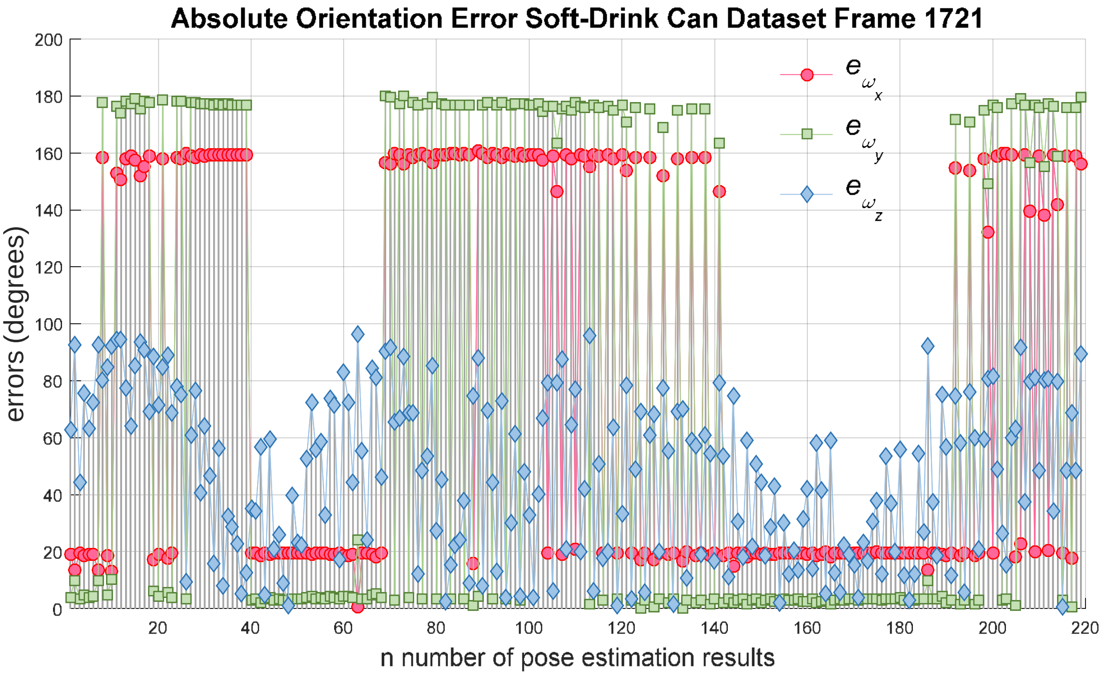


(b)

Figure 5.17. Absolute position error (a) and absolute orientation error (b) of soft-drink can dataset, frame 1120. Given 219 different initial poses, the position estimate converged to 2 different poses that can be explained as the correct standing position and a flipped down position. However, the orientation suffered from a wide variety of errors due to multimodal projection problem. In this case, the soft-drink can is symmetrical in z axis, hence ω_z cannot be retrieved based on the shape projection silhouette.



(a)



(b)

Figure 5.18. Absolute position error (a) and absolute orientation error (b) of soft-drink can dataset, frame 1721. A similar result as observed in frame 1120 shows the soft-drink can converged to 2 main positions: correct standing up position and upside down position. The orientation still experienced large error as the ω_z cannot be retrieved based on the shape projection silhouette.

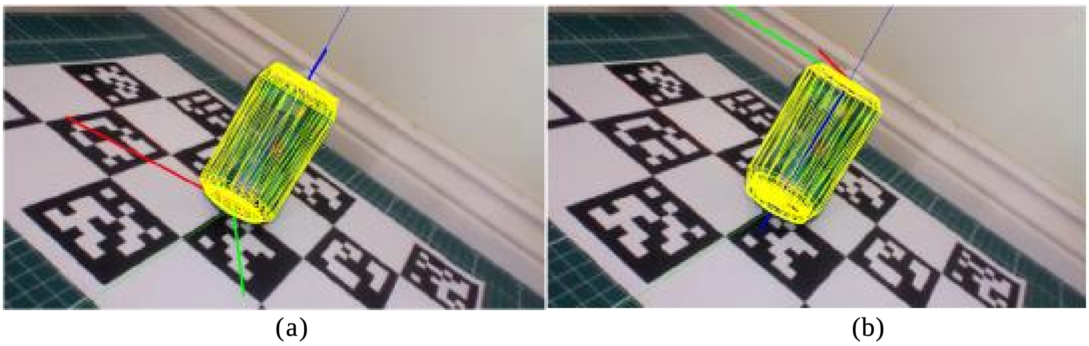


Figure 5.19. Example of soft-drink can dataset result. It shows the PWP3D-NAG converged to two main poses, that is correct standing pose (a) and upside down pose (b). However, when the pose estimate converged to the correct standing pose, it still suffered from error as can be observed from (a).

From the results of the red-box and soft-drink datasets that managed to converge to main optima points, it shows that structured orientation generator was effective as it managed to approach main optima points. At the same time, this experiment also demonstrated that multiple PWP3D-NAG execution strategy managed to refine the pose estimate and generates some better hypotheses of object’s pose.

Another interesting behaviour is the different accuracy of the estimation results given different objects also observed from this experiment. Comparing the best pose estimate of both datasets it showed that the red-box had a better pose estimation accuracy than the soft-drink can. This can be explained as the input for the PWP3D-NAG is a color histogram, the color dissimilarity between object and its background has a significant influence on the estimation result. The posterior map of the red box shows a clear distinguishable foreground-background region, along with a sharp edge making the red-box estimation resulting a lower error as presented in Figure 5.20.

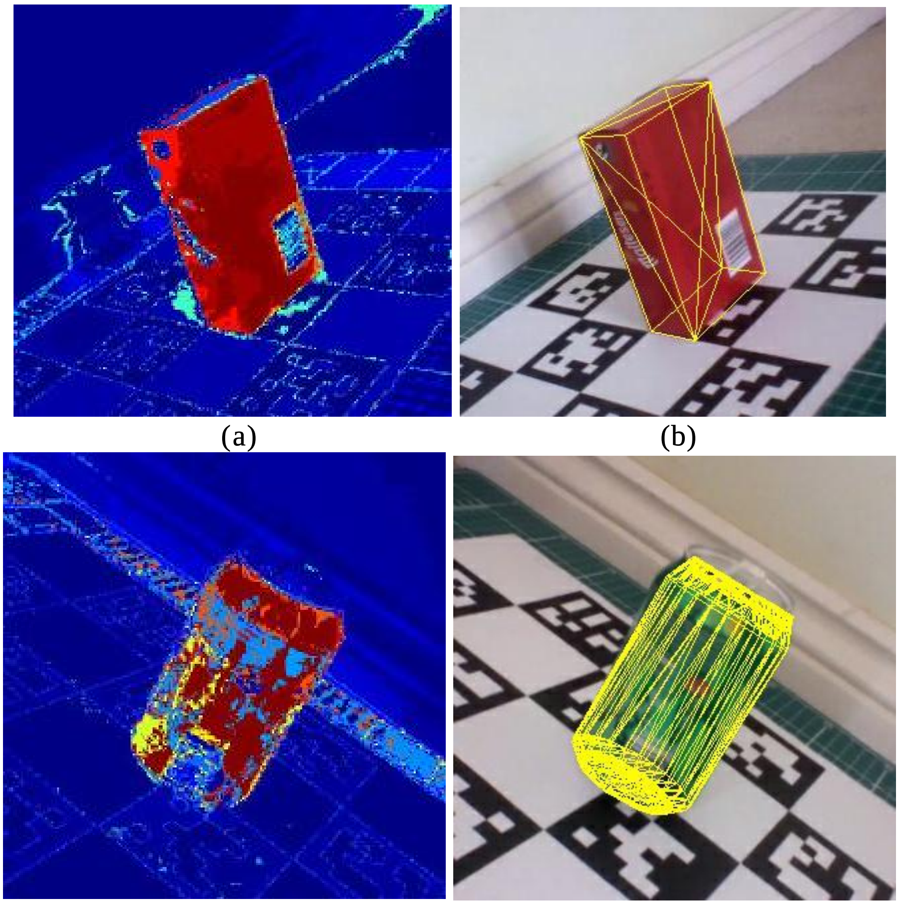


Figure 5.20. Different accuracy in pose estimation result. As the PWP3D-NAG estimates the pose based on color information, a similar color between foreground and background becomes the limit of this algorithm. The red-box dataset shows a pose estimate with low error (b) as the input was reasonably good (a). The pose estimate of soft-drink dataset (d) shows a large error as the top part of the object was falsely classified as part of the background area as shown in (c). In this case the PWP3D-NAG already reached the optimum but due to bad input the pose estimation was not good.

In contrast with the red-box result, the soft-drink can dataset result as shown in Figure 5.20 (c), the top part of the can has a silver color that is not easily distinguishable from the background. The probability map confirmed this by showing that the top part had a low probability score of being part of the object and it was falsely classified as part of the background region. Due to this condition, the PWP3D-NAG tried to avoid this part during the optimisation. From this visual observation, it shows the PWP3D-NAG managed to arrive to the good optimum as the output pose provided a good segmentation between the foreground-background. The estimation error itself came from the bad input, and for this case the PWP3D-NAG cannot refine it further. A similar color between object and the background becomes the fundamental limit of the algorithm.

The experiment demonstrated that the multiple PWP3D-NAG managed to refine the pose estimate significantly and managed to provide multiple hypotheses required for the next process. Therefore, the multiple PWP3D-NAG stage is successful and it works as expected.

5.4.5 Edge-Based Particle Filter Pose Estimation Method

The multiple PWP3D-NAG executions managed to provide some hypotheses of the object's pose. However, as presented in Section 5.3 these multiple hypotheses are needed to be selected and refined to obtain the final pose estimate. The process of selecting the best pose among all of the available hypotheses and also the requirement to improve the accuracies is done by using edge-based pose estimation based on particle filtering approach.

The experiment was done by inputting all of the refined 219 hypotheses as the initial particle states of the object, and afterward, the particle filter was executed. The plot of initial state of the particles is presented in Figure 5.21 and 5.22.

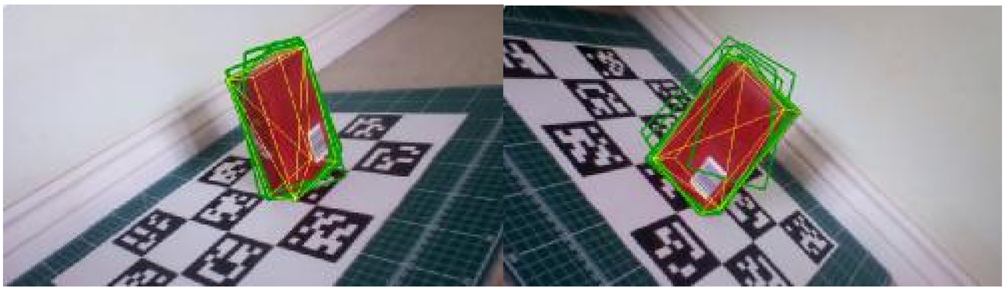


Figure 5.21. The projection of the initial particle states obtained from the output of multiple PWP3D-NAG executions. It shows that some particles were spread widely. The particle with best fitness is plotted in yellow and it shows already has a good pose estimate. This confirmed the error measurement given in Figure 5.12 and Figure 5.13 that showing some of the PWP3D-NAG outputs already resulting in a good pose estimate.

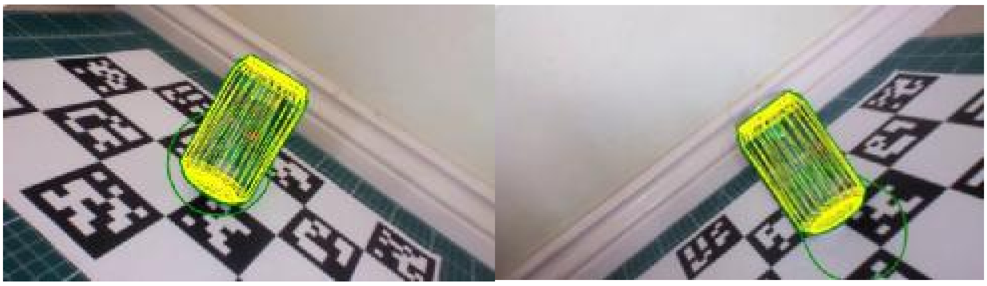
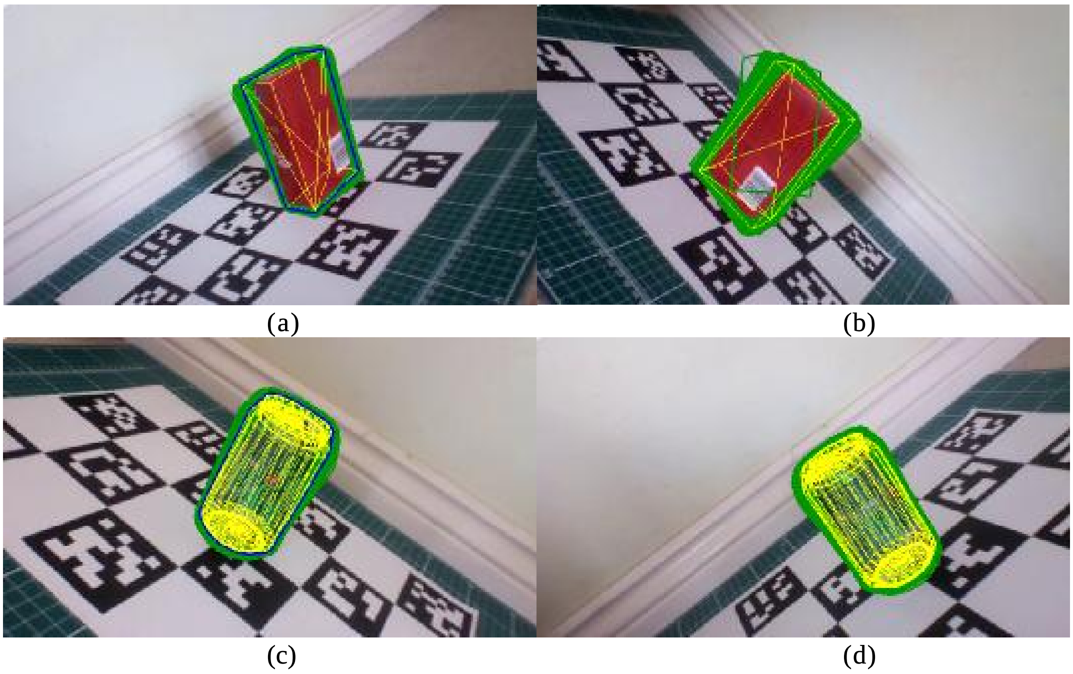


Figure 5.22. The projection of initial particle states on the soft-drink can dataset before the particle filter was executed. As can be seen most of the initial particles are already close to each other except only for a few particles. A few particles estimated the top of the soft-drink can in a direction toward the camera as shown as circle projections.

Given these initial states, the particle filter algorithm was then executed and as can be seen from Figure 5.23 the particle spread was then closer, converging to a better pose estimate. The finesses of the particles were measured from the sum of the distance to the extracted edge. Instead of calculating one by one exhaustively, the distance was obtained efficiently by performing distance transform map on the extracted edge image. The extracted image input and the distance transform map can be seen in Figure 5.24.

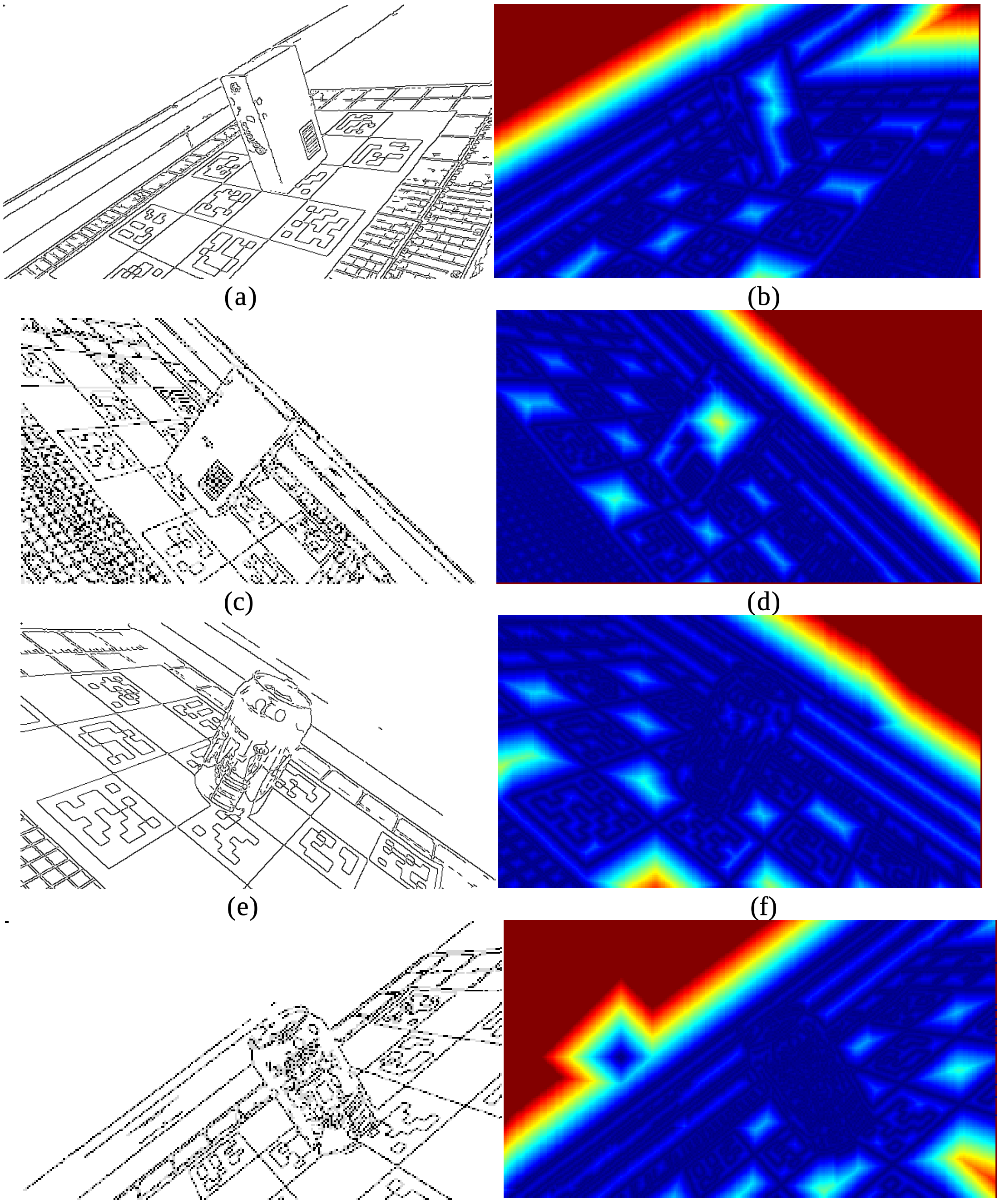

(a)

(b)

(c)

(d)

Figure 5.23. The plot of particle's state during converging time to the particular pose estimate of red-box and soft-drink can datasets. It shows the particle's state were close to each other as it tries to align to the extracted edge. The best particle is plotted in yellow color.



(a) (b)

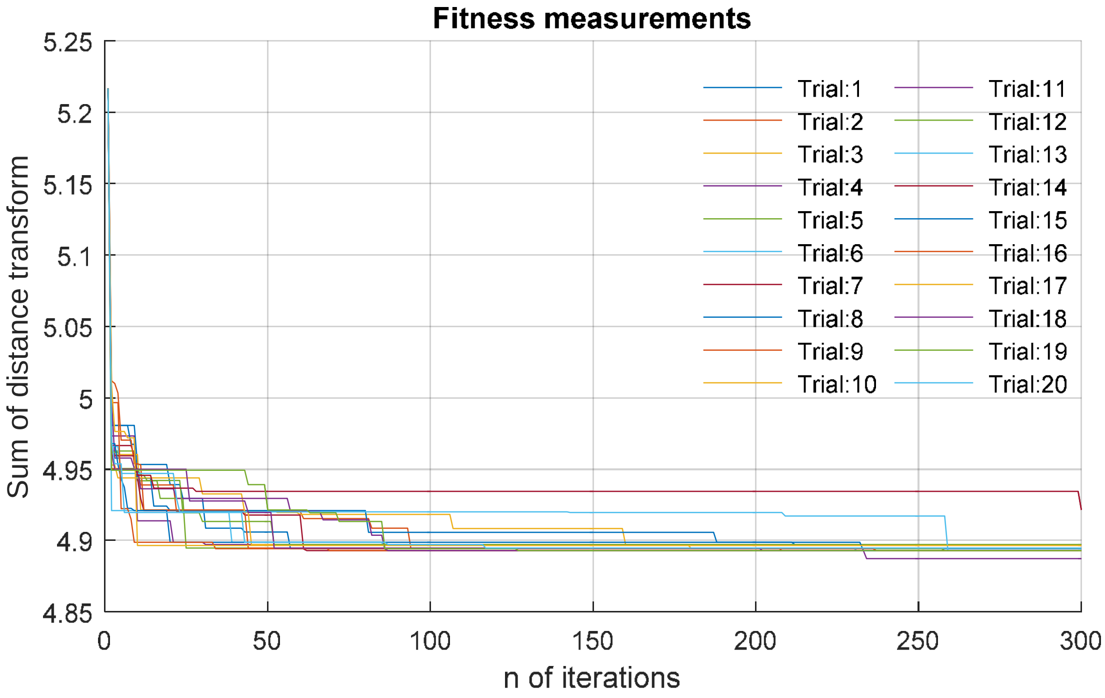
(c) (d)

(e) (f)

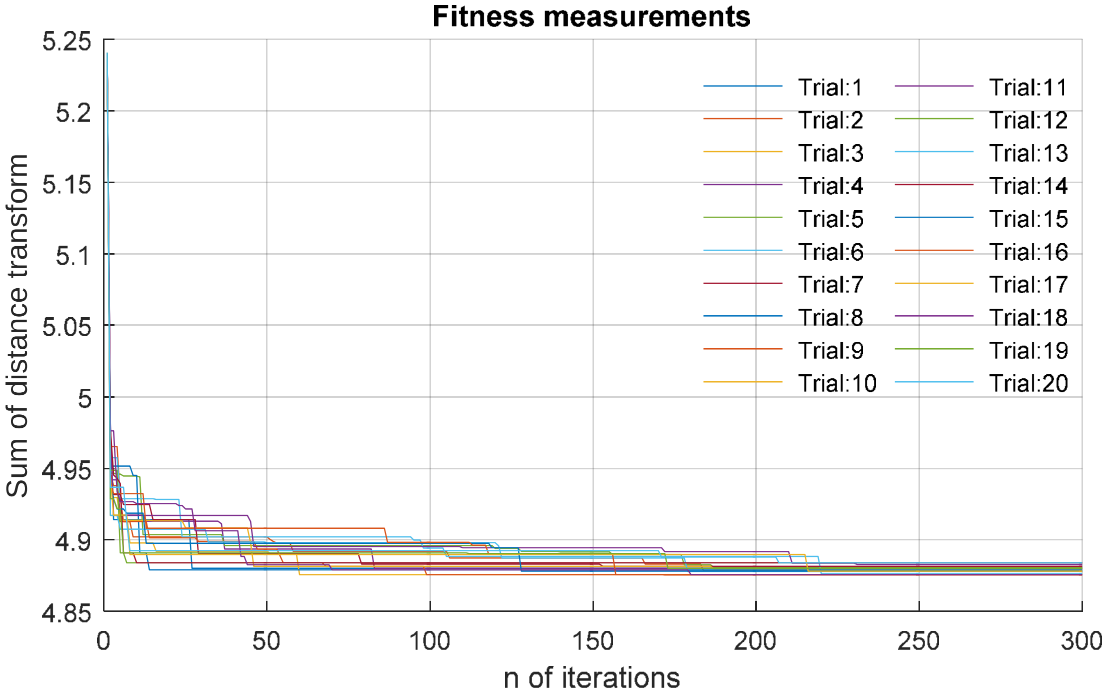
(g) (h)

Figure 5.24. The fitness measurement in particle filtering pose estimation was done by using the distance transform. The distance transform maps the distance of each pixel to the nearest detected edge. The input for the distance map is the edge image as shown in (a) and (c) for red-box dataset and in (e, g) for soft-drink can dataset. The corresponding distance map is shown in (b, d) for the red-box dataset and in (f, h) for the soft-drink can dataset.

To investigate the behaviour of the particle filter, 20 experiments were conducted for each of the frames. The outputs were recorded and then analysed. The plot of fitness measurement for each of the iterations can be seen in Figure 5.25.



(a)



(b)

Figure 5.25. The fitness evolution of red-box dataset during the particle filter iterations for frame 579 (a) and 972 (b). The total number of iterations was 300 and the experiment was repeated for 20 trials. In the first few iterations it was observed a very steep improvement of the fitness score for all trials. After these very large refinement the fitness keep improving until converged.

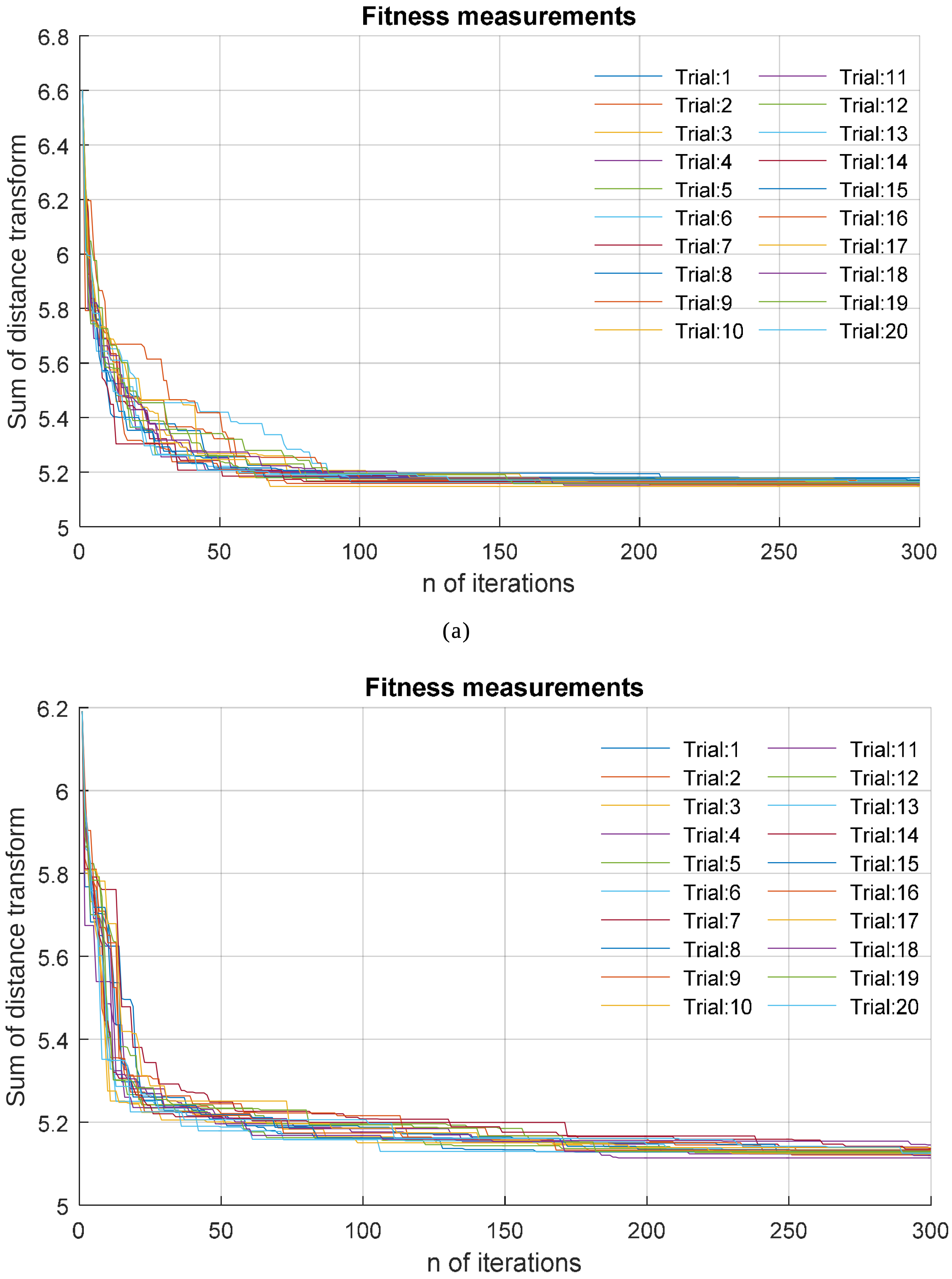
An analysis is done by taking the average and standard deviation of the convergence iterations for all of the trials. The number of iterations required to converge is computed when the observed fitness reaches and stays within 5% tolerance of the final fitness value. The average and standard deviation of convergence iteration is presented in Table 5.3

Table 5.3 Statistical measurement of the particle filter's convergence iteration for both datasets.

Dataset	Mean (iterations)	Standard Deviation
Red-box frame 579	104.9	89.3
Red-box frame 972	113.7	73.5
Soft-drink can frame 1120	101.0	37.9
Soft-drink can frame 1721	163.5	44.7

As presented in Table 5.3 the particle filter converged in less than 200 iterations. The red-box dataset fitness evolution observed an interesting behaviour. Whilst it also has mean of about 100 iterations, it shows a wide variation in the convergence time. In some trials, such as trial number 10 it converged in 5 iterations while trial 4 converged after 250 iterations. This behaviour is validated from the standard deviation that shows much higher score than the soft-drink can dataset. This result shows that the red-box had a fewer optima, but these few optima were strong and not easy to escape from and jump to other better optimum. This behaviour came from the shape of the object that has sharp edges.

A different behaviour observed from the soft-drink can dataset. Whilst the frame number 1721 has a significantly longer average number of iterations than the frame 1120, both of the fitness demonstrated a similar pattern. The pose refinement was done gradually, and the particle filter managed to jump from one optimum to other better optimum more frequently. It demonstrated that the optimum was not strong hence the longer convergence time was observed. The dull-edged shape of the soft-drink can causes this gradual pose refinement behaviour. The smaller standard deviation observation of soft-drink dataset also confirmed the fitness evolution plot that shows less variation in the convergence time. The plot of fitness measurement for each of the iterations can be seen in Figure 5.26.



(a)

(b)

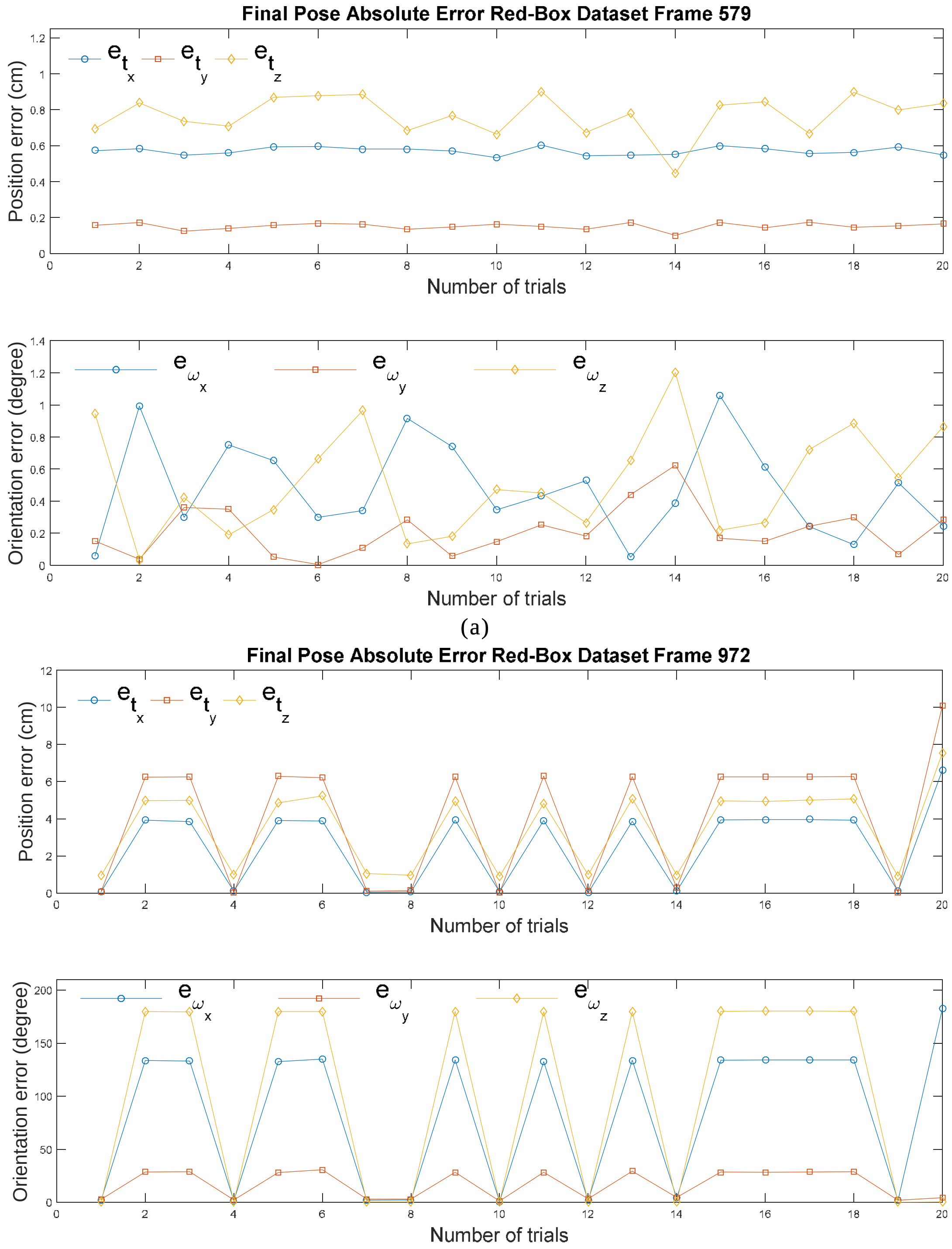
Figure 5.26. The plot of fitness score during the pose estimation given soft-drink can dataset frame 1120 (a) and frame 1721 (b). It shows the fitness was improved sharply in the first 25 iterations and then is refined gradually in the later iterations.

The accuracy of the estimation was investigated using the error from the final pose estimate of all trials. The plot of position and orientation error for the red-box dataset is presented in Figure 5.27 and the mean and standard deviation of the error is presented in Table 5.4. The result from red-box dataset frame 579 shows the final pose estimate is good with less than 1 cm position error and less than 1.2° of orientation error. However, the result of red-box dataset frame 972 demonstrated a different result. While it can estimate the pose accurately in some trials (no 1,4,6,7,10,12,14,19) with maximum position error at 1.04 cm and 4.56° for the orientation, in other trials the final pose estimated the box in upside-down direction or rotated 180° around z axis (yaw) and yielded large error. This situation is unavoidable since the projection of red-box object in the upside-down or rotated 180° pose gives the same projection edge shape. This again confirmed that vision-only pose estimation suffers from multimodal projection problem.

The statistical measurements of soft-drink can is shown in Table 5.5. The position estimate is accurate with maximum mean error at 0.87 cm. The low standard deviation on position mean error also shows that all the 20 trials were having similar behaviour. This is also confirmed from the plot of absolute error that is shown in Figure 5.28. The orientation estimate shows a different result. While the roll ω_x and pitch ω_y angle errors were small with maximum average 3.06° error. The yaw angle ω_z experienced very large error with maximum mean at 98.86°. The large score of ω_z standard deviation also shows that the yaw estimate was very varied. This behaviour was different with roll and pitch angle that tend to be more consistent and yielded a low standard deviation. This statistical measurement is also confirmed by the plot of absolute error in Figure 5.28. For both frames of soft-drink can dataset, ω_z error tended to fluctuate. The shape of the soft-drink can that is symmetrical in z axis is the reason of this result.

Table 5.4 Statistical measurement of the red-box dataset error.

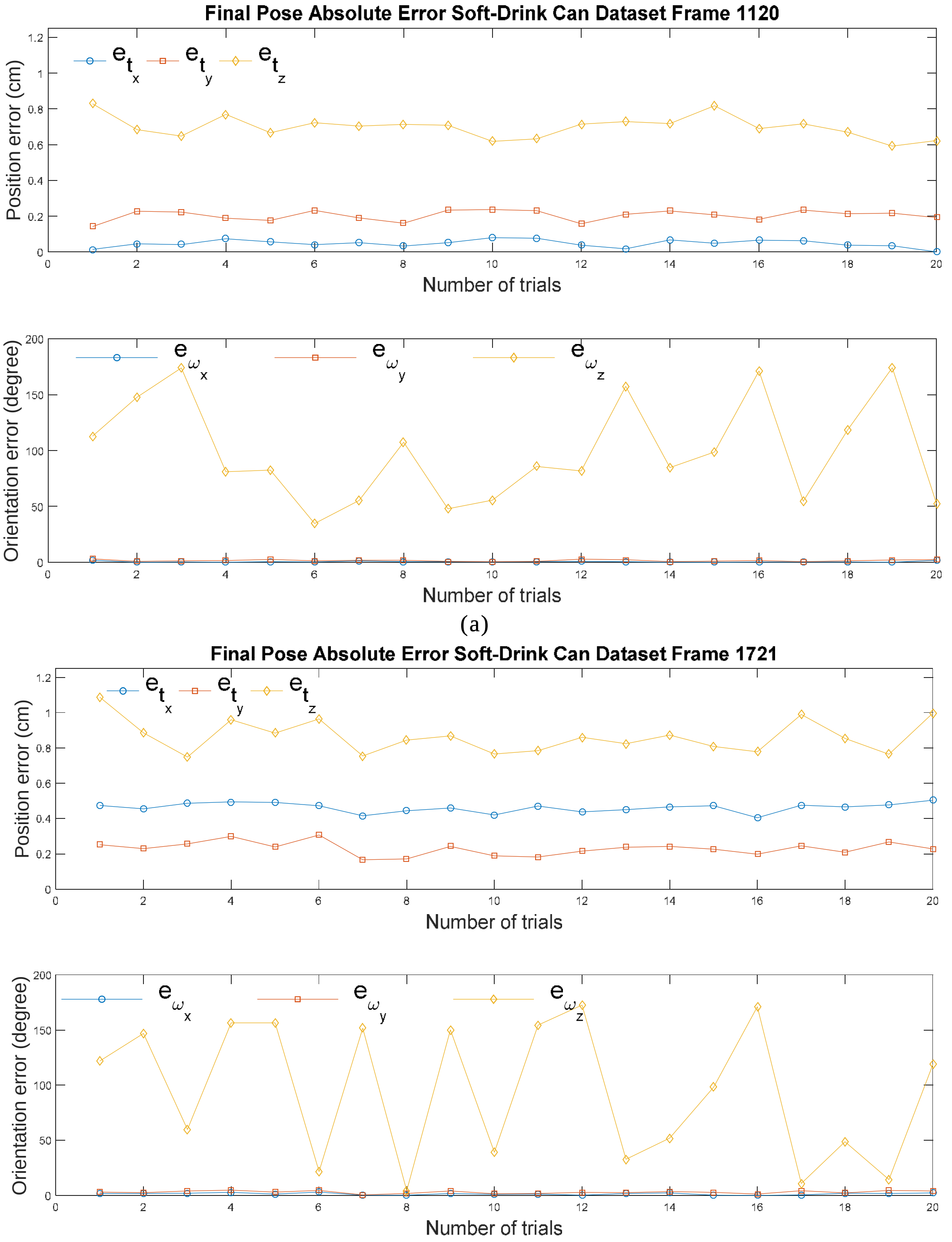
Red-box dataset	Mean Error						Standard Deviation					
	t_x	t_y	t_z	ω_x	ω_y	ω_z	t_x	t_y	t_z	ω_x	ω_y	ω_z
Frame 579	0.57	0.15	0.77	0.48	0.21	0.52	0.02	0.02	0.11	0.30	0.15	0.33
Frame 972	2.51	3.99	3.5	83.50	17.12	99.04	2.13	3.37	2.21	69.12	13.22	91.72



(a)

(b)

Figure 5.27. The plot of absolute error of red-box dataset from frame 579 (a) and frame 972 (b). The frame 579 final pose never experienced extreme pose error such as upside down or rotated 180 ° hence both position and orientation errors are small with less than 1 cm for the position error and less than 1.4 degree for the orientation error. The different result is observed in frame 972 as it shows in some trials the estimated pose orientation suffers from large errors. The large error comes since the pose estimation wrongly estimated the objects in upside-down pose or in rotated pose. Hence as can be seen in some trials the yaw angle error e_{ω_z} reached 180 ° .



(a)

(b)

Figure 5.28. Plot of orientation and position error of soft-drink can dataset for frame 1120 (a) and frame 1721 (b). Both results show position error of less than 1.2 cm. The orientation error in ω_x and ω_y are also small but very large and fluctuate in ω_z angle. The large and highly variant ω_z error is observed due to multimodal projection problem for symmetrical object, since in this case, the object is symmetrical in z axis.

Table 5.5 Statistical measurement of the soft-drink can dataset error.

Can dataset	Mean Error						Standard Deviation					
	t_x	t_y	t_z	ω_x	ω_y	ω_z	t_x	t_y	t_z	ω_x	ω_y	ω_z
Frame 1120	0.05	0.2	0.7	0.52	1.52	98.86	0.02	0.03	0.06	0.51	0.77	45.19
Frame 1721	0.46	0.23	0.87	1.46	3.06	94.12	0.03	0.04	0.09	0.88	1.21	61.80

As any rotation around the z axis produces the same projection shape, the PF converged randomly with regards to estimating the rotation around the z axis. Figure 5.29 shows some final pose estimates. The visual observation confirmed the statistical measurement and displayed that the particle filter managed to estimate position and orientation with low error, except the yaw angle that cannot be retrieved from any edge-based or region-based vision-only pose estimator.

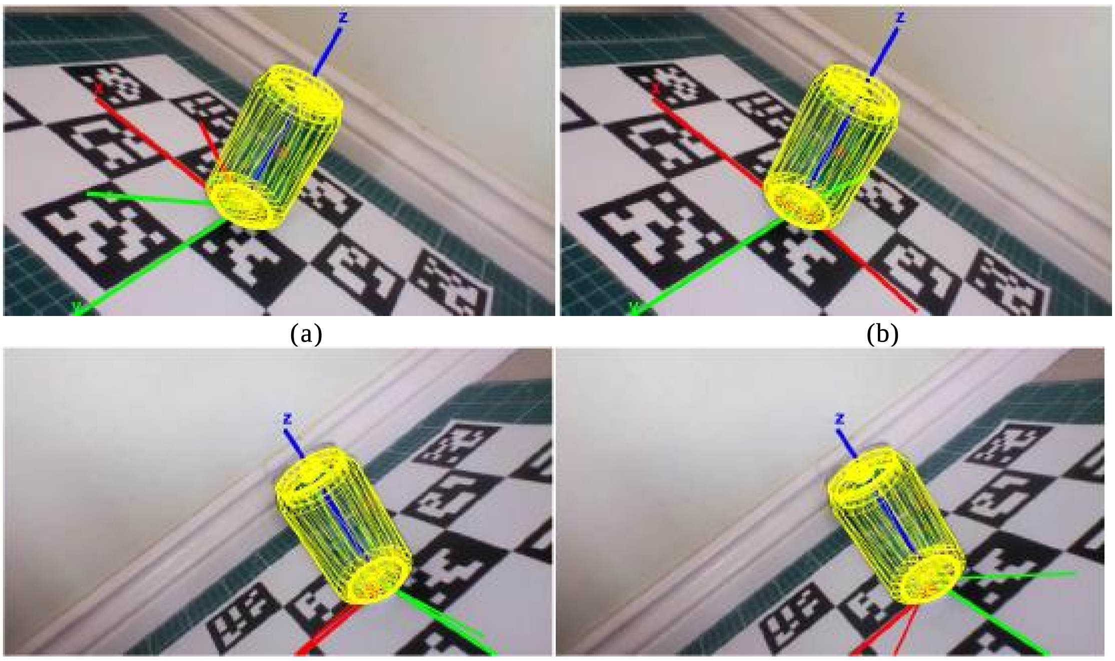
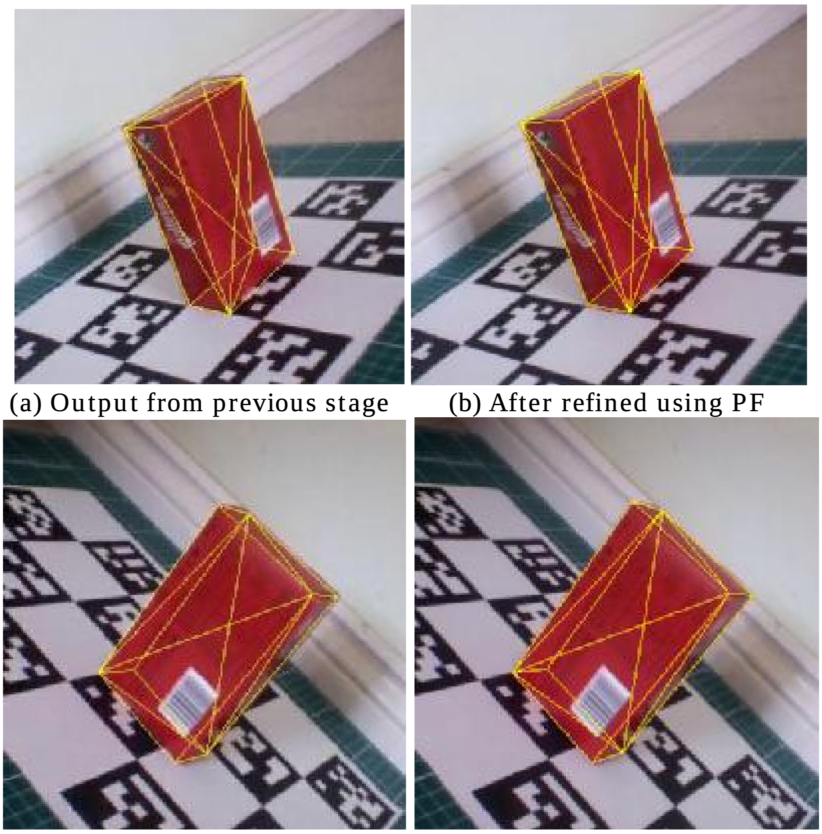


Figure 5.29. Some outputs of the soft-drink can dataset that show large variation of orientation result. As the object is symmetrical about the z axis, the vision-only pose estimation cannot recover the yaw angle ω_z. It shows that for any yaw angle, the projections were similar. However, it shows the proposed framework managed to estimate the roll ω_x and pitch angle ω_y with small error.

PWP3D-NAG and PF output comparison

The final stage of the proposed framework aimed to improve the pose estimation accuracy from earlier stage (PWP3D-NAG). While this has been confirmed from statistical measurement, a visual observation comparing the output from PWP3D-NAG step to the particle filter stage is presented in Figure 5.30 and 5.31.

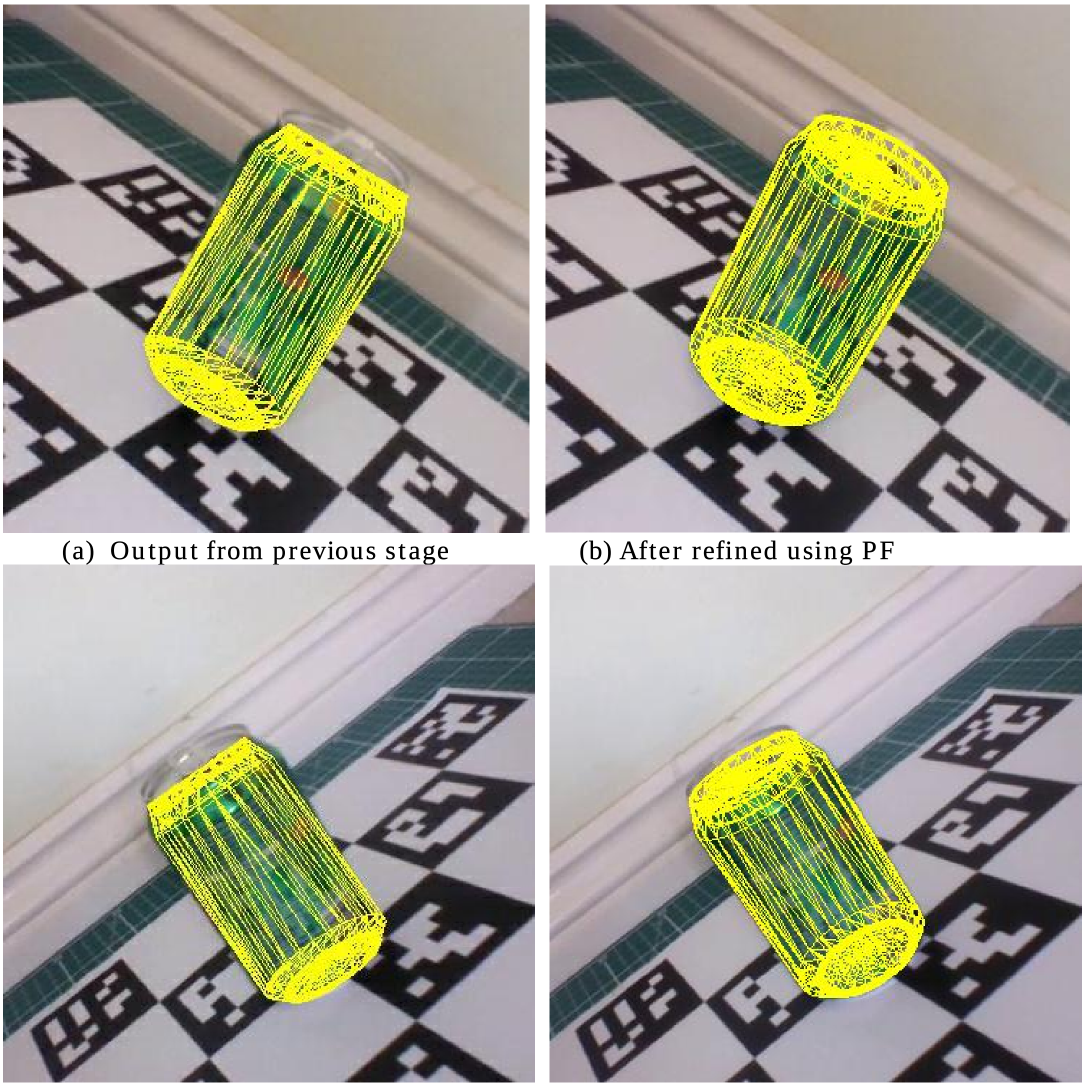
As shown in Figure 5.30 the edge-based pose orientation managed to improve the pose estimation accuracy of red-box dataset, for both of the frames. The edge-based pose orientation successfully pulled the projection to better align to the extracted edge so the accuracy was enhanced. However, a big impact was observed in soft-drink can dataset as shown in Figure 5.31. The edge-based particle filter pose estimation managed to refine the pitch angle ω_y significantly so the final pose estimation for both frames of this dataset were much more accurate.



(a) Output from previous stage (b) After refined using PF

(c) Output from previous stage (d) After refined using PF

Figure 5.30. The best pose estimate obtained from PWP3D-NAG (a, c) before being refined by edge-base pose estimation using PF were less accurate as can be seen from the wireframe plot which did not aligned correctly to the edge of the object. This result from the later stage performed a refinement by estimating the pose based on edge information. The object pose was pulled closer to the extracted edge and it shows in (b) and (d) that the final result have a better accuracy. This better accuracy of the pose estimation is also confirmed from the error measurement with respect to the ground truth provided by Aruco.



(a) Output from previous stage

(b) After refined using PF

(c) Output from previous stage

(d) After refined using PF

Figure 5.31. The comparison of the pose estimate before being refined by particle filtering pose estimate (a, c) and after the particle filtering (b, d). It clearly shows that the output from PWP3D-NAG did not manage to recover pitch angle accurately since the top part of the soft-drink can color was indistinguishable from the background. The edge-based pose estimation managed to improve the accuracy significantly by pulling the pose estimate to align to the extracted edges regardless the color.

5.4 Concluding Remarks

This chapter proposed an initialisation framework that is aimed to estimate the pose of an object as required for calculating the initial transformation between inertial/magnetic frame to object frame for the hybrid visual-inertial tracking that has been proposed in Chapter 4. The initialisation framework is intended to estimate a still object pose relative to a still camera pose. Given only statistical appearance of the model and the background (M_f and M_b respectively), a CAD model M and a very rough distance guess d the initialisation framework should be able to recover object's pose given a still image Q.

To achieve this goal, the framework performs some steps: 1. Estimates the 2D image's centre of object based on object's appearance statistical model; 2. Performs inverse projection of the 2D centre of the object to the 3D Cartesian space location and get rough 3D position estimate; 3. Generates a set of possible orientations structurally by using icosphere geometry; 4. Combines the rough position estimate and the set of generated orientations to get a set of poses. 5. Performs refinement by executing PWP3D-NAG for each of the poses and yield a refined set of pose estimate; 6. Uses this refined pose estimate to generate initial particles; and 7. Performs edge-based pose estimation on particle filter to get final pose estimate.

Each of the stages of the proposed framework has been validated and it showed its capability to improve the accuracy of the earlier stages. The final pose estimate demonstrated a small position and orientation error hence the proposed framework fulfilled the goal and can be used as the initialisation framework for the hybrid visual-inertial/magnetic pose estimate. The experiments also demonstrated that the vision-only pose estimation suffers from multimodal projection problem that has been addressed in Chapter 4.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this research, visual 3D pose estimation has been addressed. The Pixel Wise Posterior 3D Pose Estimation (PWP3D) algorithm that performs pose recovery by matching the shape of the projected model to the shape of the segmented region has been investigated since this algorithm has demonstrated its accuracy in tracking general objects with reasonably good tracking speed.

However, this state-of-the-art algorithm struggles in tracking symmetrical objects, since for symmetrical objects, different poses can have the same indistinguishable projection shape. In this case, the pose cannot be fully retrieved and PWP3D output suffers from large errors. This problem, known as multimodal projection problem, becomes the fundamental limit of the PWP3D algorithm.

Another observed weakness of PWP3D is its limited tracking capability related to motion area and the motion speed. PWP3D performs badly when the camera goes close to an object and then goes far from object in a fast motion speed. As the PWP3D implements classical Gradient Descent, the performance depends on the setting of the step size parameter. Close object requires a fine pose update to avoid divergence or unnecessary oscillation and this fine pose update can only be achieved by a small step size. However, in a same time, tracking object far from camera in a fast motion requires a large step size to be able to converge in enough time.

A new 3D pose estimation method has been developed to enhance PWP3D and deal with these problems. The strategy to deal with multimodal projection problem is by adding an additional modality from inertial/magnetic measurements. By having an additional constraint in the system of non-linear equations, the pose of symmetrical objects can be recovered. This strategy has led to an improved hybrid visual-inertial/magnetic variant of PWP3D algorithm. The limited area and speed of motion problem has been addressed by implementing a better optimisation method. Nesterov Accelerated Gradient descent (NAG) that has demonstrated a better convergence property than classical Gradient Descent has been chosen. This hybrid visual-inertial/magnetic pose estimation that implements Nesterov Accelerated Gradient descent is referred to PWP3Di-NAG.

The hybridisation between visual pose estimate and inertial/magnetic orientation estimate has been done by taking the set of non-linear equations from PWP3D visual pose estimation and developing a set of non-linear equations from inertial/magnetic orientation estimation. These equations are then all combined as a single optimisation problem and solved simultaneously. This method of integration requires an inertial/magnetic orientation estimate that is described as a pure optimisation problem. This approach is not available so this requirement has led to the development of novel inertial/magnetic orientation method and it is referred to NAG-AHRS.

This method of integration also requires all of the measurements to be carried out in the same reference system. In this case, the inertial/magnetic measurement needs to be transformed to the object reference system. This transformation can be done if the initial inertial/magnetic sensor orientation with respect to the object pose is known. Hence in this case, an additional initialisation step is required and it has led to the development of an initialisation framework to serve this purpose. The initialisation framework implements a few steps: 1. Coarse 3D position estimate. 2. Generate set of possible orientations structurally by using icosphere geometry. 3. Create a set of possible pose by combining coarse position estimate and the set of possible orientation estimate. 4. Refine each of the pose estimate by executing PWP3D-NAG multiple times. 5. Select and refine the pose estimate by edge-base pose estimation on particle filtering framework.

Therefore, in this research three main contributions have been developed: 1. Inertial/magnetic orientation estimates in a full optimisation framework NAG-

AHRS. 2. Hybrid visual-inertial/magnetic 3D pose estimation PWP3Di-NAG; and 3. An initialisation framework for PWP3Di-NAG. Each of these proposed methods has been validated by experiments.

NAG-AHRS

The NAG-AHRS has been validated using inputs from a publicly available dataset provided by Silesian University of Technology. The dataset consists of a set of comprehensive motions such as slow-fast motions, linear-nonlinear motions and structured-freehand motions. Given this dataset as input, the NAG-AHRS performance has been investigated by comparing its output to the provided ground truth. Furthermore, the NAG-AHRS has also been benchmarked to the five widely-known state-of-the-art AHRS algorithms: 1. Extended Kalman Filter-AHRS, 2. TRIAD, 3. QUEST, 4. Mahony-AHRS and 5. Madgwick-AHRS. These algorithms also represent three existing categories of method in orientation estimation which are: statistical approach (EKF-AHRS), single-frame deterministic method (TRIAD and QUEST) and complementary filter (Mahony-AHRS and Madgwick-AHRS). The experiment was done in 11 different motion scenarios and the result has been analysed thoroughly. The NAG-AHRS has demonstrated competitive results in various motion scenarios compared to the other methods. A main observed advantage of the NAG-AHRS is its capability in handling highly non-linear motions and the performance is superior among other competitive algorithms.

PWP3Di-NAG

The developed variant of PWP3D that is referred to as PWP3Di-NAG has been validated by estimating the pose of two objects: red-box and soft-drink can. The outputs of PWP3Di-NAG have been analysed and compared to the reference pose obtained from Aruco fiducial marker. The outputs have also been benchmarked to the original PWP3D algorithm. From the experiments, the PWP3Di-NAG has demonstrated its capability in tracking poorly-textured symmetrical objects, and yielded a better solution accuracy than PWP3D. This result has confirmed that the additional inertial/magnetic modality along with the proposed integration method has successfully overcome the multimodal projection problem. Other experiments have also been done in investigating the capability of PWP3Di-NAG in handling wider motion dynamics. PWP3Di-NAG managed to track an object with an extreme appearance difference (due to a significant distance difference) without changing any parameters. The hypothesis that replacing classical Gradient Descent with NAG will improve the convergence property has also been validated in this experiment. Analyses of all of these result have shown that PWP3Di-NAG has superior performance among original PWP3D.

Initialisation Framework

The developed initialisation framework consists of a few steps so the validation has been done at the final stage as well as in each of the stages to investigate its individual contribution to the overall framework. The validation was done from two datasets from different objects: red-box and soft-drink can. From each of the datasets, two frames that represent extremely different poses have been chosen randomly as the validation inputs. In the earliest stage, the position estimate has managed to obtain a coarse position estimate. Furthermore, the orientation generator strategy that utilised icosphere geometry has managed to spread the initial hypotheses. PWP3D-NAG has performed well in refining the poses and is capable to provide good hypotheses of pose. In the final stage, particle filter demonstrated the ability to select the best pose and is capable of refining the region-based pose estimation with the edge-based pose estimation. In general, the experiments have shown the pose estimation error observed a decreasing trend in every stage and the final pose estimate demonstrated a small position and orientation error. This confirmed that the proposed framework fulfilled the goal as the initialisation framework for PWP3Di-NAG.

In conclusion, the proposed NAG-AHRS, PWP3Di-NAG and initialisation framework have shown to be capable in improving the performance of their predecessor and achieved its objectives.

6.2 Future Works

Potential further improvements, include;

1. The developed PWP3Di-NAG and its initialisation framework were motivated for serving visual inspection support system. However, the developed algorithms have never been adopted to this purpose due to time the constraints. Therefore, an implementation of the developed methods will be

carried out in the future research. The implementation should include the development simulation environment using Gazebo for testing the system. Moreover, after the simulation is successful, the algorithm should be tested in real world.

2. NAG-AHRS has been developed as pure optimisation problem and the performance depends on the optimisation method. Implementing recently developed optimisation methods that have a better convergence property than Nesterov Accelerated Gradient descent can potentially improve the accuracy. Therefore, replacing the NAG with some recent optimisation method such as ADA-GRAD, ADAM that have been widely used in the training phase of neural networks need to be investigated further.
3. The proposed initialisation framework requires statistical appearance model of the object and the background. In this research, these models were generated by manually selecting parts of the image that belong to the object and part of the image belong to background. After these regions are selected, the color histograms are calculated and become the statistical appearance model. To achieve full automatic tracking without the need of any human intervention for generating the model, an algorithm that can automatically recognise the object and performs accurate segmentation is needed. A promising method for automatic object recognition and segmentation such as CRF-CNN and Mask R-CNN may be explored in further research to achieve fully automatic 3D pose estimation.
4. The experiment in initialisation framework demonstrated that region-based pose estimation can be refined significantly by complementing it with edge-based pose estimation. At the same time, the experiment in PWP3Di-NAG showed that adding some constraints (from inertial/magnetic orientation estimate in this case) to the system of non-linear equation has improved the performance. Taking this idea, a combining region-based and edge-based approach as a single optimisation problem can potentially improve the performance. Further research needs to be carried out for developing edge-based energy function and that later can be used as an additional constraints to build hybrid region-edge based pose estimation.

References

"*EasyJet's Aircraft Maintenance Lifts Off with Vicon*". (2015). Retrieved March 1, 2018, from www.vicon.com

A. J. Bray. (1990). "*Tracking Objects Using Image Disparities*". Journal of Image Vision Computing, *8*(1), 4–9.

Armstrong, M., & Zisserman, A. (1995). "*Robust Object Tracking*". In Asian Conference on Computer Vision (Vol. 1, pp. 58–61).

Bachmann, E. R., Duman, I., Usta, U. Y., McGhee, R. B., Yun, X. P., & Zyda, M. J. (1999). "*Orientation Tracking for Humans and Robots Using Inertial Sensors*". In IEEE International Symposium on Computational Intelligence in Robotics and Automation (pp. 187–194).

Barrow, H. G., Tenenbaum, J. M., Bolles, R. C., & Wolf, H. C. (1977). "*Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching*". In International Joint Conference on Artificial Intelligence (pp. 659–663).

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). "*Speeded-Up Robust Features (SURF)*". Computer Vision and Image Understanding, *110*(3), 346–359.

Besl, P. J., & McKay, N. D. (1992). "*A Method for Registration of 3-D Shapes*". IEEE Transactions on Pattern Analysis and Machine Intelligence, *14*(2), 239–256.

Bibby, C., & Reid, I. (2008). "*Robust Real-Time Visual Tracking Using Pixel-Wise Posteriors*". In Proceedings of European Conference on Computer Vision (pp. 831–844).

Bloesch, M., Omari, S., Hutter, M., & Siegwart, R. (2015). "*Robust Visual Inertial Odometry Using a Direct EKF-Based Approach*". In International Conference on Intelligent Robots and Systems (IROS) (pp. 298–304).

Borgefors, G. (1986). "*Distance Transformations in Digital Images*". Computer Vision Graphics and Image Processing, *34*(3), 344–371.

Botev, A., Lever, G., & Barber, D. (2017). "*Nesterov's Accelerated Gradient and Momentum as Approximations to Regularised Update Descent*". In International Joint Conference on Neural Networks (IJCNN) (pp. 1899–1903).

Braud, T., & Ouarti, N. (2016). "*Comparison of Nonlinear Attitude Fusion Filters*". In International Conference on Information Fusion (FUSION) (pp. 2101–2108).

Cavallo, A., Cirillo, A., Cirillo, P., De Maria, G., Falco, P., Natale, C., & Pirozzi, S. (2014). "*Experimental Comparison of Sensor Fusion Algorithms for Attitude Estimation*". IFAC Proceedings Volumes, *47*(3), 7585–7591.

Cawley, P. (2001). "*Non-Destructive Testing—Current Capabilities and Future Directions*". Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications, *215*(4), 213–223.

Chang, H., Xue, L., Qin, W., Yuan, G., & Yuan, W. (2008). "*An Integrated MEMS Gyroscope Array with Higher Accuracy Output*". In Sensors (Vol. 8, pp. 2886–2899).

Chen, Y., Zhao, S., & Farrell, J. A. (2016). "*Computationally Efficient Carrier Integer Ambiguity Resolution in Multiepoch GPS/INS: A Common-Position-Shift Approach*". IEEE Transactions on Control Systems Technology, *24*(5), 1541–1556.

Chen, Z. (2003). "*Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond*". In Technical Report.

Choi, C., & Christensen, H. I. (2011). "*Robust 3D Visual Tracking Using Particle Filtering on the SE(3) Group*". International Conference on Robotics and Automation (ICRA), (3), 4384–4390.

Choi, C., & Christensen, H. I. (2012a). "*3D Textureless Object Detection and Tracking: An Edge-Based Approach*". In International Conference on Intelligent Robots and Systems (pp. 3877–3884).

Choi, C., & Christensen, H. I. (2012b). "*Robust 3D Visual Tracking Using Particle Filtering on the Special Euclidean Group: A Combined Approach of Keypoint and Edge Features*". The International Journal of Robotics Research, *31*(4), 498–519.

Choukroun, D., Bar-Itzhack, I. Y., & Oshman, Y. (2006). "*Novel Quaternion Kalman Filter*". IEEE Transactions on Aerospace and Electronic Systems, *42*(1), 174–190.

Comport, A. I., Marchand, E., Pressigout, M., & Chaumette, F. (2006). "*Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework*". IEEE Transactions on Visualization and Computer Graphics, *12*(4), 615–628.

Crivellaro, A., Rad, M., Verdie, Y., Yi, K. M., Fua, P., & Lepetit, V. (2015). "*A Novel Representation of Parts for Accurate 3D Object Detection and Tracking in Monocular Images*". In 2015 IEEE International Conference on Computer Vision (ICCV) (pp. 4391–4399).

Crivellaro, A., Rad, M., Verdie, Y., Yi, K. M., Fua, P., & Lepetit, V. (2017). "*Robust 3D Object Tracking from Monocular Images Using Stable Parts*". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–1.

D. Black, H. (1964). "*A Passive System for Determining the Attitude of a Satellite*". American Institute of Aeronautics and Astronautics, *2*(7), 1350–1351.

Dambreville, S., Sandhu, R., Yezzi, A., & Tannenbaum, A. (2008). "*Robust 3D Pose*

Estimation and Efficient 2D Region-Based Segmentation from a 3D Shape Prior". In D. Forsyth, P. Torr, & A. Zisserman (Eds.), 10th European Conference on Computer Vision (pp. 169–182).

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). "*MonoSLAM: Real-Time Single Camera SLAM*". IEEE Transactions on Pattern Analysis and Machine Intelligence, *29*(6), 1052–1067.

Drummond, T., & Cipolla, R. (2002). "*Real-Time Visual Tracking of Complex Structures*". IEEE Transactions on Pattern Analysis and Machine Intelligence, *24*(7), 932–946.

Durrant-Whyte, H., & Bailey, T. (2006). "*Simultaneous Localization and Mapping: Part I*". IEEE Robotics & Automation Magazine, *13*(2), 99–110.

Ekvall, S., Kragic, D., & Hoffmann, F. (2005). "*Object Recognition and Pose Estimation Using Color Cooccurrence Histograms and Geometric Modeling*". Image Vision Comput., *23*(11), 943–955.

El-Sheimy, N., Hou, H., & Niu, X. (2008). "*Analysis and Modeling of Inertial Sensors Using Allan Variance*". IEEE Transactions on Instrumentation and Measurement, *57*(1), 140–149.

Engel, J., Koltun, V., & Cremers, D. (2018). "*Direct Sparse Odometry*". IEEE Transactions on Pattern Analysis and Machine Intelligence, *40*(3), 611–625.

Engel, J., Schöps, T., & Cremers, D. (2014). "*LSD-SLAM: Large-Scale Direct Monocular SLAM*". In 13th European Conference in Computer Vision (pp. 834–849).

Euston, M., Coote, P., Mahony, R., Jonghyuk Kim, & Hamel, T. (2008). "*A Complementary Filter for Attitude Estimation of a Fixed-Wing UAV*". In International Conference on Intelligent Robots and Systems (pp. 340–345).

Fang, W., Zheng, L., & Deng, H. (2016). "*A Motion Tracking Method by Combining the IMU and Camera in Mobile Devices*". In 10th International Conference on Sensing Technology (pp. 1–6).

Fang, W., Zheng, L., Deng, H., & Zhang, H. (2017). "*Real-Time Motion Tracking for Mobile Augmented/Virtual Reality Using Adaptive Visual-Inertial Fusion*". Sensors (Basel, Switzerland), *17*(5), 1037.

Farrell, J. L. (1970). "*Attitude Determination by Kalman Filtering*". Automatica, *6*(3), 419–430.

Filippeschi, A., Schmitz, N., Miezal, M., Bleser, G., Ruffaldi, E., & Stricker, D. (2017). "*Survey of Motion Tracking Methods Based on Inertial Sensors: A Focus on Upper Limb Human Motion*". Sensors (Basel, Switzerland), *17*(6), 1257.

Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). "*SVO: Fast Semi-Direct Monocular Visual Odometry*". In 2014 IEEE International Conference on Robotics and Automation (ICRA) (pp. 15–22).

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., & Marín-Jiménez, M. J. (2014). "*Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion*". Pattern Recognition, *47*(6), 2280–2292.

Gennery, D. B. (1992). "*Visual Tracking of Known Three-Dimensional Objects*". International Journal of Computer Vision, *7*(3), 243–270.

Goh, G. (2017). "*Why Momentum Really Works*". Distill.

Gui, J., Gu, D., Wang, S., & Hu, H. (2015). "*A Review of Visual Inertial Odometry from Filtering and Optimisation Perspectives*". Advanced Robotics, *29*(20), 1289–1301.

Han, P., & Zhao, G. (2015). "*CAD-Based 3D Objects Recognition in Monocular Images for Mobile Augmented Reality*". Comput. Graph., *50*(C), 36–46.

Harris, C., & Stennett, C. (1990). "*RAPID - a Video Rate Object Tracker*". In British Machine Vision Conference (p. 15.1-15.6).

Hellier, C. (2012). *Handbook of Nondestructive Evaluation, Second Edition*.

Hesch, J. A., Kottas, D. G., Bowman, S. L., & Roumeliotis, S. I. (2014). "*Camera-IMU-Based Localization: Observability Analysis and Consistency Improvement*". The International Journal of Robotics Research, *33*(1), 182–201.

Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., & Navab, N. (2013). "*Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes*". In Asian Conference on Computer Vision (pp. 548–562).

Isard, M., & Blake, A. (1998). "*CONDENSATION—Conditional Density Propagation for Visual Tracking*". International Journal of Computer Vision, *29*(1), 5–28.

Jensen, A., Coopmans, C., & Chen, Y. (2013). "*Basics and Guidelines of Complementary Filters for Small UAS Navigation*". In International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 500–507).

Jiang, W., & Yin, Z. (2017). "*Combining Passive Visual Cameras and Active IMU Sensors for Persistent Pedestrian Tracking*". Journal of Visual Communication and Image Representation, *48*(Supplement C), 419–431.

Kalman, R. E. (1960). "*A New Approach to Linear Filtering and Prediction Problems*". Transactions of the ASME – Journal of Basic Engineering, *82*(1), 35–45.

Kehl, W., Tombari, F., Ilic, S., & Navab, N. (2017). "*Real-Time 3D Model Tracking in Color and Depth on a Single CPU Core*". In Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 465–473).

Kelsey, J. M., Byrne, J., Cosgrove, M., Seereeram, S., & Mehra, R. K. (2006). "*Vision-Based Relative Pose Estimation for Autonomous Rendezvous and Docking*". In IEEE Aerospace Conference (p. 20).

Kim, H.-B., & Sim, K.-B. (2010). "*A Specified Object Tracking with Particle Filter*". In International Conference on Soft Computing and Intelligent Systems (pp. 1137–1138).

Klein, G., & Murray, D. (2007). "*Parallel Tracking and Mapping for Small AR*

Workspaces". In 6th IEEE and ACM International Symposium on Mixed and Augmented Reality.

Klein, G., & Murray, D. W. (2006). "*Full-3D Edge Tracking with a Particle Filter*". British Machine Vision Conference, 1–10.

Koller, D., Daniilidis, K., & Nagel ', H.-H. (1993). "*Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes*". International Journal of Computer Vision, *103*, 257–281.

Kuga, H. K., & Carrara, V. (2013). "*Attitude Determination with Magnetometers and Accelerometers to Use in Satellite Simulator*". Mathematical Problems in Engineering, 1–6.

Kumar, S., & Mahto, D. G. (2013). "*Recent Trends in Industrial and Other Engineering Applications of Non Destructive Testing: A Review*". International Journal of Scientific & Engineering Research, *4*(9).

La, H. M., Gucunski, N., Dana, K., & Kee, S.-H. (2017). "*Development of an Autonomous Bridge Deck Inspection Robotic System*". Journal of Field Robotics, *34*(8), 1489–1504.

Lebeda, K., Matas, J., & Bowden, R. (2012). "*Tracking the Untrackable: How to Track When Your Object Is Featureless*". In Asian Conference on Computer Vision (pp. 347–359).

Lefferts, E. J., Markley, F. L., & Shuster, M. D. (1982). "*Kalman Filtering for Spacecraft Attitude Estimation*". Journal of Guidance, Control, and Dynamics, *5*(5), 417–429.

Leonard, J. J., & Durrant-Whyte, H. F. (1991). "*Simultaneous Map Building and Localization for an Autonomous Mobile Robot*". In IEEE/RSJ International Workshop on Intelligent Robots and Systems (pp. 1442–1447).

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2014). "*Keyframe-Based Visual–Inertial Odometry Using Nonlinear Optimization*". The International Journal of Robotics Research, *34*(3), 314–334.

Li, M., & Mourikis, A. I. (2013). "*High-Precision, Consistent EKF-Based Visual-Inertial Odometry*". The International Journal of Robotics Research, *32*(6), 690–711.

Ligorio, G., & Sabatini, A. M. (2013). "*Extended Kalman Filter-Based Methods for Pose Estimation Using Visual, Inertial and Magnetic Sensors: Comparative Analysis and Performance Evaluation*". Sensors (Basel, Switzerland), *13*(2), 1919–1941.

Liu, T., Inoue, Y., & Shibata, K. (2011). "*Simplified Kalman Filter for a Wireless Inertial-Magnetic Motion Sensor*". In IEEE SENSORS Proceedings (pp. 569–572).

Lowe, D. G. (2004). "*Distinctive Image Features from Scale-Invariant Keypoints*". International Journal of Computer Vision, *60*(2), 91–110.

Madgwick, S. O. H., Harrison, A. J. L., & Vaidyanathan, R. (2011). "*Estimation of IMU and MARG Orientation Using a Gradient Descent Algorithm*". In IEEE

International Conference on Rehabilitation Robotics (pp. 179–185).

Mahony, R., Hamel, T., & Pflimlin, J. M. (2005). "*Complementary Filter Design on the Special Orthogonal Group SO(3)*". In 44th IEEE Conference on Decision and Control (pp. 1477–1484).

Mahony, R., Hamel, T., & Pflimlin, J. M. (2008). "*Nonlinear Complementary Filters on the Special Orthogonal Group*". IEEE Transactions on Automatic Control, *53*(5), 1203–1218.

Marins, J. L., Yun, X., Bachmann, E. R., McGhee, R. B., & Zyda, M. J. (2001). "*An Extended Kalman Filter for Quaternion-Based Orientation Estimation Using MARG Sensors*". In IEEE/RSJ International Conference on Intelligent Robots and Systems (Vol. 4, pp. 2003–2011).

Markley, F. L. (2003). "*Attitude Error Representations for Kalman Filtering*". Journal of Guidance, Control, and Dynamics, *26*(2), 311–317.

Merriaux, P., Dupuis, Y., Boutteau, R., Vasseur, P., & Savatier, X. (2017). "*A Study of Vicon System Positioning Performance*". Sensors, *17*(7), 1591.

Munoz, E., Konishi, Y., Murino, V., & Del Bue, A. (2016). "*Fast 6D Pose Estimation for Texture-Less Objects from a Single RGB Image*". In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5623–5630).

Mur-Artal, R., & Tardos, J. (2016). "*Visual-Inertial Monocular SLAM with Map Reuse*". IEEE Robotics and Automation Letters, *2*(2).

Mur-Artal, R., & Tardos, J. D. (2017). "*ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras*". IEEE Transactions on Robotics, *33*(5), 1255–1262.

Nesterov, Y. (1983). "*A Method of Solving a Convex Programming Problem with Convergence Rate O(1/k2)*". In Soviet Mathematics Doklady 27(2) (pp. 372–376).

Neunert, M., Blösch, M., & Buchli, J. (2015). "*An Open Source, Fiducial Based, Visual-Inertial State Estimation System*". In International Conference on Information Fusion.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., … Fitzgibbon, A. (2011). "*KinectFusion: Real-Time Dense Surface Mapping and Tracking*". In 2011 10th IEEE International Symposium on Mixed and Augmented Reality (pp. 127–136).

Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., & Siegwart, R. (2013). "*A UAV System for Inspection of Industrial Facilities*". In IEEE Aerospace Conference (pp. 1–8).

Nowicki, M., Wietrzykowski, J., & Skrzypczynski, P. (2015). "*Simplicity or Flexibility? Complementary Filter vs EKF for Orientation Estimation on Mobile Devices*". In IEEE 2nd International Conference on Cybernetics (CYBCONF) (pp. 166–171).

Ozaslan, T., Loianno, G., Keller, J., Taylor, C. J., Kumar, V., Wozencraft, J. M., &

Hood, T. (2017). "*Autonomous Navigation and Mapping for Inspection of Penstocks and Tunnels With MAVs*". IEEE Robotics and Automation Letters, *2*(3), 1740–1747.

Ozuysal, M., Calonder, M., Lepetit, V., & Fua, P. (2010). "*Fast Keypoint Recognition Using Random Ferns*". IEEE Transactions on Pattern Analysis and Machine Intelligence, *32*(3), 448–461.

P. Davenport. (1965). "*Attitude Determination and Sensor Alignment via Weighted Least Squares Affine Transformations*". In Nasa Technical Report.

Petkov, P., & Slavov, T. (2010). "*Stochastic Modeling of MEMS Inertial Sensors*". Bulgarian Academy of Sciences: Cybernetics and Information Technologies, *10*.

Pham, N. H., & La, H. M. (2016). "*Design and Implementation of An Autonomous Robot for Steel Bridge Inspection*". In 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton) (pp. 556–562).

Prisacariu, V. A., Kähler, O., Murray, D. W., & Reid, I. D. (2015). "*Real-Time 3D Tracking and Reconstruction on Mobile Phones*". IEEE Transactions on Visualization and Computer Graphics, *21*(5), 557–570.

Prisacariu, V. A., & Reid, I. D. (2012). "*PWP3D: Real-Time Segmentation and Tracking of 3D Objects*". International Journal of Computer Vision, *98*(3).

Psiaki, M. L., Martel, F., & Pal, P. K. (1990). "*Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data*". Journal of Guidance, Control, and Dynamics, *13*(3), 506–514.

Pupilli, M., & Calway, A. (2006). "*Real-Time Camera Tracking Using Known 3D Models and a Particle Filter*". In 18th International Conference on Pattern Recognition (ICPR’06) (pp. 199–203).

Qu, G., & Li, N. (2017). "*Accelerated Distributed Nesterov Gradient Descent for Convex and Smooth Functions*". In IEEE 56th Annual Conference on Decision and Control (CDC) (pp. 2260–2267).

Rad, M., & Lepetit, V. (2017). "*BB8: A Scalable, Accurate, Robust to Partial Occlusion Methodfor Predicting the 3D Poses of Challenging Objects without Using Depth*". In IEEE International Conference on Computer Vision.

Renfro, B. A., Terry, A., & Boeker, N. (2013). "*An Analysis of Global Positioning System (GPS) Standard Positioning System (SPS) Performance for 2013*". In GPS Instituition Report.

Roberts, J. M., Corke, P. I., & Buskey, G. (2003). "*Low-Cost Flight Control System for a Small Autonomous Helicopter*". In IEEE International Conference on Robotics and Automation (Vol. 1, pp. 546–551).

Rosten, E., & Drummond, T. (2006). "*Machine Learning for High-Speed Corner Detection*". In 9th European Conference on Computer Vision (pp. 430–443).

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). "*ORB: An Efficient Alternative to SIFT or SURF*". In International Conference on Computer Vision (pp. 2564–2571).

Sa, I., Hrabar, S., & Corke, P. (2015). "*Outdoor Flight Testing of a Pole Inspection UAV Incorporating High-Speed Vision*". Springer Trasaction in Advanced Robotics, *105*, 107–121.

Sabatini, A. M. (2006). "*Quaternion-Based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing*". IEEE Transactions on Biomedical Engineering, *53*(7), 1346–1356.

Sabatini, A. M., & Maria, A. (2011). "*Estimating Three-Dimensional Orientation of Human Body Parts by Inertial/Magnetic Sensing*". Sensors, *11*(2), 1489–1525.

Scaramuzza, D., & Fraundorfer, F. (2011). "*Visual Odometry [Tutorial]*". IEEE Robotics & Automation Magazine, *18*(4), 80–92.

Schempf, H., Chemel, B., & Everett, N. (1995). "*Neptune: Above-Ground Storage Tank Inspection Robot System*". IEEE Robotics & Automation Magazine, *2*(2), 9–15.

Segal, A., Haehnel, D., & Thrun, S. (2009). "*Generalized-ICP.*". In Robotics: Science and Systems (Vol. 2).

Senyurek, V., Baspinar, U., & S Varol, H. (2014). "*A Modified Adaptive Kalman Filter for Fiber Optic Gyroscope*". In Revue Roumaine des Sciences Techniques - Serie Électrotechnique et Énergétique (Vol. 59, pp. 153–162).

Seo, B.-K., Park, H., Park, J.-I., Hinterstoisser, S., & Ilic, S. (2014). "*Optimal Local Searching for Fast and Robust Textureless 3D Object Tracking in Highly Cluttered Backgrounds*". IEEE Transactions on Visualization and Computer Graphicss, *20*(1), 99–110.

Seo, B.-K., Park, J., Park, H., & Park, J.-I. (2013). "*Real-Time Visual Tracking of Less Textured Three-Dimensional Objects on Mobile Platforms*". Optical Engineering, *51*(12), 127202.

Seo, B.-K., & Wuest, H. (2016). "*A Direct Method for Robust Model-Based 3D Object Tracking from a Monocular RGB Image*". In European Conference on Computer Vision (pp. 551–562).

Shiau, J.-K., Huang, C.-X., & Chang, M.-Y. (2012). "*Noise Characteristics of MEMS Gyro's Null Drift and Temperature Compensation*". Journal of Applied Science and Engineering, *15*, 239–246.

Shuster, M. D., & Oh, S. D. (1981). "*Three-Axis Attitude Determination from Vector Observations*". Journal of Guidance, Control, and Dynamics, *4*(1), 70–77.

Sirtkaya, S., Seymen, B., & Alatan, A. A. (2013). "*Loosely Coupled Kalman Filtering for Fusion of Visual Odometry and Inertial Navigation*". In Proceedings of the 16th International Conference on Information Fusion (pp. 219–226).

Skrypnyk, I., & Lowe, D. (2004). "*Scene Modelling, Recognition and Tracking with Invariant Image Features*". In 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (pp. 110–119).

Sobel, I. (2014). *An Isotropic 3x3 Image Gradient Operator*. Presentation at Stanford A.I. Project 1968.

Stumm, E., Breitenmoser, A., Pomerleau, F., Pradalier, C., & Siegwart, R. (2012). "*Tensor-Voting-Based Navigation for Robotic Inspection of 3D Surfaces Using Lidar Point Clouds*". The International Journal of Robotics Research, *31*(12), 1465–1488.

Su, W., Chen, L., Wu, M., Zhou, M., Liu, Z., & Cao, W. (2017). "*Nesterov Accelerated Gradient Descent-Based Convolution Neural Network with Dropout for Facial Expression Recognition*". In 11th Asian Control Conference (ASCC) (pp. 1063–1068).

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). "*On the Importance of Initialization and Momentum in Deep Learning*". In 30th International Conference on Machine Learning (ICML) (pp. 1139–1147).

Szczęsna, A., & Pruszowski, P. (2016). "*Model-Based Extended Quaternion Kalman Filter to Inertial Orientation Tracking of Arbitrary Kinematic Chains*". SpringerPlus, *5*(1), 1965.

Szczęsna, A., Skurowski, P., Pruszowski, P., Pęszor, D., Paszkuta, M., & Wojciechowski, K. (2016). "*Reference Data Set for Accuracy Evaluation of Orientation Estimation Algorithms for Inertial Motion Capture Systems*". In International Conference on Computer Vision and Graphics (pp. 509–520).

Takayama, L., Ju, W., & Nass, C. (2008). "*Beyond Dirty, Dangerous and Dull*". In 3rd International Conference on Human robot interaction (p. 25).

Teixeira, L., Alzugaray, I., & Chli, M. (2018). "*Autonomous Aerial Inspection Using Visual-Inertial Robust Localization and Mapping*" (pp. 191–204).

Tian, Z., Li, J., Li, Q., & Cheng, N. (2017). "*A Visual-Inertial Navigation System Based on Multi-State Constraint Kalman Filter*". In 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) (Vol. 1, pp. 199–202).

Timothy Dozat. (2015). "*Incorporating Nesterov Momentum into Adam*". In Technical Report.

Tjaden, H., Schwanecke, U., & Schömer, E. (2016). "*Real-Time Monocular Segmentation and Pose Tracking of Multiple Objects*". In 14th European Conference of Computer Vision (pp. 423–438).

Tjaden, H., Schwanecke, U., & Schömer, E. (2017). "*Real-Time Monocular Pose Estimation of 3D Objects Using Temporally Consistent Local Color Histograms*". In International Conference on Computer Vision.

Tjaden, H., Schwanecke, U., Schömer, E., & Cremers, D. (2018). "*A Gauss-Newton Approach to Real-Time Monocular Multiple Object Tracking.*". CoRR. Retrieved from http://arxiv.org/abs/1807.02087

Trawny, N., & Roumeliotis, S. I. (2005). "*Indirect Kalman Filter for 3D Attitude Estimation*". In Multiple Autonomous Robotic Systems Laboratory Technical Report.

Vacchetti, L., Lepetit, V., & Fua, P. (2004a). "*Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking*". In 3rd IEEE/ACM

International Symposium on Mixed and Augmented Reality (pp. 48–57).

Vacchetti, L., Lepetit, V., & Fua, P. (2004b). "*Stable Real-Time 3D Tracking Using Online and Offline Information*". IEEE Transaction on Pattern Analysis and Machine Intelligence, *26*(10), 1385–1391.

Valenti, R. G., Dryanovski, I., & Xiao, J. (2016). "*A Linear Kalman Filter for MARG Orientation Estimation Using the Algebraic Quaternion Algorithm*". IEEE Transactions on Instrumentation and Measurement, *65*(2), 467–481.

Valinetti, A., Fusiello, A., & Murino, V. (2001). "*Model Tracking for Video-Based Virtual Reality*". In 1th International Conference on Image Analysis and Processing (pp. 372–377).

von Stumberg, L., Usenko, V., & Cremers, D. (2018). "*Direct Sparse Visual-Inertial Odometry Using Dynamic Marginalization*". In IEEE International Conference on Robotics and Automation (ICRA).

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., & Schmalstieg, D. (2008). "*Pose Tracking from Natural Features on Mobile Phones*". In 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (pp. 125–134).

Wahba, G. (1965). "*A Least Squares Estimate of Satellite Attitude*". SIAM Review, *7*(3).

Wang, G., Wang, B., Zhong, F., Qin, X., & Chen, B. (2015). "*Global Optimal Searching for Textureless 3D Object Tracking*". The Visual Computer, *31*(6–8), 979–988.

Wang, L., Zhang, Z., & Sun, P. (2015). "*Quaternion-Based Kalman Filter for AHRS Using an Adaptive-Step Gradient Descent Algorithm*". International Journal of Advanced Robotic Systems, *12*(9), 131.

Wang, R., Schworer, M., & Cremers, D. (2017). "*Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras*". In IEEE International Conference on Computer Vision (ICCV) (pp. 3923–3931).

Wenner, C., Spencer, F., & Drury, C. G. (2003). "*The Impact of Instructions on Aircraft Visual Inspection Performance: A First Look at the Overall Results*". Proceedings of the Human Factors and Ergonomics Society Annual Meeting, *47*(1), 51–55.

Worrall, A. D., Marslin, R. F., Sullivan, G. D., & Baker, K. D. (1991). "*Model-Based Tracking*". In Proceedings of the British Machine Vision Conference (pp. 310–318).

Wuthrich, M., Pastor, P., Kalakrishnan, M., Bohg, J., & Schaal, S. (2013). "*Probabilistic Object Tracking Using a Range Camera*". In IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3195–3202).

Yan, D., & Hu, H. (2017). "*Application of Augmented Reality and Robotic Technology in Broadcasting: A Survey*". Robotics, *6*(3), 18.

Yang, J., Li, H., Campbell, D., & Jia, Y. (2016). "*Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration*". IEEE Transactions on Pattern Analysis and Machine Intelligence, *38*(11), 2241–2254.

Yang, Y., & Cao, Q.-X. (2012). "*Monocular Vision Based 6D Object Localization for Service Robot's Intelligent Grasping*". Computers & Mathematics with Applications, *64*(5), 1235–1241.

Yun, X., & Bachmann, E. R. (2006). "*Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking*". IEEE Transactions on Robotics, *22*(6), 1216–1227.

Yun, X., Bachmann, E. R., & McGhee, R. B. (2008). "*A Simplified Quaternion-Based Algorithm for Orientation Estimation From Earth Gravity and Magnetic Field Measurements*". IEEE Transactions on Instrumentation and Measurement, *57*(3), 638–650.

Zhang, Z.-Q., Meng, X.-L., & Wu, J.-K. (2012). "*Quaternion-Based Kalman Filter With Vector Selection for Accurate Orientation Tracking*". IEEE Transactions on Instrumentation and Measurement, *61*(10), 2817–2824.