Tampere University

IHTISHAM ALI

Methods, Models, and Datasets for Visual Servoing and Vehicle Localisation

Tampere University Dissertations 747

Tampere University Dissertations 747

IHTISHAM ALI

Methods, Models, and Datasets for Visual Servoing and Vehicle Localisation

ACADEMIC DISSERTATION
To be presented, with the permission of
The Faculty of Information Technology and Communication Sciences
of Tampere University,
for public discussion in the auditorium TB109
of the Tietotalo, Korkeakoulunkatu 1, Tampere,
on 24 February 2023, at 12 o'clock.

ACADEMIC DISSERTATION
Tampere University,
Faculty of Information Technology and Communication Sciences,
Finland

Responsible supervisor and Custos	Professor Atanas Gotchev Tampere University Finland	
Supervisor	University Lecturer, Dr. Sari Peltonen Tampere University Finland	
Pre-examiners	Professor Ahmet Enis Cetin University of Illinois Chicago USA	Dr. Sven Fleck SmartSurv Vision Systems Germany
Opponents	Assistant Professor Juho Kannala Aalto University Finland	Dr. Sven Fleck SmartSurv Vision Systems Germany

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Cover design: Roihu Inc.

ISBN 978-952-03-2763-7 (print)
ISBN 978-952-03-2764-4 (pdf)
ISSN 2489-9860 (print)
ISSN 2490-0028 (pdf)
http://urn.fi/URN:ISBN:978-952-03-2764-4

Carbon dioxide emissions from printing Tampere University dissertations have been compensated.

PunaMusta Oy – Yliopistopaino
Joensuu 2023

Dedicated to my parents, to whom I owe everything.

PREFACE

This research represents the culmination of many years of diligent work and dedication to the field of machine vision, and it has been a privilege to be able to conduct in-depth research and exploration in this area. The findings of this study provide new insights and perspectives on Visual Servoing and Localisation and offer potential implications for industrial applications and academic research.

Throughout the process of conducting this research, I have been fortunate to have the support and guidance of many individuals. I would like to extend my sincere gratitude to my supervisors, Prof. Atanas Gotchev and Dr. Sari Peltonen, for their invaluable guidance and support throughout this journey. Additionally, I would like to extend my gratitude to an incredible colleague and mentor, Olli. J. Suominen.

I want to thank my pre-examination reviewers, Prof. Ahmet Enis Cetin from the University of Illinois Chicago and Dr. Sven Fleck from SmartSurv Vision Systems GmbH in Germany, for providing valuable feedback and comments on this thesis. I am also thankful to Assist. Prof. Juho Kannala from Aalto University and Dr. Sven Fleck for agreeing to serve as opponents during the thesis defense.

It would be my pleasure to express my appreciation to my colleagues, Dr. Robert Bregovic, Ahmed Durmush, Laura Goncalves, Dr. Erdem Sahin, Sarianne Niemelä, Jani Mäkinen, Ugur Akpinar, Filipe Da Graca Gama, Sergio Moreschini, and all the many members of the 3D Media Research Group that I have had the pleasure of working with over the years.

I would like to acknowledge the funding provided by Business Finland, ITER, the Finnish Cultural Foundation, and the Nokia Foundation in the form of grants and achievement awards. These awards made this research possible.

I want to thank my family and friends for their love and support. Without their encouragement, this dissertation would not have been possible.

It is with honor and pleasure that I share this work with the academic community.

ABSTRACT

Machine autonomy has become a vibrant part of industrial and commercial aspirations. A growing demand exists for dexterous and intelligent machines that can work in unstructured environments without any human assistance. An autonomously operating machine should sense its surroundings, classify different kinds of observed objects, and interpret sensory information to perform necessary operations.

This thesis summarizes original methods aimed at enhancing machine's autonomous operation capability. These methods and the corresponding results are grouped into two main categories. The first category consists of research works that focus on improving visual servoing systems for robotic manipulators to accurately position workpieces. We start our investigation with the hand-eye calibration problem that focuses on calibrating visual sensors with a robotic manipulator. We thoroughly investigate the problem from various perspectives and provide alternative formulations of the problem and error objectives. The experimental results demonstrate that the proposed methods are robust and yield accurate solutions when tested on real and simulated data. The work package is bundled as a toolkit and available online for public use. In an extension, we proposed a constrained multiview pose estimation approach for robotic manipulators. The approach exploits the available geometric constraints on the robotic system and infuses them directly into the pose estimation method. The empirical results demonstrate higher accuracy and significantly higher precision compared to other studies.

In the second part of this research, we tackle problems pertaining to the field of autonomous vehicles and its related applications. First, we introduce a pose estimation and mapping scheme to extend the application of visual Simultaneous Localization and Mapping to unstructured dynamic environments. We identify, extract, and discard dynamic entities from the pose estimation step. Moreover, we track the dynamic entities and actively update the map based on changes in the environment. Upon observing the limitations of the existing datasets during our earlier work, we

introduce FinnForest, a novel dataset for testing and validating the performance of visual odometry and Simultaneous Localization and Mapping methods in an unstructured environment. We explored an environment with a forest landscape and recorded data with multiple stereo cameras, an IMU, and a GNSS receiver. The dataset offers unique challenges owing to the nature of the environment, variety of trajectories, and changes in season, weather, and daylight conditions. Building upon the future works proposed in FinnForest Dataset, we introduce a novel scheme that can localize an observer with extreme perspective changes. More specifically, we tailor the problem for autonomous vehicles such that they can recognize a previously visited place irrespective of the direction it previously traveled the route. To the best of our knowledge, this is the first study that accomplishes bi-directional loop closure on monocular images with a nominal field of view. To solve the localisation problem, we segregate the place identification from the pose regression by using deep learning in two steps. We demonstrate that bi-directional loop closure on monocular images is indeed possible when the problem is posed correctly, and the training data is adequately leveraged.

All methodological contributions of this thesis are accompanied by extensive empirical analysis and discussions demonstrating the need, novelty, and improvement in performance over existing methods for pose estimation, odometry, mapping, and place recognition.

CONTENTS

ABBREVIATIONS

ADAS	Advance Driver Assistance System
AR	Augmented Reality
BOW	Bag of Words
BRISK	Binary Robust Invariant Scalable Key points
CNN	Convolutional Neural Network
DoF	Degrees of freedom
ECL	endpoint closed-loop
EOL	endpoint open-loop
GMM	Gaussian Mixture Model
GNSS	Global Navigation and Satellite System
HD	Homography Decomposition
HE	Hand-Eye
IBVS	Image based Visual Servoing
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
IPPE	infinitesimal plane-based pose estimation
ITER	International Nuclear Fusion Research
LBP	Local Binary Pattern
MMS	Monocular multi shot
MSS	Monocular single shot
PBVS	Position/Pose-based Visual Servoing

PnP	Perspective from n-points
ReLU	Rectified Linear Unit
RWHE	Robot-World-Hand-Eye
S-PTAM	Stereo Parallel Tracking and Mapping
SFM	Stucture from Motion
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SSFM	Spatial Sensing for Machines
SSS	Stereo single shot
SURF	Speeded Up Robust Feature
SVD	Single Value Decomposition
TCP	Tool centre point
VLAD	vector of locally aggregated descriptors

ORIGINAL PUBLICATIONS

Publication I — I. Ali, O. Suominen, A. Gotchev, and E. R. Morales, "Methods for simultaneous robot-world-hand–eye calibration: A comparative study," *Sensors*, vol. 19, no. 12, p. 2837, 2019. DOI: 10.3390/s19122837.

Publication II — I. Ali, O. J. Suominen, E. R. Morales, and A. Gotchev, "Multi-view camera pose estimation for robotic arm manipulation," *IEEE Access*, vol. 8, pp. 174 305–174 316, 2020. DOI: 10.1109/ACCESS.2020.3026108.

Publication III — I. Ali, O. Suominen, and A. Gotchev, "Discrimination of active dynamic objects in stereo-based visual SLAM," *Electronic Imaging*, vol. 2018, no. 13, pp. 463-1–463-6, 2018. DOI: 10.2352/ISSN.2470-1173.2018.13.IPAS-463.

Publication IV — I. Ali, A. Durmush, O. Suominen, J. Yli-Hietanen, S. Peltonen, J. Collin, and A. Gotchev, "FinnForest dataset: A forest landscape for visual SLAM," *Robotics and Autonomous Systems*, vol. 132, p. 103 610, 2020. DOI: 10.1016/j.robot.2020.103610.

Publication V — I. Ali, S. Peltonen, and A. Gotchev, "Bi-directional loop closure for visual SLAM," *arXiv:2204.01524*, 2022.

Author's contribution

Publication I — The study was motivated by the need to solve problems pertaining to robot manipulation and tool placement for ITER funded project. The candidate conducted a comprehensive study on Robot-World-Hand-Eye calibration techniques and proposed novel approaches. Additionally, he designed and conducted the experi-

ments, developed a toolkit, prepared an open-source dataset and drafted the manuscript. The authors Olli J. Suominen, Emilio Ruiz Morales, and Prof. Atanas Gotchev contributed significantly with invaluable technical guidance, recommendations and internal revisions of the manuscript.

Publication II — This study is an extension of the Robot-World-Hand-Eye calibration work and is focused on subsequent pose estimation of fiducial markers in the environment. The contributions of the authors are the same as in Publication I.

Publication III — The candidate contributed to the idea and methodology of the work, implemented and conducted experimental analysis, and drafted the manuscript. Olli J.Suominen and Prof. Atanas Gotchev polished the experimental process and supervised the study.

Publication IV — The original idea was proposed by the first author after observing the need for such a dataset while working on the project Spatial Sensing for Machines (SSFM). All the authors contributed significantly in developing the idea and providing valuable recommendations. The first author primarily designed the experiments, prepared the hardware, post-processed the data, set up experimental benchmarks, and drafted the manuscript. Ahmed Durmush developed the software for recording multiple streams of videos using RGB cameras. The first author and Olli J. Suominen diligently recorded the data in various sessions over a period of a year. Dr. Jussi Collin provided valuable guidance in the preparation of the ground truth poses using GNSS and IMU data. Dr. Jari Yli-Hietanen, Dr. Sari Peltonen, and Prof. Atanas Gotchev supervised the dataset preparation and revision of the manuscript.

Publication V — The work is based on the challenges mentioned in the Publication IV. All the authors contributed in part towards the inception of the idea. The first author contributed towards proposing a solution, designing the experiments, implementing the algorithms, and drafting the manuscript. Dr. Sari Peltonen and Prof. Atanas

Gotchev actively oversaw the experimentation, discussed the results, and reviewed the technical ingredients.

1 INTRODUCTION

This chapter provides a brief overview of the topic and the motivation for researching the topic. Thereafter, the research questions are provided that drove and directed the research findings for this dissertation. Finally, the chapter explains the structure of the thesis and visually relates the preliminaries to various topics in this dissertation.

1.1 Motivation

The use of a robotic systems, which was once a speculation, has become a key contributor to our work and individual personal lives. The story of the transition from the use of bulky robotic manipulators in manufacturing industries to compact commercialized products has been rapid and ubiquitous due to the inherent capability of the machine to achieve high accuracy with better efficiency compared to manual work. The rapid adoption is motivated primarily by the shift from pre-programmed automation to increased autonomous operation capability. This can be broken down, into three categories, based on the capacity to perceive and interact with the environment. The first category includes operation in a highly structured and static environment. The second class is built on the perception capability and attempts to attain semi-autonomy in real-world scenarios (e.g., self-driving/autonomous cars). The last category yearns towards total autonomy, i.e., attaining a stable performance by operating continuously, learning, and adapting to the changes in the environment. The demand for autonomous machines and vehicles can be found in many application fields, such as transportation, heavy work machines in forestry, construction, mining, and even in maritime operations. Irrespective of the nature of the task, medium of travel, mode of sensing, and level of autonomy, all these vehicles share a common base from where they branch off to their modular and industry-specific tasks. The core commonality they share is the ability of the system to localise itself against some reference system by perceiving the environment, possibly recognizing

relevant information, and estimating its pose against a reference frame in the world. In a non-static world, the robotic system effectively has to respond to the changes in the environment that may be caused by movement of the robot, some moving object in the scene, physical contact, and interaction with an object, or by any other dynamic entity. Such dynamic environments strongly motivate the need for robotic systems and algorithms that can perceive, plan and act accordingly. The scope of this thesis spans the extent of the second class of autonomy where we may face unstructured and dynamic environments and attempt to perceive, localise, and respond based on observations. More specifically, we address the development and deployment of pose estimation methods for various applications ranging from indoor robotic arm manipulation to outdoor Simultaneous Localisation and Mapping (SLAM) to assist autonomous vehicles and advanced driver assistance systems (ADAS). The study aims at providing concrete theoretical contributions along with innovative solutions for applied industrial problems, assisted and validated by novel high-quality datasets and benchmarks.

1.2 Research Questions

The questions that motivated the research and subsequent contributions, summarized in this thesis, are as follows:

(I) Can we improve the performance of visual servoing by improving the individual components such as Hand-Eye calibration? Does formulating the problem in an alternative manner yields better results, and can we segregate the sources contributing to the uncertainty?

(II) Can modeling geometric constraints based on manipulator kinematics aid in improving the pose estimates for visual servoing? Would Multiview pose estimation provide any significant improvement to a single shot approach?

(III) Can we segregate dynamic entities from the map in Visual SLAM and use it as a prior knowledge to discard outliers when estimating the next pose in odometry and updating the map?

(IV) What are the limitations of existing datasets for pose estimation and SLAM? Can new datasets add value to the validation and challenge the performance of state-of-the-art methods?

(V) Is loop closure possible from other perspectives, such as in a view from the opposite direction, for vehicular motion, and can this aid in improving visual odometry results?

1.3 Structure of Thesis

The thesis is organized in five chapters. Chapter 2 presents the required theoretical preliminaries and provides the research background that is relevant to the topics. The author reviews the core concepts of camera modeling, camera pose parametrization and estimation, scene feature extraction, and correspondence. Furthermore, the chapter provides state-of-the-art on the topics of Hand-Eye calibration, visual servoing, visual odometry, place recognition, and SLAM.

The main contributions of this thesis are presented in Chapters 3 and 4. Chapter 3, summarizes contributions towards enhancing the ability of autonomous operation of robotic manipulators. The chapter formulates the calibration problem and summarizes the contribution toward the calibration phase from Publication I. Subsequently, it discusses the pose estimation and manipulation of the robotic arm from a constrained Multi-view perspective, as proposed in Publication II. Chapter 4 is dedicated to the contributions made in Visual SLAM. The chapter follows and discusses the constituent blocks of the SLAM pipeline. The contributions toward odometry and mapping in a dynamic environment are summarized from Publication III. This is followed by an introduction, empirical analysis, and discussion on the challenges provided by the FinnForest dataset proposed in Publication IV. Chapter 4 concludes with a novel approach proposed in Publication V toward place recognition and pose estimation for the case of Bi-directional loop closure.

Finally, Chapter 5 summarizes the work and offers concluding remarks. Figure 1.1 categorizes and illustrates the relationships among the topics discussed in this thesis and the research contribution from Publications I – V.

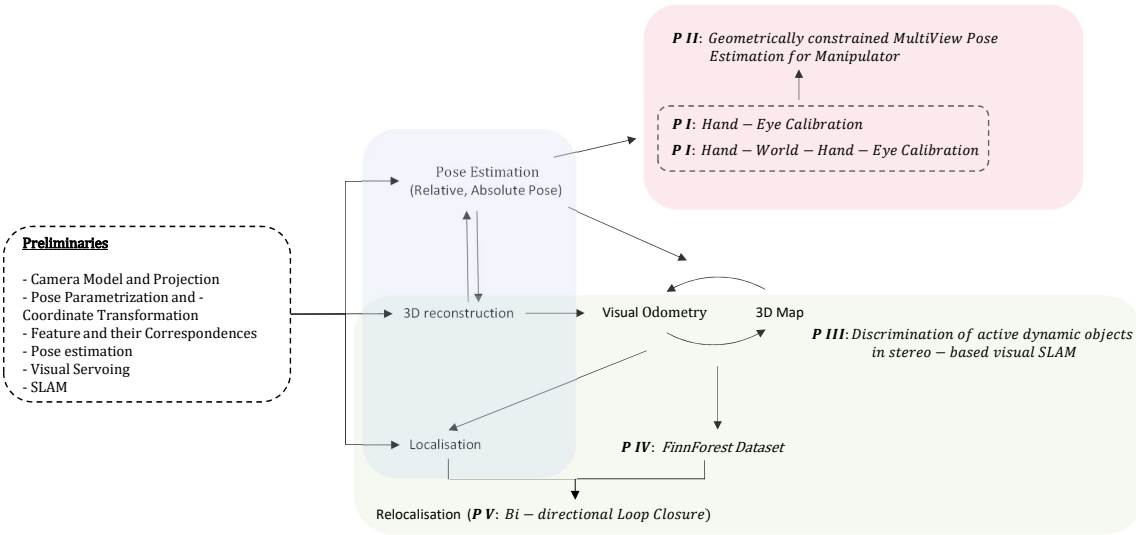


Figure 1.1 Overview of the topics in this thesis.

2 BACKGROUND

This chapter presents the general theoretical framework which forms the basis for developing the novel contributions presented later in the thesis. It briefly overviews the core concepts of camera geometry such as camera modeling, pose parameterization, and estimation and progresses toward broader topics; namely visual servoing and SLAM. Furthermore, it overviews the most recent research studies on the topics of Hand-Eye calibration, visual servoing, visual odometry (VO), loop closure, and visual SLAM.

2.1 Preliminaries

2.1.1 Camera Model and Projection

In general, most imaging and computer vision applications relay information to consumers in two-dimensional (2D) form either as direct images or with information overlaid over the images to form augmented reality. The process of two-dimensional image formation from a three-dimensional world itself follows specific geometric relations that are compiled to form a camera model.

The reduction in the dimension of 3D information to 2D is the result of a process known as projection in which a point in space is drawn from a 3D world point through a fixed point in space, the center of projection. A plane is assumed/placed in space in the path of the ray. The intersection of the ray with the plane represents the image point for the 3D world point on the image plane. However, if the 3D structure itself lies on a plane i.e., it is a 2D object then there can be no drop in dimension for the object. This basic relation forms the basis of modern cameras that deploy lenses as an intermediate medium to focus multiple rays and direct them to a film or digital sensor, producing an image of the point. Ignoring such effects as focus and lens thickness, a reasonable approximation is that all the rays pass through

a single point, the center of the lens to form a Pinhole Camera model. However, in practice, it is essential to incorporate these effects, especially radial distortions (due to lenses) to accurately model the image formulation. The corrections are estimated in a calibration step and can be applied later on the images.

Let's model the world as a 3D projective space, equal to $\mathbb{R}^3$ along with points at infinity and the image is the 2D projective plane $\mathbb{P}^2$. Then by definition projection is the mapping of the world points from $\mathbb{R}^3$ to $\mathbb{P}^2$. The world points can be represented as homogenous coordinates of the form $W = (X, Y, Z, 1)^T$. If we have the center of projection at the origin $(0, 0, 0, 1)^T$ then the transformation is given as

$$\begin{pmatrix} w\,u \\ w\,v \\ w \end{pmatrix} = M_{3\times4} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

Here, M is the transformation matrix, known as the Camera Matrix, that linearly maps the world points to image points u and v with a scale factor w. Moreover, if all the world points lie on a plane (effectively Z = 0), then the linear mapping reduces in dimension and $M_{3\times4}$ is replaced by the Homography Matrix $H_{3\times3}$.

In this thesis, we use the widely adopted Pinhole camera model for which the Camera matrix M is constructed with the aid of Figure 2.1.

As before, the center of projection is considered to be the origin of a Euclidean coordinate system. The image plane is defined at f and termed as the focal plane. For the pinhole camera model, a point in camera coordinate space with coordinates $W_c = (X_c, Y_c, Z_c)^T$ is mapped to the point on the image plane where a line joining the point W_C to the center of projection C meets the image plane.

The center of projection is also termed the camera center or the optical center. The line from the optical center perpendicular to the image plane is called the principal axis or principal ray. The point where the principal axis meets the image plane is called the principal point p. The plane through the camera center parallel to the image plane is called the principal plane of the camera. In the above relation, the camera center is placed at the coordinate origin. [1]

In practice, for most cameras, pixels are arranged in a grid with the indices starting at the top left corner $(0, 0)$ and not from the center of the image. The offset can be

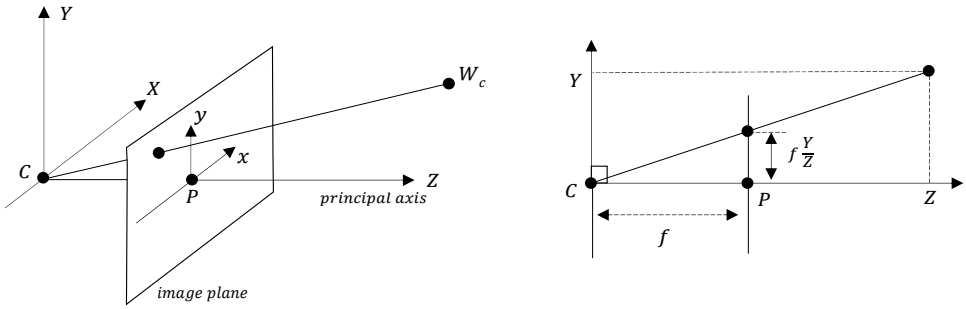


Figure 2.1 Pinhole camera geometry relating a world point and the image point on image plane.

accommodated by translating the projected points to the assumed center of the digital camera. The accommodated mapping relationship can be written as

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x X_c + Z_c p_x \\ f_y Y_c + Z_c p_y \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}.$$

The terms f_x and f_y are the focal lengths of the camera in terms of pixel dimensions in the x-axis and the y-axis, respectively, and (p_x, p_y) are the coordinates of the principal point. The matrix containing these terms is generally known as camera intrinsic matrix K. A more general form includes a skew parameter s, however, it is zero for most of the modern cameras. We use a subscript c with the 3D points to emphasize that the points are in the *camera coordinate frame*. This essentially means that the camera is assumed to be located at the origin of a Euclidean coordinate system with the principal axis of the camera pointing straight down the Z-axis.

2.1.2 Pose Parameterization and Coordinate Transformation

A common task in computer vision and robotics is to identify specific objects in an image and determine each object's position and orientation within a specified coordinate system. The combination of position and orientation is called a *pose*. Similarly,

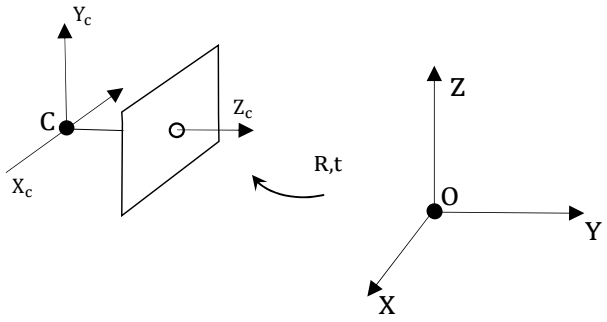


Figure 2.2 The transformation between the world coordinate frame and camera coordinate frames.

camera pose refers to the position and orientation of the camera in relation to a reference/world coordinate system. Any single point in space has to be transformed into the camera coordinate frame and projected using the camera model: from 3D space coordinates to 2D image coordinates. The homogeneous transformation consists of a rotation and translation part as shown in Figure 2.2 and is expressed as

$$W_c = [R \;\; t]\, W, \tag{2.1}$$

where R is a 3×3 rotation matrix and t is a 3×1 translation vector. This transformation, generally, expresses the pose of the camera against some reference and is termed as camera extrinsic. Hence, for a camera located anywhere in the world with known camera extrinsic, the camera perspective projection/view can be obtained by

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x X_c + Z_c p_x \\ f_y Y_c + Z_c p_y \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K[R \;\; t]\, W,$$

2.1.3 Features and Their Correspondence

Feature identification and its correspondence is the problem of identifying key points/regions, in two or more different images, which correspond to the same world point. Feature detection, its correspondence, and tracking are the building block for many

vision-based applications such as place recognition, structure from motion, etc. Extensive research has been conducted on this topic and many variants and approaches have been proposed over the years. The selection of a suitable approach greatly depends on the system's requirements, type of sensor, and challenges posed by the working environment.

Fiducial Markers Typically, applications such as photogrammetry, which requires accurate pose estimation, make use of carefully manufactured Fiducial Markers. Such markers make use of the projective invariant features such as corner points, lines, and planes. The most common pattern used, especially for camera calibration, is checkerboard. Checkerboard can provide multiple accurate corner points in images whose scale for world points is already known. However, a checkerboard pattern is only useful if it is viewed in its entirety. It is next to impossible to automatically identify which calibration point is which unless the full pattern is visible. Moreover, it certainly fails in the case of partial visibility due to clipping against the image boundary, and due to partial occlusion. To overcome this limitation alternative approaches, make use of quadrilateral blocks that encode a binary pattern for identification and error correction. The approach has many successful variants and is widely used for Augmented Reality applications. The most common among these are ArUco [2], AR Toolkit [3], and ChArUco [4]. Cal-tag [5] follows the principles of the AR toolkit; however, the implementation claims lower corner detection inaccuracies and a more flexible licensing. Additionally, some applications such as photogrammetry prefer circular marker-based tag designs. This is due to the fact that the position of the ellipse's centroid can be more accurately retrieved in comparison to the position of the center of a square. For a square, we estimate the center using its four corner points, whereas more pixels along the perimeter are used for fitting an ellipse and estimating its center. This makes the center of an ellipse statistically more stable and thus more accurate. Among the ellipse based markers, V-STAR [6], Rune-Tag [7], and PI-Tag [8] are the most prominent studies due to their stable implementation and occlusion resilience.

Feature Points In some applications such as mobile Augmented Reality and visual SLAM , it is desirable to extract features from the structures and textures in the images. These are considered *key points*. They have to be well localised in image space and should be detectable with high repeatability. These feature points can be defined in different ways such as using brightness of regions in images (analyzed

through image derivative) or boundary extraction (through edge detection or curvature analysis). These feature points are then further encoded using feature descriptors for effective and efficient matching. Lowe [9] proposed the famous scale invariant detector and descriptor named Scale Invariant Feature Transform (SIFT) based on the Laplacian of Gaussians (LoG) transform . The algorithm is quite powerful, however, computationally expensive. In 2006, Bay et al. [10] presented the Speeded Up Robust Feature (SURF) detector and descriptor using Haar wavelet transform and integral image. Other well-known feature detectors and descriptors include Local Binary Pattern (LBP) [11], Binary Robust Invariant Scalable Key points (BRISK) [12], Binary Robust Independent Elementary Features (BRIEF) [13], Oriented FAST, and rotated BRIEF (ORB) [14].

Deep Features Recent studies exhibit a strong trend in the use of learning-based approaches for feature detection and description. Learning-based approaches can be used to extract local features similar to SIFT, SURF, etc., or global descriptions of the entire image. The results obtained are shown to exhibit better accuracy and efficiency compared to conventional approaches for image retrieval [15] and object recognition tasks [16]. McManus et al. [17] proposed learning features from image patches and called them scene signatures which were used to match and retrieve scenes when the appearance of the scene changes. While this method was accurate, it required an extensive training phase with data from the test environment under all possible environmental conditions. There are studies that directly use the intermediate representations that are learned by a model during training [15], [18]. The underlying concept is that features from higher layers of a Convolutional Neural Network (CNN) encode semantic information about a place while features from the lower layer encode more descriptive information about the geometry of the scene. A careful combination of these descriptions from various intermediate layers can provide a powerful description of the image.

A visual illustration is provided in Figure 2.3 where we extract different types of local information from an image. Similar approaches are employed in this thesis. We adopt Fiducial Marker-based approaches for our Hand-Eye calibration and Camera Pose Estimation work in Publications I and II. For the studies published in Publications III and IV, we use various local feature extractors for keypoint tracking, camera pose estimation, and sparse reconstruction tasks. Finally, a deep learning-based approach is utilized in Publication V for learning global image description for

(a) (b) (c)

Figure 2.3 Extraction of feature information from a sample images using (a) ArUco Marker detected in the scene (b) ORB features detected in the image (c) Activation map of the deep features, extracted using VGG-16 network, overlaid over the sample image.

place recognition.

2.1.4 Pose Estimation

Camera pose estimation with respect to a target object/scene has been widely researched in the fields of computer and machine vision, photogrammetry, and robotics. Accurate pose estimation is needed in numerous applications such as camera calibration [19], localisation [20], reconstruction [21], robot visual servoing [22], and augmented reality [23]. Recent advances in these fields have greatly enhanced the accuracy and efficiency towards a wide range of applications. Even with much progress, there remains a need for improvement in application-specific methods. For example, a reconstruction technique that is ideal for achieving a visually pleasing result might not yield accurate results in localisation. In any case, the use of the appropriate method is heavily dictated by the type of data and application under consideration.

3D-3D Correspondences: In the case when we have point clouds of a 3D structure from a scene acquired at different locations in the environment, we can use point registration algorithms like Iterative Closest Point (ICP). ICP plays a crucial role in localisation and mapping in modern mobile robotics [24], [25]. The algorithm estimates a 3D rigid transformation that aligns a reading point cloud to a reference point cloud (or more generally a model or a surface). It is widely used for registering the outputs of 3D scanners, which typically only scan an object/scene from one direction/position at a time. The standard ICP starts with two sets of points or point clouds and an initial guess for their relative rigid-body transform. The initial guess helps to reach convergence quickly and is naturally provided in mobile robotics by odometry [26], [27] based on wheel speeds, inertial sensors, or vision.

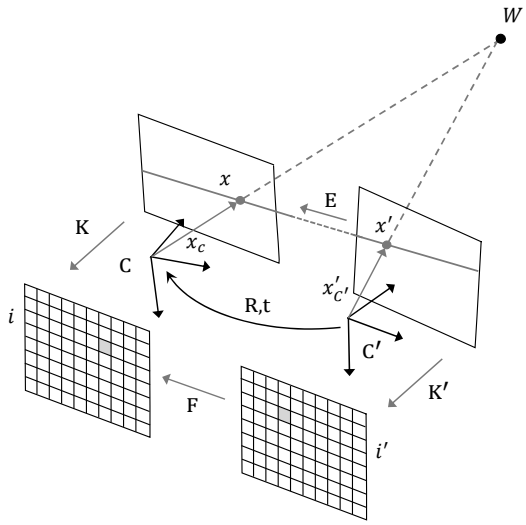


Figure 2.4 Relationship of Essential matrix and Fundamental matrix with the world point, normalised camera coordinates and image points.

The algorithm then iteratively refines the transform by repeatedly generating pairs of corresponding points in the clouds and minimizing a user-chosen error metric. In general, ICP approach can be extended to be used with Line Segment Sets, Implicit Curves, Parametric Curves, Triangle Sets, Implicit Surfaces, Parametric Surfaces, etc [28]. Robust variants of ICP can be used with complex and noisy data to reconstruct 2D or 3D surfaces from different scans and localise robots to achieve optimal path planning.

2D-2D Correspondences: This set of methods aims at finding the relative pose of the camera by estimating the plane-to-view mapping. In other words, it employ a mapping between two planar projections of an image using a 3x3 transformation matrix in a homogenous coordinate space, which is then decomposed to obtain the pose. This set of methods is known as *Homography Decomposition* (HD) methods [29]–[31]. A more generalized form of Homography is the *Essential matrix* [32]. In contrast to a Homography in which the points in the image space are coplanar, an Essential matrix is able to relate any set of points in one image to those in another taken by the same camera. Since the Essential matrix is more generic, calculating the Essential matrix requires more points than calculating a homography. The *Fundamental matrix* is a further generalization of the Essential matrix where the assumption of calibrated cameras is removed [33]. The Essential matrix operates on image points expressed in normalized coordinates while the Fundamental matrix is directly related to pixel coordinates. Figure 2.4 illustrates the geometric relationship.

Let W be the world point viewed by a camera at two positions C and C' with

camera intrinsics K and K', respectively. The corresponding points x and x' in the normalized image planes are related by the relative transformation $[R|t]$ between C and C'. Then, in terms of vectors, the equation for the epipolar plane can be written as

$$(x'_{C'} \times t) \,.\, (R\, x_C) = 0.$$

This can be re-written in term of matrices as

$$x'^{T}_{C'}\, [t]_\times\, R\, x_C = 0$$

Here, $[t]_\times$ is a skew symmetric matrix of the baseline t. The Essential matrix is thus defined as $[t]_\times\, R$. Then by Longuet-Higgins [34] equation, we can directly replace the viewing rays/vectors with their corresponding intercepted points from the normalized image plane, thus giving us the relation

$$x'^{T}\, E\, x = 0. \tag{2.2}$$

The Essential matrix E represents the epipolar constraint on the corresponding normalized points. The epipolar constraint on image points is naturally connected to the Essential matrix by the calibration matrices K and K'. From Figure 2.4, we can extract $x_C = K^{-1}\, i$ and $x'_{C'} = K'^{-1}\, i' \implies x'^{T}_{C'} = i'^{T}\, K'^{-T}$. Replacing the terms in Equation 2.2 gives us

$$i'^{T}\, K'^{-T}\, E\, K^{-1}\, i = 0.$$

This defines the Fundamental matrix $F = K'^{-T}\, E\, K^{-1}$, giving us the epipolar constraint relation in the image space as

$$i'^{T}\, F\, i = 0. \tag{2.3}$$

2D-3D Correspondences: The final category of pose estimation methods employs 2D-3D correspondences to estimate a rigid transformation. This method is often referred to as *Perspective from n-points* (PnP) [35] and is able to estimate the pose of a calibrated camera from a set of 3D points in the world and their 2D observations in the image. PnP methods minimize the cost function of the correspondence transfer error. This error refers to the difference between predicted and measured

positions of point correspondences. The PnP problem in its minimal form is called P3P and can be solved with three-point correspondences. P3P, however, produces four possible solutions based on three-point correspondences alone. When noise levels are low, a fourth correspondence is effective in removing ambiguity.

Alternatively, many studies consider data from multiple perspectives, also referred to as multi-view, for improved accuracy. By observing the feature points or regions of interest from many different perspectives, a model can be created more accurately and coherently. Through robust tracking, we can link features across views and align them through relative geometric transformations. With the ability to efficiently and accurately match feature points across multiple views, many studies have opted for *Structure-from-Motion* (SFM) based approaches, also known as Full Multi-View. Martinec and Pajdla proposed a SFM based method that computes the rotation and translation separately for relative views [36]. Subsequently, a bundle adjustment approach is used to optimize the relative poses and distribute the pose errors evenly across the poses.

We have used all the aforementioned approaches of pose estimation at different stages of research that contribute to this thesis. Our calibration of the cameras and certain studies in the state-of-the-art used for comparison employ 2D-2D correspondences (Publications I-II). Additionally, we employed 2D-3D correspondences in Publication I for our proposed approach and Publication IV for benchmarking other studies. Thereafter, we found the 3D-3D correspondence based relative pose estimation approach more suitable for our proposed odometry and mapping methodology in Publication III.

2.1.5 Visual Servoing

Visual servoing is an approach of controlling and manipulating a robot with the aid of camera vision as the primary sensing mechanism to enhance the robot's control mechanism. In general, visual servoing can be categorized based on configurations of its constituent blocks [37]. The first segregation of the category is based on the perception mechanism. Visual servoing directly based coordinates of image features, such as lines or moments of regions, is known as *Image based Visual Servoing* (IBVS). The approach works on the error between current and desired features on the image plane and does not involve any estimate of the pose of the target. In IBVS, large rotations produce difficulties, which are solved through a phenomenon

called *camera retreat* [38]. The second approach is a model-based technique known as *Position/Pose-based Visual Servoing* (PBVS) [37], where the pose of the object of interest is estimated with respect to the camera and then a command is issued to the robot controller for manipulation. The approach involves extraction of the image features as well, however, they are subsequently used to estimate 3D or 6D information (pose of the object in Cartesian space). A combination of the above two approaches can also be used which is termed as the *Hybrid approach*. The second categorization is based on the observation model of the system. A system that only observes the target object is known as *endpoint open-loop* (EOL) system or *Eye-in-Hand* configuration, since the camera is affixed to the robot in such cases. In contrast, if the system observes both the target object and the end-effector of the robot, it is termed as *endpoint closed-loop* (ECL) system or *Eye-on-Hand* configuration. Both configurations have their own advantages and limitation which dictate their adoption. Generally, for mobile robots it is favorable to use EOL/Eye-in-Hand configuration since ECL requires a constrained work space.[39]

The final set of classification is based on the control architecture. The first subcategory is a hierarchical control architecture, where the vision system provides setpoint inputs to the robot controllers and the system makes use of joint feedback or end-effector pose to internally stabilizing the robot. Such an architecture is referred to as a *dynamic look-and-move* system [40]. On the other hand, *Direct visual servoing* [41], entirely replaces the robot controller with a controller that computes command input directly for the manipulator joints, thus stabilizing the mechanism solely through vision. Generally, most of the approaches favor dynamic look-and-move approach over Direct visual servoing due to several factors. First and foremost, the relatively low sampling rates from vision sensors make direct control of a robot end-effector an extremely challenging control problem due to its complex and nonlinear dynamics. Additionally, many robots already have a method for estimating Cartesian velocity or incremental position information. As a result, the visual servo system is easier to develop, and it is also more portable. Moreover, the use of look-and-move bypasses the need to explicitly handle the kinematic singularities of the system from the visual controller, which normally has a specialized mechanism, thus greatly simplifying the operation.

To align the research work targeted for the dissertation with the requirements of partnering industries, we adopted and investigated systems that employ Eye-in-Hand

configuration and use PBVS with a dynamic look-and-move control system.

2.1.6 Odometry from Pose

The word odometry finds its roots in Greek literature and refers to measuring routes [42]. As evident from the name, in odometry, data is gathered from motion sensors to estimate a change in position over time. The term VO is used when the motion is estimated using visual sensors, such as RGB camera, time-of-flight, etc., in any configuration for example monocular, stereo, omnidirectional, etc. Odometry is not exclusive to visual sensors, other sensors such as wheel odometer, Inertial Navigation System (INS), GNSS, magnetometer, etc. can also be used to estimate the change in pose [43]. VO is based upon the pose estimation step where the relative pose estimates are accumulated over time, with respect to a reference coordinate system, to localize a robot. Visual sensors are widely adopted for odometry since they offer a good balance among cost, reliability, and implementation complexity. Uneven terrain or other unfavorable conditions do not affect VO. Additionally, VO works robustly in GPS-deficient environments [44]. In comparison to wheel encoders and low-precision INS [43], local drift under VO is much lower. Moreover, cameras offer a rich amount of information that opens unbounded opportunities for semantic understanding of the environment which is crucial to autonomous operation. Nonetheless, VO has its own set of limitations where the performance is often strongly affected by extreme weather and light conditions. State-of-the-art approaches achieve accurate results by integrating VO with GPS and INS to complement each other's limitations.

2.1.7 Mapping

The robotic mapping problem is that of constructing an accurate spatial model of the robot's surroundings. To create a map, local sensor inputs of the robot, or local maps, could be registered into a common coordinate system if the poses of the robot are known. The poses can be retrieved through VO. Unfortunately, mere odometry estimates can suffer from imprecision and drift over a period of time. A more robust technique is the use of SLAM where the maps are simultaneously generated along with localisation. In contrast to visual odometry where the relative poses are estimated from images, SLAM approaches prefer to estimate the pose by

projecting the active landmarks/key points from the map onto subsequent temporal images [45]. The key points are feature points in images that are unique, robust, and consistently seen over a period of time. The features are triangulated and added to the map along with a track of their feature description. If an image has a significantly high amount of unique key points, then it is regarded as a keyframe. A keyframe in turn can be used for place recognition and subsequent re-localisation. Hence, the localisation and mapping are performed in a continuous cyclic operation. Depending on the task at hand, the SLAM pipeline can generate a sparse map [46] or a dense map [47]. A sparse map is generally populated with 3D points pertaining to the landmarks and stores the minimum representation of the 3D structure of the scene for effective SLAM. On the other hand, dense maps contain a rich spatial representation and provide a continuous structural form of the environment. Dense maps are crucial for understanding and interacting with the objects in the world and have numerous applications in AR.

2.1.8 SLAM

Early studies in robotics approached localisation and mapping independently. However, later researchers observed the cyclic interdependency of localisation and mapping for mobile robotics. This is true since a map can only be created when the robot's pose is known against a reference. Similarly, we need a precise map representation to provide that reference to perform localisation. However, both tasks need to be performed simultaneously, hence, the term Simultaneous Localisation and Mapping. The interdependency has a deteriorating effect as well. It is particularly difficult to perform SLAM since inaccuracies in the ego-motion estimate will occur which will propagate towards the generation of the map. A poor ego-motion estimate will result in a poor map quality, which biases the next ego-motion estimate and so on. Many studies adopt different sensors such as Camera [46], Radar [48], LiDAR [49], Sonar [50], Global Navigation Satellite System (GNSS) [51], and Inertial Measurement Units (IMU) [52] to perform SLAM. These sensors can be used independently or in combination to achieve better results. Many companies that strive for an autonomous vehicle adopted LiDAR as the primary sensor for SLAM since LiDAR can be used to estimate odometry and build a sparse 3D map of the environment using relatively noise-free measurement. However, the technology has its limitations since the hardware itself is relatively expensive and fragile compared

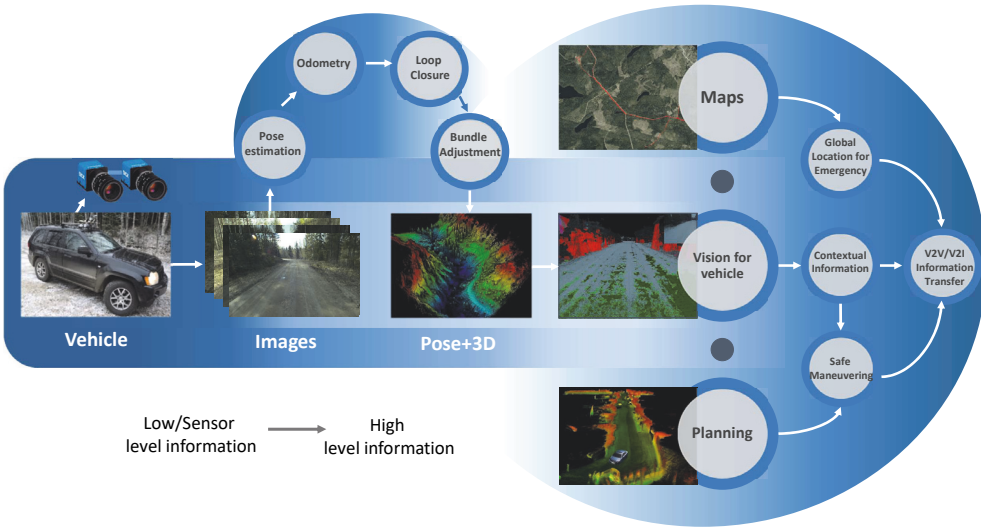


Figure 2.5 A pipeline of operations that illustrate the processing blocks of autonomous vehicle and the achieved functionalities. Here, V2V and V2I indicate vehicle–to-vehicle and vehicle-to-infrastructure communication, respectively.

to other sensors and not suitable for mass production [53]. On another size, the current advancements in imaging technologies and algorithms have made the use of cameras for SLAM (Visual SLAM) affordable, effective, flexible and modular. Their combination with high processing capabilities opens the possibility to develop new visual SLAM methods with untapped potential. Using merely visual information, one can estimate the position, orientation and speed of the vehicle, enforce safety regulations, understand the environment as a human does, and take safety critical decisive actions. Such a system will be affordable yet robust and effective. A typical Visual SLAM framework constitutes of the following nodes: 1) estimation of vehicle position and orientation (odometry), 2) mapping of the environment, 3) refining and distributing error (Bundle Adjustment), and 4) re-identification of a previously visited place (loop closure). The solutions of these problems form blocks of an effective SLAM pipeline (see Figure 2.5).

We tabulate a list of advantages and limitations of various sensors that can be used for SLAM in Table 2.1. It can be observed that different sensors and their combinations provide different advantages and disadvantages. However, in order to focus on solving crucial problems that inhibit the capabilities of SLAM, we restrict the scope of this thesis and contribute to methods pertaining only to visual SLAM. The research questions III, IV, and V are associated with visual SLAM and are addressed

in the Publications III, IV, and V respectively.

2.2 State-of-the-Art and Challenges

2.2.1 Hand and Eye Calibration and Pose Estimation for Visual Servoing

In order to relay information observed by the camera to the robotic manipulator for successful operation, it requires that a geometric relation is known between the robot hand/end-effector and the optical frame of the camera. The problem can be posed as *Hand-Eye* or *Robot-World-Hand-Eye* (RWHE) calibration. The former estimates the homogenous transformation between the end effector and the camera [54]–[57] while the latter additionally finds the relation between the robot base and the target [58]–[61]. A renowned study in RWHE calibration was conducted by Shah [62]. The author proposed a closed form solution involving Kronecker product of the rotational matrices and decomposing the result using singular value decomposition (SVD) to yield rotation and subsequently translation estimates. The Kronecker product enables the estimation of the optimal transformation in these cases. However, the resulting rotational matrices might not follow orthogonality. To compensate for this issue, the best approximations for orthonormal rotational matrices are obtained using Singular Value Decomposition (SVD). Separate approaches suffer from a core limitation that the errors in the orientation step propagate towards the translation estimation step as a result deteriorating the position estimate.

In contrast, Tabb and Khalil [63] tackled the problem of Hand-Eye calibration from an optimization perspective and estimates the orientation and translation components simultaneously. Moreover, the study investigates the effects of using different rotation representations, such as Euler, rotation matrix, and quaternions, on the accuracy of estimates.

Following Hand-Eye calibration, the subsequent step towards robot manipulation is to acquire the target pose. We state three state-of-the-art studies that were used for comparison, with the method proposed in Publication II. Using Collins and Bartoli's [64] method, the problem is analytically solved after the homography is computed. The method they developed is called IPPE, or infinitesimal plane-based pose estimation. According to this proposition, even if the estimated homography is noisy, at some regions of the model plane, it will still approximate the true trans-

Table 2.1 A comparison of sensors for SLAM

Type	Sensor	Advantages	Limitations
Range	LiDAR	- Clean measurements - Direct 3D information - Smooth maps - Sparse representation	- Expensive - Fragile - Power consumption - Lack of semantic information - limit on LiDAR's signal strength for safety
	Radar	- Reliable detection unaffected by weather	- Low range - Low angular resolution - High noise
Camera	Monocular	- Physically smallest - Low power consumption - Cheap - Minimal Calibration	- Scale unobservable - Scale drift - 3D only from multi-view - No map under pure rotation - Non-trivial SLAM initialization
	Stereo	- 3D from one stereo frame - Trivial SLAM initialization	- More processing compared to monocular - Extrinsic calibration
	RGB-D	- Directly provides dense depth map - Trivial SLAM initialization - Dense maps	- Active Sensor (interference) - Range limitation - Only indoors - Complex calibration with other sensors - Power consumption
Camera + IMU	IMU	- High frequency - Interframe motion estimation - Pitch and Roll are observable - Scale for monocular SLAM	- Varying sensor biases - Gravity vector compensation - Observability issues - Visual-Inertial calibration - Synchronization
Camera + GNSS	GNSS	- Global consistency - Measurable quality of signal	- Only Outdoors - Low accuracy in urban area

formation between the image and the model. Points are taken from those regions to solve for the pose using first-order partial differential equation (PDE). In contrast, Collet and Srinivasa [65] proposed a modified version of full multi-view, referred to as introspective multi-view. First, a single-view method is used to estimate object and camera poses. Following the initial estimates, the points are grouped, and the outliers removed from the matches. A final step involves reoptimizing the poses based on the filtered matches using bundle adjustments. The authors assert that the approach achieves a good balance between computation speed and accuracy. Geiger et al. [66] presented a stereo vision-based approach to generate dense 3D maps in real-time from high-resolution stereo sequences. The authors claim that their method yields accurate pose estimates and subsequent odometry by constraining the objective function. The pose is estimated by reprojecting the triangulated world points onto the previous stereo pair and optimizing them using the Gauss-Newton algorithm.

The presence of numerous approaches makes it difficult to select a suitable approach for any use case. Keeping this in mind and in accordance with our Research Question I, we conducted a detailed comparative study in Publication I to draw a comparison between various Hand-Eye and RWHE approaches. Additionally, we propose a new cost function for achieving better accuracy and provide a novel dataset for testing and validation with real and synthetic data. Moreover, we provide open-source code of the implementation to assist other users. Similarly, while exploring solutions for Research Question II, we were able to contribute to the pose estimation block for robot manipulation by proposing a geometrically constrained approach of camera pose estimation, which is detailed in Publication II.

2.2.2 Visual Odometry and SLAM

The need and demand for state-of-the-art research in this field is growing. According to KITTI Vision Benchmark Suite [67], among the leading methods for odometry estimation and sparse map generation are SOFT-SLAM2 [68] and ORB-SLAM2 [46]. The methods use carefully selected sparse features from images to estimate the pose of the camera from consecutive temporal views. A temporal registration of previous camera information to the following camera information is prone to drift over time. The drift is typically reduced using joint filtering. Additionally, the methods employ loop closure techniques to recognize previously visited places to reduce navigation errors. Until recently, the core assumption for SLAM odometry has been

that the environment under observation is static. As a result, this assumption leads to an inconsistent map, erroneous localisation, residual noise, and possible failure in registration of vehicle poses in dynamic environments. A few studies have attempted to incorporate the dynamic entities while performing SLAM. Keller et al. [69] and Whelan et al. [70] proposed a point-based fusion approach that overcomes the previously mentioned limitations of non-functionality in a dynamic environment. The approach utilizes ICP algorithm to compute the successive poses for the camera using weighted features. The approach is effective as it successfully reconstructs a static map while updating the dynamic entities in the map. The methods in [69], [70] rely on the dense and compact depth maps provided by an RGB-D camera, typically Kinect or its updated version. These RGB-D cameras are good solutions for an indoor environment; however, they perform poorly in an outdoor environment [71]. To overcome these limitations, we build upon the base concepts proposed in the earlier studies and extend the work to outdoor environment with the use of stereo vision. This facilitates the deployment of the concept in the field of autonomous vehicles/machines. The study successfully addresses our Research Question III and is detailed in the publication III.

2.2.3 Place Recognition for Loop Closure

State-of-the-art methods still face problems when it comes to recognizing a place from a perspective view that shows a high variation to its earlier corresponding perspective pairs from a nearby location. The likelihood of place recognition is maximized when the loop closure occurs in an environment that has been previously viewed from a similar perspective, for example, a vehicle traveling toward the north passes by the same location in the same direction. A wide range of strategies can be used to approach this problem. Some state-of-the-art techniques still find the classical approaches robust and effective. Such techniques employ feature descriptors such as SIFT [9], SURF [10], ORB [14], etc. to extract key points from images and further encode, compress, and organize them with descriptors and dictionaries such as bag-of-visual-words [72], vector of locally aggregated descriptors (VLAD) [73], or Fisher vector [74]. Fisher Vector adopts the Gaussian mixture model (GMM) to build a visual word dictionary and is assumed to encode more image information and outperform Bag of Words (BoW) in some computer vision tasks. VLAD, on the other hand, is a simplification of Fisher Vector and offers a trade-off between

performance and computational efficiency. In most cases, VLAD performs similarly to Fisher Vector with better efficiency. Generally, all these methods have experimentally proved to be effective in traditional urban settings and normal test scenarios. Alternatively, multiple recent studies have shown that the use of CNNs reduces complexity and improves accuracy. Models trained on very large datasets significantly outperform local descriptors such as SIFT in a variety of applications such as scene [75] and object recognition [76]. McGann et al. [17] suggested learning scene signatures from image patches, which would be used to match and retrieve scenes with appearance changes. However, this method required an extensive training procedure with data inclusive of the test environment under all possible situations. In some studies, the intermediate representations that are learned at different layers from an object recognition dataset are directly used for scene identification [15], [18]. Generally, it is observed that feature information acquired from higher layers of a CNN encodes semantic information about a place, while feature information acquired from the lower layers encodes finer descriptive information about the geometry of the scene. The authors in [15] experiment and tune various combinations of the encoded feature vectors and subsequently search for nearest neighbors, for a query image from the database, based on cosine distance. However, as stated earlier, these studies are designed for uni-directional motion cases and fail in high perspective variation cases such as bi-directional motion. To the best of our knowledge, the only study that addressed bi-directional loop closure attempted to solve the problem using panoramic images [77]. We believe that the use of panoramic images reduced the complexity of the problem by providing roughly similar views to a unidirectional case. The panoramic images were captured in an enclosed structural environment with a circular trajectory. This means that the reverse motion captures a substantial overlap of the forward motion scenes with some spatial offsets of regions in images with marginally different perspectives. Additionally, this is evident from the illustrations in their study [77] which only depict spatial changes in the scene. However, even in panoramic images, traditional methods like FAB-MAP [78] were not able to find potential matches for loop closure for the bi-directional case. This concern was raised in the Research Question V and significant contributions were made in Publication V which are also summarized in this thesis. We extend the place recognition and loop closure detection capability to bi-directional cases and test our proposed methods in both indoor and outdoor scenarios.

2.3 Summary

In this chapter, we have briefly overviewed the fundamentals of image formation based on the pinhole camera model and related camera pose representations, transformations, and projections. We have analyzed the strengths and weaknesses of feature selection approaches that are essential for identifying and tracking robust landmarks in the scene from its image in order to properly use them in our approaches. We have completed the overview with the topics of visual servoing and SLAM, which are in the focus of the thesis. Furthermore, we have critically analyzed the challenges and limitations of the state of the art in the fields of Hand and Eye Calibration, Pose Estimation for Visual Servoing, Visual Odometry, Place Recognition, and SLAM, with the aim to effectively address them in the thesis.

3 VISION FOR ROBOTICS WITH MARKERS

This chapter presents a thorough investigation into the various formulations of Hand-Eye, Robot-World-Hand-Eye, and constrained camera pose estimation methods for robotic arm manipulation. These constituent blocks of visual servoing are thoroughly investigated and the performance of the proposed contributions is compared against state-of-the-art methods. The contributions in this chapter are directed to address the research questions I and II. In addition, the chapter contributes a novel dataset for testing and validation of the calibration methods.

3.1 Problem Formulation for Calibration

We illustrate the two configurations that can be used for calibrating a camera with a robotic manipulator with the aid of Figure 3.1 extracted from Publication I. We will represent the homogeneous transformations matrix using T and various sub-indices throughout this article. The sub/super-scripts b, t, c, and w represent the coordinate frames for the robot base, robot tool, camera, and calibration pattern, respectively. The sub-indices i and j indicate the state of the system in time. The problem formulations are discussed in the following subsections.

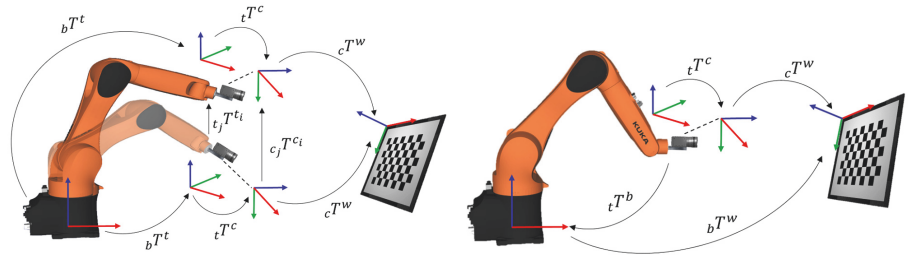


Figure 3.1 Formulations relating geometrical transformation for calibration; (a) hand–eye calibration; (b) robot-world-hand–eye calibration. (P.I)

3.1.1 Hand-Eye Calibration

One approach of formulating the calibration problem is $AX = XB$ illustrated in Figure 3.1a, where ${}_{b}T^{t_i}$ is equivalent to A_i and indicates the homogenous transformations from robot base to the tool center point (TCP)/end-effector. Similarly, ${}_{c_i}T^{w}$ is the equivalent of B_i and indicates the transformation from camera to the world/calibration pattern. This formulation makes use of the relative transformations A $({}_{t_j}T^{t_i})$ and B $({}_{c_j}T^{c_i})$ to estimate the unknown X or ${}_{t}T^{c}$, which is the required hand-to-eye homogeneous transformation. Then, from Figure 3.1a, we can form the following relationship

$$ {}_{b}T^{t_2}{}^{-1}\ {}_{b}T^{t_1}\ {}_{t}T^{c} = {}_{t}T^{c}\ {}_{c_2}T^{w}\ {}_{c_1}T^{w^{-1}} \leftarrow \quad ({}_{t_1}T^{c_1} = {}_{t_2}T^{c_2}). \tag{3.1} $$

Generalizing the relation in 3.1, we get

$$ {}_{t_j}T^{t_i}\ {}_{t}T^{c} = {}_{t}T^{c}\ {}_{c_j}T^{c_i}. \tag{3.2} $$

In order to ensure reliable results and a solution, it is recommended to record data for at least three positions with non-parallel movements of the rotational axis [79]. It is possible to estimate the unknown parameters in the relation 3.2 by directly minimizing the errors of the cost function

$$ \{q_{(t,c)},\ {}_{t}t^{c}\} = \underset{q_{(t,c)},t^{c}}{\operatorname{argmin}} \sum_{i=1,j=i+1}^{n-1} \|\tilde{n}({}_{t_j}T^{t_i}[q_{(t,c)},\ {}_{t}t^{c}]_{HT} - [q_{(t,c)},\ {}_{t}t^{c}]_{HT}\ {}_{c_j}T^{c_i})\|_2^2. \tag{3.3} $$

Here, the symbol $[\ ]_{HT}$ indicates homogeneous transformation representation. We minimize the cost function with the rotation parameters of the unknown ${}_{t}T^{c}$ in quaternion representation $q_{(t,c)}$ and translation ${}_{t}t^{c}$. The operation $\tilde{n}$ denotes the aggregation of the 4×4 error matrix into a scalar value by summation of normalized values of quaternion angles and normalized translation vector. The solver minimizes the residual scalar values with L2-norm using the Levenberg–Marquardt algorithm.

In light of recommendation of [63], we can also re-arrange Equation 3.3 in the

following manner

$$\{q_{(t,c)}, {}_{t}t^{c}\} = \underset{q_{(t,c)}, {}_{t}t^{c}}{\operatorname{argmin}} \sum_{i=1,j=i+1}^{n-1} \|\bar{n}({}_{t_j}T^{t_i} - [q_{(t,c)}, {}_{t}t^{c}]_{HT\ c_j}T^{c_i}\ [\tilde{q}_{(t,c)}, {}_{t}\tilde{t}^{c}]_{HT})\|_2^2. \quad (3.4)$$

The terms $\tilde{q}_{(t,c)}$ and ${}_{t}\tilde{t}^{c}$ are the quaternion and translation vector obtained from the inverse of ${}_{t}T^{c}$. In our results, we refer to the cost functions in Equations 3.3 and 3.4 as *Xc*1 and *Xc*2, respectively.

Since rotation and translation are solved simultaneously in Equations 3.3 and 3.4, these solutions fall under the category of simultaneous solution of Hand-Eye calibration. The objective function successfully converges to a solution without any initial estimates for the $q_{(t,c)}$ and ${}_{t}t^{c}$.

The objective function can also be expressed in terms of projection errors instead of direct pose optimization. In a wide variety of pose estimation problems, projection error minimization has been shown to produce promising results [80], [81]. Tabb and Khalil [63] presented a reprojection based cost function for the $AX = ZB$ formulation. Here, we generalize and expand the approach towards the case of the $AX = XB$ formulation. Let W be the 3D points in the world frame and P^{c} be the corresponding 2D points in the image plane. Then, the cost function for minimizing the reprojection errors of the 3D points from pose i to pose j is

$$\{q_{(t,c)}, {}_{t}t^{c}\} = \underset{q_{(t,c)}, {}_{t}t^{c}}{\operatorname{argmin}} \sum_{i=1,j=i+1}^{n-1} \|\bar{P}_j - \Pi(K, [\tilde{q}_{(t,c)}, {}_{t}t^{c}]_{HT\ t_j}T^{t_i}\ [q_{(t,c)}, {}_{t}t^{c}]_{HT}, P_i^{c})\|_2^2. \quad (3.5)$$

The relationship given in Equation 3.5 is referred to as RX here onwards.

A projection operation, Π, uses the camera intrinsic K and the camera extrinsic obtained using the homogeneous transformations given in Equation 3.5 to project 3D points from world space to image space. The cost function is minimized as before with the residual obtained from the difference of the observed 2D points $\bar{P}_j$, in the j^{th} image, and the corresponding projected points. Note that the reprojection error minimization approach is not invariant to the initial estimates used by the solver. The nonlinear optimization of reprojection error is more accurate when a decent initial estimate is provided.

3.1.2 Robot-World-Hand-Eye Calibration

The alternative formulation makes use of absolute poses. This formulation is expressed as $AX = ZB$ in literature and is illustrated in Figure 3.1b (extracted from Publication I). This formulation uses homogeneous transformations A ($_tT^b$) and B ($_cT^w$) from their respective coordinate frames. The unknown X ($_bT^w$) and Z ($_tT^c$) are the homogeneous transformations from robot base to the world frame and the end effector to the camera frame, respectively. It is important to note that the Hand-Eye transformation is referred to as Z in this formulation to adhere to representation in the literature. However, we will make use of conventional transformation notation with their subscript for the sake of clarity in representations. From Figure 3.1b we can form a straightforward geometrical relationship as:

$$_tT^b\,{}_bT^w = {}_tT^c\,{}_cT^w. \tag{3.6}$$

Similar to the previous cases, we can directly use the relationship in aforementioned equations to obtain $_tT^c$ and $_bT^w$ using nonlinear minimization of their respective costs

$$\{q_{(t,c),t}t^c, q_{(b,w)}, {}_bt^w\} = \underset{q_{(t,c),t}t^c, q_{(b,w)}, {}_bt^w}{\operatorname{argmin}} \sum_{i=1}^{n} \|\tilde{n}({}_tT_i^b\,[q_{(b,w)}, {}_bt^w]_{HT} - [q_{(t,c),t}t^c]_{HT}\,{}_cT_i^w)\|_2^2. \tag{3.7}$$

The parameterization adopted involves optimizing 14 parameters, of which the two quaternions and translation vectors contribute 8 and 6, respectively. Despite the higher number of unknowns in the RWHE formulation, it nevertheless yields higher accuracy due to stricter constraints on the geometry. Modern nonlinear solvers provide efficient approaches to solve optimization problems with large number of unknowns. As before, the objective function in Equation 3.7 can be re-arranged in the form

$$\{q_{(t,c),t}t^c, q_{(b,w)}, {}_bt^w\} = \underset{q_{(t,c),t}t^c, q_{(b,w)}, {}_bt^w}{\operatorname{argmin}} \sum_{i=1}^{n} \|\tilde{n}({}_tT_i^b - [q_{(t,c),t}t^c]_{HT}\,{}_cT_i^w\,[\tilde{q}_{(b,w)}, {}_b\tilde{t}^w]_{HT})\|_2^2. \tag{3.8}$$

The objective functions in Equations 3.7 and 3.8 are referred to as $Zc1$ and $Zc2$, respectively, in the study by Tabb and Khalil [63].

The objective function successfully converges to a solution for $q_{(t,c)}$, ${}_{t}t^{c}$, $q_{(b,w)}$ and ${}_{b}t^{w}$. However, the formulation requires initialization due to increased number of unknowns. A rough initial estimate, obtained from a fast closed-form method such as Tsai [54] or Shah [62], is sufficient since the formulation is not a high-dimensional optimization problem. As before, the formulation can be expressed as a reprojection error minimization problem as

$$\{q_{(t,c),t}t^{c}, q_{(b,w)}, {}_{b}t^{w}\} = \underset{q_{(t,c)},{}_{t}t^{c},q_{(b,w)},{}_{b}t^{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\bar{P}_i - \Pi(K, [\tilde{q}_{(t,c),t}\tilde{t}^{c}]_{HT}\ {}_{t}T_{i}^{b}\ [q_{(b,w),b}t^{w}]_{HT}, W)\|_2^2 \tag{3.9}$$

and is referred to as *rp*1 in [63]. The equation minimizes the reprojection of the 3D world points W onto the image space in camera frame, where $\bar{P}_i$ are the observed 2D points in the i-th image.

The formulation in Equation 3.9 has an added advantage that it no longer depends directly on the camera poses, which are required by the reprojection error based cost function of the formulation $AX = XB$ in Equation 3.5. If the camera intrinsic parameters are accurate enough, then the extrinsic can be indirectly computed as a transformation through ${}_{t}T^{c}$, ${}_{t}T^{b}$ and ${}_{b}T^{w}$ through the minimization of the objective function. Nonetheless, the reprojection error cost function presented for problem formulation $AX = XB$ proves to be more robust to robot pose errors given good images.

A marginal improvement in the results can be observed in various cases by using $log(cosh(x))$ as the loss function. $log(cosh(x))$ approximates $\frac{x^2}{2}$ for small value of x and $abs(x) - log(2)$, for large values. This essentially means that $log(cosh(x))$ emulates the behavior of the mean squared error, with better robustness to noise and outliers. Additionally, the function is twice differentiable everywhere and therefore does not deteriorate the convexity of the problem. The revised cost function, referred to as RZ hereafter, is

$$\{q_{(t,c),t}t^{c}, q_{(b,w)}, {}_{b}t^{w}\} = \underset{q_{(t,c)},{}_{t}t^{c},q_{(b,w)},{}_{b}t^{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\log(\cosh(E(x)))\|_2^2 \tag{3.10}$$

where $E(x)$ is the error in terms of difference between the observed points and the reprojected points.

3.1.3 Dataset for Calibration

We present a dataset in Publication I with multiple sequences to test and assess the performance of the calibration methods in laboratory and near-field settings. In this dataset, we acquire the data with multiple combinations of camera, lens, calibration patterns, and robot poses. Moreover, we provide both real and simulated data with synthetic images for calibration and testing. Table 3.1 provides an overview of the dataset. Excerpts from the dataset are provided in Figure 3.2.

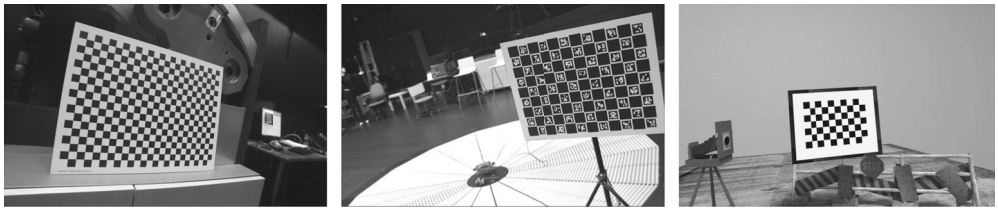


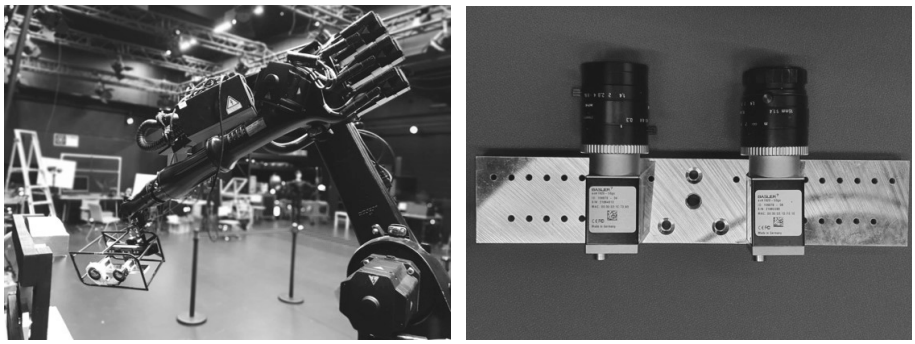
Figure 3.2 Example of calibration images from the dataset (a) checkerboard pattern (b) ChArUco pattern (c) synthetic image with checkerboard pattern.

Table 3.1 Overview of the dataset acquired and generated for testing.

No.	Dataset	Data Type	Lens Focal Length [mm]	Square Size [mm]	Image Size	Robot	Poses
1	kuka_1	Real	12	20	1928 × 1208	KR16L6-2	30
2	kuka_2	Real	16	15	1920 × 1200	KR16L6-2	28
3	kuka_3	Real	12	60	1928 × 1208	KR16L6-2	29
4	CS_synthetic_1	Simulated	18	200	1920 × 1080	N/A	15
5	CS_synthetic_2	Simulated	18	200	1920 × 1080	N/A	19
6	CS_synthetic_3	Simulated	18	200	1920 × 1080	N/A	30

The dataset was recorded using a Basler acA1920-50gc camera with 12 mm and 16 mm lenses. The camera was mounted and maneuvered with the aid of the KUKA KR16L6-2 serial 6-degrees of freedom (DoF) robot arm (see Figure 3.3). We use both checkerboard and ChArUco board of varying square sizes recorded with different camera lenses at varying robot/camera poses with the aim to provide a thorough dataset which is still convenient to use.

Real data allows us to observe all of the uncertainties associated with a real system for Hand-Eye calibration; however, it restricts us from acquiring ground truth information for comparison. Manually obtaining the ground truth TCP-to-camera



(a) (b)

Figure 3.3 An example of the setup for acquiring the datasets; (a) robotic arm moving in the workspace (b) cameras and Lenses for data acquisition.

transformation is not feasible, since the frame of the camera lies within the camera. In such cases, the absolute pose error is always absent for the quantification of accuracy and other metrics for relative errors and error distribution are utilized. In contrast, the use of simulated data provides us with ground truth information to assess the absolute performance of our calibration method. To the best of our knowledge, Publication I is the first paper to provide simulated dataset with synthetic images, for Hand-Eye and RWHE calibration, that are available for public use.

The complexity of the dataset is varied for each subset by acquiring images from different positions and orientations of the camera. A 3D computer graphics software, Blender, is used to render the synthetic scenes. To simplify the case, we assume that the robot's TCP position is the same as the camera position. In such a case, the homogenous transformation from hand-to-eye is the result of the orientation difference between the Blender world frame and Blender camera frame.

3.1.4 Experimental Results

In this section, we report and discuss the main experimental results to provide an insightful comparison between our study and six others. Figure 3.4 illustrates the results from simulated data in sequence 5 (Table 3.1) over varying visual noise in the presence of the pseudo-realistic robotic arm pose noise. The modelling of pseudo-realistic robotic arm pose noise and visual noise into the system is discussed in detail in Publication I.

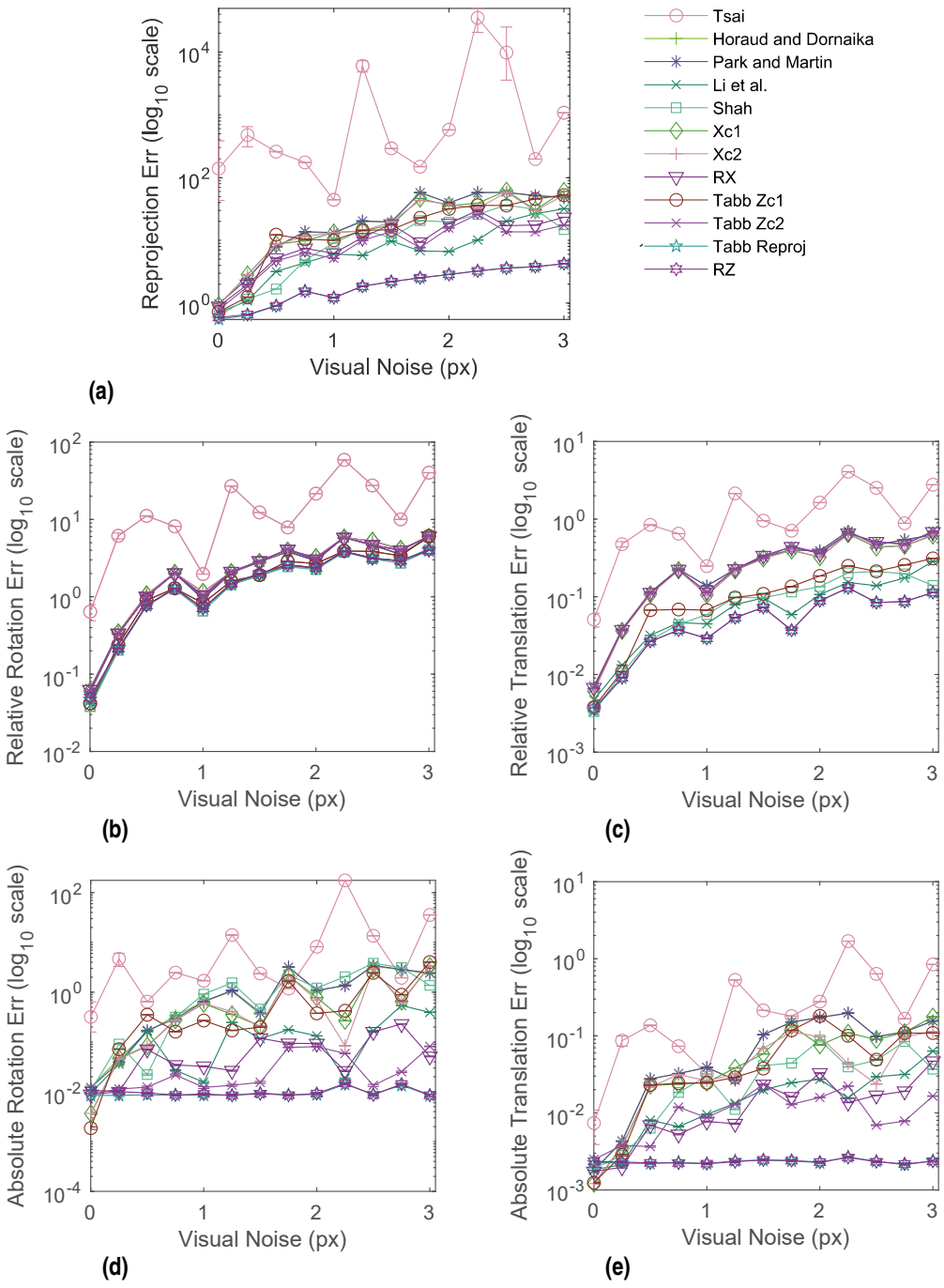


Figure 3.4 Metric error results for sequence 5 with constant robot pose noise; (a) reprojection error (b) mean rotation error (c) mean translation error (d) absolute rotation error (e) absolute translation error. (P.I)

To achieve a stable response, the plots represent the averaged results over 1000 iterations. This figure also shows the 95% confidence intervals from all the iterations for each experimentation point. With the exception of the response from Tsai [54] over reprojection error metrics, the confidence intervals are quite narrow. The narrow range of confidence interval indicates that we are 95% sure that our true mean lies somewhere within that narrow interval. Furthermore, this implies that the noise introduced during the iterative process is consistent and represents a coherent response from the methods. Based on the results in Figure 3.4 in conjunction with the results on real data in Publication I, we remark that Tabb rp1 [63] and the proposed RZ are quite robust to the increments in visual noise compared to other methods over all error metrics. It is noteworthy that despite the increase in relative rotation, translation and reprojection error, the absolute rotation and translation errors stay much more the same for Tabb rp1 [63] and RZ. When noise in the data is present, the Tsai's algorithm [54] performs poorly and erratically. In the absence of visual noise Zc1 [63], Xc1, RX and Shah [62] can achieve lower errors compared to rp1 [63] and RZ for multiple metrics. There is always some degree of visual noise in real data and different approaches might be affected differently by visual noise. To summarize our observations, the nonlinear reprojection-based methods achieve the best results among the methods under consideration even in the presence of visual and hand pose noise. RX yields good results with high accuracy under realistic visual noise with respect to reprojection error. In addition, RZ is more robust to visual noise and yields more consistent results for a greater range of visual noise. For detailed results, analysis, and discussion, we refer the readers to Publication I.

3.2 Geometrically constrained Multi-View Pose Estimation for Manipulator

In photogrammetry, robotics, and computer and machine vision, the pose of a camera has been extensively studied with respect to a target object. With the advances in this area, users have been able to accomplish a wide variety of tasks accurately. Despite this, application-specific methods can still be improved to achieve better accuracy and robustness. For instance, a reconstruction technique that is suited to achieving visually pleasing results might not be suited to accurate localization.

This section overviews the contributions from Publication II that are aimed at

solving the Research Question II. To limit the scope of the work, we explore solutions in a test case of a robotic manipulator (see Figure 3.5). The robotic arm is equipped with cameras to perceive the environment for visual servoing with the aid of a marker.

3.2.1 Problem Formulation

To model the physical system more accurately, we propose a modified form of the monocular multishot/multiview (MMS) approach in which we constrain the free pose optimization of cameras. We estimate only the transformation ${}_{c}T_{1}^{w}$ and use the prior information (robot poses ${}_{b}T_{i}^{t}$ and Hand-Eye transformation ${}_{t}T^{c}$) to constrain and geometrically relate the camera views from n poses. As opposed to traditional MMS methods, the proposed approach does not require estimating $n-1$ transformations ${}_{c}T_{i}^{w}$.

From Figure 3.5, we can form the following relationship among n manipulator poses

$$\begin{aligned} {}_{b}T_{1}^{t}\ {}_{t}T^{c}\ {}_{c}T_{1}^{w} &= {}_{b}T_{2}^{t}\ {}_{t}T^{c}\ {}_{c}T_{2}^{w} \\ &= {}_{b}T_{3}^{t}\ {}_{t}T^{c}\ {}_{c}T_{3}^{w} \\ &\vdots \\ &= {}_{b}T_{n}^{t}\ {}_{t}T^{c}\ {}_{c}T_{n}^{w}. \end{aligned} \tag{3.11}$$

We optimize only for one homogeneous transformation ${}_{c}T^{w}$ in the estimation step which transforms a point from the camera frame position in the initial/reference view to the fixed coordinate frame associated with the object/world. The geometric relationship in equation 3.11 can be generalized as an accumulation of all the transformations from the world to camera frames, excluding the reference pose. $\overline{T}_{i}$ transforms the 3D world points from the object/world coordinate frame, through the first reference pose, to the camera frames at each of the remaining $n-1$ poses.

$$\overline{T}_{i} = {}_{c}T^{t}\ {}_{t}T_{i}^{b}\ {}_{b}T_{1}^{t}\ {}_{t}T^{c}\ {}_{c}T^{w}. \tag{3.12}$$

Since we use quaternion and translation vector representation during optimization,

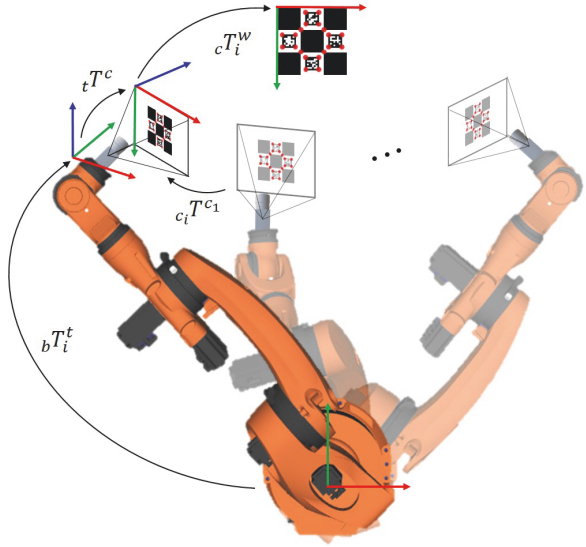


Figure 3.5 Illustration of the setup explaining the geometrical relation among various coordinate frames. (P.II)

we re-write equation 3.12 as

$$\overline{T}_i = {}_tT^{c^{-1}}\ {}_tT_i^b\ \ {}_tT_1^{b^{-1}}\ {}_tT^c\ \ [q_{(c,w)}\ ,\ {}_ct^w]_{HT}. \tag{3.13}$$

We can now estimate ${}_cT^w$ by optimizing the following expression

$$\{q_{(c,w)}\ ,\ {}_ct^w\} = \underset{q_{(c,w)}\ ,\ {}_ct^w}{\operatorname{argmin}} \sum_{i=1}^{n} ||P_i - \Pi(K, \overline{T}_i, W)||_2^2. \tag{3.14}$$

3.2.2 Experimental Results and Discussion

In our experiments, we compare the performance of our proposed method to the results of other studies using synthetic images and real data. The motivation for using simulated data is to check the response of the method against actual ground truth. Real data, in contrast, can be used to assess the effectiveness of a method in an actual working environment containing more data perturbations.

A quantitative comparison is provided between the proposed method and four other state-of-the-art methods. Among the methods used for comparison, IPPE [64] and Zhang [31] are based on monocular single shot (MSS) approaches. A MMS

approach is proposed by Collet and Srinivasa [65], while Geiger et al. [66] propose a single shot of stereo (SSS) cameras. Here we present the experimental results on real data acquired for pose estimation using the setup shown in Figure 3.6. The KUKA KR16L6-2 is equipped with an adaptor that houses two cameras and a custom tool. The tool is an aluminum bar with a Polycarbonate sheet at the end. A crosshair marker is drawn on this sheet. The tool is used to measure as accurately as possible the location and orientation of the target object. The intersection of the cross-hair marker helps to pinpoint the position while the planar surface of the tool sheet aids in measuring the orientation of the planer target. The tool is used both for initial ground truth measurements and for the evaluation of estimated poses. To measure the ground truth, the tool marker is manually aligned on the target object and robot pose information is recorded. Afterwards, these estimated poses are compared with the recorded poses.

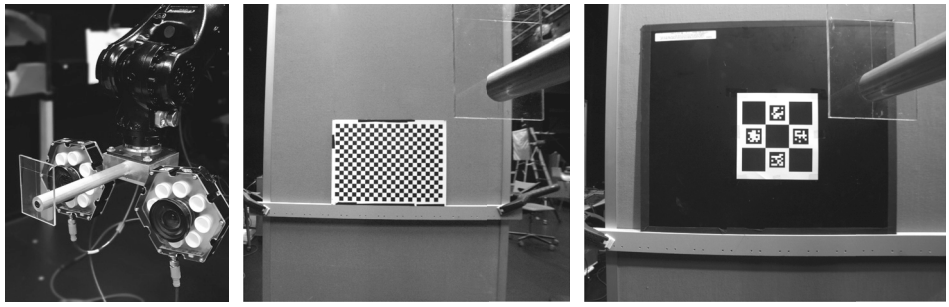


Figure 3.6 Illustration of the experimental setup (a) Close up of the adaptor with the tool, stereo camera pair and lights affixed to the manipulator using customized hardware (b) Checkerboard from camera view (c) ChAruCo board from camera view. (P.II)

It can be observed from the tabulated results in Table 3.2 that the proposed method yields the least absolute rotation error (μ_R) and absolute reprojection error (μ_{re}). The least absolute translation error (μ_t) is obtained by the stereo approach in [66], however, the proposed approach yields a comparative result with the second-best translation estimate. The results obtained for μ_R , μ_t ,and μ_{re} using [31], [64] and [65] are quite similar for the given set of experiments. Moreover, the proposed method yields significantly lower deviations over translation (σ_t) and reprojection (σ_{re}) estimates. The least standard deviation for rotation (σ_R) is achieved by SSS-Gieger et al. [66].

Table 3.2 Comparative results using checker board as target object. (P.II)

Methods	Abs. Rotation Error, $\mu_R\ (deg)$	Abs. Translation Error, $\mu_t\ (mm)$	Abs. Reprojection Error, $\mu_{re}\ (px)$	Rotation std. dev., $\sigma_R\ (deg)$	Translation std. dev., $\sigma_t\ (mm)$	Reprojection std. dev., $\sigma_{re}\ (px)$
MSS-Zhang [31]	1.9546	3.4217	1.3194	0.036274	0.19947	0.21217
MSS-IPPE [64]	1.9586	3.6378	1.3345	0.047422	0.2165	0.22098
SSS-Geiger et al. [66]	1.9325	**2.931**	1.1621	**0.027213**	0.22121	0.26702
MMS-Collet et al. [65]	1.9498	3.4847	1.3306	0.03223	0.20967	0.2204
MMS-Proposed	**1.6796**	3.1285	**1.0733**	0.045619	**0.15001**	**0.12447**

Table 3.3 Comparative results using diamond marker as target object. (P.II)

Methods	Abs. Rotation Error, $\mu_R\ (deg)$	Abs. Translation Error, $\mu_t\ (mm)$	Abs. Reprojection Error, $\mu_{re}\ (px)$	Rotation std. dev., $\sigma_R\ (deg)$	Translation std. dev., $\sigma_t\ (mm)$	Reprojection std. dev., $\sigma_{re}\ (px)$
MSS-Zhang [31]	2.508	4.0691	2.3638	0.92636	1.0597	0.20078
MSS-IPPE [64]	2.3373	4.1017	2.2399	0.52881	0.47025	0.18418
SSS-Geiger et al. [66]	2.3903	4.2761	2.1124	0.20142	0.30155	**0.13553**
MMS-Collet et al. [65]	2.3095	4.0959	2.2096	0.25623	0.50482	0.17876
MMS-Proposed	**2.1837**	**4.0628**	**2.105**	**0.14443**	**0.21076**	0.15846

Table 3.3 presents the results for the second set of experiments, using the diamond marker as the target object. As before, the tabulated results exhibit that the proposed method achieved the best results over all metrics except for σ_{re}, where it yields a result comparable to the stereo approach. SSS-Gieger et al. [66] shows a comparable error distribution to the proposed method. It appears that the MSS approach by Zhang [31] has the least consistent performance. The standard deviation of the estimate errors (σ_R, σ_t, and σ_{re}) is the highest in this case. MSS-IPPE performs comparatively well among MSS approaches and shows comparative results to MMS approach by Collet et al. [64].

We refer the readers to Publication II for more experimental results, including the tests on simulated dataset with synthetic images, and discussions that corroborate the conclusion drawn here.

3.3 Summary

In this chapter, we presented elements that were essential for manipulating a robotic arm with the aid of visual sensors. We started with the calibration of the robotic arm with a camera to mimic the mammalian Hand-Eye coordination capability. We presented a collection of novel methods to address the Robot-World-Hand-Eye calibration problem in its two alternative geometrical interpretations. The methods were

extensively tested on real and simulated datasets acquired through setups specifically developed to test and assess the robustness and accuracy of the methods. The later part of the chapter presented a monocular multi-shot approach to estimate the 6-DoF pose of the camera against a target object. The proposed approach modeled the geometric relations among various coordinate systems and explicitly incorporated the robotic manipulator poses into the formulation as implicit constrains. The proposed solutions were thoroughly tested and their performances were compared against the state-of-the-art methods on the novel datasets provided along with the studies. The empirical results demonstrated that our methods yielded significant improvement in accuracy and precision over other methods.

4 VISUAL ODOMETRY AND SLAM

In this chapter, we overview visual SLAM and relate some of its elemental blocks namely, VO, mapping, and re-localisation for loop closure. Hence, the chapter addresses Research Questions III-V. A novel approach is provided to map the environment in the presence of actively dynamic entities in the scene. A unique training, testing, and validation dataset is contributed to assist visual SLAM algorithms. Finally, a novel approach is proposed to enhance place recognition capabilities for loop closure using deep learning-based approaches.

4.1 Discrimination of Active Dynamic Entities

Generally, SLAM approaches have the fundamental assumption that the observed environment is static and that objects do not exhibit any change in dynamics or shape. However, this assumption is valid only in a controlled environment and is invalidated altogether in busy urban areas. In such cases, it is important to identify, extract, and discard dynamic entities from the pose estimation step and track them for future reference. In any other case, a SLAM pipeline that is unable to discriminate outliers can possibly result in inconsistent map, erroneous localisation, or altogether failure in registration.

Here, we present the results of our investigation (P.III) on the Research Question III, formulated in this thesis, and demonstrate that dynamic entities can indeed be segregated from the local map, in the case of a dynamic environment, and used as prior knowledge to discard outliers pertaining to the moving object when estimating the next pose in odometry.

In Publication III, we demonstrate the application with a stereo camera for localizing and mapping an active dynamic environment without any prior knowledge about the dynamics in the scene. For each image, disparity maps are obtained and checked for consistency from one camera to another in order to eliminate wrong disparities.

Each pixel position $(x, y)^T \in \mathbb{R}^2$ has its computed disparity. Point clouds are formed from the 3D positions of all valid disparity points. To simplify the computational and memory related complexities of SLAM, the point clouds are uniformly down-scaled by a grid filter. Each individual point cloud PtC_t has the associated description of each point i.e. location (X_k, Y_k, Z_k), Color (R_k, G_k, B_k) and Normal vectors to the plane (Nx_k, Ny_k, Nz_k) stored along with it.

4.1.1 Data Association and Pose Estimation

An initial registration is performed by finding the correspondences using the nearest neighbors in the point clouds PtC_t and $PtC_{(t-1)}$. ICP is used to register the point clouds by iteratively minimizing the error metric

$$e^i = \sum_{i=1}^{N} d_s^2(T^i p_k,\ S_j^k). \tag{4.1}$$

Here d_s is the signed distance from a set of points p_k in the point cloud PtC_t to the tangent plane formed of the set of points q_j in $PtC_{(t-1)}$. T^i is the transformation computed in the iteration i of the error minimization process. The 6-DoF transformation matrix T_t encapsulates the rotational matrix and translation vector used to register the two-point clouds. We adopt a hard threshold-based approach in the correspondence search for the nearest points to remove the outstanding wrong correspondences (outliers). The outliers can be present in the form of 3D point noise that may originate from erroneous depth estimation, different sampling of an entity, or motion of the objects. The new point cloud is transformed and merged with the global map with two new descriptions; a confidence metric C_k and frame presence F_k. The confidence C_k of a point informs us about the integrity of the point for being static and valid while the frame presence F_k stores the information about time the point was initially introduced to the system.

4.1.2 Confidence change for Dynamic Entities

The confidence metric is defined to measure the stability of 3D points in the map. A high confidence value indicates that the 3D point has remained a static and stable part of the world. However, removal of unstable points is necessary to limit the computational and memory load and at the same time discriminate and discard ac-

tively dynamic objects. Although a point may find multiple valid matches, however, only the nearest points are physically merged, and their confidence is accumulated and raised by some constant. The remaining valid associations (among the inliers) are not merged physically but temporarily added (with low confidence and original properties) to the global map since they might represent a different view of the same object. Additionally, all of the inliers' frame presences F_k are incremented to indicate that they have been observed.

A global map must be updated accurately not only with confidence gain but also with confidence loss. The 3D points should be updated logically when a stationary object in the scene starts moving. Such a representation of the dynamic entities can be realized by continuously reducing the confidence of all the points that are in the camera view by a small factor. The 3D points from the global map are projected to the image plane using the camera intrinsic K and the inverse of global camera pose T_{gt}^{-1} at time t. The points that are projected within the bounds of the plane are assumed to be in the camera perspective and, therefore, reduced in confidence. Hence there is a continuous struggle by the 3D points to maintain their confidence by merging with nearest neighbors in the presence of the confidence leak. This tug-of-war of confidence enables us to smoothly transition between a static and dynamic representation every time an object moves in and out of the scene.

We provide some results from Publication III in Figures 4.1 and 4.2 for stationary and moving cameras, respectively. The results exhibit successful discrimination, identification, and subsequent extraction of the clusters of points pertaining to dynamic entities in the map and their corresponding segmented regions in the 2D images. Moreover, the results exhibit an effective transition and merger of the moving person into the static map and the converse transition and removal from the map once the person starts moving. More details about these experiments can be found in Publication III. The observations from the study validate the hypothesis we formulated to address the Research Question III and provide a solid ground for expansion of the work towards a scalable solution for real-time accurate mapping applications.

4.2 Visual SLAM in Forest Landscape and Benchmarking

Research into autonomous vehicles has been greatly influenced by the fierce competition to develop a safe, marketable self-driving car. With many companies already ex-

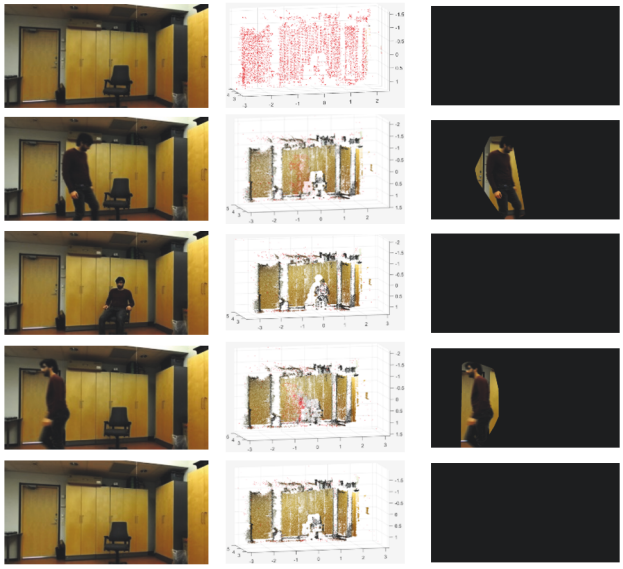


Figure 4.1 Test sequence with stationary camera. (P.III)

perimenting with self-driving vehicles, other companies are now looking into other forms of autonomous vehicles for automation of various industrial processes like mining, shipping, and agriculture. Getting from Advanced Driver Assistance System (ADAS) to autonomous vehicles relies heavily on advancements in many operations, such as object detection [82], reconstruction quality [83], and semantic understanding of the environment [84]. It is, however, odometry, relocalisation and mapping [85] that are the base capabilities required by an autonomous vehicle. To do this, all scenarios that a vehicle would face in real operation need to be tested and validated in a simulated environment.

In this section, we summarize our extensive investigation into existing datasets that was conducted in light of the Research Question IV. We discuss the contributions of the Publication IV, which proposes a novel dataset, towards the Research Question IV in specific and the thesis in general.

A number of datasets are publicly available for testing SLAM methods under a wide range of conditions and locations. Several well-known datasets will be mentioned in order to illustrate the range of the collection and discover a horizon. Most of these datasets focus on urban environments, for example [86]–[91], in order to facilitate testing on public roads in urban areas. When combined, these datasets provide a fair amount of data to test the methods in short and long trajectories at

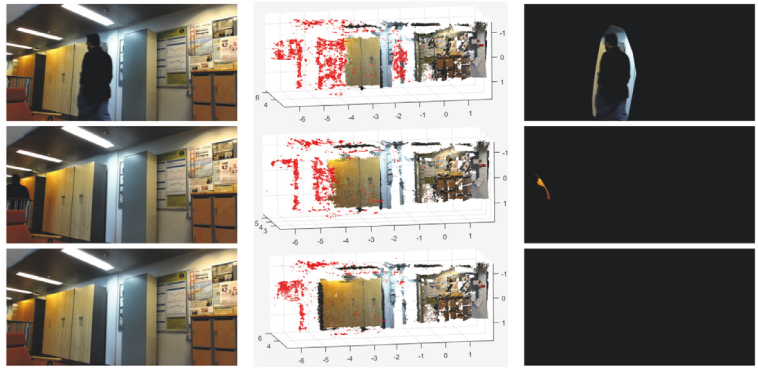


Figure 4.2 Test sequence with moving camera.(P.III)

various speeds [90]. Additionally, they consider weather and seasonal changes [92], long-term changes in city structure [92] as well as gradual/abrupt variations in illumination [93]. However, all these datasets target indoor or outdoor urban environment with the exception of some studies that target land and water-based terrains [94]–[98]. We observed a gap in work towards testing data for non-urban settings and decided to target terrain environments. For this reason, we recorded data in a forest landscape that benefits both self-driving cars and heavy work machines that navigate and/or operate in a dense forest landscape.

4.2.1 Dataset Overview

In Publication IV, we present an original dataset that explores a real forest landscape located in the outskirts of Tampere, Finland (see Figure 4.3a, extracted from P.IV). The goal is to provide testing data in order to facilitate the research towards increasing the autonomy of vehicles traversing rural areas and heavy machines working in the forest. Unlike urban settings, a terrain environment provides fewer discriminate landmarks and more repetitive textures in the scene. Presumably, such a situation strengthens VO to some extent, however, affects adversely relocalisation algorithms. This dataset provides semi-structured forest routes under different conditions (i.e. lighting, weather, vegetation, and infrastructure) in a highly self-similar natural environment. Furthermore, the sequences include scenes that best replicate the motions (i.e. stationary, sharp motion, bumps and potholes, slopes, and back-and-forth motion) and environments (i.e. log piles, close-up of trees, and off-road routes) involved in actual forestry operations. The dataset includes unique trajectories to test

both VO and visual SLAM algorithms thoroughly. Moreover, each path is traversed in two different conditions, namely sunny summer and snowy winter. The dataset provides images acquired with four cameras (affixed on a sensor rig shown in Figure 4.3b) that form three stereo pairs. The ground truth poses are acquired using a tightly coupled solution of GNSS and IMU information. We provide processed rectified images, calibration data and ground truth at three sampling rates i.e., 40, 13.33 and 8 Hz except for two sequences which are sampled at 20, 10 and 7 Hz. Furthermore, we provide raw images (40 Hz), camera calibration images, development tools to process the raw data, and evaluation tools to facilitate comparison against the benchmarked results.

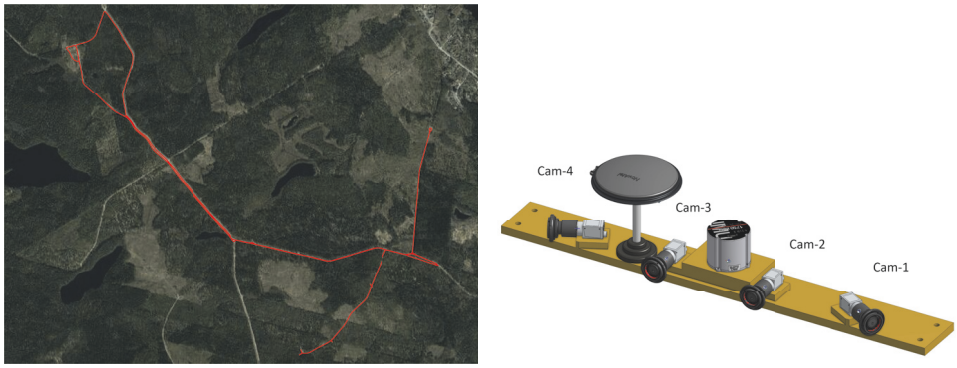


Figure 4.3 Illustrations from P.IV (a) The GPS trajectory of our recordings in the forest area in the outskirts of Tampere, Finland. (b) Rendered 3D model of the sensor rig.

4.2.2 Ground Truth and Benchmarking

Ground truth information can be difficult to obtain in enclosed environments. In order for the ground truth solution to be globally accurate, GNSS links must be available. An open area has high signal strength and accuracy, whereas enclosed areas, including narrow city streets and forests, have weak signals. A tightly coupled pose estimation framework employing GNSS and IMU data is used to acquire a ground truth solution using NovAtel's PwrPak7^{TM} module.

Based on the results in Publication IV, we observed that the average standard deviations in position for the winter sequences in the East and North directions were less than 2 cm, with occasional larger deviations. When vehicles traverse an

area densely covered in trees for a prolonged period of time, they produce larger deviations. Compared to winter recordings, summer sequences exhibit slightly larger standard deviations. All summer sequences in East and North generally have errors of less than 20 cm; the highest error is seen in elevation. There is a logical reason for the degrading GNSS results in the summertime. The increase in foliage can cause a 24 to 35 % attenuation of the GNSS signal at L-band [99]. This attenuation of signals is due to the combined effect of signal absorption and scattering from the canopy and trunk of the trees. When the GNSS signal is absent, the ground truth pose estimation system primarily relies on the IMU. In spite of this, the results obtained for the summer sequence are impressive and serve as a useful reference for further experimentation.

Among the state-of-the-art SLAM implementations, we chose ORBSLAM2 [46] and Stereo-Parallel Tracking and Mapping (S-PTAM) [100] for testing. These studies provide open-source implementation of a stereo based visual SLAM method which facilitates the testing phase of our study. Using the state-of-the-art method to test the dataset gives the reader an understanding of the challenges presented by the dataset and provides a benchmark for other algorithms. We tabulate the experimental results from ORBSLAM-2 and S-PTAM in Tables 4.1 and 4.2, respectively. It is clear from the results extracted from Publication IV that despite covering short distances, large drift and scale errors are observed for VO sequences. These errors result in large deviations and errors in the estimated trajectory. We discuss more about the unique challenges posed by the dataset in the forthcoming section. Additionally, to facilitate testing, a development and evaluation toolkit is included with the dataset, which can be used to process raw data or to compare the odometry obtained from one's algorithm with ground truth poses.

4.2.3 Challenges and Impact

In this section, we discuss the challenges posed by the novel dataset and its contribution to the thesis and research community. The following remarks are intended to assist future research and experiments with the data.

Forest environments present unique challenges for tracking features. The self-similarity and repetition of patterns make it difficult to match and track features accurately. While recording the dataset, we traverse rough terrain as opposed to urban routes. The combined effect of data sampling, speed, and erratic motion introduce

Table 4.1 Quantitative results of ORBSLAM2 for the FinnForest dataset at different sampling rates. (P.IV)

Data Sampling	40/20 Hz			13/10 Hz			8/7 Hz		
Seq.No	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)
*W*01	3.35	2.1785	0.00014197	3.6738	2.3016	0.00019584	3.4914	2.4092	0.00021607
*W*03	12.266	9.1805	0.00012107	12.025	9.2299	0.00013344	12.249	9.1962	0.0001253
*W*04	17.421	7.7753	9.7778e-05	17.244	7.8332	9.935e-05	20.666	7.6746	0.00011482
*W*05	55.422	9.2678	0.0001298	56.323	9.4365	0.00013865	75.715	9.7977	0.00022451
*W*06*	21.789	32.14	0.00020608	TL	TL	TL	TL	TL	TL
*W*07*	37.933	7.2208	0.00011185	34.324	7.2193	0.00013786	48.88	7.2107	0.00016175
*S*01	4.3677	1.9672	0.00022474	3.8189	1.917	0.00019894	6.2793	2.1462	0.00027508
*S*02	26.132	4.2061	0.00017796	26.874	4.2181	0.0001728	TL	TL	TL
*S*03	12.633	5.873	0.00020877	10.986	5.6022	0.00018197	9.8899	5.5459	0.00018258
*S*04	30.053	5.5827	0.0001988	26.825	5.4608	0.00018299	TL	TL	TL
*S*05	228.88	9.4575	0.00025169	191.52	8.8165	0.00020505	200.81	8.9426	0.00021338

∗ indicates that the data is subsampled at 20/10/7 Hz TL: Tracking lost

Table 4.2 Quantitative results of S-PTAM for the FinnForest dataset at different sampling rates. (P.IV)

Data Sampling	40/20 Hz			13/10 Hz			8/7 Hz		
Seq.No	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)
*W*01	TL	TL	TL	TL	TL	TL	TL	TL	TL
*W*03	19.709	10.166	0.00011828	27.663	12.63	0.0004809	28.369	14.819	0.00063508
*W*04	25.852	9.4934	0.00014839	45.091	14.9	0.00071498	48.944	14.914	0.00073208
*W*05	TL	TL	TL	79.774	11.312	0.00011181	TL	TL	TL
*W*06*	TL	TL	TL	TL	TL	TL	TL	TL	TL
*W*07*	TL	TL	TL	102.54	8.319	0.00019895	TL	TL	TL
*S*01	7.3247	2.883	0.00018821	9.4022	4.0914	0.00066569	8.652	3.9342	0.00030787
*S*02	34.391	9.2735	0.0005317	44.68	11.63	0.00061402	34.752	9.2786	0.00020271
*S*03	21.779	7.0644	0.00025365	31.418	11.105	0.00025333	47.392	14.82	0.00031883
*S*04	31.891	7.1297	0.00023556	39.749	9.703	0.00019259	TL	TL	TL
*S*05	130.41	10.182	0.00022272	171.55	14.517	0.00032586	201.65	17.9	0.00038022

∗ indicates that the data is subsampled at 20/10/7 Hz TL: Tracking lost

challenges for tracking and localisation.

We recorded the data at low driving speeds, around 25-30 km/h, and a short exposure time for image acquisition to prevent motion blur. Following a recommendation in [101], we included the skyline in the scene which is expected to enhance feature matching possibilities. The features that are extracted farther away from the forest near the skyline contribute significantly to the accurate estimation of rotations, especially pitch and yaw.

The dataset contains various opportunities to test the robustness of the visual SLAM methods for estimating the pose and tracking objects moving in a scene with varying illumination. The sequences W07, S04, and W06 offer notable opportunities regarding illumination change. In W07, illumination gradually changes as the

day grows darker. As the sequence was recorded at dusk, the lighting significantly changes between the start and end of the sequence. The sequence S04, however, displays a more rapid change in illumination because of direct sunlight. Lastly, W06 sequence provides data collected at night in the presence of directed light sources.

The dataset provides three sequences with loop closure opportunities. Two of these routes, S01, and W01, repeat the same route three times, twice in one direction, and once in the opposite direction. Additionally, there are odometry sequences that do not offer a direct closure of loops. In such cases, scale and pose drift are prominent especially in the benchmarked results of S03 and W03.

This dataset and subsequent experimental results appear to be affected by seasonal changes in different ways. As described earlier, the ground truth accuracy is slightly lower in the summer compared to the winter because of the seasonal foliage effect. In addition, it was a challenge from the perspective of recording as well since the varying density of trees at different areas resulted in different level of scene illumination. Due to the fixed aperture, this posed a challenge to avoid over or underexposure of the scenes as is evident from the sequence S04.

Altogether, we believe that the dataset proposed in Publication IV along with the detailed experimental results contributes to answering our Research Question IV. The study successfully highlights the limitations of state-of-the-art methods and provides us with the opportunity to develop new techniques to handle the aforementioned challenges in a unique testing environment.

4.3 Loop Closure Detection and Relocalisation

A key challenge for mobile robotics, navigation, and augmented reality is determining where you are in your local world. The problem is commonly referred to as the kidnapped or lost robots with the solution termed as Relocalisation. For any intelligent transportation system, loop closure is crucial to achieving robust navigation as it helps to reduce the errors accumulated in visual navigation [102]. The process involves recognizing previously visited places and determining the current pose in comparison to the previous pose from the visual representation of the scene. We describe these two problems, in detail, later in this chapter.

4.3.1 Uni-directional vs Bi-directional

Typically, loop closures are detected by visually recognizing places, from images, that have already been visited and viewed from somewhat similar perspectives. This problem has been approached in much the same way as the image retrieval problem. As a rule of thumb, a query image whose location needs to be determined is compared to a large database of geo-tagged images. In Section 2.2.3, we discussed in detail the classical and state-of-the-art methods developed to handle place recognition for loop closure. In the case of mobile robots/vehicles, the motion expected is along a specific route and roughly linear in direction for longer periods. For simplification, the problem has always been approached as unidirectional e.g., a vehicle when passing by the same location is always assumed to be roughly oriented along the same direction as it was during its earlier visit. This limits the change in visual perspectives and simplifies the problem. However, such an assumption is highly dependent on the availability of uni-directional landmarks. The system is entirely insensitive to odometry sequences where routes are traveled from both directions. We formulated this problem in our Research Question V and further experimentally validated the failure of the state-of-the-art methods on our FinnForest dataset for bi-directional loop closure during benchmarking in Publication V. Bi-directional loop closure is a relatively new term with initial work conducted in [77] on panorama images. However, as we discussed earlier, the use of panoramic images reduces the complexity of the problem by providing roughly similar views as a uni-directional case. In most cases, monocular cameras are used for visual navigation [46], [100]. In Publication V, we propose a deep learning-based approach that successfully recognizes places and estimates relative poses in a bi-directional motion configuration. We validate the performance of the proposed approach on both indoor and outdoor data.

4.3.2 Data Preparation

Developing a deep learning model that can generalize well for all cases requires a lot of data. It is often expensive to produce new data every time, which can drive researchers away from the actual problem. Therefore, researchers in several fields have introduced many public datasets. There are many datasets available for testing purposes when it comes to visual SLAM [31]-[35]. Most datasets, however, do not allow for bi-directional movement since they were designed to deal with the problem

from a uni-directional perspective. For the bi-directional loop closure work in Publication V, we found that parts of PennCOSYVIO dataset [103] and our FinnForest dataset, from Publication IV, can be used for training and testing.

Our localisation and pose regression models are trained by passing both datasets through a specialized data preparation phase. We generate sub-datasets out of the original datasets and use them for training. We can use the data generated for localisation in the pose regression block since the two are performed on the same scenes. We split and consider two cases of the bi-directional localisation problem for the sake of simplification. Assuming forward motion, an anchor sample is collected at the query location. Images that share a perspective view are set as positive samples, and images that are distant are set as negative samples. Together, anchor, positive and negative image samples form a triplet. A triplet set can be expressed as

$$S = \{(s_i, {s_i}^+, {s_i}^-) | (s_i, {s_i}^+ \in S^+); (s_i, {s_i}^- \in S^-), i = 1, ..., M\}. \tag{4.2}$$

Here, S^+ refers to the set of relevant image pairs, S^- refers to negative image pairs, and M indicates the span of the entire triplet set.

We compute the relative translation and rotation (in quaternions) for these triplet samples, which are used as ground truth in training and testing. To ascertain the quality of chosen samples, we leverage structure from motion to autonomously generate training triplets. This is performed by triangulating image features from a query image and projecting them using the ground truth poses on nearby positive target samples from the generated triplets. We use the relation

$$E_{px} = ||P_{s_i^+} - \Pi(K, [q_{(s_i, s_i^+)}, {}_{s_i}t^{s_i^+}]_{HT}, W_{s_i})||_2^2. \tag{4.3}$$

Samples with reprojection errors E_{px} higher than a threshold are discarded. This is important for rejecting data samples that might be affected by camera jitter, motion blur, sudden overexposure, or sun flare in the camera view.

Similarly, we attempt to find pairs of query images in the backward motion part of the sequence using the query image samples initially selected and filtered for the forward case. For a backward motion case, only samples ahead of the query location can be positive. The potential positive samples should all be oriented in the opposite direction from the camera orientation at the query point. If a camera pose is slightly

ahead of the query point, it is likely that it would see more of the same scene, even from the opposite point of view. It is this similarity in the scene within this small range that we want our model to recognize and discriminate. Camera poses too far forward or too far back from the query pose would have little and no overlap with the query perspective, respectively. For the backward case, the traditional feature detector cannot detect and track features with such high perspective changes, so the reprojection-based verification is impossible.

4.3.3 Place Recognition

We adopt a Siamese network [104] to train our models, as shown in Figure 4.4, for place recognition. The network is built using a VGG-16 as the base CNN model that takes three inputs: query image samples I_q, positive image samples I_p, and negative image samples I_n from the database I_D. With VGG-16, an input image of 224×224 pixels is propagated through a series of convolution layers and pools, where the layers are connected through a Rectified Linear Unit (ReLU) as an activation function. A neural network descriptor known as the NetVLAD pooling layer [105] is fed the normalized outputs of the base model. In its simplest sense, VLAD encodes information about local descriptors' statistics aggregated in a given image in terms of feature distance from a cluster center.

For N D-dimensional local image descriptors $\overrightarrow{x_i}$ as input, and K cluster centres (visual words) c_k as VLAD parameter, the output VLAD image representation V is $K \times D$-dimensional. The differentiable L2-Normalized vector form of V with elements (j, k) is

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^T \overrightarrow{x_i} + b_k}}{\sum_{k'} e^{w_{k'}^T \overrightarrow{x_i} + b_{k'}}} ((x_i(j) - c_k(j)). \tag{4.4}$$

Here, $x_i(j)$ and $c_k(j)$ are the j-th dimensions of the i-th descriptor and k-th cluster centre, respectively. w_k and b_k are sets of trainable parameters for each cluster k which are learned in an end-to-end manner during training. Conceptually, the weight that the descriptor $\overrightarrow{x_i}$ is assigned to the cluster c_k is proportional to their proximity. Moreover, the relative proximity to other cluster centres also plays a part in the relation.

We employ a triplet loss that takes input from the outputs of each branch of the

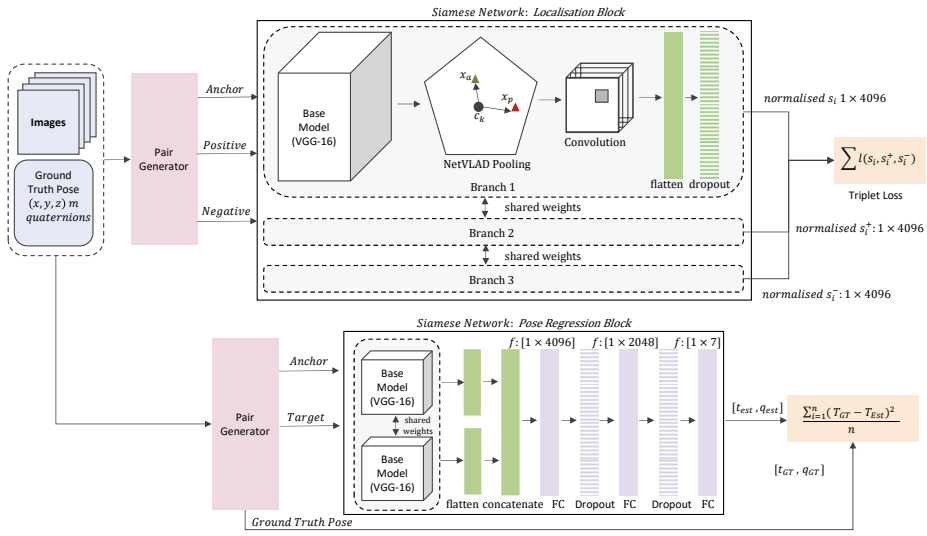


Figure 4.4 Illustration of the system pipeline. A siamese network constituting a VGG-16 base model topped with NetVLAD pooling layer is used to learn similarity in the scenes using a triplet loss. The pose regression network (lower) is independently trained to directly regress the 6-DoF relative camera poses between the query and the retrieved match.

triplet siamese. During training and validation, the model learns the representation of the input image samples that minimize the distance between the I_q and I_p samples, and maximizes the distance between the I_q and I_n samples, simultaneously, for each triplet sample in feature space. For place recognition, the triplet loss is given as

$$\ell(s_i, s_i^+, s_i^-) = max(0, m + \left\|f(s_i) - f(s_i^+)\right\|_2^2 - \left\|f(s_i) - f(s_i^-)\right\|_2^2). \tag{4.5}$$

Here, margin m is a scalar that defines an offset between positive and negative pairs, and $f(.)$ is an embedding of an image sample. The global loss over all triplet samples is given as

$$L = \sum_{(s_i, s_i^+, s_i^-) \in S} \ell(s_i, s_i^+, s_i^-). \tag{4.6}$$

Once we learn the optimum feature representation, we can use any branch of the Siamese Network to encode our query and database images. This enables us to have the representation in the same embedding space for all our images. A coherent representation then enables us to use a nearest neighbor approach to retrieve the top N-ranked database images, $d = (d_n | d_n \in D, n = 1...N)$ based on a distance metric (the squared Euclidean in our case) in the embedding space. Depending on the number of

keyframes generated during earlier exploration of the environment, a query image may have one, many, or no matches in the database.

The place recognition network is ready at this point to be incorporated into a SLAM pipeline to find and retrieve potential matches for loop closure. However, for complex scenes the problem can become significantly more challenging when the environment contains repetitive textures even for distinct locations such as in FinnForest dataset. To ascertain that the loop closure is robust in operation, we perform an additional confidence sharing between candidate query images. Using a confidence-sharing scheme, the previously localised points' confidence is propagated to their neighbors in a causal manner. In order to determine the sanity of a potential match for a query point, we consider the distance between the neighbors. A new query point is valid if (1) it matches the image in the database well (in embedding space), and (2) it is also surrounded by nearby localised neighbors whose distances agree with the estimates from the odometry. New neighbors found far away from a nearby localised neighbor are rejected as potentially wrong matches if the odometry estimates do not agree in general.

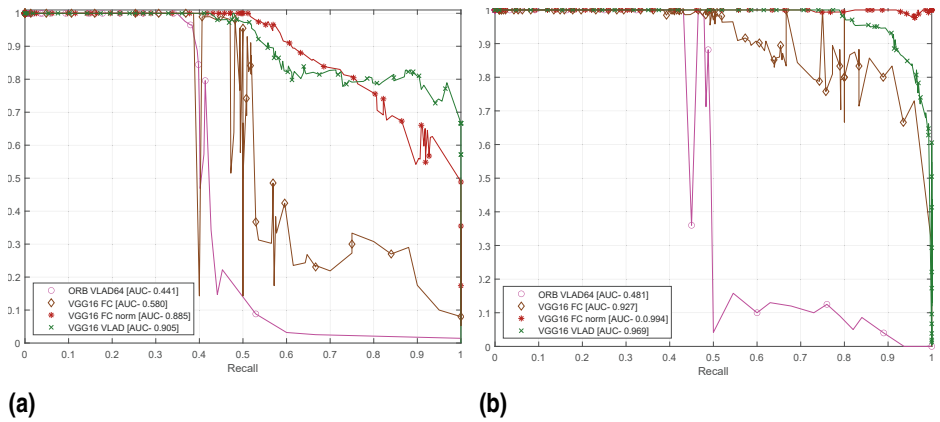


Figure 4.5 Precision-recall curves for bi-directional loop closures in the (a) FinnForest dataset and (b) PennCOSYVIO dataset.

We present here some experimental results, from Publication V, for the localisation tests on FinnForest and PennCOSYVIO datasets in the form of a Precision-Recall (PR) curve (shown in Figure 4.5). Observing the results, it can be seen that the proposed approach VGG16-VLAD and VGG16-FCnorm outperform other tested approaches/methods. For both datasets, the area under the curve (AUC) for

VGG16-VLAD and VGG16-FCnorm is nearly the same. Nevertheless, we believe that the VGG16-VLAD is better suited to the task. The proposition is based on the observation that VGG16-VLAD performs better on the FinnForest dataset, which is considerably more challenging than the PennCOSYVIO dataset. The FinnForest dataset is set over a much larger spatial area with more repetitive textures and fewer discriminating landmarks. Meanwhile, the PennCOSYVIO dataset presents the same indoor scene in all sequences, while the route and motion speed are slightly different. As a result, we can anticipate a high correlation between the training and testing data in the case of PennCOSYVIO. The FinnForest dataset, on the other hand, represents a variety of routes and scene, resulting in a lower correlation between training and testing data and a higher data center distribution (in the space that contains the encoded data clusters). Hence, we can infer that VGG16-VLAD has better generalization capability compared to the other methods.

4.3.4 Pose Regression

The pose estimation block proposed in Publication V, uses a VGG-based Siamese architecture that takes two monocular images as input and then predicts a relative 6-DoF transformation between the two camera poses. The Siamese regression block is shown in Figure 4.4. The Siamese architectures share a common weight between its branches which we initialized with the weights of a network pre-trained for large-scale place classification task [106] using the Places 365 dataset. The weights are finetuned and learned for our regression task during the training phase. The feature outputs from each branch are vectorized and concatenated to form a single encoded description. The encoded description is passed to three fully connected (FC) layers with dropout layers placed in between them for regularization.

Different studies opt for different representations while regressing for a solution. In study [107], the authors use deeply learned key points to estimate the Fundamental matrix between the camera views while in UnDeepVO [108], the authors estimate directly the relative pose, where the rotations are represented in Euler angles. Despite its shorter representation than the fundamental matrix, Euler angles are still subject to limitations such as discontinuities in the form of gimbal lock. On the other hand, representations such as a rotation matrix require that a Euclidean embedding is determined for its distance estimation. In our approach, we adopt the use of quaternions, similar to the work in [30], to represent the rotations.

Quaternions are defined on a unit sphere, however, during training/optimization the difference between spherical distance and Euclidean distance becomes insignificant. Exploiting this advantage, we refrain from using constraints based on spherical geometry to avoid complicating the optimization process. In such cases, the Euclidean L2-norm can be used to determine the distance between any two quaternions $\|q_{GT} - q\|$. A popular study PoseNet [109] and its derivative study [110] use a decoupled approach with a weighted parameterization of the angle, by a scale factor β, to balance the loss function

$$L = \|\Delta t_{GT} - \Delta t\|_2^2 + \beta \left\|\Delta q_{GT} - \Delta q\right\|_2^2. \tag{4.7}$$

Here Δq_{GT} and Δt_{GT} are the ground-truth relative orientation and translation, respectively. From the discussion in the aforementioned studies and our experimentation, we observed that determining a good value of β for every dataset can be quite cumbersome. Hence, in Publication V we propose an approach aimed at avoiding the use of such scaling factor during optimization by providing pre-scaled translation vectors and quaternions. These inputs are independently scaled down to same ranges during the preprocessing stage of the dataset preparation. In our work, we used an adaptation of ReLu activation for the FC layer, hence we rescaled both the quaternions and the translation vectors between $[0, 1]$. The training phase is then invariant to the scale factor. The scale factors can be extracted from the range of the data in the dataset and applied using relation

$$d_{scaled} = \frac{(sc_{max} - sc_{min}) * (d - d_{min})}{d_{max} - d_{min}}. \tag{4.8}$$

where d denotes the data array and sc indicates the scaler values of the desired range for scaling. The model is trained to predict an arbitrarily scaled version of the pose where the scale is restored in a post-processing step after the prediction. The MSE is then given as

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (T_{scaled\ GT} - T_{Est.})^2. \tag{4.9}$$

Here, $T_{scaled\ GT}$ is the pose constituting scaled $[t_x, t_y, t_z]$ and scaled $[q_w, q_x, q_y, q_z]$. $T_{Est.}$ is a similar vector to $T_{scaledGT}$ which is predicted by the model.

For the FinnForest and PennCOSYVIO datasets, some of the experimental re-

sults (from Publication V) are presented in Tables 4.3 and 4.4, respectively. To compare the proposed network's performance, we replace the base model (VGG16) with Resnet50. Additionally, the relative impact of weight initialization for classification tasks is also examined by using weights from models pre-trained on ImageNet and Places 1365 (an extension of Places 365). Tests are performed on individual sequences and the combined case to evaluate the pose regression network's performance. Accuracy is determined by comparing the predicted and the ground truth values for location and orientation.

It can be observed that the network that has VGG16 as the base model and initialized with the weights of Places 1365 yields the best results followed by VGG16 initialized with ImageNet. We can infer from this slight improvement that the relevancy of scenes in Places 1365 does convey a more direct impact on the training phase compared to ImageNet which includes more diverse, however, at times irrelevant scenes to the task at hand. Moreover, a comparatively poor performance was observed from Resnet50 on both datasets. Keeping in mind that the task at hand is localisation and not odometry, the results obtained for the relative pose are good and effective for loop closure. Traditional methods fail when we consider bi-directional cases of localisation. The results obtained for bi-directional pose regression in this study match the performance of other state-of-the-art approaches that are reported in studies conducted for uni-directional loop closure [109], [110]. For more experimental results, we refer the readers to Publication V.

Table 4.3 Comparison of pose estimation results from the regressor model trained on FinnForest dataset. (P.V)

Sequence	Test Samples	Spatial Extent (m)	Resnet50 Imagenet	VGG-Imagenet	VGG-Places 1365
S1	2044	47 x 193	5.38m, 1.02°	2.42m, 0.352°	2.26m, 0.3°
S3	2706	800 x 190	5.26m, 1.00°	2.38m, 0.33°	2.31m, 0.29°
S4	3566	812 x 568	5.68m, 1.06°	2.54m, 0.39°	2.36m, 0.32°
S5	8866	1826 x 1883	7.35m, 0.84°	3.35m, 0.44°	3.23m, 0.53°
Combined	17182	2633 x 2014	5.92m, 0.98°	2.67m, 0.38°	2.54m, 0.36°

Table 4.4 Comparison of pose estimation results from the regressor model trained on PennCOSYVIO dataset. (P.V)

Sequence	Test Samples	Spatial Extent (m)	Resnet50 Imagenet	VGG-Imagenet	VGG-Places 1365
C2-af	3361	144 x 36	3.79m, 0.71°	1.51m, 0.21°	1.35m, 0.22°
C2-bs	3330	144 x 36	3.85m, 0.72°	1.51m, 0.20°	1.33m, 0.21°
C2-bf	3090	144 x 36	3.81m, 0.73°	1.49m, 0.19°	1.36m, 0.22°
C2-bs	3375	144 x 36	5.75m, 0.80°	2.20m, 0.40°	1.81m, 0.26°
Combined	13156	144 x 36	4.3m, 0.74°	1.68m, 0.25°	1.46m, 0.22°

4.4 Summary

In this chapter, we discussed the deployment of pose estimation techniques toward larger problems namely odometry, mapping, and localization. We proposed an approach for discriminating dynamic objects in the scene when performing odometry and mapping with the aid stereo camera. The approach generated an active map based on the estimated odometry while observing and maintaining states of moving/dynamic entities using a confidence scheme. Moreover, the study demonstrated successful segregation of the dynamic entities from the pose estimation. Later, we presented the FinnForest Dataset that contributes unique sequences recorded in a forest environment in various light and weather conditions for VO and SLAM systems. We briefly discussed the specifics of route planning, data processing, and sampling, ground truth generation, and the challenges provided by the dataset. In addition, we discussed the benchmark performance of various state-of-the-art algorithms on the dataset. Finally, the chapter concluded with contributions to the topic of localisation where we presented a learning-based method to solve the bi-directional loop closure problem by separately training two deep models in an end-to-end manner for place identification and pose regression. The performance of the model was validated with unseen data and the results demonstrated that the networks generalized well and learned geometric and spatial relations in images rather than memorizing scenes/locations. A comparison of the proposed approach was provided against other deep learning and classical methods using qualitative and quantitative results which exhibited the effectiveness of the proposed approach against other well-established methods.

5 CONCLUSIONS

This thesis has provided a concept-driven and technical journey from camera positioning to location-aware exploration. It proposes several solutions to enhance a machine's autonomy, which can be grouped into two main categories. First, we propose methods that contribute to the overall improvement of a visual servoing system for robotic manipulators. We dissect the Hand-Eye calibration problem and extensively study various configurations of the system, alternate approaches, simulate models, analyze a real system and compare the results of the proposed methods to other state-of-the-art studies. Moreover, we introduce a novel dataset with real data and simulated data with synthetic images to aid systematic testing of the methods. In addition, we proposed a robotic arm error modeling approach to be used along with the simulated datasets for generating a realistic response. The methods proposed in Publication I were posed as an optimization problem and solved with an iterative algorithm. The empirical results demonstrated that calibration methods that incorporate camera geometry and projection directly in the calibration step yield more consistent and accurate results compared to approaches that rely on distributing error over pose estimates. Furthermore, parameterization of angles in quaternions and estimating the rotation and translation simultaneously is less prone to the propagation of errors from earlier steps.

In the study presented in Publication II, we proposed a constrained multiview pose estimation approach for robotic manipulators to aid visual servoing. The approach exploits the available geometric constraints on the robotic system and infuses them directly into the pose estimation method. As a result, the nonlinear optimizer that minimizes the reprojection error-based cost functions, yielded estimates with better accuracy and significantly more precision compared to other methods. The Hand-Eye calibration and multiview pose estimation schemes proposed are not limited to industrial robotic arms and can be extended to other form of machines such as excavators, reach stacker etc. that mimic robotic manipulators and are equipped

with cameras.

The second category of solutions is based on the contributions in Publication III-V where we address the corresponding research questions raised in Chapter 1 by identifying the limitations in existing methods for visual odometry and SLAM techniques. In many cases, robots are required to navigate and interact with the environment based on a map that is previously generated. Often the map can become outdated either due to actively moving objects in the scene or gradual changes to the outlook of the environment. This may render the map outdated and produce mismatches to the current observations. Moreover, the active dynamic objects can induce errors in pose estimation for odometry. To address these limitations, we present an approach in Publication III that actively segregates dynamic entities in the scene by assigning confidence measures to every point. A scheme is presented that aids in discarding the dynamic points as outliers to the pose estimation step and aids in transitioning the state of objects as stationary or dynamic based on current observations from the scene.

One of the biggest problems encountered in research is the limitations in examining all test cases thoroughly due to lack of resources. For robust systems, it is essential that test data properly relates to the problem at hand. Observing the limitations during our literature review and initial experimentation, we present the FinnForest dataset in Publication IV. In contrast to existing datasets that target the urban environment, we explore an unregulated natural environment to exemplify sub-urban and forest scenes. The dataset offers an actual forest landscape with routes traveled under different conditions (i.e. lighting, weather, vegetation, and infrastructure). Furthermore, the sequences include scenes that best replicate the motions (i.e. stationary, sharp motion, bumps and potholes, slopes, and back-and-forth motion) and environments (i.e. log piles, close-up of trees, and off-road routes) involved in the actual forestry operations. The dataset aims at facilitating research towards increasing the autonomy of vehicles traversing rural areas and heavy machines working in the forest. In addition, we provide highly accurate ground truth poses achieved using a tightly coupled solution of accurate IMU and GNSS data. Furthermore, we provide benchmark performance of the state-of-the-art methods on the dataset. The dataset provides unique and challenging test data and continues to receive considerable attention and appreciation in the research community.

Another research question that was identified during the testing phase of the

FinnForest dataset is how to effectively localize an observer under high perspective changes particularly to aid the case of bi-directional motion of vehicles over the same route. To our knowledge, the work presented in Publication V is the first to achieve bi-directional loop closure on monocular images with a nominal Field of View. The study proposes a two-step solution to the localisation problem, where a deep learning approach is employed stepwise for the place recognition and the pose regression tasks. We show that the networks generalize well and learn geometric and spatial relationships in images rather than memorizing scenes or locations. This is demonstrated by the performance of the model on unseen data. The results show that bi-directional loop closure is indeed possible on monocular images when the problem is adequately posed and training data is properly leveraged.

Overall, the thesis identified a number of serious gaps in the literature, and the proposed contributions addressed these gaps effectively, meeting the intent and scope of the thesis. The thesis also revealed certain limitations that can be addressed in future research work. The study conducted in Publication III regarding the discrimination of dynamic entities is limited to offline processes at the moment. The approach can prove to be computationally expensive as the map grows. A prospective contribution could be sparsification of the problem by using a hybrid approach that uses features for pose estimation while being aided by the sparse active map in the removal of potential outliers. Moreover, a tracking mechanism can be developed to keep track of moving bodies once they exit the camera view. The study already proposes the approach for relaying the information about the dynamic objects from 3D to 2D images for segmenting the objects; an extension can track objects in the images once identified.

Regarding the development of datasets, FinnForest is an excellent contribution as it offers numerous challenges for users to test and validate their odometry and localisation methods for accuracy. Future contributions to FinnForest dataset can be made by adding additional benchmarks such as depth estimation and completion, object tracking in 2D and 3D, and semantic segmentation of the scene.

Similar extensions are also possible by validating the performance of the proposed localisation method in Publication V over new datasets. Even though most of the available datasets are from urban environments, at the moment, they do not provide suitable motion sequences for testing bi-directional loop closure as they are tailored specifically for forward loop closure.

REFERENCES

[1] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[2] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marıén-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[3] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, IEEE, 1999, pp. 85–94.

[4] D. Hu, D. DeTone, and T. Malisiewicz, "Deep charuco: Dark charuco marker pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8436–8444.

[5] B. Atcheson, F. Heide, and W. Heidrich, "Caltag: High precision fiducial markers for camera calibration.," in *VMV*, vol. 10, 2010, pp. 41–48.

[6] *Vstar geodetic systems, inc*, https://www.geodetic.com/products/, Accessed: accessed: 17.11.2021.

[7] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in *CVPR 2011*, IEEE, 2011, pp. 113–120.

[8] F. Bergamasco, A. Albarelli, and A. Torsello, "Pi-tag: A fast image-space marker design based on projective invariants," *Machine vision and applications*, vol. 24, no. 6, pp. 1295–1310, 2013.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[11] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[12] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*, Ieee, 2011, pp. 2548–2555.

[13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*, Springer, 2010, pp. 778–792.

[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*, IEEE, 2011, pp. 2564–2571.

[15] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2015, pp. 4297–4304.

[16] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[17] C. McManus, B. Upcroft, and P. Newman, "Scene signatures: Localised and point-less features for localisation," *Robotics: Science and Systems X*, pp. 1–9, 2014.

[18] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *arXiv preprint arXiv:1411.1509*, 2014.

[19] A. Richardson, J. Strom, and E. Olson, "Aprilcal: Assisted and repeatable camera calibration," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 1814–1821.

[20] J. A. Castellanos and J. D. Tardos, *Mobile robot localization and map building: A multisensor fusion approach*. Springer Science & Business Media, 2012.

[21] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, vol. 1, 2006, pp. 363–370.

[22] W. Pan, M. Lyu, K.-S. Hwang, M.-Y. Ju, and H. Shi, "A neuro-fuzzy visual servoing controller for an articulated manipulator," *IEEE Access*, vol. 6, pp. 3346–3357, 2018.

[23] S. Dong, A. H. Behzadan, F. Chen, and V. R. Kamat, "Collaborative visualization of engineering processes using tabletop augmented reality," *Advances in Engineering Software*, vol. 55, pp. 45–55, 2013.

[24] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[25] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-d," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 110–124, 2015.

[26] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot SLAM system for 3D lidars," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 1004–1011.

[27] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "Lips: Lidar-inertial 3D plane SLAM," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 123–130.

[28] P. Li, R. Wang, Y. Wang, and W. Tao, "Evaluation of the icp algorithm in 3D point cloud registration," *IEEE Access*, vol. 8, pp. 68 030–68 048, 2020.

[29] T. Collins, J.-D. Durou, P. Gurdjos, and A. Bartoli, "Singleview perspective shape-from-texture with focal length estimation: A piecewise affine approach," *3D data processing visualization and transmission (3DPVT10)*, 2010.

[30] P. Sturm, "Algorithms for plane-based pose estimation," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, IEEE, vol. 1, 2000, pp. 706–711.

[31] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.

[32] J. Zhao, "An efficient solution to non-minimal case essential matrix estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[33] X. Armangué and J. Salvi, "Overall view regarding fundamental matrix estimation," *Image and vision computing*, vol. 21, no. 2, pp. 205–220, 2003.

[34] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.

[35] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[36] D. Martinec and T. Pajdla, "Robust rotation and translation estimation in multiview reconstruction," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–8.

[37] F. Chaumette and S. Hutchinson, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[38] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.

[39] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[40] F. Conticelli and B. Allotta, "Two-level visual control of dynamic look-and-move systems," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, IEEE, vol. 4, 2000, pp. 3784–3789.

[41] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, "A direct visual servoing-based framework for the 2016 iros autonomous drone racing challenge," *Journal of Field Robotics*, vol. 35, no. 1, pp. 146–166, 2018.

[42] D. Fernandez and A. Price, "Visual odometry for an outdoor mobile robot," in *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, IEEE, vol. 2, 2004, pp. 816–821.

[43] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: Types, approaches, challenges, and applications," *SpringerPlus*, vol. 5, no. 1, pp. 1–26, 2016.

[44] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[45] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. J. Berlles, "S-ptam: Stereo parallel tracking and mapping," *Robotics and Autonomous Systems*, vol. 93, pp. 27–42, 2017.

[46] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. DOI: 10.1109/TRO.2017.2705103.

[47] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," 2012.

[48] Z. Hong, Y. Petillot, and S. Wang, "RadarSLAM: Radar based large-scale SLAMm in all weathers," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 5164–5170.

[49] D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3D lidar-based online mapping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 5000–5007.

[50] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard, "Relocating underwater features autonomously using sonar-based slam," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 3, pp. 500–513, 2013.

[51] Z. Gong, R. Ying, F. Wen, J. Qian, and P. Liu, "Tightly coupled integration of gnss and vision SLAM using 10-dof optimization on manifold," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12 105–12 117, 2019.

[52] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[53] Z. Li, Y. Yan, Y. Jing, and S. Zhao, "The design and testing of a lidar platform for a uav for heritage mapping," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 1, p. 17, 2015.

[54] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.

[55] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax= xb," 1987.

[56] H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form ax= yb," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549–554, 1994.

[57] R.-h. Liang and J.-f. Mao, "Hand-eye calibration with a new linear decomposition algorithm," *Journal of Zhejiang University-SCIENCE A*, vol. 9, no. 10, pp. 1363–1368, 2008.

[58] R. L. Hirsh, G. N. DeSouza, and A. C. Kak, "An iterative approach to the hand-eye and base-world calibration problem," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, IEEE, vol. 3, 2001, pp. 2171–2176.

[59] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

[60] A. Li, L. Wang, and D. Wu, "Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product," *International Journal of Physical Sciences*, vol. 5, no. 10, pp. 1530–1536, 2010.

[61] W. Li, M. Dong, N. Lu, X. Lou, and P. Sun, "Simultaneous robot–world and hand–eye calibration without a calibration object," *Sensors*, vol. 18, no. 11, p. 3949, 2018.

[62] M. Shah, "Solving the robot-world/hand-eye calibration problem using the kronecker product," *Journal of Mechanisms and Robotics*, vol. 5, no. 3, p. 031 007, 2013.

[63] A. Tabb and K. M. Ahmad Yousef, "Solving the robot-world hand-eye (s) calibration problem with iterative methods," *Machine Vision and Applications*, vol. 28, no. 5, pp. 569–590, 2017.

[64] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *International journal of computer vision*, vol. 109, no. 3, pp. 252–286, 2014.

[65] A. Collet and S. S. Srinivasa, "Efficient multi-view object recognition and full pose estimation," in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 2050–2055.

[66] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *2011 IEEE intelligent vehicles symposium (IV)*, Ieee, 2011, pp. 963–968.

[67] KITTI. "The KITTI vision benchmark suite." (), [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_odometry.php (visited on 09/03/2022).

[68] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "Soft-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *Journal of field robotics*, vol. 35, no. 4, pp. 578–595, 2018.

[69] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *2013 International Conference on 3D Vision-3DV 2013*, IEEE, 2013, pp. 1–8.

[70] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense SLAM without a pose graph," Robotics: Science and Systems, 2015.

[71] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *2015 International Conference on Advanced Robotics (ICAR)*, IEEE, 2015, pp. 388–394.

[72] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE conference on computer vision and pattern recognition*, IEEE, 2007, pp. 1–8.

[73] R. Arandjelovic and A. Zisserman, "All about vlad," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.

[74] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1704–1716, 2011.

[75] A. R. Memon, H. Wang, and A. Hussain, "Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems," *Robotics and Autonomous Systems*, vol. 126, p. 103 470, 2020.

[76] P. Azad, T. Asfour, and R. Dillmann, "Combining harris interest points and the sift descriptor for fast scale-invariant object recognition," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 4275–4280.

[77] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Gámez, "Bidirectional loop closure detection on panoramas for visual navigation," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, IEEE, 2014, pp. 1378–1383.

[78] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[79] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *2006 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2006, pp. 4647–4653.

[80] O. Edlund, "A software package for sparse orthogonal factorization and updating," *ACM Transactions on Mathematical Software (TOMS)*, vol. 28, no. 4, pp. 448–482, 2002.

[81] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) method for pnp," in *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 383–390.

[82] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" In *European Conference on Computer Vision*, Springer, 2014, pp. 613–627.

[83] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian conference on computer vision*, Springer, 2010, pp. 25–38.

[84] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[85] R. Mur-Artal and J. D. Tardós, "Orb-SLAM2: An open-source SLAM system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[86] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," *arXiv preprint arXiv:1607.02555*, 2016.

[87] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[88] H. Jung, Y. Oto, O. M. Mozos, Y. Iwashita, and R. Kurazume, "Multi-modal panoramic 3D outdoor datasets for place categorization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4545–4550.

[89] M. Cordts, M. Omran, S. Ramos, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[90] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.

[91] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[92] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[93] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[94] M. Ferrera, J. Moras, P. Trouvé-Peloux, V. Creuze, and D. Dégez, "The aqualoc dataset: Towards real-time underwater localization from a visual-inertial-pressure acquisition system," *arXiv preprint arXiv:1809.07076*, 2018.

[95] M. Miller, S.-J. Chung, and S. Hutchinson, "The visual–inertial canoe dataset," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 13–20, 2018.

[96] A. Mallios, E. Vidal, R. Campos, and M. Carreras, "Underwater caves sonar data set," *The International Journal of Robotics Research*, vol. 36, no. 12, pp. 1247–1251, 2017.

[97] K. Leung, D. Lühr, H. Houshiar, *et al.*, "Chilean underground mine dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 16–23, 2017.

[98] P. Furgale, P. Carle, J. Enright, and T. D. Barfoot, "The devon island rover navigation dataset," *The International Journal of Robotics Research*, vol. 31, no. 6, pp. 707–713, 2012.

[99] K.-D. Park and J. Won, "The foliage effect on the height time series from permanent gps stations," *Earth, planets and space*, vol. 62, no. 11, pp. 849–856, 2010.

[100] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. Jacobo Berlles, "S-ptam: Stereo parallel tracking and mapping," *Robotics and Autonomous Systems (RAS)*, vol. 93, pp. 27–42, 2017, ISSN: 0921-8890. DOI: 10.1016/j.robot.2017.03.019.

[101] J. Garforth and B. Webb, "Visual appearance analysis of forest scenes for monocular SLAM," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 1794–1800.

[102] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: Direct sparse odometry with loop closure," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2198–2204.

[103] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland, "Penncosyvio: A challenging visual inertial odometry benchmark," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 3847–3854.

[104] S. K. Roy, M. Harandi, R. Nock, and R. Hartley, "Siamese networks: The tale of two manifolds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3046–3055.

[105] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[106] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[107] Y.-Y. Jau, R. Zhu, H. Su, and M. Chandraker, "Deep keypoint-based camera pose estimation with geometric constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 4950–4957.

[108] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 7286–7291.

[109] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[110] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 929–938.

PUBLICATIONS

PUBLICATION I

Methods for simultaneous robot-world-hand–eye calibration: A comparative study

I. Ali, O. Suominen, A. Gotchev, and E. R. Morales

Sensors, vol. 19, no. 12, p. 2837
DOI: 10.3390/s19122837

sensors

MDPI

Article

Methods for Simultaneous Robot-World-Hand–Eye Calibration: A Comparative Study

Ihtisham Ali [1,*], Olli Suominen [1], Atanas Gotchev [1] and Emilio Ruiz Morales [2]

[1] Faculty of Information Technology and Communication, Tampere University, 33720 Tampere, Finland; olli.j.suominen@tuni.fi (O.S.); atanas.gotchev@tuni.fi (A.G.)

[2] Fusion for Energy (F4E), ITER Delivery Department, Remote Handling Project Team, 08019 Barcelona, Spain; Emilio.Ruiz@f4e.europa.eu

* Correspondence: ihtisham.ali@tuni.fi; Tel.: +358-417-268-110

Received: 1 June 2019; Accepted: 21 June 2019; Published: 25 June 2019

Abstract: In this paper, we propose two novel methods for robot-world-hand–eye calibration and provide a comparative analysis against six state-of-the-art methods. We examine the calibration problem from two alternative geometrical interpretations, called 'hand–eye' and 'robot-world-hand–eye', respectively. The study analyses the effects of specifying the objective function as pose error or reprojection error minimization problem. We provide three real and three simulated datasets with rendered images as part of the study. In addition, we propose a robotic arm error modeling approach to be used along with the simulated datasets for generating a realistic response. The tests on simulated data are performed in both ideal cases and with pseudo-realistic robotic arm pose and visual noise. Our methods show significant improvement and robustness on many metrics in various scenarios compared to state-of-the-art methods.

Keywords: robot-world-hand–eye calibration; hand–eye calibration; optimization

1. Introduction

Hand–eye calibration is an essential component of vision-based robot control also known as visual servoing. Visual servoing effectively uses visual information from the camera as feedback to plan and control action and motion for various applications such as robotic grasping [1] and medical procedures [2]. All such applications require accurate hand–eye calibration primarily to complement the accurate robotic arm pose with the sensor-based measurement of the observed environment into a more complete set of information.

Hand–eye calibration requires accurate estimation of the homogenous transformation between the robot hand/end-effector and the optical frame of the camera affixed to the end effector. The problem can be formulated as $\boldsymbol{AX} = \boldsymbol{XB}$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the robotic arm and camera poses between two successive time frames, respectively, and $\boldsymbol{X}$ is the unknown transform between the robot hand (end effector) and the camera [3,4].

Alternatively, the estimation of a homogeneous transformation from the robot base to the calibration pattern/world coordinate system can be obtained as a byproduct of the problem solution widely known as robot-world-hand–eye (RWHE) calibration, formulated as $\boldsymbol{AX} = \boldsymbol{ZB}$. In this formulation, we define $\boldsymbol{X}$ as the transformation from robot base to world/pattern coordinate and $\boldsymbol{Z}$ is the transformation from the tool center point (TCP) to the camera frame. These two notations might be opposite in some other studies. The transformations $\boldsymbol{A}$ and $\boldsymbol{B}$ no longer represent the relative motion poses between different time instants. Instead, they now represent the transformation from TCP to the robot base frame, and the transformation from the camera to the world frame.

A considerable number of studies have been carried out to solve the problem of hand–eye calibration. While the core problem has been well addressed, the need for improved accuracy and robustness has increased with time as the hand–eye calibration problem expands to finds its uses in various fields of science.

The earliest approach presented for hand–eye calibration estimated the rotational and translational parts individually. Due to the nature of the approach, the solution is known as *separable* solution. Shiu and Ahmed [4] presented a closed-form approach to finding the solution for the problem formulation $\boldsymbol{AX} = \boldsymbol{XB}$ by separately estimating the rotation and translation from robot wrist to the camera in that order. The drawback of the approach presented was that the linear system doubles at each new entry of the image frame. Tsai [3] approached the problem from the same perspective, however, they improved the efficiency of the method by keeping the number of unknowns fixed irrespective of the number of images and robot poses. Moreover, the derivation is both simpler and computationally efficient compared to [4]. Zhuang [5] adopted the quaternion representation for solving the rotation transformation from hand to eye and robot base to the world. The translation components are then computed using linear least squares. Liang et al. [6] proposed a closed-form solution by linearly decomposing the relative poses. The implementation is relatively simple; however, the approach is not robust to noise in the measurements and suffers intensely in terms of accuracy. Hirsh et al. [7] proposed a separable approach that solves for $\boldsymbol{X}$ and $\boldsymbol{Z}$ alternatingly in an iterative process. The approach makes an assumption that one of the unknown is pseudo-known for that time being and estimates the best possible values for the other unknown by distributing the error. In the first case, it assumes that $\boldsymbol{Z}$ is known by the system and estimates $\boldsymbol{X}$ by averaging over the equation $\boldsymbol{X} = \boldsymbol{ZB_nA^{-1}}$ for all n poses of $\boldsymbol{B}$. Similarly, an estimation for $\boldsymbol{Z}$ is obtained by using the previously obtained $\boldsymbol{X}$. This process continues until the system reaches the condition to terminate the iterative estimation. In a recent study, Shah [8] proposed a separable approach that forms its bases on the methods presented by Li et al. [9]. Shah suggests using the Kronecker product to solve the hand–eye calibration problem. The method first computes the rotational matrices for the unknown $\boldsymbol{X}$, followed by computing the translation vectors. Kronecker product is an effective approach to estimate the optimal transformation in this problem. However, the resulting rotational matrices might not follow orthogonality. To compensate for this issue, the best approximations for orthonormal rotational matrices are obtained using Singular Value Decomposition (SVD). The primary difference between the work of [8] and [9] is that Li et al. do not update the positions that were only optimal for the rotational transformation before the orthonormal approximation. This augments to any errors that might already be present in the solution. In contrast, Shah [8] explicitly re-computes the translations based on the new orthonormal approximations of the rotations $\boldsymbol{R_X}$ and $\boldsymbol{R_Z}$. Earlier studies have shown that separable approaches have a core limitation, which results in slightly high position errors. Since the orientations and translation are computed independently and in the mentioned order, the errors from orientations step propagate to the position estimation step. Typically, separate solution based approaches have good orientation accuracy; however, the position accuracy is often compromised.

The second class of solutions is based on *simultaneous* computation of the orientation and position. Chen [10] argued that rotation and translation are interdependent quantities and, therefore, should not be estimated separately. He proposed a simultaneous approach to the hand–eye problem based on screw theory where both the rotation and translation components are computed altogether. In his work, Chen estimates a rigid transformation to align the camera screw axis to the robot screw axis. Dornaika and Horaud [11], proposed a nonlinear least square based approach to solve the hand–eye calibration problem. The optimization approach solved for an abundant number of parameters that represent rotations in the form of matrices. The cost function constrained the optimization to solve for orthonormal rotation matrices. It was observed that the nonlinear iterative approach yielded better results to linear and closed form solution in term of accuracy. Henceforth, many studies have opted for nonlinear cost minimization approach since they are more tolerant to nonlinearities present in measurements in the form of noise and errors. Shi et al. [12] proposed to replace the rotation

matrices with quaternion representation to facilitate the iterative optimization approach towards a solution. In [13], Wei et al. contributed an approach for an online hand–eye calibration approach that estimate the transformations through active motion. The method discards degenerative cases where no or little rotation cases induce high errors into the system. Strobel and Hirzinger [14], proposed an adaptive technique for hand–eye calibration using nonlinear optimization. The approach adjusts weights that are assigned to the rotation and translation errors during the cost minimization step. In [15], Fassi and Legnai construed a geometrical interpretation of the hand–eye calibration problem for the formulation $\boldsymbol{AX} = \boldsymbol{XB}$. They argued that the general formulation can lead to an infinite solution and therefore a constrained multi-equation based system is always suitable to optimize. Some cases that result in singularity were also discussed. Zhao [16] presents a convex cost function by employing the Kronecker product in both rotational matrix and quaternion form. The study argues that a global solution can be obtained using linear optimization without specifying any initial points. This serves as an advantage over using L2 based optimization. Heller et al. [17] proposed a solution to the hand–eye calibration problem using the branch-and-bound (BnB) method introduced in [18]. The authors minimize the cost function under the epipolar constraints and claim to yield a globally optimum solution with respect to $L_\infty-$norm. Tabb [19] tackled the problem of hand–eye calibration from the iterative optimization based approach and compared the performance of various objective functions. The study focused on $\boldsymbol{AX} = \boldsymbol{ZB}$ formulation and solved for the orientation and translation both separately and simultaneously using the nonlinear optimizer. Moreover, a variety of rotation representations was adopted including Euler, rotation matrix and quaternion in order to study their effect on accuracy. The study explored the possibility of a robust and accurate solution by minimizing pose and reprojection errors using different costs. The authors used the nonlinear optimizer Ceres [20] to solve for a solution using the Levenberg-Marquardt algorithm.

In this study, we present a collection of iterative methods for the hand–eye calibration problem under both $\boldsymbol{AX} = \boldsymbol{XB}$ and $\boldsymbol{AX} = \boldsymbol{ZB}$ formulations. We adopt the iterative cost minimization based approach similar to Tabb [19]. However, the geometrical formulation is reverted to the generic form for better coherence. Moreover, we study the problem from $\boldsymbol{AX} = \boldsymbol{XB}$ formulation, which is not present in [19]. The prospects of a new cost functions for the non-linear regression step are also studied. Each method is quantified from pose optimization and reprojection error minimization perspective. The main contributions of this study are as follows:

(1) We provide a comprehensive analysis and comparison of various cost functions for various problem formulations.
(2) We provide a dataset composed of three simulated sequences and three real data sequence, which we believe is handful for testing and validation by the research community. To the best of our knowledge, this is the first simulated data set for hand–eye calibration with synthetic images that are available for public use. Moreover, the real data sequences include chess and ChArUco calibration board of varying sizes. The datasets are available from [21].
(3) We provide extensive testing and validation results on a simulated dataset with realistic robot (position and orientation) and camera noise to allow comparisons between the estimated and true solutions more accurately.
(4) We provide an open-source code of the implementation of this study along with the surveyed approaches to support reproducible research. The code is available from [21].

The article is organized as follows: In Section 2, we present in detail the problem formulations for robot-world-hand–eye calibration. In Section 3, we discuss the development of real and synthetic dataset for evaluation purpose. Section 4 presents the error metrics used to quantify the performance of the calibration methods. Section 5 summarizes the experimental results using both synthetic and real datasets against the aforementioned error metrics. Finally, Section 6 concludes the article.

2. Methods

For the needs of our study, we introduce notations, as illustrated in Figure 1. Throughout this article, we will represent homogenous transformations by $\boldsymbol{T}$ supported with various sub-indexes. The sub-indexes b, t, c and w indicate the coordinate frames associated with robot base, robot tool, camera and the calibration pattern, respectively. The sub-indexes i and j are associated with time instants of the state of the system. For the first general formulation $\boldsymbol{AX} = \boldsymbol{XB}$ illustrated in Figure 1a, ${}_{b}\boldsymbol{T}^{\boldsymbol{b}_i}$ is the equivalent to $\boldsymbol{A_i}$ and denotes the homogenous transformations from robot base to the tool center point (TCP)/end-effector. ${}_{\boldsymbol{c_i}}\boldsymbol{T}^{\boldsymbol{w}}$ is the equivalent of $\boldsymbol{B_i}$ and denotes the homogenous transformation from camera to the world/calibration pattern. The formulation uses the relative transformation $\boldsymbol{A}\ ({}_{\boldsymbol{t_j}}\boldsymbol{T}^{\boldsymbol{t_i}})$. and $\boldsymbol{B}\ ({}_{\boldsymbol{c_j}}\boldsymbol{T}^{\boldsymbol{c_i}})$ from their respective previous pose to another pose. The unknown $\boldsymbol{X}$ or ${}_{\boldsymbol{t}}\boldsymbol{T}^{\boldsymbol{c}}$ is the required homogenous transformation from the end effector to the camera.

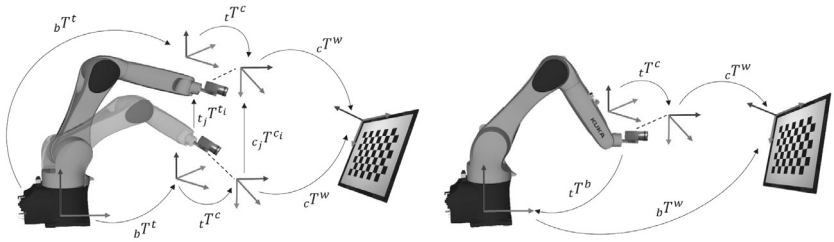


Figure 1. Formulations relating geometrical transformation for calibration; (**a**) hand–eye calibration; (**b**) robot-world-hand–eye Calibration.

The second general formulation, $\boldsymbol{AX} = \boldsymbol{ZB}$ is illustrated in Figure 1b. The formulation uses absolute transformation $\boldsymbol{A}({}_{\boldsymbol{t}}\boldsymbol{T}^{\boldsymbol{b}})$ and $\boldsymbol{B}({}_{\boldsymbol{c}}\boldsymbol{T}^{\boldsymbol{w}})$ from their respective coordinate frames. The unknown $\boldsymbol{X}({}_{\boldsymbol{b}}\boldsymbol{T}^{\boldsymbol{w}})$ and $\boldsymbol{Z}({}_{\boldsymbol{t}}\boldsymbol{T}^{\boldsymbol{c}})$ are the homogenous transformations from robot base to the world frame and the end effector to the camera frame, respectively. The hand–eye transformation is referred to as $\mathbf{Z}$ in this formulation for coherence in literature, since many studies opt for such notation.

In this section, we focus on various cost functions for the two general problem formulations with the aim to analyze their performance under real situations. For both cases, we consider solving the problem by minimizing pose error and reprojection error. Some studies including [19] propose to optimize the camera's intrinsic parameters using the nonlinear solver to yield better results. However, Koide and Menegatti [22] argue that the approach involving camera intrinsic optimization overfits the model on the data for the reprojection error; consequently, the results are poor for other error metrics including reconstruction accuracy. Following the insight from [22], we solve for the transformation by minimizing the reprojection error.

The main information required for hand–eye calibration are the Tool Centre Point (TCP)/end effector poses and the camera poses. The TCP pose of the robotic arm is directly provided by the software of the robotic arm against the base of the arm. The pose is typically quite accurate due to the high accuracy of the encoders in the robotic arm that provide feedback for the angles of the joints. In general, for many robotic arms, the precision for the end effector's position is around 0.1–0.2 mm. On the other hand, the camera pose against the world frame can be obtained through various methods. The common approach is to use a calibration pattern for simultaneously calculating the calibration parameters of the camera and the pose of the camera against the pattern or in this case world frame. Many researchers favor this approach since the calibration pattern is easy to acquire and its use yields good results. In contrast, some studies [23,24] prefer Structure from Motion (SFM) to acquire the relative camera transformation when the camera is moved from one point to another. The approach is independent of the calibration pattern and can acquire the correspondences from the feature-rich environment. However, SFM based camera calibration and camera pose computation

are prone to errors. The approach inherits additional errors into the hand–eye calibration process and reduces the overall accuracy of the system. To compensate for these errors, the process must include additional steps to mitigate the effects. The added efforts deviate the focus from the core target, which is accurate hand–eye calibration. In this study, we utilize industrial-grade calibration boards in order to estimate the camera intrinsic parameters and camera extrinsics for the robot-world-hand–eye calibration problem. The camera calibration approach used in this study is based on the widely adopted method by Zhang [25].

2.1. Hand–Eye Formulation

This mathematical problem formulation involves estimating one unknown with the help of two known homogenous transformations in a single equation. Let ${}_{b}\boldsymbol{T}^{t_i}$ be the homogenous transformation from the base of the robot to the robot TCP. The homogenous transformation relating the camera coordinate frame to the world coordinate frame affixed to the calibration patters is ${}_{c_i}\boldsymbol{T}^{w}$. The unknown homogenous transformation from the tool to the camera coordinate frame to be estimated is represented by ${}_{t}\boldsymbol{T}^{c}$. Then from Figure 1a, we can form the following relationship

$$ {}_{b}\boldsymbol{T}^{t_2{-}1}\ {}_{b}\boldsymbol{T}^{t_1}\ {}_{t}\boldsymbol{T}^{c} = {}_{t}\boldsymbol{T}^{c}\ {}_{c_2}\boldsymbol{T}^{w}\ {}_{c_1}\boldsymbol{T}^{w-1} \leftarrow \left({}_{t_1}\boldsymbol{T}^{c_1} = {}_{t_2}\boldsymbol{T}^{c_2}\right) \tag{1} $$

$$ {}_{t_2}\boldsymbol{T}^{b}\ {}_{b}\boldsymbol{T}^{t_1}\ {}_{t}\boldsymbol{T}^{c} = {}_{t}\boldsymbol{T}^{c}\ {}_{c_2}\boldsymbol{T}^{w}\ {}_{w}\boldsymbol{T}^{c_1}. \tag{2} $$

Generalizing Equation (2) gives us Equation (3).

$$ {}_{t_j}\boldsymbol{T}^{t_i}\ {}_{t}\boldsymbol{T}^{c} = {}_{t}\boldsymbol{T}^{c}\ {}_{c_j}\boldsymbol{T}^{c_i} \tag{3} $$

$$ \begin{bmatrix} {}_{t_j}\boldsymbol{R}^{t_i} & {}_{t_j}\boldsymbol{T}^{t_i} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} {}_{t}\boldsymbol{R}^{c} & {}_{t}\boldsymbol{t}^{c} \\ 0_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} {}_{t}\boldsymbol{R}^{c} & {}_{t}\boldsymbol{t}^{c} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} {}_{c_j}\boldsymbol{R}^{c_i} & {}_{c_j}\boldsymbol{t}^{c_i} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{4} $$

Equation (4) represents the direct geometrical relationship between various coordinate frames involved in the system. In order to attain a solution and achieve dependable results it is required that the data is recorded for at least 3 positions with non-parallel movements of the rotational axis [14]. We can directly minimize the relationship in Equation (4) to estimate the unknown parameters presented in Equation (5). In the experimentation section, we refer to the cost functions in Equations (5) and (6) as Xc1 and Xc2, respectively.

$$ \left\{q_{(t,c)}, {}_{t}\boldsymbol{t}^{c}\right\} = \underset{q_{(t,c)},\ {}_{t}\boldsymbol{t}^{c}}{\operatorname{argmin}} \sum_{i=1,j=i+1}^{n-1} \left\| \overline{n}\left({}_{t_j}\boldsymbol{T}^{t_i} \left[q_{(t,c)}, {}_{t}\boldsymbol{t}^{c}\right]_{HT} - \left[q_{(t,c)}, {}_{t}\boldsymbol{t}^{c}\right]_{HT} {}_{c_j}\boldsymbol{T}^{c_i} \right) \right\|_2^2 \tag{5} $$

In light of recommendation of [19], we can also re-arrange Equation (5) in the following manner.

$$ \left\{q_{(t,c)}, {}_{t}\boldsymbol{t}^{c}\right\} = \underset{q_{(t,c)},\ {}_{t}\boldsymbol{t}^{c}}{\operatorname{argmin}} \sum_{i=1,j=i+1}^{n-1} \left\| \overline{n}\left({}_{t_j}\boldsymbol{T}^{t_i} - \left[q_{(t,c)}, {}_{t}\boldsymbol{t}^{c}\right]_{HT} {}_{c_j}\boldsymbol{T}^{c_i} \left[\widetilde{q}_{(t,c)}, \widetilde{{}_{t}\boldsymbol{t}^{c}}\right]_{HT} \right) \right\|_2^2 \tag{6} $$

Here, the symbol $[]_{HT}$ denotes the conversion of the parameters to homogenous transformation representation. The solver optimizes the parameters in quaternion representation $q_{(t,c)}$ of the rotational matrix ${}_{t}\boldsymbol{R}^{c}$ and translation ${}_{t}\boldsymbol{t}^{c}$. The operation $\overline{n}$ denotes the aggregation of the 4×4-error matrix into a scalar value by summation of normalized values of quaternion angles and normalized translation vector. The terms $\widetilde{q}_{(t,c)}$ and $\widetilde{{}_{t}\boldsymbol{t}^{c}}$ are the quaternion and translation vector obtained from the inverse of ${}_{t}\boldsymbol{T}^{c}$. The objective functions minimize the L2-norm of the residual scaler values. The solutions in Equations (5) and (6) belong to the simultaneous solution category of hand–eye calibration because the rotation and translation are solved at the same time. We use the Levenberg –Marquardt algorithm to search for a minimum in the search space. The objective function successfully converges to a solution without any initial estimates for the $q_{(t,c)}$ and ${}_{t}\boldsymbol{t}^{c}$. We have observed that the cost function in

Equation (6) enjoys a slight improvement in some cases over Equation (5), which will be discussed in the experimental results and discussion section.

The second approach to seek a solution is based on reprojection-based methods. Reprojection error minimization has shown promising results for pose estimation in various problem cases [26,27]. Tabb [19] examined the reprojection-based method for the $\boldsymbol{AX} = \boldsymbol{ZB}$ formulation. We generalize this approach for the case of the $\boldsymbol{AX} = \boldsymbol{XB}$ formulation. Let W be the 3D points in the world frame and P^c be the same points in the camera frame. In the case of the chessboard pattern, these points are the corners of the chessboard. The following relationship represents the objective function for minimizing the reprojection error of the 3D points from pose i to pose j. The cost function in Equation (7) is referred to as RX here onwards.

$$\left\{q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}\right\} = \underset{q_{(t,c)}\,,\,{}_t\boldsymbol{t}^{\boldsymbol{c}}}{\operatorname{argmin}} \sum_{i=1,j=i+1}^{n-1} \| \overline{P}_j \quad - \Pi\Big(\quad K, \quad \left[\widetilde{q}_{(t,c)}, {}_t\widetilde{\boldsymbol{t}}^{\boldsymbol{c}}\right]_{HT} \; {}_{t_j}\boldsymbol{T}^{\boldsymbol{t_i}} \left[q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}\right]_{HT}, \; P_i^c \Big) \|_2^2 \tag{7}$$

In the equation, Π represents the operation that projects the 3D points from world space to image space using the camera intrinsic K and the camera extrinsic obtained using the homogenous transformations given in Equation (7), while $\overline{P}_j$ are the observed 2D points in the j-th image.

It is important to note that the reprojection error minimization based approach is not invariant to the choice of initial estimates for the solver. However, if a good initial estimate is provided, the nonlinear optimization of reprojection error can provide a more accurate solution with a fine resolution.

2.2. Robot-World-Hand–Eye Formulation

This mathematical formulation involves the estimation of an additional homogenous transformation that is between the robot base frame and world frame. Therefore, we have two known and two unknown homogenous transformations. Let ${}_t\boldsymbol{T}^{\boldsymbol{b}}$. be the homogenous transformation from robot TCP to the base of the robot. The homogenous transformation relating the camera coordinate frame to the world coordinate is ${}_c\boldsymbol{T}^{\boldsymbol{w}}$.The additional unknown homogenous transformation from the robot base frame to the world frame is ${}_b\boldsymbol{T}^{\boldsymbol{w}}$. Then from Figure 1b, we can form a straightforward geometrical relationship as:

$${}_t\boldsymbol{T}^{\boldsymbol{b}}\,{}_b\boldsymbol{T}^{\boldsymbol{w}} = {}_t\boldsymbol{T}^{\boldsymbol{c}}\,{}_c\boldsymbol{T}^{\boldsymbol{w}} \tag{8}$$

$$\begin{bmatrix} {}_t\boldsymbol{R}^{\boldsymbol{b}} & {}_t\boldsymbol{t}^{\boldsymbol{b}} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} {}_b\boldsymbol{R}^{\boldsymbol{w}} & {}_b\boldsymbol{t}^{\boldsymbol{w}} \\ 0_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} {}_t\boldsymbol{R}^{\boldsymbol{c}} & {}_t\boldsymbol{t}^{\boldsymbol{c}} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} {}_c\boldsymbol{R}^{\boldsymbol{w}} & {}_c\boldsymbol{t}^{\boldsymbol{w}} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{9}$$

Similar to the previous cases, we can directly use the relationship in aforementioned equations to obtain ${}_t\boldsymbol{T}^{\boldsymbol{c}}$ and ${}_b\boldsymbol{T}^{\boldsymbol{w}}$ using nonlinear minimization of their respective costs

$$\left\{q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}\right\} = \underset{q_{(t,c)}\,,\,{}_t\boldsymbol{t}^{\boldsymbol{c}},\, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}}{\operatorname{argmin}} \sum_{i=1}^{n} \| \overline{n}\Big({}_t\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{b}} \left[q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}\right]_{HT} - \left[q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}\right]_{HT} {}_c\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{w}} \Big) \|_2^2 \tag{10}$$

We can observe from Equation (10), that we are attempting to solve for two unknown homogenous transformations. The adopted parametrization involves optimizing over 14 parameters, where the two quaternions and translation vectors contribute to 8 and 6 parameters, respectively. While the robot-world-hand–eye calibration involves more unknowns for estimation, nonetheless, it constrains the geometry with more anchor points and helps to converge closer to the global minimum. With the advent of modern nonlinear solvers, the problem of optimizing for a large number of unknowns has become simpler and more efficient. As before, the objective function in Equation (10) can be re-arranged in the form of Equation (11). The cost functions in Equations (10) and (11) are referred to as Zc1 and Zc2, respectively, in Tabb [19]

$$\left\{q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}\right\} = \underset{q_{(t,c)}\,,\,{}_t\boldsymbol{t}^{\boldsymbol{c}},\, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}}{\operatorname{argmin}} \sum_{i=1}^{n} \| \overline{n}\Big({}_t\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{b}} \; - \left[q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}\right]_{HT} \; {}_c\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{w}} \left[\widetilde{q}_{(b,w)}, {}_b\widetilde{\boldsymbol{t}}^{\boldsymbol{w}}\right]_{\boldsymbol{HT}} \Big) \|_2^2 \tag{11}$$

The objective function successfully converges to a solution for $q_{(t,c)}$, ${}_t\boldsymbol{t}^{\boldsymbol{c}}$, $q_{(b,w)}$ and ${}_b\boldsymbol{t}^{\boldsymbol{w}}$. However, the primary difference here is that the solver depends on initialization. In case of bad initial estimates, the optimization algorithm might not converge to a stable solution. However, the formulation presented is not a high dimensional optimization problem and therefore, a rough initial estimate is sufficient. The initial estimates can be acquired from any fast closed-form method such as Tsai [3] or Shah [8].

This formulation can also be viewed as reprojection error minimization problem. The following equation presents a cost function that minimizes the reprojection of the 3D world points W onto the image space in camera frame, where $\overline{P}_i$ are the observed 2D points in the i-th image. The cost functions in Equation (12) is referred to as rp1 in [19].

$$\left\{q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}\right\} = \underset{q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\overline{P}_i - \Pi\Big(K, \left[\widetilde{q}_{(t,c)}, \widetilde{\boldsymbol{t}^{\boldsymbol{c}}}\right]_{HT} {}_t\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{b}} \left[q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}\right]_{HT}, W\Big)\|_2^2 \tag{12}$$

In contrast to the reprojection error cost function for problem formulation $= XB$, this formulation from [19] has the added advantage that it is not explicitly affected by the errors in pose estimation caused by blurred images or low camera resolution. If the camera intrinsic parameters are accurate enough, then the extrinsic can be indirectly computed as a transformation through ${}_t\boldsymbol{T}^{\boldsymbol{c}}$, ${}_t\boldsymbol{T}^{\boldsymbol{b}}$ and ${}_b\boldsymbol{T}^{\boldsymbol{w}}$ through the minimization of the objective function. On the contrary, the reprojection error cost function presented for problem formulation $\boldsymbol{AX} = \boldsymbol{XB}$ is more robust to robot pose errors given good images.

A marginal improvement in the results can be observed in various cases by using $\log(\cosh(x))$ as the loss function. The relative improvement is discussed in detail in Section 5. $\log(\cosh(x))$ approximates $\frac{x^2}{2}$ for small value of x and $abs(x) - \log(2)$, for large values. This essentially means that $\log(\cosh(x))$ imitates the behavior of the mean squared error but is more robust to noise and outliers. Moreover, the function is twice differentiable everywhere and therefore does not deteriorate the convexity of the problem. The modified version is given as followed, where $E(x)$ is the error in terms of difference between the observed points and the reprojected points. The cost function in Equation (13) is referred to as RZ hereafter.

$$\left\{q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}\right\} = \underset{q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}, q_{(b,w)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}}{\operatorname{argmin}} \sum_{i=1}^{n} \| \log(\cosh(\, E(x)))\|_2^2 \tag{13}$$

3. Performance Evaluation Using Datasets

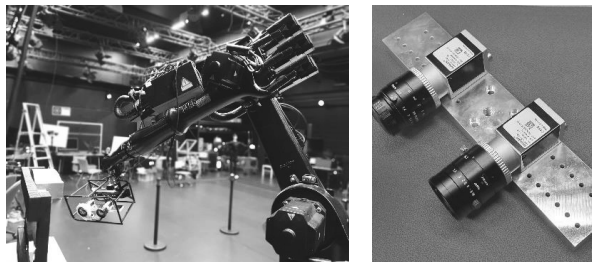
In order to assess the performance of the robot-world-hand–eye calibration methods, we present multiple datasets to test the methods in laboratory and near field settings. These datasets contain data acquired using various combinations of camera, lens, calibration patterns and robot poses. A detailed description of datasets is provided in Table 1. The table also lists the length of each side of square of the calibration patterns, focal length of the lenses, and number of robot poses used to acquire images. The datasets include real data and simulated data with synthetic images. To the best of our knowledge, this study is the first to provide simulated robot-world-hand–eye calibration dataset with synthetic/rendered images as open source for public use. A more detailed explanation of the datasets is presented in the following subsections.

Table 1. Description of the dataset acquired and generated for testing.

No.	Dataset	Data Type	Lens Focal Length [mm]	Square Size [mm]	Image Size	Robot	Poses
1	kuka_1	Real	12	20	1928 × 1208	KR16L6-2	30
2	kuka_2	Real	16	15	1920 × 1200	KR16L6-2	28
3	kuka_3	Real	12	60	1928 × 1208	KR16L6-2	29
4	CS_synthetic_1	Simulated	18	200	1920 × 1080	N/A	15
5	CS_synthetic_2	Simulated	18	200	1920 × 1080	N/A	19
6	CS_synthetic_3	Simulated	18	200	1920 × 1080	N/A	30

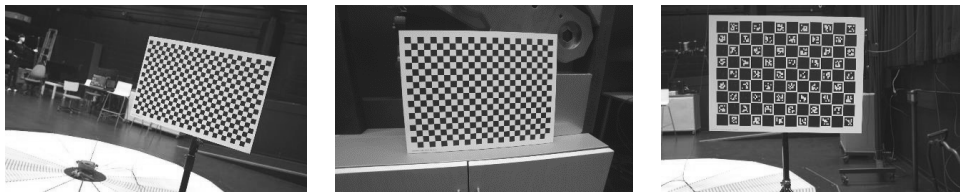
3.1. Real Datasets

To acquire real data for this experiment, a KUKA KR16L6-2 serial 6-DOF robot arm was used with Basler acA1920-50gc camera using 12 mm and 16 mm lenses as shown in Figure 2. The primary aim in recording these datasets was to collect real data for various robot-world-hand–eye calibration tests. With this aim, the collection provides three real datasets with varying robot poses and calibration patterns as shown in Figure 3. In this study, we primarily use the chessboard pattern for accurate camera calibration and robot-world-hand–eye calibration. A minor yet significant difference between the datasets [28], used in [19], is that the robot hand/camera orientation changes are quite gentle. This is done to facilitate the OpenCV camera calibration implementation used in [19], therefore the aforementioned implementation is not invariant to significant orientation changes and as a result, it flips the origin of the calibration pattern. For our experiments, we utilized MATLAB's implementation of [25], which can correctly detect the orientation of the pattern in any given pose. However, this neat trick requires that the calibration pattern is asymmetric in the number of rows and columns and that one of the sides has an even number of squares while the other side has odd. This requirement makes the datasets in [28], which have chessboard patterns with even number of rows and columns, unusable in our tests.



(**a**) (**b**)

Figure 2. An example of the setup for acquiring the datasets; (**a**) robotic arm moving in the workspace; (**b**) cameras and Lenses for data acquisition.



(**a**) (**b**) (**c**)

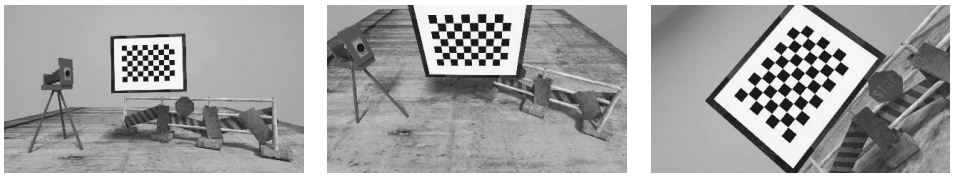
Figure 3. Example of captured images from the dataset 1 through 3; (**a**) checkerboard from dataset 1; (**b**) checkerboard from dataset 2; (**c**) ChArUco from dataset 3.

In addition, the calibration board used in the third dataset is a ChArUco pattern with square size of 60 mm, shown in Figure 3c. ChArUco tries to combine the benefit of both chessboard and ArUco markers and tends to facilitate the calibration process by fast, robust and accurate corner detection even in occluded or partial views [29]. The ChArUco dataset is only provided as open source material for future testing and has not been utilized in this study.

3.2. Simulated Dataset with Synthetic Images

The real data has the advantage of encapsulating all the uncertainties of a real system; however, in such cases we do not have any ground truth information. It is not possible to acquire the ground truth TCP-to-camera transformation, since the camera frame lies inside the camera. While various metric

for relative errors and error distribution can be used, nonetheless, the absolute pose error is always missing to quantify accuracy. The main purpose of using simulated data is to quantify the accuracy of the estimated poses against ground truth pose for various robot-world-hand–eye calibration methods. We provide three simulated datasets as part of the dataset package excerpts of which are shown in Figure 4. Each dataset provides different number of poses and complexity through the orientation of the camera. The simulated data comprises of synthetic images generated in Blender [30], a 3D computer graphics software, of the specifications mentioned in Table 1. For simplification, we assume that the camera position is the same, as the robot TCP position. Then the homogenous transformation from hand-to-eye constitutes of rotation resulting from the orientation difference between the Blender world frame and Blender camera frame.



(**a**) (**b**) (**c**)

Figure 4. Example of rendered images for simulated datasets from the datasets 4 through 6; (**a**) excerpt from dataset 4; (**b**) excerpt from dataset 5; (**c**) excerpt from dataset 6.

3.3. Pseudo-Real Noise Modeling

While simulated data carries the advantage of providing the ground truth information for various robot-world-hand–eye calibration, the limitation is that it lacks the uncertainties of the real world situations. These uncertainties could originate from either robot TCP pose errors or camera pose errors. Many studies [19,22,31] suggest testing the robustness of the methods by inducing one type of noise at a time into the system and evaluating its performance based on the response. Unfortunately, these uncertainties are mostly co-existent and co-dependent in real-world cases. In this study, we propose to model the uncertainties in terms of pose and pixel errors and induce a realistic amount of noise simultaneously into the simulated dataset for testing. The motivation behind inducing such type of noise is to carry the advantage of testing simulated data for accuracy and adjoining it with the robustness of testing on real data.

We aim at introducing a realistic amount of noise. The robot position repeatability is generally provided in the datasheets, which ranges from 0.1–0.3 mm for various robots. However, the orientation repeatability is absent since it cannot be measured for real robots at such a fine resolution. Here, we propose a reverse engineering approach to acquire a statistically valid amount of orientation noise. The position and orientation error of the TCP arises from the accumulated errors of the individual joints of the robotic arm due to robot flexibility and backlash. Using inverse kinematic we can find the joint angles for any position of TCP within its workspace.

Once the joint angles are available, we can introduce noise into the individual joints through trial and error until it produces the end-effector position error comparable to the realistic error. Through forward kinematics, we can then estimate the position and orientation of the end-effector under various arm configurations. Figure 5 shows the operation flow for computing the error range of the new orientations.

For our test, we used the position error of the KUKA KR16L6-2 computed through highly accurate laser sensor. The mean of the errors in X, Y and Z axes were 0.06 mm, –0.05 mm and –0.04 mm, while the standard deviation of the errors were 0.22 mm, 0.18 mm and 0.17 mm. A normally distributed error for each axis is generated based on these values and introduced to the system to estimate the corresponding effects in the orientation of the TCP. The range of realistic valid error for the TCP position is shown in Figure 6a, while the output of the orientation error using the aforementioned framework is shown in Figure 6b.

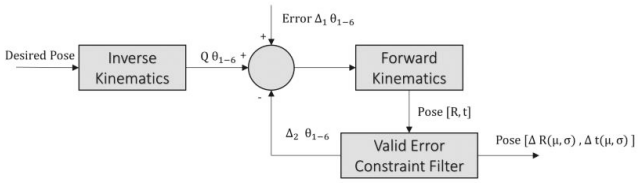
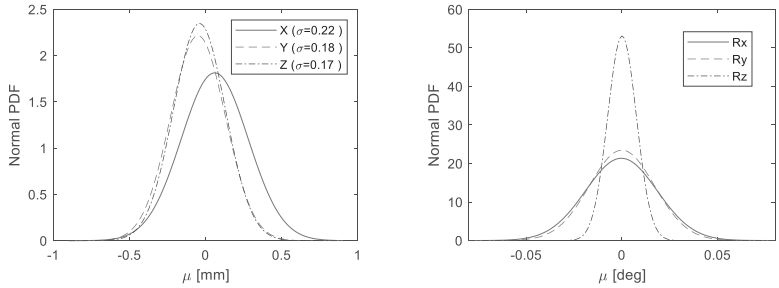


Figure 5. Flowchart of the orientation noise modelling approach.



(**a**) (**b**)

Figure 6. Probability distributions functions; (**a**) the measured position error from the robotic arm; (**b**) the modeled orientation error for the robotic arm.

4. Error Metrics

In order to compare the results of all the methods with other existing studies, we suggest to use mean rotation error (deg), mean translation error (mm), reprojection error (px), absolute rotation error (deg), and absolute translation error (mm). Each error metric has its own merits and demerits. We have avoided the use of reconstruction error since it involves further estimation of valid 3D points from the reprojected 2D points. This can be achieved by searching the space for such 3D points using nonlinear minimization, as before. However, it is not possible to segregate the error that arises from the pose estimation step and the reconstruction step, while using the error metric.

The first error is the mean rotation error derived from Equations (4) and (9) for $\boldsymbol{AX} = \boldsymbol{XB}$ and $\boldsymbol{AX} = \boldsymbol{ZB}$ formulation, respectively. Equation (16) gives the mean rotation error, which takes its input from Equations (14) and (15) for their respective formulation. Here, the angle represents the conversion from a rotation matrix to axis-angle for simpler user interpretation.

$$\Delta \boldsymbol{R} = ({}_t\boldsymbol{R}^{\boldsymbol{c}}\ {}_c\boldsymbol{R}^{\boldsymbol{w}})^{-1}\ {}_t\boldsymbol{R}^{\boldsymbol{b}}\ {}_b\boldsymbol{R}^{\boldsymbol{w}} \tag{14}$$

$$\Delta \boldsymbol{R} = \left({}_t\boldsymbol{R}^{\boldsymbol{c}}\ {}_{c_j}\boldsymbol{R}^{\boldsymbol{c_i}}\right)^{-1}\ {}_{t_j}\boldsymbol{R}^{\boldsymbol{t_i}}\ {}_t\boldsymbol{R}^{\boldsymbol{c}} \tag{15}$$

$$e_{rR} = \frac{1}{n}\sum_{i=1}^{n} \|angle(\Delta \boldsymbol{R})\|_2^2 \tag{16}$$

The second error metric focuses on computing the translation errors. As above, the translation errors emerge from the same Equations (4) and (9).

$$e_{rt} = \frac{1}{n}\sum_{i=1,j=i+1}^{n-1} \|({}_{t_j}\boldsymbol{R}^{\boldsymbol{t_i}}\ {}_t\boldsymbol{t}^{\boldsymbol{c}}) + {}_{t_j}\boldsymbol{t}^{\boldsymbol{t_i}} - ({}_t\boldsymbol{R}^{\boldsymbol{c}}\ {}_{c_j}\boldsymbol{t}^{\boldsymbol{c_i}}) + {}_t\boldsymbol{t}^{\boldsymbol{c}}\|_2^2 \tag{17}$$

$$e_{rt} = \frac{1}{n}\sum_{i=1}^{n} \|({}_t\boldsymbol{R}^{\boldsymbol{b}}\ {}_b\boldsymbol{t}^{\boldsymbol{w}}) + {}_t\boldsymbol{t}^{\boldsymbol{b}} - ({}_t\boldsymbol{R}^{\boldsymbol{c}}\ {}_c\boldsymbol{t}^{\boldsymbol{w}}) + {}_t\boldsymbol{t}^{\boldsymbol{c}}\|_2^2 \tag{18}$$

The third metric to measure the quality of the results is the reprojection root mean squared error (*rrmse*). The *rrmse* is measured in pixels and is a good metric to measure the quality of the results in the absence of ground truth information. The *rrmse* provides an added advantage that it back-projects the 3D points from the calibration board onto the images by first transforming them through the robotic arm. In case, if the hand eye calibration is not correct, the reprojection errors will be large. The *rrmse* for both the formulations are given in Equations (19) and (20).

$$e_{rrmse} = \sqrt{\frac{1}{n-1}\sum_{i=1,j=i+1}^{n-1} \|\overline{P}_j - \Pi\Big(K,\ \left[\widetilde{q}_{(t,c)}, {}_t\widetilde{\boldsymbol{t}}^{\boldsymbol{c}}\right]_{HT}\ {}_{t_j}\boldsymbol{T}^{\boldsymbol{t_i}}\left[q_{(t,c)}, {}_t\boldsymbol{t}^{\boldsymbol{c}}\right]_{HT},\ P_{\boldsymbol{i}}^{\boldsymbol{c}}\Big)\|_2^2} \tag{19}$$

$$e_{rrmse} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\overline{P}_i - \Pi\Big(K,\ \left[\widetilde{q}_{(t,c)}, {}_t\widetilde{\boldsymbol{t}}^{\boldsymbol{c}}\right]_{HT}\ {}_t\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{b}}\left[q_{(b,W)}, {}_b\boldsymbol{t}^{\boldsymbol{w}}\right]_{HT},\ W\Big)\|_2^2} \tag{20}$$

For the case of simulated data, we have accurate ground truth pose from the robot TCP to the camera. We can effectively utilize that information to acquire the absolute rotation error and absolute translation errors. The absolute rotation error can be obtained using Equation (21), while the absolute translation error is given using Equation (22). Here, ${}_t\boldsymbol{R}_{\boldsymbol{gt}}^{\boldsymbol{c}}$ and ${}_t\boldsymbol{t}_{\boldsymbol{gt}}^{\boldsymbol{c}}$ are the ground truth values.

$$e_{aR} = \|angle\Big({}_t\boldsymbol{R}^{\boldsymbol{c}\ -1}\ {}_t\boldsymbol{R}_{\boldsymbol{gt}}^{\boldsymbol{c}}\Big)\|_2^2 \tag{21}$$

$$e_{at} = \|{}_t\boldsymbol{t}_{\boldsymbol{gt}}^{\boldsymbol{c}} - {}_t\boldsymbol{t}^{\boldsymbol{c}})\|_2^2 \tag{22}$$

5. Experimental Results and Discussion

In this section, we report the experimental results for various cases and discuss the obtained results. We tabulate the results obtained for these cases using our own and six other studies to provide a clear comparison. Tables 1–4 summarize the results using the error metrics described in Section 4, over the datasets presented in Section 3. To elaborate on the naming, Xc1, Xc2, RX, and RZ refer to the optimization of the cost function based on Equations (5)–(7) and (13), respectively. In addition, Figure 7 illustrates the results from simulated data in dataset 5 over varying visual noise in the presence of the pseudo-realistic robotic arm pose noise. Tables 2 and 3 shows the evaluation of the aforementioned methods on datasets 1 and 2, respectively. Both datasets vary in the use of camera lenses and robot arm poses. It can be observed that the method by Shah [8] provides a better distribution of the rotational error and hence has the lowest relative rotation error (e_{rR}) values, while the method by Li et al. [9] yields a comparable result. The lowest relative translation error (e_{rt}) varies for both datasets and is yielded by the proposed method Xc2 and Park and Martin [32]. However, for dataset 2, it seems that Xc2 has not converged properly and has obtained a local minimum. On the other hand, the method by Park and Martin [32], still yields a relatively low e_{rt}. Moreover, for both datasets 1 and 2, the method by Horaud and Dornaika [11] provides comparable results to Park and Martin [32].

For the reprojection root mean squared error e_{rrmse}, the best results are obtained using the proposed RX approach for both tests. This is aided by the fact that the recorded datasets do not have large visual errors and as a result, RX performs comparably better. Moreover, since the cost function has only one unknown transformation to minimize for, the optimizer distributes the errors more evenly for the reprojection based cost function. Other reprojection based approaches namely Tabb's rp1 [19] and RZ achieve quite close results to RX. It is noteworthy, that in spite of being a closed-form approach, Shah [8] obtains quite good e_{rrmse} that are at a competitive level to the reprojection errors based approaches.

Table 2. Comparison of methods using the described error metrics for dataset 1.

Method	Evaluation Form	Relative Rotation (e_{rR})	Relative Translation (e_{rt})	Reprojection Error e_{rrmse}
Tsai [3]	AXXB	0.051508	1.1855	2.5386
Horaud and Dornaika [11]	AXXB	0.051082	1.0673	2.5102
Park and Martin [32]	AXXB	0.051046	1.0669	2.5091
Li et al. [9]	AXZB	0.043268	1.6106	2.5135
Shah [8]	AXZB	**0.042594**	1.5907	2.4828
Xc1	AXXB	0.11619	7.0582	17.806
Xc2	AXXB	0.075211	**0.71369**	3.3834
Tabb Zc1 [19]	AXXB	0.051092	1.1315	2.5796
Tabb Zc2 [19]	AXZB	0.10205	3.6313	5.2324
RX	AXXB	0.076491	1.7654	**2.3673**
Tabb rp1 [19]	AXZB	0.066738	1.9455	2.4004
RZ	AXZB	0.079488	2.0806	2.4114

Table 3. Comparison of methods using the described error metrics for dataset 2.

Method	Evaluation Form	Relative Rotation (e_{rR})	Relative Translation (e_{rt})	Reprojection Error e_{rrmse}
Tsai [3]	AXXB	0.046162	0.48363	1.9944
Horaud and Dornaika [11]	AXXB	0.042587	0.4104	1.3804
Park and Martin [32]	AXXB	0.042639	**0.41033**	1.3807
Li et al. [9]	AXZB	0.040297	39.535	61.466
Shah [8]	AXZB	**0.04028**	0.6078	1.5767
Xc1	AXXB	1.2697	10.038	54.436
Xc2	AXXB	9.7461	24.908	197.96
Tabb Zc1 [19]	AXXB	0.61435	4.9182	16.103
Tabb Zc2 [19]	AXZB	0.48439	13.518	23.672
RX	AXXB	0.092173	0.6726	**1.1234**
Tabb rp1 [19]	AXZB	0.16515	0.84439	1.1438
RZ	AXZB	0.14824	0.81163	1.1567

Table 4. Comparison of methods using the described error metrics for dataset 6.

Method	Evaluation Form	Relative Rotation (e_{rR})	Relative Translation (e_{rt})	Reprojection Error e_{rrmse}	Absolute Rotation Error (e_{aR})	Absolute Translation Error (e_{at})
Tsai [3]	AXXB	0.65051	50.062	20.423	1.1567	8.2512
Horaud and Dornaika [11]	AXXB	0.049173	6.2428	0.60685	0.028066	2.0674
Park and Martin [32]	AXXB	NaN	NaN	NaN	NaN	NaN
Li et al. [9]	AXZB	0.031909	3.6514	0.44024	0.012108	1.0889
Shah [8]	AXZB	0.032997	1.5195	0.18418	0.021235	1.0213
Xc1	AXXB	0.051304	5.7074	0.50083	0.0079584	0.73682
Xc2	AXXB	0.051239	5.7076	0.493	**0.0075352**	0.75278
Tabb Zc1 [19]	AXXB	0.049653	5.8363	0.45621	0.01299	0.97462
Tabb Zc2 [19]	AXZB	0.033778	1.9665	0.31189	0.011335	0.69158
RX	AXXB	0.049583	5.8213	0.34127	0.01078	0.25753
Tabb rp1 [19]	AXZB	**0.031857**	**1.0829**	**0.057526**	0.0085848	**0.19154**
RZ	AXZB	0.032432	1.1072	0.05826	0.0084204	0.21121

We further study the performance of the methods using our simulated datasets. The primary difference between dataset 4 and 6 is the number and complexity of the unique camera poses for image acquisition. During experimentation, we observed that the resolution of the accuracy slightly improved with the increased number of images acquired from significantly different poses. However, none of the methods suffered significantly from comparably less information in dataset 4, therefore, we consider datasets 5 and 6 for extensive quantitative comparison of the methods. In addition to the previous tabulated results, Tables 4 and 5 provide experimental results on simulated data with synthetic images from dataset 6. The main difference between the two tests is that the first test (Table 4) considers ideal simulated data, while the second test (Table 5) has visual and robot pose noise induced. The robot pose noise is derived from the process explained in Section 3.3, while the visual noise is selected such that the resultant reprojection error amounts to the reprojection errors of real data tests.

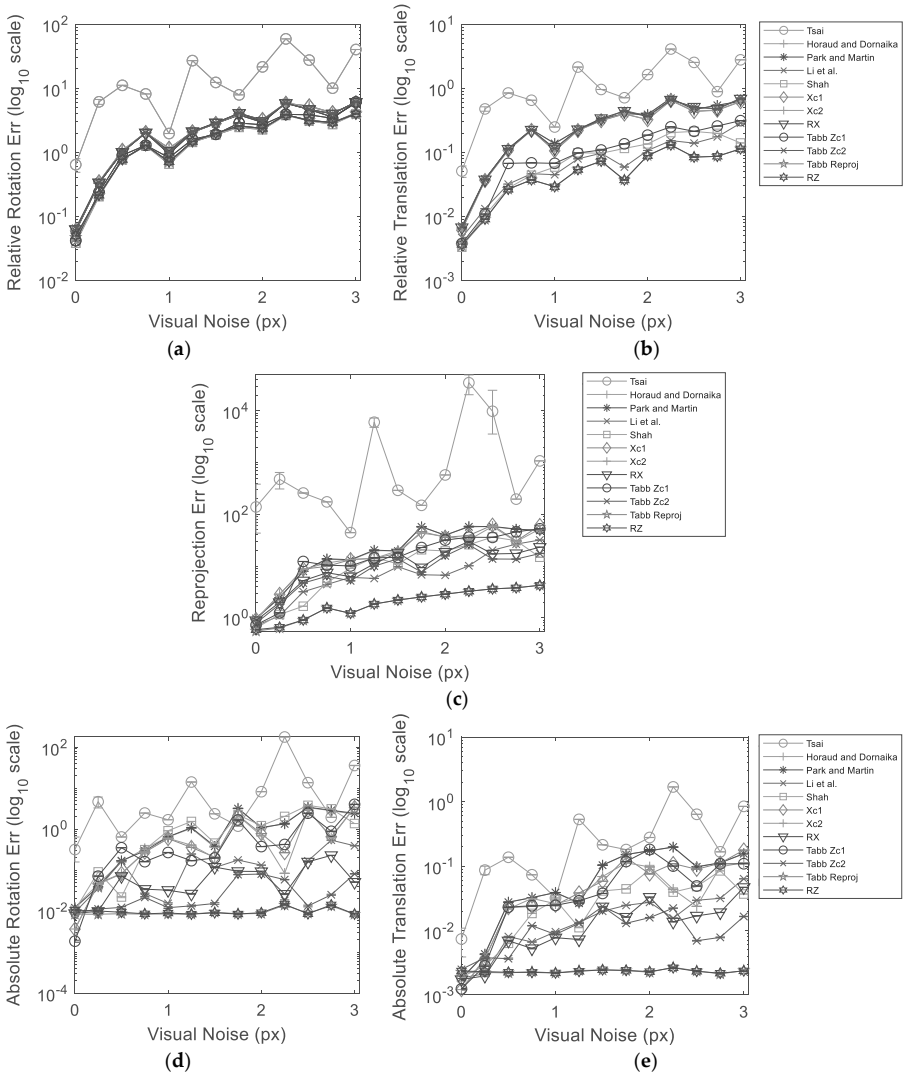


Figure 7. Metric error results for Dataset 5 with constant robot pose noise; (**a**) mean rotation error; (**b**) mean translation error; (**c**) reprojection error; (**d**) absolute rotation error against ground truth; (**e**) absolute translation error against ground truth.

Tables 4 and 5, present two absolute errors due to the presence of ground truth information for the simulated cases. It can be observed that Tabb's rp1 [19] achieves the least e_{rR}, e_{rt}, e_{rrmse} and e_{at}. Xc2 yields minimum Absolute Rotation Error (e_{aR}). For this dataset, the method by Park and Martin [32], failed to find a solution as it suffered from singularity. It is important to note for an ensued comparison that the proposed method RZ yields the second best results over most of the error metrics with minor differences from the least errors. This is important in a sense that all the errors are equally distributed and restricted close to their minimum values.

The backend experiments for the results in Table 5 use the same methods, metrics and dataset, as for Table 4. In agreement with the results of real data, Shah [8] yields the least e_{rR} for this dataset as well. In addition to a validation on the performance of Shah [8], this indicates that a realistic amount

of orientation noise is present in the system for the method to emanate similar response. The proposed method *RZ* yields the minimum e_{rt}, e_{rrmse} and, e_{at} and the second best result for e_{aR}. Tabb Zc1 [19] obtains the minimum e_{aR}.

Table 5. Comparison of methods using the described error metrics for dataset 6 with robot pose and visual noise.

Method	Evaluation Form	Relative Rotation (e_{rR})	Relative Translation (e_{rt})	Reprojection Error e_{rrmse}	Absolute Rotation Error (e_{aR})	Absolute Translation Error (e_{at})
Tsai [3]	AXXB	34.925	2476.4	99190	28.04	747.48
Horaud and Dornaika [11]	AXXB	1.723	199.92	18.764	0.43124	47.913
Park and Martin [32]	AXXB	1.7208	199.98	18.916	0.43819	47.733
Li et al. [9]	AXZB	1.177	80.061	7.8757	0.0029485	23.272
Shah [8]	AXZB	**1.1767**	58.552	8.5123	0.51765	8.3389
Xc1	AXXB	1.7752	192.86	17.442	0.12827	37.068
Xc2	AXXB	1.8026	193.22	19.031	0.20831	40.4
Tabb Zc1 [19]	AXXB	1.7989	206.01	13.445	**0.0042828**	11.368
Tabb Zc2 [19]	AXZB	1.2571	86.844	13.891	0.050182	27.247
RX	AXXB	1.8087	204.06	12.534	0.027714	7.0139
Tabb rp1 [19]	AXZB	1.2093	44.982	1.5463	0.0075401	0.95904
RZ	AXZB	1.2079	**44.932**	**1.546**	0.0069577	**0.95845**

This comparison demonstrates that the proposed *RZ* is more robust to outliers present in the data and performs marginally better compared to Tabb's rp1 [19] in the presence of noise.

Figure 7 shows the evaluation results for dataset 5 composed of simulated data. As before, the dataset is injected pseudo-realistic robotic arm pose noise and tested over varying realistic range of visual noise. The plots represent the averaged results over 1000 iterations in order to achieve a stable response. The 95% confidence interval from all the iterations for each experimentation point is also shown in Figure 7. It can be observed that the confidence intervals are quite narrow with the exception of the response of Tsai [3] over reprojection error metric. The narrow range of confidence interval indicates that we are 95% sure that our true mean lies somewhere within that narrow interval. Moreover, this implies that the noise introduced during different iterations is consistent in behavior and emulates a coherent response from the methods. The plot curves for each method pass through the mean values at each experimentation point. The results show that Tabb rp1 [19] and the proposed RZ are quite robust to the increments in visual noise compared to other methods over all error metrics. Moreover, at high visual noise RZ shows a slight improvement over Tabb rp1 [19]. It is noteworthy that despite the increase in relative rotation, translation and reprojection error, the absolute rotation and translation errors stay much more the same for Tabb rp1 [19] and RZ. Tsai [3] performs poorly and erratically in the presence of noise in data. In the absence of visual noise Tabb's Zc1 [19], Xc1, RX and Shah [8] can achieve lower errors compared to Tabb rp1 [19] and RZ for multiple metrics. However, real data always contains some magnitude of visual noise due to various reasons. The presence of visual noise may affect each method differently depending on the approach used. Nonetheless, the nonlinear reprojection based methods of the formulation $\boldsymbol{AX} = \boldsymbol{ZB}$ show good estimation results under visual noise and hand pose noise.

6. Conclusions

This study has examined the robot-world-hand–eye calibration problem in its two alternative geometrical interpretations, and has proposed a collection of novel methods. It benefits from non-linear optimizers that iteratively minimize the cost function and determine the transformations. We have conducted a comparative study to quantify the performances of optimizing over pose errors and reprojection errors. The code for the presented methods is provided as open-source for further use. Our collection of methods was evaluated with respect to state-of-the-art methods. The study also contributes new datasets for testing and validation purposes. These include subsets of three real data and three simulated data with synthetic images. Simulated data are beneficial as they provide ground truth. We have proposed a noise modeling approach to generate realistic robot TCP orientation noise to study the robustness of methods under realistic conditions. We showed that our methods perform

well under different testing conditions. RX yields good results with high accuracy under realistic visual noise with respect to reprojection error. In addition, RZ is more robust to visual noise and yields more consistent results for a greater range of visual noise.

Author Contributions: Conceptualization, I.A. and O.S.; Methodology, I.A.; Software, I.A.; Experiments and Validation, I.A.; Writing—Original Draft Preparation, I.A.; Writing—Review & Editing, E.R.M. and A.G.; Supervision, O.S. and A.G.; Funding Body Member, E.R.M.

Funding: The work presented in this paper was funded by Fusion for Energy (F4E) and Tampere University under the F4E grant contract F4E-GRT-0901. The results are intended to be integrated in advanced camera-based systems attached to manipulator arms in order to achieve complex Remote Handling maintenance operations in a safe and efficient way.

Conflicts of Interest: The authors declare no conflict of interest. This article reflects the views of the authors. F4E and TUNI cannot be held responsible for any use which may be made of the information contained herein.

References

1. Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; Quillen, D. Learning hand–eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* **2018**, *37*, 421–436. [CrossRef]
2. Pachtrachai, K.; Allan, M.; Pawar, V.; Hailes, S.; Stoyanov, D. Hand–eye calibration for robotic assisted minimally invasive surgery without a calibration object. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 2485–2491.
3. Tsai, R.; Lenz, R. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. Robot. Autom.* **1989**, *5*, 345–358. [CrossRef]
4. Shiu, Y.; Ahmad, S. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB. *IEEE Trans. Robot. Autom.* **1989**, *5*, 16–29. [CrossRef]
5. Zhuang, H.; Roth, Z.; Sudhakar, R. Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX=YB. *IEEE Trans. Robot. Autom.* **1994**, *10*, 549–554. [CrossRef]
6. Liang, R.; Mao, J. Hand–eye calibration with a new linear decomposition algorithm. *J. Zhejiang Univ. Sci. A* **2008**, *9*, 1363–1368. [CrossRef]
7. Hirsh, R.; DeSouza, G.; Kak, A. An iterative approach to the hand–eye and base-world calibration problem. In Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 21–26 May 2001; pp. 2171–2176.
8. Shah, M. Solving the Robot-world-hand–eye Calibration Problem Using the Kronecker Product. *J. Mech. Robot.* **2013**, *5*, 031007. [CrossRef]
9. Li, A.; Wang, L.; Wu, D. Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product. *Int. J. Phy. Sci.* **2010**, *5*, 1530–1536.
10. Chen, H. A screw motion approach to uniqueness analysis of head-eye geometry. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Maui, HI, USA, 3–6 June 1991; pp. 145–151.
11. Horaud, R.; Dornaika, F. Hand–eye Calibration. *Int. J. Robot. Res.* **1995**, *14*, 195–210. [CrossRef]
12. Shi, F.; Wang, J.; Liu, Y. An approach to improve online hand–eye calibration. In *Pattern Recognition and Image Analysis, Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Estoril, Portugal, 7–9 June 2015*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 647–655.
13. Wei, G.; Arbter, K.; Hirzinger, G. Active self-calibration of robotic eyes and hand–eye relationships with model identification. *IEEE Trans. Robot. Autom.* **1998**, *14*, 158–166. [CrossRef]
14. Strobl, K.H.; Hirzinger, G. Optimal hand–eye calibration. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4647–4653.
15. Fassi, I.; Legnani, G. Hand to sensor calibration: A geometrical interpretation of the matrix equation. *J. Robot. Syst.* **2005**, *22*, 497–506. [CrossRef]
16. Zhao, Z. Hand–eye calibration using convex optimization. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2947–2952.
17. Heller, J.; Havlena, M.; Pajdla, T. Globally Optimal Hand–eye Calibration Using Branch-and-Bound. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1027–1033. [CrossRef] [PubMed]

18. Hartley, R.; Kahl, F. Global Optimization through Rotation Space Search. *Int. J. Comput. Vis.* **2009**, *82*, 64–79. [CrossRef]
19. Tabb, A.; Ahmad Yousef, K. Solving the robot-world hand–eye(s) calibration problem with iterative methods. *Mach. Vis. Appl.* **2017**, *28*, 569–590. [CrossRef]
20. Agarwal, S.; Mierle, K. Ceres Solver — A Large Scale Non-linear Optimization Library. Available online: http://ceres-solver.org/ (accessed on 22 June 2019).
21. Ali, I. RWHE-Calib. Available online: https://github.com/ihtishamaliktk/RWHE-Calib (accessed on 31 May 2019).
22. Koide, K.; Menegatti, E. General Hand–Eye Calibration Based on Reprojection Error Minimization. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1021–1028. [CrossRef]
23. Zhi, X.; Schwertfeger, S. Simultaneous hand–eye calibration and reconstruction. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2017, Vancouver, BC, Canada, 24–28 September 2017; pp. 1470–1474.
24. Li, W.; Dong, M.; Lu, N.; Lou, X.; Sun, P. Simultaneous Robot–World and Hand–Eye Calibration without a Calibration Object. *Sensors* **2018**, *18*, 3949. [CrossRef] [PubMed]
25. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [CrossRef]
26. Edlund, O. A software package for sparse orthogonal factorization and updating. *ACM Trans. Math. Softw.* **2002**, *28*, 448–482. [CrossRef]
27. Hesch, J.; Roumeliotis, S. A Direct Least-Squares (DLS) method for PnP. In Proceedings of the International Conference on Computer Vision 2011, Barcelona, Spain, 6–13 November 2011; pp. 383–390.
28. Tabb, A. Data from: Solving the Robot-World Hand–eye(s) Calibration Problem with Iterative Methods, National Agricultural Library. Available online: http://dx.doi.org/10.15482/USDA.ADC/1340592 (accessed on 28 April 2019).
29. OpenCV: Detection of ChArUco Corners. Available online: https://docs.opencv.org/3.1.0/df/d4a/tutorial_charuco_detection (accessed on 28 April 2019).
30. Foundation, B. Blender.org - Home of the Blender Project - Free and Open 3D Creation Software. Available online: https://www.blender.org/ (accessed on 28 April 2019).
31. Lee, J.; Jeong, M. Stereo Camera Head-Eye Calibration Based on Minimum Variance Approach Using Surface Normal Vectors. *Sensors* **2018**, *18*, 3706. [CrossRef] [PubMed]
32. Park, F.; Martin, B. Robot sensor calibration: Solving AX=XB on the Euclidean group. *IEEE Trans. Robot. Autom.* **1994**, *10*, 717–721. [CrossRef]

PUBLICATION
II

Multi-view camera pose estimation for robotic arm manipulation

I. Ali, O. J. Suominen, E. R. Morales, and A. Gotchev

IEEE Access, vol. 8, pp. 174 305–174 316
DOI: 10.1109/ACCESS.2020.3026108

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 4, 2020, accepted September 10, 2020, date of publication September 23, 2020, date of current version October 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026108

Multi-View Camera Pose Estimation for Robotic Arm Manipulation

IHTISHAM ALI[1], OLLI J. SUOMINEN[1], EMILIO RUIZ MORALES[2], AND ATANAS GOTCHEV[1]
[1]Faculty of Information Technology and Communication Sciences, Tampere University, 33720 Tampere, Finland
[2]ITER Delivery Department, Remote Handling Project Team, Fusion for Energy (F4E), 08019 Barcelona, Spain

Corresponding author: Ihtisham Ali (ihtishamaliktk@gmail.com)

This work was supported in part by the Fusion for Energy (F4E) under Contract F4E-GRT-0901, and in part by Tampere University, Finland.

ABSTRACT This article proposes a novel approach aimed at estimating the pose of a camera, affixed to a robotic manipulator, against a target object. Our approach provides a way to exploit the redundancy of the robotic arm kinematics by directly considering manipulator poses in the model formulation for camera pose estimation. We adopt a single camera multi-shot technique that minimizes the reprojection error over all the rigid poses. The results of the proposed method are compared to four other studies employing either monocular or stereo setup. The experimental results on synthetic and real data show that the proposed monocular approach achieves better and in some cases comparable results to the stereo approach. Moreover, the proposed approach is significantly more robust and precise compared to other methods.

INDEX TERMS Pose estimation, multi-view, multi-shot, machine vision, robotic arm, visual servoing.

I. INTRODUCTION

Camera pose estimation with respect to a target object/scene has been widely researched in the fields of computer and machine vision, photogrammetry and robotics. Accurate pose estimation is needed in numerous applications such as camera calibration [1], localization [2], reconstruction [3], robot visual servoing [4], and augmented reality (AR) [5]. The advances in these fields have significantly benefited users to accomplish a variety of tasks with good accuracy. Despite much progress, there is still need of improvement for application specific methods to improve accuracy and robustness. For example, an approach suited for achieving visually pleasing reconstruction might not be well suited for accurate localization.

In this study, we focus on the prerequisites of visual servoing of a robotic arm for accurate manipulation. Visual servoing uses visual information acquired from cameras to get spatial and semantic understanding of the surrounding to plan the motion of the robot. The most common applications are robotic grasping [6] and medical procedures [7]. Visual servoing depends on many independent components such as accuracy of robot positioning, hand-eye calibration, and target pose estimation. For this study, we restrict our scope to the accuracy of target pose estimation. Pose estimation of the camera against a target position/object can be achieved through various approaches that incorporate different algorithms and/or hardware configurations. Among these, monocular approaches are widely adopted for AR applications [8]. This primarily means that 6-DoF pose is obtained using a single monocular image. The depth of the object with respect to the camera can be estimated from a scaling approach by forming a geometric relationship between the camera and the known metric size of an object in view of the camera.

The generic approaches for pose estimation of a single camera with respect to the object, or vice versa, can be categorized into two groups. The first category of methods finds the solution by estimating the plane-to-view homography and then decomposing it to obtain the pose. This set of methods is known as Homography Decomposition (HD) methods [9]–[11]. Collins and Bartoli [12] proposed a method that analytically solves the problem after the homography is computed. They named their method Infinitesimal Plane-based Pose Estimation (IPPE). The underlying concept is that even when the estimated homography is noisy, it will still be close to the true transform between the image and the model plane at some regions on the plane. The method takes the points on those regions to solve for a pose using 1st order PDE. The second category of methods treats it as a rigid pose estimation problem. It uses 2D-3D point correspondence

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro Neto.

for estimating the pose of the camera relative to the object. This approach is commonly known as Perspective from n points (PnP) [13]. PnP methods work by minimizing the cost function of the correspondence transfer error to estimate a rigid pose. The correspondence transfer is the error between the predicted positions of point correspondences compared with their measured positions. Collins and Bartoli [12], also makes the argument that IPPE has a deep connection with PnP problem, where the n points can be centered at infinitely small separation from each other using the estimated homography. Lu *et al.* [14], proposed a provably convergent method called RPP that iteratively solves the PnP problem. The method minimizes the collinearity error to estimate the rotation part of the pose followed by its associated translation. The method is quite efficient and usually converges in 5-10 iterations from a random geometric configuration. Schweighofer and Pinz [15], extended the work presented in [14] and introduced RPP-SP to handle ambiguous cases that results in the case of planer targets. The method first computes the pose solution in a similar way to [14] and then estimates a second pose solution by minimizing the reprojection error along 1-DoF rotation and translation at a time. The aim is to find the second local minimum if such a minimum exists. The limitations of [15] are that if the first solution is poor, then the second solution suffers as well. Moreover, it is very difficult to physically characterize the ambiguous cases since the second solution is obtained from the roots of a 4th order polynomial, where two of the roots are imaginary. Li *et al.* [16], proposed a non-iterative method that solves the PnP problem numerically in O(n) by producing subsets of three points. Each subset is then solved as a separate P3P problem. The final solution is obtained from the group of solutions that best fits the model.

Alternatively, many studies consider multi-view approaches to achieve better accuracy. In a multi-view approach, the feature points or parts of interest are observed through several views to generate a coherent and accurate model. These features can be linked across views through robust tracking and subsequently aligned through relative geometric transformation. Federico *et al.* presented a closed-form method to estimate the pose of an object from multiple views [17]. The method requires at least one point-point and two point-ray correspondences from two or more views to solve a generalized PnP problem. With the ability to efficiently and accurately match feature point across multiple views, many studies have opted for structure-from-motion (SFM) based approaches, also known as full multi-view. Daniel and Tomas proposed an SFM method that computes the rotation and translation separately for relative views [18]. The approach then optimizes the relative poses globally and evenly distributes the pose errors using bundle adjustment. Typically, approaches that opt for separate estimation of rotation and translation yield good orientation accuracy. However, the position accuracy is often compromised as the errors from rotations estimation step propagate to the translation estimation step. Nonetheless, in the case of study [18], these are compensated for in the bundle adjustment step. Collet and Srinivasa [19] introduced a modified version of full multi-view, which they termed as introspective multi-view approach. This multistep approach first estimates object and camera pose using a single-view method. Once the initial estimates are obtained, the points are clustered and the outliers from matches are removed. Finally, the poses are re-optimized in a bundle adjustment step using the filtered matches. According to the authors, the approach provides a good tradeoff between computational speed and accuracy. This study is important for our comparative analysis since it demonstrates its use for robot grasping application.

Some studies utilize multi-camera approaches to solve the pose estimation problem. Theoretically, multi-camera systems are similar to multi-view approaches for specific cases where time is not a relevant factor. Furthermore, Stereoscopic approach is a specific case of multi-camera approaches where two cameras are separated by a fixed baseline. In such a case, there must exist a considerable overlap between the views. Stereo approaches can obtain highly accurate results due to the inherent advantage of constrained two or more views. The depth estimated from stereo can be considerably more accurate compared to traditional monocular approaches. Clipp *et al.* [20], proposed an approach that estimates the pose in two steps. First, the absolute rotation and up to scale translation are estimated using a 5-point algorithm [21] in one of the cameras. The correction factor for the scale is then computed separately from an additional point correspondence in the second camera. However, the scale retrieval approach is not robust and absolute translation cannot be obtained all the time. Later Clipp *et al.* presented a modified approach that estimates the relative pose of a stereo pair by employing constraints on the feature point selection for pose estimation [22]. The pose is estimated using a selection of four feature points, where the first point is observed in all four views (both stereo-pairs). Two more points should be observed in two-views of one of the cameras (left or right), while the last point is observed in both views from the other camera. The results show improvement over a random selection of points; however, the study lacks comprehensive testing over real data. Geiger *et al.*, proposed a novel approach that generates dense 3D maps from high-resolution stereo sequences in real-time [23]. The authors claim that the presented approach achieves state-of-the-art accuracy in terms of pose estimation and its sub-sequent odometry. The method estimates pose by reprojecting the world points simultaneously on the stereo views and thereby constraining the objective function. The objective function is iteratively optimized using the Gauss-Newton method. Igor *et al.* presented a stereo approach for ego-motion estimation called SOFT [24]. The approach focuses on a careful selection of features and robust tracking for improving the overall accuracy. The author estimates the rotation with the 5- point algorithm [21] and translation with a 1-point stereo method that is iteratively optimized in both views. Raul and Juan presented a similar approach to SOFT for ego-motion estimation with slightly loose constraint on feature

selection [25]. The approach first computes the relative camera pose followed by a local bundle adjustment among a few recent poses. Later, a full bundle adjustment is performed to optimize the camera locations by minimizing the reprojection errors in all the observed views. The approaches in [24] and [25] are more suited for a large amount of data where the tradeoff is maintained between local accuracy and error distribution among all the views.

Though multi-camera approaches provide considerable advantages over monocular approaches, in many cases the additional hardware and software resources required can exceed the allocated resource budget of the task. This study is driven by the motivation to develop an accurate and robust pose estimation method for the International Thermonuclear Experimental Reactor (ITER) project using an eye-in-hand monocular approach. The goal is to perform certain tasks autonomously using a robotic arm with high precision and accuracy. In this work, we attempt to achieve comparable results to stereo approaches by proposing a multi-view monocular approach considering the case of robotic arm manipulation.

The article is organized as follows: In Section 2, we present the problem and define the preliminaries for its formulation. In Section 3, we formulate the proposed method along with other methods considered for comparative analysis. Section 4 presents the experimental setup, error metrics and experimental results using both synthetic and real datasets. Finally, Section 5 concludes the article.

II. PROBLEM FORMULATION

In this study, we attempt to elucidate the approach through geometrical relationships for thorough understanding. We adopt various notations to help us describe the problem and use them consistently throughout the study. We represent the homogeneous transformation matrix by the standard notation T and support it through various sub-indices. The sub-indices b, t, c, and w correspond to the robot base, robot tool/tool center point (TCP), camera optical center and world coordinate frames, respectively. These notations are exemplified in Fig. 1.

The TCP/end-effector pose from the base of the robot, denoted by ${}_bT^t$, is provided readily by the control system associated with the robot. Generally, the robot pose is highly accurate due to the high precision encoders used in the robotic arm at each joint. These robots, especially industrial robotic arms, are designed to perform tasks that require accuracy and high repeatability with precision around $0.1 - 0.2$ mm of the end effector's position. The transformation from the robot TCP to the camera coordinate frame ${}_tT^c$ is known as hand-eye transformation. We have discussed in detail various approaches for hand-eye and robot-world-hand-eye calibration methods in an earlier study [26]. The aforementioned article can be studied for a thorough understanding of the methods and their MATLAB implementation. For this study, we will adopt the reprojection based approach of the robot-world-hand-eye formulation to estimate ${}_tT^c$.

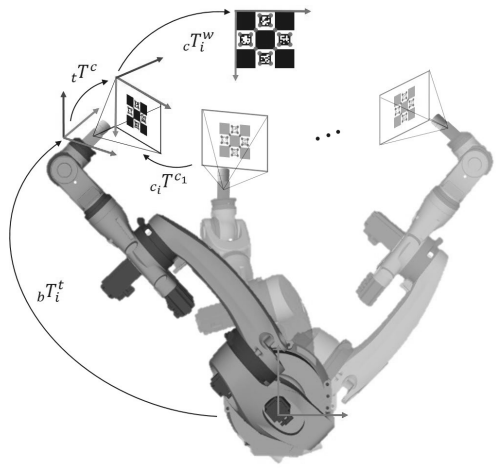


FIGURE 1. Illustration of the setup explaining the geometrical relation among various coordinate frames.

Finally, the unknown homogeneous transformation ${}_cT^w$ from camera coordinate frame to the world coordinate frame affixed to the target object needs to be estimated. The estimation of this transformation matrix defines the pose estimation problem in the described arrangement. Generally, the pose of the camera against the target object or vice versa can be computed independently of the robotic arm pose. For the case of monocular camera with one image (monocular single shot - MSS), only one camera pose ${}_cT_1^w$ exists at $i = 1$ pose. This can be estimated directly with the state-of-the-art methods mentioned in the Introduction Section. Similarly, in the case of stereo camera with one stereo image (stereo single shot - SSS), we estimate only the camera pose ${}_cT_1^w$ which should be able to validate the constrained views of the stereo image pair. The constraint between the stereo pair is a fixed transformation obtained during the calibration of the stereo camera.

In contrast, we can find the reference ${}_cT_1^w$ using multiple images from different views (monocular multi shot - MMS). For $i = (1, 2, \ldots, n)$, we estimate n ${}_cT_i^w$ transformations from camera frame, at each pose, to world frame. In the arrangement shown in Fig. 1, the camera positions at various poses of the end-effector of the robot can be considered independent camera bodies floating in space. The position and orientation of the camera at these n camera poses can be estimated and optimized freely, independent of the robot poses. The optimization is typically performed as local and/or full bundle adjustment [25], where the goal is to distribute the errors and obtain the best possible 3D reconstruction of the object/scene. Theoretically, the transformation ${}_cT_1^w$ estimated through MMS approach is more accurate as it is constrained with the help of the remaining n-1 ${}_cT_i^w$ transformations.

In this study, we propose a modified form of MMS approach where we constrain the free pose optimization of

cameras in an attempt to model the physical system more adequately. We estimate only the transformation ${}_cT_1^w$ and use the prior information (robot poses ${}_bT_i^t$ and Hand-Eye transformation ${}_tT^c$) to constrain and geometrically relate the camera views from n poses. Unlike the traditional MMS approaches, the proposed approach does not need to estimate the additional $n-1$ ${}_cT_i^w$ transformations.

III. METHODS

In this section we discuss the stereo single-shot approach presented in [23], the monocular multi-shot approach presented in [19] and our proposed monocular multi-shot approach. All these methods solve the problem of camera pose estimation in the image space. Such a method takes the world points and estimates a suitable transformation that enables us to reproject the 3D points to the image space at their corresponding views.

The stereo single-shot approach presented by Geiger *et al.* [23] was briefly discussed in Section I. We present here the mathematical relation that we use to estimate the stereo pose. The relationship is given as

$$\begin{aligned}&\{q_{(c,w)},\ {}_ct^w\}\\&\quad=\underset{q_{(c,w)},\,{}_ct^w}{\operatorname{argmin}}\,(||P^l-\Pi(K^l,[q_{(c,w)},\ {}_ct^w]_{HT},W)||_2^2\\&\qquad+||P^r-\Pi(K^r,[q_{(c,w)},\ {}_ct^w]_{HT}*{}_lT^r,W)||_2^2).\end{aligned}\tag{1}$$

Here, Π is the perspective projection function that projects the 3D points $W=(X,Y,Z,1)^T$ from world frame space to image space using the camera intrinsic (K^l and K^r) and the stereo extrinsic ${}_lT^r$. The superscript T indicates the transpose of a vector. The subscript or superscript l and r indicate the camera to which the corresponding parameters relate in the stereo pair. The cameras intrinsic and extrinsic are estimated using Zhang's stereo camera calibration approach [11]. The perspective projection yields $\tilde{x}=(\tilde{u},\tilde{v},1)^T$ in the image space of the camera at the pose of interest. The reprojected points $\tilde{x}$ are compared directly against the observed/tracked 2D points (P^l and P^r) in the corresponding left and right image pair. The symbol $[\ ]_{HT}$ indicates the conversion from quaternion $q_{(c,w)}$ and translation vector ${}_ct^w$ to the homogeneous transformation matrix ${}_cT^w$. The solver minimizes the error function in quaternion representation of angles. This helps to reduce the number of unknowns from 12 to 7 parameters. We use the Levenberg–Marquardt algorithm to search for a minimum in the search space by minimizing the L2-norm ($\|\ \|_2^2$) of the residual scalar values.

The second method is a MMS approach known as Introspective Multiview Approach [19]. The method first extracts feature points from the scene and estimates a camera pose for each view using a single-view method. The points are then clustered, filtered, and matched across the views. Finally, the individual poses are re-optimized in a bundle adjustment step using the filtered matches/clusters. The study presents two mathematical relationships for solving the problem; one is based on reprojection and the other is based on back-projection. It then argues that both relations are equivalent in Euclidean space and one may use either of the approaches. The reprojection based approach is generally preferred since it is invariant to projective transformations, while the back-projection does not provide useful information in projective space [19]. The authors opted for back-projection based approach to extend their implementation to be used with other sensors e.g. LIDAR data. However, we use the relationship provided for reprojection based approach as it concurs to the approach we have adopted throughout this study. In the original implementation, feature points from multiple objects in the scene were extracted and multiple hypotheses are generated; one for each cluster of points tracked across views. Since we are using one target pattern, the formulation can be simplified to a single hypothesis optimization problem. In line with the argument in Problem Formulation Section, we estimate and re-optimize n ${}_cT_i^w$ transformations in this MMS approach. The mathematical relationship is given as

$$\{h^*\}=\underset{{}_cT_i^w}{\operatorname{argmin}}\sum_{i=1}^{n}\delta_j^i||P_1^i-\Pi(K,\ {}_cT_i^w,W)||_2^2.\tag{2}$$

where $h^*=\{h_1^*,h_2^*,..h_n^*\}$ indicates the set of optimal hypotheses and δ_j^i is a logical operator that switches to 1 when P_1^i has points in a cluster and 0 otherwise. We have fixed the lower subscript to 1 since we assume one cluster of points i.e. the target pattern.

In the proposed method, we use data from n poses and explicitly take the robot poses into consideration. The primary difference between our proposed approach and [19] is that we recommend introducing the robotic arm transformations in the optimization step to constrain the model and minimize the number of unknow transformations from n ${}_cT_i^w$ to ${}_cT_1^w$.

From Fig. 1, we can form the following relationship among n manipulator poses.

$$\begin{aligned}{}_bT_1^t\ {}_tT^c\ {}_cT_1^w&={}_bT_2^t\ {}_tT^c\ {}_cT_2^w\\&={}_bT_3^t\ {}_tT^c\ {}_cT_3^w\\&\vdots\\&={}_bT_n^t\ {}_tT^c\ {}_cT_n^w.\end{aligned}\tag{3}$$

During the estimation step, we optimize only for one homogeneous transformation ${}_cT^w$ that transforms a point from camera frame position in the first/reference view to the fixed object/world coordinate frame. Hence, we curtail the geometric relationship in (3) and accumulate the transformations from world frame to the camera frames at all poses except the reference pose. The resultant transformation $\overline{T}_i$ transforms the 3D world points from object/world coordinate frame, through the first reference pose, to the camera frames at the remaining $n-1$ poses.

$$\overline{T}_i={}_cT^t\ {}_tT_i^b\ {}_bT_1^t\ {}_tT^c\ {}_cT^w.\tag{4}$$

Since we use quaternion and translation vector representation during optimization, we re-write (4) as

$$\overline{T}_i = {}_tT^{c^{-1}}\ {}_tT_i^b\ \ {}_tT_1^{b^{-1}}\ {}_tT^c\ \ [q_{(c,w)}\ ,\ {}_ct^w]_{HT}. \tag{5}$$

We can now estimate ${}_cT^w$ by optimizing the following expression

$$\{q_{(c,w)}\ ,\ {}_ct^w\} = \underset{q_{(c,w)}\ ,\ {}_ct^w}{\operatorname{argmin}} \sum_{i=1}^{n} ||P_i - \Pi(K, \overline{T}_i, W)||_2^2. \tag{6}$$

Many studies suggest optimizing the camera intrinsic parameters along the solution estimation to achieve better results [27]. On the other hand, Koide and Menegatti [28] contend this argument as an overfitting problem. The rationale that [28] provides is that upon optimizing the intrinsic parameters for the reprojection error, the model overfits to the given samples. This will yield poor results for all error metrics other than the reprojection error and carry the estimate away from the true solution. We observed a similar response while optimizing for ${}_tT^c$. As mentioned before, this transformation is obtained from the robot-world-hand-eye calibration. The calibration is performed on a significantly higher number of poses (10-20) compared to the number of poses used for object pose estimation (3-5). Due to fewer poses, the result deteriorates and the errors propagate to the final solution. Based on the presented argument and experimental results, we opted for a single camera intrinsic and robot-world-hand-eye calibration.

It is noteworthy that the proposed approach is not invariant to the choice of initial estimates for the solver. However, we have successfully constrained the number of unknown parameters to just 7, which improves the convergence of the solution, even with a rough initial estimate. The initial estimates for [19] and our proposed method are obtained from MATLAB's implementation of Zhang's method [11] for camera calibration and monocular pose estimation.

IV. EXPERIMENTAL SETUP AND RESULTS

To assess the performance of our proposed method against other studies, we carry out tests on simulated data with synthetic images and real data. The motivation for using simulated data is to check the actual response of the method against actual ground truth. In contrast, the real data is used to assess the performance of the methods in a real working environment where perturbations in the data are higher and the ground truth is always an approximation.

When estimating the object pose by relaying information through the image space, the selection of feature points for tracking plays a significant role in the overall accuracy of the system. Many studies prefer a markerless approach similar to SFM approaches [18], [29]–[31] to make the system independent of special fiducial patterns. These approaches use feature point correspondences from the feature-rich scene and track them through the views. This is immensely useful for the case where the environment is unregulated and the use of markers is difficult. However, the drawback of such an approach is that the feature correspondence step is prone to outliers. Even in the presence of powerful consensus generator algorithms such as RANSAC [32], the approach inherits additional errors in the form of weak feature correspondence due to tracking or matching. As a result, the accuracy is always an approximation of what it can be in the presence of specialized markers. The study aims to develop an accurate pose estimation method for robotic arm manipulation, where the environment is moderately regulated. It is to our advantage to use specialized markers.

For this study, we use classical checkerboard and ChArUco diamond marker [33], for simplicity referred hereafter to as *diamond marker*. The diamond marker consists of 3×3 squares with 4 ArUco markers placed inside the white squares. This pattern and its detection approach are more robust compared to the use of only markers and compact compared to the original ChArUco pattern.

In this section, we provide a quantitative analysis of the proposed method against four other state-of-the-art methods. Moreover, we discuss the error metrics used to assess the performance of each method. Among the methods used for comparison, IPPE [12] and Zhang [11] are based on monocular single shot (MSS) approaches. Collet and Srinivasa [19] is a monocular multi-shot/multi-view (MMS) approach, while Geiger *et al.* [23] requires a single shot from stereo (SSS) camera pair. We have not considered a comparison with a stereo multi-shot approach as it is redundant for this task and is in conflict with the aim of this study i.e. improving results with reduced hardware. Finally, we discuss and report the experimental results of all these methods for each test case.

A. ERROR METRICS

To estimate the error in the computed pose against the ground truth poses, we use absolute rotation error (deg), absolute translation error (mm), and reprojection error (px). The absolute errors require that the ground truth poses are known. In the case of synthetic data, the ground truth poses are exactly known. In the case of real data, the approximates of ground truth are found through dedicated manual steps as explained in later sections. The absolute rotation error e_{aR} and the absolute translation error e_{at} are given as follows

$$e_{aR} = ||\ angle\ ({}_bR^{w^{-1}}\ {}_bR_{gt}^w)\ ||_2^2, \tag{7}$$

$$e_{at} = ||\ {}_bt_{gt}^w\ -\ {}_bt^w\ ||_2^2. \tag{8}$$

Here the angle() represents the conversion from a rotation matrix to axis-angle for simpler user interpretation. ${}_bR^w$ and ${}_bt^w$ are the rotation and translation from the base of the robot to the world frame.

The final metric that we use, is the reprojection root mean squared error (rrmse). This metric is measured in pixels and is a good way to assess the quality of the results in image space. Moreover, it consolidates the absolute error metrics since the proposed model of reprojection error back-projects the 3D points onto the images by first transforming them through the robotic arm. Hence, each pose has to be accurate and in

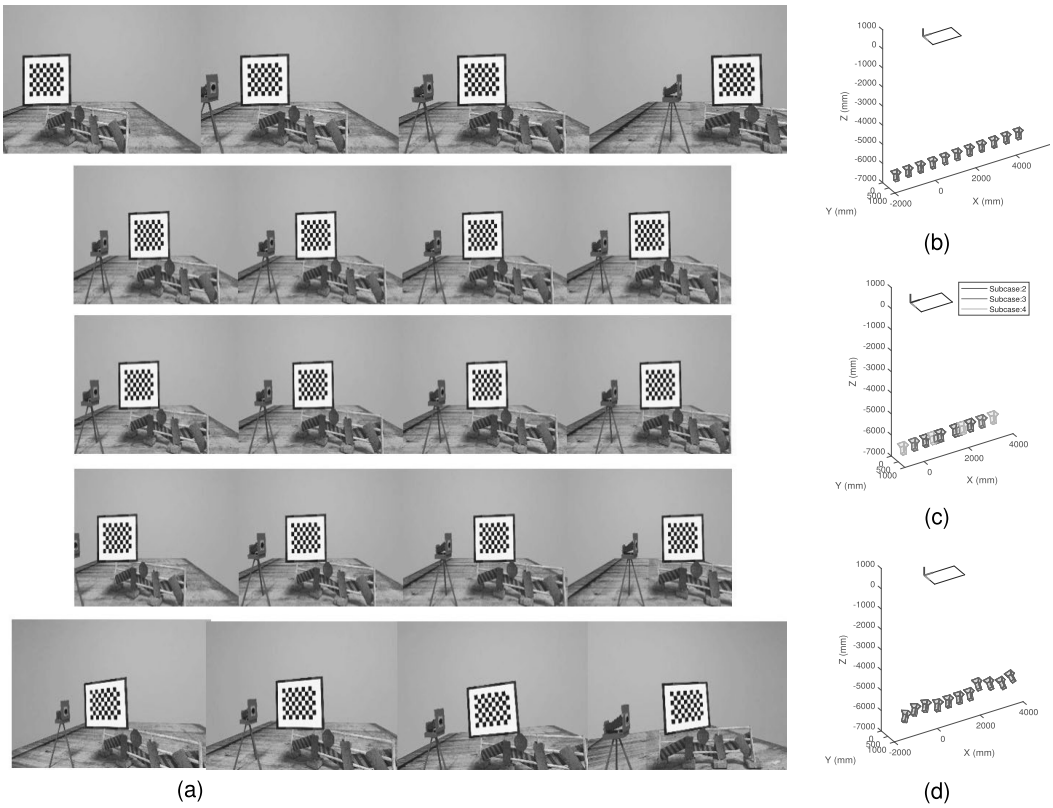


FIGURE 2. Example of the rendered images from the simulated cases and the 6DoF poses of the camera. (a) The first row of images exemplifies case 1 with varying number of poses with motion along one axis. Second to the fourth row include extracts from case 2 (subcase 2-4) with varying inter-pose distance. The last row of images exhibits case 3 where all axes are excited during camera motion and rotation (b-d) Camera poses against the target pattern for each synthetic case shown in (a).

agreement to the overall geometric relationship for the 3D points to back-project precisely onto these different views from corresponding robot poses. The transformation is the same as used in (6) and shown in (4). The reprojection error is given as

$$e_{rrmse} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\overline{P}_i - \Pi(K, \overline{T}_i, W)||_2^2}. \tag{9}$$

B. TESTS ON SYNTHETIC DATA

A significant advantage of using simulated data is the availability of exact ground truth information. In the case of simulated data, we have the ground truth robot poses hand-to-eye transformation and the camera to world object transformation. In real cases, the ground truth hand to eye transformation is not available as it is not feasible to estimate the exact location of the optical frame in a physical setup. Moreover, any ground truth robot pose and the camera pose is only the best possible approximation of the actual information. For the simulated case, we generated high-resolution synthetic images instead of simulated points, as shown in Fig. 2a. These synthetic images were generated using Blender, a 3D computer graphics software. It should be noted that the tripod in the images is part of the scene and is not to be confused with the virtual camera that is capturing the scene. To simplify the experimentation, we assume that the camera and the robot TCP position is the same for the simulated test cases. This means the virtual position of the camera is the position of the robot TCP. Then the homogeneous transformation from hand-to-eye constitutes of just rotation. This rotation is the result of the transformation between the Blender world frame and Blender camera frame.

To study the effect of various parameters, we set up three test cases for the simulated data based experimentation. The excerpts from these cases are shown in Fig. 2. The results from these simulated data aids in selecting suitable parameters to use for real data acquisition and testing. Moreover, we induce visual noise to the points detected for pose estimation. The noise was introduced to study how well the methods can converge towards an accurate solution in the presence of uncertainties. The generated noise has a Gaussian distribution

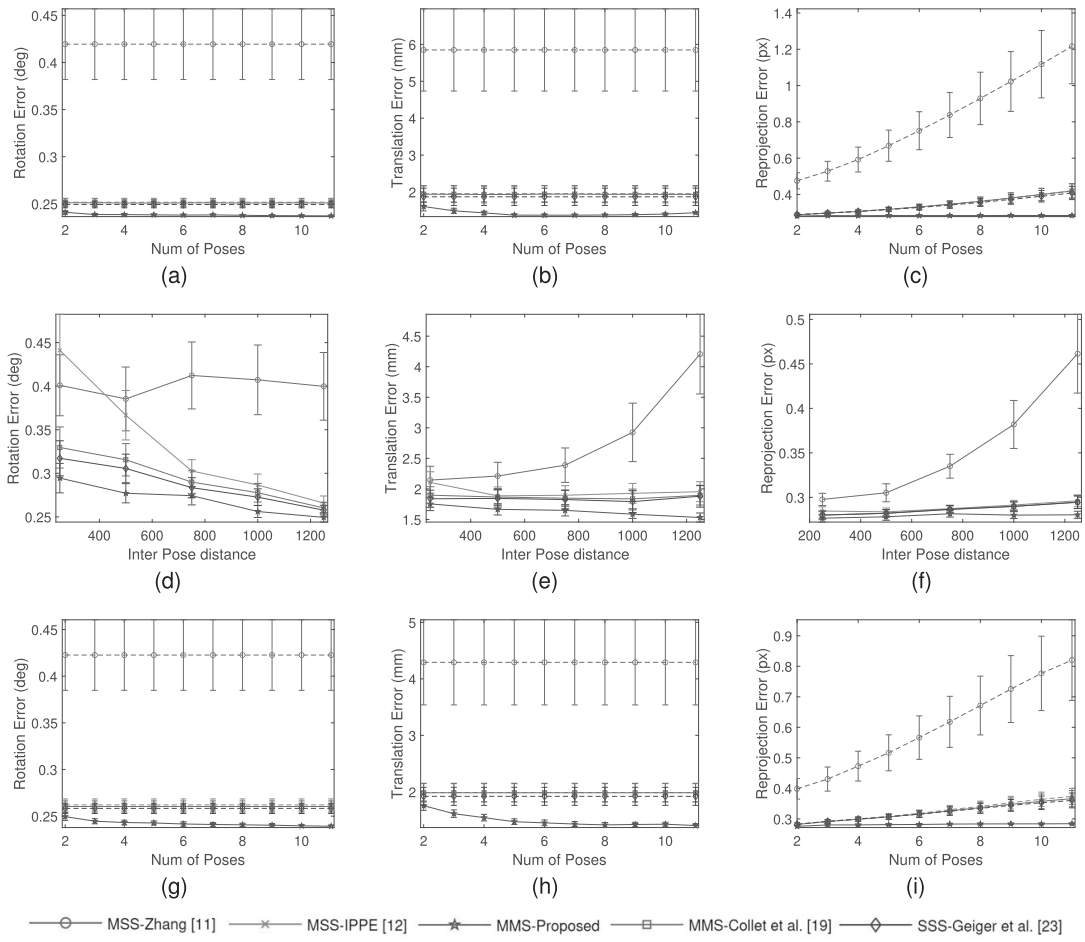


FIGURE 3. Metric error results for the synthetic data with added Gaussian noise to images (a-c) Results for case 1 of the synthetic data where the camera moves along one axis (d-f) Results for case 2 of the synthetic data with varying inter-pose distance and one axis motion (g-i) Results for case 3 of the synthetic data where all axes are excited during camera motion and rotation. The dashed line exhibit extrapolation from the first pose estimate for the case of monocular methods.

with a mean of 0.5 and standard deviation of 0.5. To avoid a biased result due to the addition of noise to synthetic data, we repeat the experiment for 50 iterations. In each iteration, we introduce the same level of noise, randomly generated, with a Gaussian distribution. Finally, the mean performance over these 50 iterations is considered a stable response of the corresponding method over the given data.

In the first case, we study the effect of varying the number of poses while moving the camera only in one axis. We choose the horizontal axis. Few images from this case are shown in the first row of Fig. 2a, where the camera moves in one axis only. The camera pose distribution against the calibration pattern for case 1 can be observed from Fig. 2b. We analyze the response of the methods when we increase the number of poses from where the object is viewed. The response of the methods can be observed in Fig. 3a, 3b and 3c. The rotation and translation error show slight improvement especially in the case of the proposed method. It is noteworthy, that the confidence interval of the proposed method is the smallest, which correlates to good precision over varying noise. On the contrary, Zhang [11] show significant deviation from its mean results. The rising trend in the reprojection error can be explained by the fact that the single shot (MSS and SSS) estimates the pose from one image only. The estimate might be accurate for that specific viewpoint, however, its global accuracy is poor as we attempt to use that geometrical information to transform and reproject the points onto other poses. Moreover, Collets and Srinivasa [19] shows a similar increase in the reprojection error, even though, it is based on a multi-shot based approach similar to the proposed method. IPPE [12] shows significantly better results despite being a single-shot approach. The proposed method begins

to reach the minimum error using 5 unique poses for image acquisition. It is noteworthy, that the rotation and translation errors are constant over the increasing number of poses for the single-shot methods. This is because they are extrapolated from the first/reference view for which the camera pose is estimated against the target pattern. The extrapolated dashed line is intended to assist readers in visually comparing the single-shot methods with multi-shot methods along the increasing number of poses. In contrast, the reprojection error for the single-shot methods is not constant over the varying number of poses. This is because the mean reprojection error is estimated over all the views for all methods by using the estimated first/reference pose and the ground truth poses of the relative views. This is done to ensure that the estimated pose from a method is not the result of a local minimum rather the solution is globally consistent and accurate.

In the second case, we attempt to examine the effect of interpose distance on object pose estimation. We fix the number of images for estimating the pose so that the only varying parameter is the interpose distance. The second case has further six sub-sequences where each sequence has four images. In each of the sub-cases, the interpose distance is varied. In Fig. 2a, we show three sub-sequences (of case 2) in row 2, 3 and 4, where the interpose distance is 400, 600 and 800mm, respectively. We visualized the camera pose distribution against the calibration pattern for the aforementioned sub-cases of case 2 in Fig. 2c. The response of the methods on the data from case 2 can be observed in Fig. 3d, 3e and 3f. A noticeable change can be observed between the results of case 1 and case 2. The responses on the case 2 are more sharply varying especially for the single-shot approaches. This is because the camera pose for the reference image (first image) changes as we increase the pose distance from the middle of the scene. In case 1, we started from one side of the scene and moved the camera along the horizontal axis by adding more frames. As a result, the reference frame always remained the same. In contrast, the reference pose/view point in case 2 changes as we move further from the center of the scene. Pose estimates of the same object from different view points may incorporate different levels of uncertainties. As a consequence, we observe in Fig. 3 (d-f) that the estimates between two consecutive data points exhibit a sharp change in response as we vary the interpose distance. This effect strongly points toward the data dependency of many single-camera methods. This data dependency results in the form of imprecise solutions. Here, the primary factor causing this dependency is variation in the poses chosen for calibration, however, such an effect may also be observed due to the model of the robot, and how the robot is mounted, which may introduce new errors. Nonetheless, the proposed method yields the best result over varying interpose distance followed by Geiger *et al.* stereo based approach [23]. The overall trend shows that increasing the interpose distance improves the accuracy of the estimate, with the exception for Zhang's response [11]. Moreover, MMS and SSS approaches exhibit more stable response compares to MSS approaches.

The final test case of the simulated data focuses on studying the impact of position and orientation change in more than one axis. The dominant motion is the same as in case 1. However, small position and orientation changes are also introduced in other axes as well. Few images from this case are shown in the last row of Fig. 2a and the camera poses are given in Fig. 2d. An apparent change between case 1 and case 3 can be observed in the images and the camera poses in the form of change in yaw angle. All other movements are minute and cannot be observed from the images. The response of the methods on the data from case 3 can be observed from Fig. 3g, 3h and 3i. The results follow the trend of case 1, where IPPE [12], Collet and Srinivasa [19], and Geiger *et al.* [23] show almost similar responses with Geiger *et al.* [23] method yielding the lowest errors among them. Zhang [11] shows the largest error while the proposed method yields the best results on all the error metrics. It is noteworthy that the mean errors for case 3 are marginally lower than the errors in case 1. This exemplifies that it is important to excite motion and rotation around all axes to yield better results.

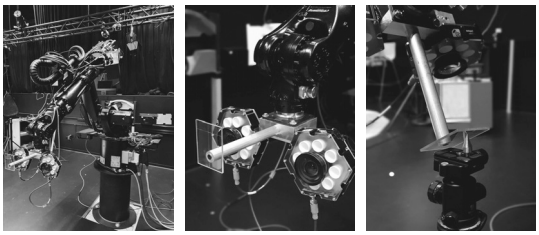


FIGURE 4. An example of the experimental setup (a) KUKA KR16L6-2 robotic manipulator used for recording data (b) Close up of the adaptor with the tool, stereo camera pair and lights affixed to the manipulator using customized hardware (c) A snapshot from the tool 4 point XYZ-calibration step.

C. TESTS ON REAL DATA

We further study the performance of the methods using real data. The real data is acquired using industrial-grade equipment for high accuracy. The experimental setup is shown in Fig. 4. A custom adaptor was designed to fix two Basler acA1920-50gc cameras to KUKA KR16L6 -2 serial 6-DoF robot arm, as shown in Fig. 4b. We use 6mm lens with each camera. The stereo pair has a baseline of 14 cm. Moreover, we use dedicated lamps to uniformly light the target. The use of these lamps is not mandatory; however, they are convenient in maintaining a uniform brightness irrespective of the room lighting condition. The adaptor not only houses the cameras but also holds a custom tool. The tool is an aluminum bar with a Polycarbonate sheet at the end. A cross-hair marker is drawn on this sheet. The purpose of this tool is to manually measure the position and orientation of the target object as accurately as possible. The intersection of the cross-hair marker helps to pinpoint the position while the planer surface of tool sheet aids in measuring the orientation of the planer target. Since the study focuses on accurately estimating the

TABLE 1. Comparative results using checker board as target object.

Methods	Abs. Rotation Error, $\mu_R(deg)$	Abs. Translation Error, $\mu_t(mm)$	Abs. Reprojection Error, $\mu_{re}\ (px)$	Rotation std. dev., $\sigma_R\ (deg)$	Translation std. dev., $\sigma_t\ (mm)$	Reprojection std. dev., $\sigma_{re}\ (px)$
MSS-Zhang [11]	1.9546	3.4217	1.3194	0.036274	0.19947	0.21217
MSS-IPPE [12]	1.9586	3.6378	1.3345	0.047422	0.2165	0.22098
SSS-Geiger et al. [23]	1.9325	**2.931**	1.1621	**0.027213**	0.22121	0.26702
MMS-Collet et al. [19]	1.9498	3.4847	1.3306	0.03223	0.20967	0.2204
MMS-Proposed	**1.6796**	3.1285	**1.0733**	0.045619	**0.15001**	**0.12447**

pose of the camera against the target without target handling; the tool effectively fulfills the purpose. The tool is calibrated for the robotic arm using KUKA's XYZ 4-point method for position and ABC 2-point method for orientation calibration as illustrated in Fig. 4c.

We utilize the tool for both initial ground truth measurement and evaluation of the estimated pose. The ground truth is measured by manually aligning the tool marker on the target object and recording the robot pose information. The estimated poses from the experiments are then compared to this recorded pose.

However, we observed a marginal instability at the base of the robotic arm used for experimentation. The base instability is observed near the maximum reach of the arm inside the workspace. As a result, the manual measurement of the ground truth has small uncertainties (around 2mm). Therefore, we refer to it as the desired pose instead of the ground truth in this work. Nonetheless, the evaluation of our estimates against the desired pose provides us with invaluable results for relative comparison of the methods under consideration in this study.

FIGURE 5. Example of the captured images using checkerboard and diamond marker as target objects.

We perform two sets of experiments using different patterns as the target objects, shown in Fig. 5. The first set of experiments uses a checkerboard of size 18×25, while the other uses a diamond marker. Based on our earlier results from synthetic data, we can infer that 5 poses are sufficient for a multishot approach to converge to a stable solution. The distance of the target object from the tool is kept around 0.76 meters since the reach of the robot with the custom tool is approximately 2.15 meters. It is not possible to manually measure the desired pose on the target object for evaluation after 2.15 meters with the current setup. Each experiment is repeated multiple times from randomly initialed pose with varying additional poses. At each pose, the stereo pair takes images of the target object. We can easily see the distribution of the poses in 3D space in relation to the calibration pattern for the real test cases from Fig. 6.

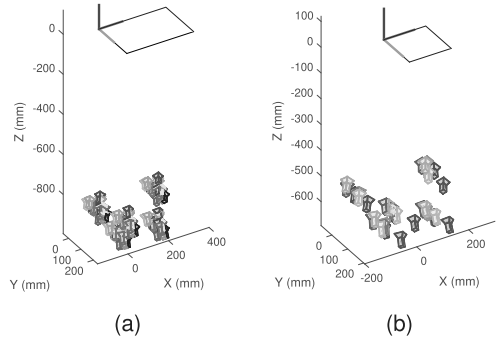


FIGURE 6. Camera poses against target pattern for multiple set of experiments where each color represent a unique set of experiment (a) Poses for the checkerboard (b)Poses for the diamond marker.

The experimental results for the tests using checkerboard as the target pattern are shown in Table 1 and Fig. 7. We compute the arithmetic mean (μ) and standard deviation (σ) of the corresponding errors from all the test iterations. The tabulated results show that the proposed method yields the least absolute rotation error (μ_R) and absolute reprojection error (μ_{re}). The least absolute translation error (μ_t) is obtained by the stereo approach in [23], however, the proposed approach yields a comparative result with the second-best translation estimate. The results obtained for μ_R, μ_t,and μ_{re} using [11], [12] and [19] are quite similar for the given set of experiments.

In addition to accuracy, the system must be consistent in realizing its accuracy over varying data samples. If a method achieves good results only half of the time then the system is not robust and requires improvement in precision. We also tabulate the standard deviation of the estimates for quantitative analysis of the robustness of the methods. It can be observed from Table 1, that the proposed method yields significantly lower deviations over translation (σ_t) and reprojection (σ_{re}) estimates. The least standard deviation for rotation (σ_R) is achieved by SSS-Gieger *et al.* [23]. A comparable result is obtained by the proposed method for σ_R

TABLE 2. Comparative results using diamond marker as target object.

Methods	Abs. Rotation Error, $\mu_R(deg)$	Abs. Translation Error, $\mu_t(mm)$	Abs. Reprojection Error, $\mu_{re}\ (px)$	Rotation std. dev., $\sigma_R\ (deg)$	Translation std. dev., $\sigma_t\ (mm)$	Reprojection std. dev., $\sigma_{re}\ (px)$
MSS-Zhang [11]	2.508	4.0691	2.3638	0.92636	1.0597	0.20078
MSS-IPPE [12]	2.3373	4.1017	2.2399	0.52881	0.47025	0.18418
SSS-Geiger et al. [23]	2.3903	4.2761	2.1124	0.20142	0.30155	**0.13553**
MMS-Collet et al. [19]	2.3095	4.0959	2.2096	0.25623	0.50482	0.17876
MMS-Proposed	**2.1837**	**4.0628**	**2.105**	**0.14443**	**0.21076**	0.15846

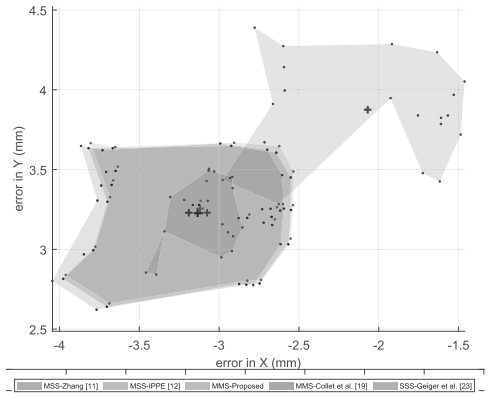


FIGURE 7. A 2D precision plot for the set of experiments using a checkerboard target. Each data point is the error of the estimate against the desired pose without the measure from the axis containing the depth information. The corresponding information from each method is shown in a unique color. The mean of each set of data points is presented by the + symbol and the region encompassing the scattered estimates shows the spread of estimates from respective methods.

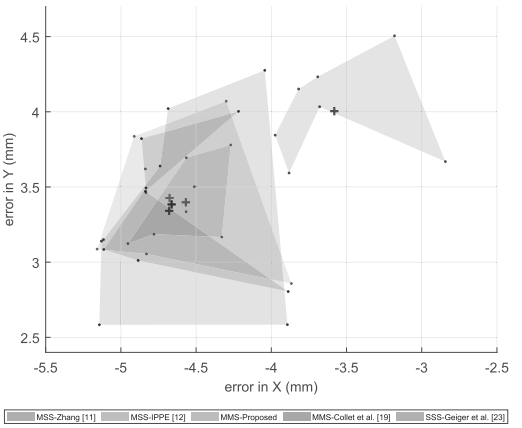


FIGURE 8. A 2D precision plot for the set of experiments using a diamond marker as target object. Each data point is the error of the estimate against the desired pose without the measure from the axis containing the depth information. The corresponding information from each method is shown in a unique color. The mean of each set of data points is presented by the + symbol and the region encompassing the scattered estimates shows the spread of estimates from respective methods.

with a significantly small deviation value. To better illustrate the effect of the standard deviation, we plot the translation estimates error without its depth dimension in Fig. 7. The mean of the data points is presented by the + symbol and the region encompassing the scattered estimates is shown in a unique color. It can be easily observed from the plot that the proposed method shows the most consistent results compared to other aforementioned methods.

We can see from the results that an offset is observed as the estimated errors lie in the quadrant formed by the negative X-axis and positive Y-axis. All the estimate errors lie within this quadrant, which is not common for a natural distribution of error. The offset indicates an uncertainty produced by a more direct cause, which is the instability of the robot base, as mentioned earlier. The effect is more apparent as the tool moves away from the robot base causing a marginal flex. However, the comparative performance of the methods under consideration and their statistical analysis are not directly affected by this problem.

The results for the second set of experiments using the diamond marker as the target object are presented in a similar structure in Table 2 and Fig. 8. The tabulated results show that the proposed method achieved the best results over all metrics except for σ_{re}, where it yields a result comparable to the stereo approach. SSS-Gieger *et al.* [23] shows a comparable error distribution to the proposed method.

The monocular single-shot approach by Zhang [11] seems to have the least consistent performance. The standard deviation of the estimate errors (σ_R, σ_t, and σ_{re}) is the highest in this case. The MSS-IPPE [12] performs comparatively better among the MSS approaches and yields a comparative result to the monocular multi-shot approach by Collins and Bartoli [12].

Moreover, it can be noted that the overall error of this set of experiments is larger by some factor compared to the results obtained for the experiments using the checkerboard. This might be due to the reason that we extract 408 corner points from a checkerboard of size 18×25. On the other hand, we use only 20 points from the diamond marker as illustrated in Fig. 1. The number of points is almost 20 times less for the case of a diamond marker with comparatively small spatial distribution to the checkerboard. In our opinion, the increase

in error for such a situation is in accordance with the stated reason.

The distribution of error estimates can be observed in Fig. 8 for the case of the diamond marker. As before, the proposed approach yields the most consistent result with its estimates being more precise and uniformly distributed.

V. CONCLUSION

In this article, we proposed a monocular multi-shot approach to estimate the 6-DoF pose of the camera against a planar target (object). The proposed approach models the geometric relation among various coordinate systems and explicitly incorporates the robotic manipulator poses into the formulation. It uses a non-linear optimizer to iteratively minimize the reprojection error based cost function. The experimental results were compared to four other existing studies, which included two monocular single shot, one monocular multi-shot, and one stereo approach. The tests were performed on both simulated data with synthetic images and real data. Two target patterns were considered for real data testing. Our method demonstrates significant improvement and robustness on many metrics in various test cases against other methods. In addition to improved accuracy, our approach achieves the most precise results.

ACKNOWLEDGMENT

The results are intended to be integrated in advanced camera-based systems attached to robotic manipulators in order to achieve complex remote handling for maintenance and operation in a safe and efficient way.

This article reflect the views of the authors. F4E and Tampere University (TAU) cannot be held responsible for any use which may be made of the information contained herein.

REFERENCES

[1] A. Richardson, J. Strom, and E. Olson, "AprilCal: Assisted and repeatable camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1814–1821.

[2] J. A. Castellanos and J. D. Tardos, *Mobile Robot Localization Map Building: A Multisensor Fusion Approach*. Boston, MA, USA: Springer, 1999.

[3] I. Ali, O. Suominen, and A. Gotchev, "Discrimination of active dynamic objects in stereo-based visual SLAM," *Electron. Imag.*, vol. 2018, no. 13, pp. 1–6, 2018.

[4] W. Pan, M. Lyu, K.-S. Hwang, M.-Y. Ju, and H. Shi, "A neuro-fuzzy visual servoing controller for an articulated manipulator," *IEEE Access*, vol. 6, pp. 3346–3357, 2018.

[5] S. Dong, A. H. Behzadan, F. Chen, and V. R. Kamat, "Collaborative visualization of engineering processes using tabletop augmented reality," *Adv. Eng. Softw.*, vol. 55, pp. 45–55, Jan. 2013.

[6] L. Minati, N. Yoshimura, and Y. Koike, "Hybrid control of a vision-guided robot arm by EOG, EMG, EEG biosignals and head movement acquired via a consumer-grade wearable device," *IEEE Access*, vol. 4, pp. 9528–9541, 2016.

[7] M. Zhou, X. Hao, A. Eslami, K. Huang, C. Cai, C. P. Lohmann, N. Navab, A. Knoll, and M. A. Nasseri, "6DOF needle pose estimation for robot-assisted vitreoretinal surgery," *IEEE Access*, vol. 7, pp. 63113–63122, 2019.

[8] V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, "Fully automated and stable registration for augmented reality applications," in *Proc. 2nd IEEE ACM Int. Symp. Mixed Augmented Reality*, Oct. 2003, pp. 93–102.

[9] T. Collins, J.-D. Durou, P. Gurdjos, and A. Bartoli, "Singleview perspective shape-from-texture with focal length estimation: A piecewise affine approach," in *Proc. 5th Int. Symp. 3D Data Process., Vis. Transmiss.* Paris, France: Espace Saint Martin, May 2010.

[10] P. Sturm, "Algorithms for plane-based pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2000, pp. 706–711.

[11] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[12] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 252–286, Sep. 2014.

[13] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 610–622, Jun. 2000.

[15] G. Schweighofer and A. Pinz, "Robust pose estimation from a planar target," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2024–2030, Dec. 2006.

[16] S. Li, C. Xu, and M. Xie, "A robust O(n) solution to the Perspective-n-Point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1444–1450, Jul. 2012.

[17] F. Camposeco, T. Sattler, and M. Pollefeys, "Minimal solvers for generalized pose and scale estimation from two rays and one point," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 202–218.

[18] D. Martinec and T. Pajdla, "Robust rotation and translation estimation in multiview reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[19] A. Collet and S. S. Srinivasa, "Efficient multi-view object recognition and full pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 2050–2055.

[20] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley, "Robust 6DOF motion estimation for non-overlapping, multi-camera systems," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2008, pp. 1–8.

[21] D. Nister, "An efficient solution to the five-point relative pose problem," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2003, p. 195.

[22] B. Clipp, C. Zach, J.-M. Frahm, and M. Pollefeys, "A new minimal solution to the relative pose of a calibrated stereo camera with small field of view overlap," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1725–1732.

[23] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 963–968.

[24] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *J. Field Robot.*, vol. 35, no. 4, pp. 578–595, 2018.

[25] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[26] I. Ali, O. Suominen, A. Gotchev, and E. R. Morales, "Methods for simultaneous robot-world-hand–eye calibration: A comparative study," *Sensors*, vol. 19, no. 12, p. 2837, 2019.

[27] A. Tabb and K. M. A. Yousef, "Solving the robot-world hand-eye (s) calibration problem with iterative methods," *Mach. Vis. Appl.*, vol. 28, nos. 5–6, pp. 569–590, 2017.

[28] K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1021–1028, Apr. 2019.

[29] T. Nöll, A. Pagani, and D. Stricker, "Markerless camera pose estimation-an overview," in *Proc. Vis. Large Unstructured Data Sets-Appl. Geospatial Planning, Model. Eng. (IRTG)*. Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum füer Informatik, 2011, pp. 55–129.

[30] P. Vicente, L. Jamone, and A. Bernardino, "Towards markerless visual servoing of grasping tasks for humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3811–3816.

[31] Y.-M. Wei, L. Kang, B. Yang, and L.-D. Wu, "Applications of structure from motion: A survey," *J. Zhejiang Univ. Sci. C*, vol. 14, no. 7, pp. 486–494, Jul. 2013.

[32] K. G. Derpanis, "Overview of the RANSAC algorithm," *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.

[33] A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ With the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, Inc, 2016.

IHTISHAM ALI received the B.Sc. degree in mechatronics engineering from the University of Engineering and Technology, Pakistan, in 2014, and the M.Sc. degree in automation engineering from Tampere University, Finland, in 2017. He is currently a Ph.D. Researcher with the 3D Media Group, Tampere University. He has worked on several industrial projects pertaining to machine automation using visual cues. His research interests include computer vision and robotics, specifically object pose estimation, 3-D reconstruction, and visual SLAM.

OLLI J. SUOMINEN received the B.Sc. and M.Sc.(Tech.) degrees in information technology, with a major in signal processing, from the Tampere University of Technology (TUT), in 2011 and 2012, respectively, where he is currently pursuing the Ph.D. degree with the Laboratory of Signal Processing, 3D Media Group. He also manages the construction and development of the Centre for Immersive Visual Technologies. After starting the B.Sc. thesis using only one camera (depth image-based rendering) and the M.Sc. thesis using two cameras (stereo depth estimation), he has scaled up to 40 cameras for the Ph.D. with research interests in multicamera systems, 3-D reconstruction, multimodal sensor fusion, SLAM, and light field capture. He focuses on applications in heavy mobile work machines, leading several industry driven research projects, and developing relations with the industry.

EMILIO RUIZ MORALES received the M.Sc. degree in electro-mechanical engineering and telecommunications from the École Polytechnique, Université Libre de Bruxelles, in 1990. He is currently the Project Manager for remote handling control systems of the ITER Project at the EU Fusion for Energy Agency. He has dedicated his career to the design and development of robotics control systems and advanced robotics applications in the fields of remote handling, nuclear, and surgical robotics.

ATANAS GOTCHEV received the M.Sc. degrees in radio and television engineering, in 1990, and in applied mathematics, in 1992, the Ph.D. degree in telecommunications from the Technical University of Sofia, in 1996, and the D.Sc.(Tech.) degree in information technologies from the Tampere University of Technology, in 2003. He is currently a Professor of Signal Processing and the Director of the Centre for Immersive Visual Technologies, Tampere University. His recent work concentrates on the algorithms for multisensor 3-D scene capture, transform-domain light-field reconstruction, and Fourier analysis of 3-D displays.

PUBLICATION III

Discrimination of active dynamic objects in stereo-based visual SLAM

I. Ali, O. Suominen, and A. Gotchev

Electronic Imaging, vol. 2018, no. 13, pp. 463-1–463-6
DOI: 10.2352/ISSN.2470-1173.2018.13.IPAS-463

https://doi.org/10.2352/ISSN.2470-1173.2018.13.IPAS-463

Discrimination of active dynamic objects in stereo-based visual SLAM

Ihtisham Ali, Olli Suominen, Atanas Gotchev ; Tampere University of Technology, Tampere, Finland

Abstract

Over the years, the problem of simultaneous localization and mapping have been substantially studied. Effective and robust techniques have been developed for mapping and localizing in an unknown environment in real-time. However, the bulk of the work presumes that the environment under observation is composed of static objects. In this study, we propose an approach aimed at localizing and mapping an environment irrespective of the motion of the objects in the scene. A hard threshold based Iterative Closest Point algorithm is used to compute transformations between point clouds that are obtained from dense stereo matching. The dynamic entities along with system noise are identified and isolated in the form of outliers of the data correspondence step. A confidence metric is defined that helps in identifying and transitioning a 3D point from static to dynamic and vice versa. The results are then verified in a 2D domain with the aid of a modified Gaussian Mixture Model based motion estimation. The dynamic objects are segmented in 3D and 2D domains for any possible analysis and decision making. The results demonstrate that the proposed approach effectively eliminates noise and isolates the dynamic objects during the mapping of the environment.

Introduction

In recent years, the approaches pertaining to Visual Simultaneous Localization and Mapping (SLAM) have been developed significantly; although it is a relatively new field. The research in this field was significantly aided by the release of Microsoft Kinect RGB-D (Red, Green, Blue, and Depth) camera. This field has proved to be of great interest to research and business minds alike, due to its impact applications. The state of the art methods are now capable of running the application in real time with robust performance. However, much improvement needs to be done towards handling problems such as expanded spatial volume with loop closure [1], dense mapping [2], and managing dynamic objects in a scene [3].

A variety of SLAM implementations exist. Each implementation may adopt a different type of sensor or methodology. A typical SLAM approach relies on the Iterative Closest Point (ICP) for registration of point clouds, and loop closing techniques for drift compensation [4]. Apart from RGB-D sensors, simple time-of-flight (TOF), monocular and stereo cameras can also be used for obtaining point clouds. Each of these sensors has its own advantages, coupled with inherent data processing challenges.

Until recently the core assumption for SLAM has been that the environment under observation is static, i.e. none of the observable objects in scene propose any change in their dynamics or shape.

As a result, this assumption leads to inconsistent map, erroneous localization, residual noise and possible failure in registration, when the environment is dynamic. Nevertheless, a few studies have successfully dealt with dynamic objects in the scene. Many of these studies use Kinect to obtain the depth maps [5].

Typically, dynamic objects in a scene can be detected and isolated for SLAM using CAD models or other form of prior knowledge with the use of commercial RGB-D sensors. However, such an approach limits the applications of the system. In this study, we demonstrate the application with a stereo camera for localizing and mapping an active dynamic environment without any prior knowledge about the dynamics in the scene.

Related Work

Davison et al. [6] introduced a real-time camera tracking system known as monoSLAM (monocular Simultaneously Localization and Mapping) to localize and map a freshly explored environment. It uses an extended Kalman filter (EKF) to estimate the camera pose. Later, in [7], Newcombe and Davison adopted structure from motion (SFM) to find the ego-motion and reconstructed a detailed model of the environment. Along with the prior mentioned studies, the approaches presented in [8], [9] and [10] maintain the assumption that the underlying environment is stationary and suggests to discard the dynamic element points by considering them outliers to the systems.

Nonetheless, the research for developing SLAM algorithms in a static environment has matured considerably. Hence, many researchers are now focused on implementing SLAM in a dynamic environment. In [11] Andrade-cetto et al. used a stereo camera to build a map for mobile robot localization. It uses strength augmentation of features and robot localization to learn in a moderately dynamic indoor environment. The landmarks used for mapping are low (approx. 50) and provide little information about the nature of the environment. In an attempt to capture more information about the dynamic object, Aguiar et al. [12] suggested a multi-view camera approach. This technique utilizes eight cameras to track a person and reconstruct a spatio-temporally consistent shape, texture, and motion of the performer at a high quality. Through a different approach, Zollhofer et al. [13] proposed to reconstruct a non-rigid body in real time with a single RGB-D camera. The non-rigid registration of RGB-D data to the template is performed using an extended non-linear As Rigid As Possible (ARAP) framework by implementing on an efficient GPU pipeline. Unfortunately, like many other implementations, [12] and [13] require an initial static model/template of the body that is later tracked and reconstructed. The template is then deformed over time based on the rigid registration and non-rigid fitting of points. However, the limiting factor is that the spatial extent of the scene is limited to a single object of interest. Additionally, the system may fail at registration and tracking in case of occlusion, sparse or noisy data.

The aforementioned limitations were successfully removed by Keller et al. [14]. The authors proposed a Point-Based Fusion approach to reconstruct a dynamic scene in real-time using Kinect/PMD Camboard. The approach considers outliers from ICP

as possible dynamic points and assigns a confidence value which later determines if the point is static or dynamic. The dynamic points are used as seeds for region growing method in order to segment the entire dynamic object in its corresponding depth map. The implementation proves to work effectively as it can reconstruct both static and dynamic scenes at a considerably good quality. Unlike the previous methods, it can map a comparatively larger spatial area and has been tested in an indoor environment. However, the use of these commercial RGB-D cameras is only suitable for the indoor environment, and it is very difficult to obtain meaningful data even with Kinect V2 in an outdoor environment. The maximum range of depth camera in Kinect V2 diminishes to 1.9m under the most favourable conditions with only two-thirds of the data being reasonably accurate [15]. In the presence of direct sunlight, the operation range falls below 0.8m [15]. The stated figures were obtained empirically with the data being processed and effectively denoised for better operation [15].

System Overview

An overview of the proposed approach is shown in Figure 1 in the form of a flowchart. The process pipeline takes in stereo images to compute the disparity maps and reconstruct 3D points after preprocessing. A six degree of-freedom (6DoF) pose of camera is computed for consecutive steps using the ICP in order to transform the 3D points from camera coordinates to a global map based in the global coordinate system. The outliers of the data association are not discarded. Instead, the outliers are used to understand the dynamics of the objects in the environment under observation. A Gaussian Mixture Model (GMM) based motion estimation method is used to corroborate the results obtained for the dynamic environment. The input data to GMM is preprocessed to extend its validity to moving sensor applications.

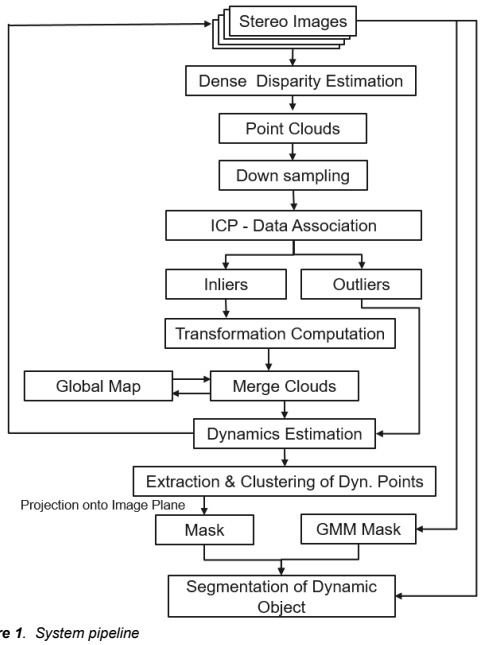


***Figure 1**. System pipeline*

Approach

Preprocessing

The stereo pair obtained for a scene is used to compute disparity estimates after rectification of the images. We employe the approach of dense disparity computation compared to sparse feature matching. Though feature matching based approaches can provide more consistent and accurate depth estimates, for pose estimation based on ICP and applications like 3D reconstruction, dense disparity estimate prove more useful. The disparity maps were computed using the Semi-Global algorithm as it offers a good compromise between computational speed and global optimality. Each pixel position $(x, y)^T \in R^2$ has its computed disparity $D_i \in R$. The disparity maps are obtained for both the images and a consistency check is performed from one camera to the other in order to remove false disparities. The 3D positions of the valid disparity points are recovered in the form of dense points clouds. However, to ease the computation and memory complexities for SLAM, the point clouds are uniformly downsampled using a grid filter. Each individual point cloud PtC_t has the associated description of each point i.e. Location (X_k, Y_k, Z_k), Color (R_k, G_k, B_k) and Normal vectors to the plane (Nx_k, Ny_k, Nz_k) stored along with it.

Data Association and Pose Estimation

The data association and pose estimation are the constituent steps of point cloud registration. During the registration step, the points from PtC_t are searched for correspondence with points from PtC_{t-1}.The algorithm Iterative Closest Point (ICP) is used to select the optimum points by iteratively minimizing the error metric e^i given in equation (1).

$$e^i = \sum_{i=1}^{N} d_s^2(T^i p_k , S_j^k) \qquad (1)$$

where d_s is the signed distance from a point to the plane, T^i is the transformation computed in the iteration i of the error minimization process, p_k are points from PtC_t and S_j depicts the tangent plane of at point q_j for the points in PtC_{t-1}.The transformation matrix T_t depicts the 6DOF camera pose change between the time t and t-1, where T_t is composed of a rotational matrix $R_t \in R^3$ and translational vector $tr_t \in R^3$.The 3D points and the associated normal are converted to global coordinate using the transformation matrix.

Generally, a percentage of closest points are selected as inliers for minimizing the error metric and computation of camera pose. However, we adopted a hard threshold based approach that filters the nearest points selected in each iteration, thereby removing most of the wrong correspondences (outliers) from the process that are present either due to noise (erroneous depth estimation, different sampling of an entity or motion of the objects. The points that help to obtain the correct transformation during the iterative process are known as the inliers.

Once spatially transformed, the new point cloud is merged with the global cloud or the 3D map. The global cloud in our work stores additional two descriptions for each 3D point in addition to the original three properties of a point in a point cloud. A confidence metric C_k and frame presence F_k is defined for each 3D point. The confidence C_k of a point informs us about the integrity of the point for being static and valid while the frame presence F_k stores the information about first time the point was introduced to the system.

Merging and Confidence Gain

The addition of new points in each iteration adds a significant amount of computational complexity and memory load on the processing pipeline. Therefore, it is advisable to remove any redundant or erroneous 3D points from the global map. Merging of points serve as one of the two steps that help in reducing the number of points in the global map. A point q_n in PtC_t may find multiple valid inlier correspondences in PtC_{t-1} during the transformation computation, however, only the closest single point p_n is merged physically after registering of the point clouds. The physical properties (location, color and normal vectors) are averaged to create a new point, which helps to remove the redundant duplicate. The confidences of both p_n and q_n are summed and increased by a constant (0.1 for our experimentation). The frame presence F_k is incremented once for all the inliers, to signify that it has been observed in the scene. In contrast the remaining valid associations (among the inliers) of q_n are not merged physically but added with the original properties to the global map since they might represent a different view of the same object. However, the confidence of these points is merged with q_n and raised through a gaussian distribution as shown in Figure 2. The maximum confidence is set equal to the constant 0.1 for the closest points. The further the points are from each other, the lower confidence it gains. The standard deviation of the gaussian distribution is set to be half of the correspondence threshold used during ICP.

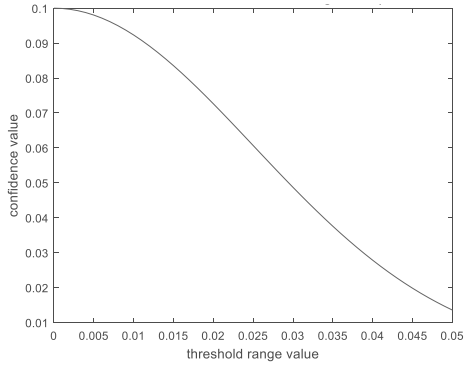


***Figure 2**. Gaussian distribution based confidence assignment at thresh=0.05*

Confidence Reduction and Removal

Confidence gain during merging of the points helped to determine the stability of points in the map. The higher the confidence, the more stable and static the point has been in the scene. However, the reverse is equally important in order to accurately update the global map. If a stationary object in the scene starts to move, the associated 3D points should logically change its position in the global map. The dynamic nature of the points is obtained by continuously reducing the confidence of all the points by a constant (1/10 of the confidence gain in this study) that are in view of the camera and therefore expected to be seen.

The 3D points from the global map are projected to the image plane using the camera intrinsics K and the inverse of global camera pose $T_{g\,t}^{-1}$ at time t. The points that are projected within the bounds of the plane are assumed to be in the camera perspective and, therefore, reduced in confidence. Among these points, those that have been associated with other points would still have a positive confidence change, however, the points that did not find any association would only be reduced in confidence. If the confidence of a point falls below 1, it is assumed to be unstable or dynamic.

In order to maintain and update the map, unstable points representing noise and dynamic 3D points are removed in each timestamp. For this study, the maximum confidence that a point can acquire is 1.25 which was selected empirically while keeping into consideration the confidence gain and reduction values. The global map is searched for these unstable points that have remained unstable for more than a threshold time t_{max} and are removed from the map.

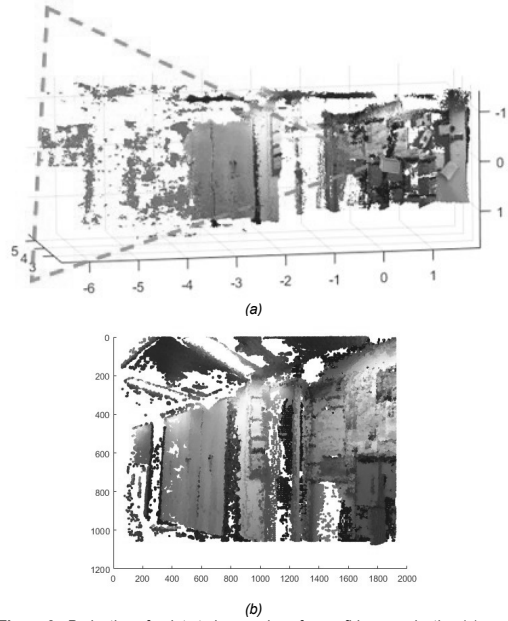


***Figure 3**. Projection of points to image plane for confidence reduction (a) illustration of the camera viewing the global cloud (b) projected points onto the image plane from the perspective view*

Dynamic Estimation

The Global Cloud is composed of both static and unstable/dynamic points. The static points have high confidence measure that is obtained through the continuous merging of points from close timestamps. The unstable noise or points from a dynamic object are observe at different position and with less consistency, hence, they do not accumulate enough confidence. It is essential to discriminate the unstable points due to point cloud reconstruction inaccuracy or slightly off localization and the points pertaining to true dynamic entities.

For an image frame at time t, the low confidence points from the global point cloud are projected on to the image using the accumulated transformations $T_{g\,t}^{-1}$ computed during the registration. The points that lie within the bounds of the image plane are indexed and clustered in 3D space based on their distances. For each frame at time t multiple k clusters $C_{t\,k}$ might

be created. Clusters with fewer points than a threshold are removed. This threshold may vary depending on the downsampling of the original point clouds. A 2D mask is generated from the boundaries of the projected clusters on the image plane. This mask may contain the bounds of both dynamic points and the unstable noise. In order to discriminate between the two, another verification step is included in the process.

A GMM based motion detection approach [16] is adopted to highlight moving objects in the scene. GMM is a background modelling technique that is trained on images to learn the background model at pixel level and describe each pixel with K gaussian distributions. The approach detects any moving object that does not fit the model's description in the form of foreground. However, a limitation exists to the direct application of GMM. The approach is only applicable for static camera systems. In this study, we extended the use of GMM with specific preprocessing steps. With a moving camera, the background changes frequently, therefore, the model is continuously trained with few images as function of displacement. For this study we used 3 to 10 images based on the displacement from current scene. Moreover, the training images are geometrically transformed to the current frame by tracking salient features in the images in order to maintain the assumption of static camera for GMM application. The approach provides a clean highlight of the moving objects for small translation between consecutive images. The mask obtained using motion detection with GMM is used as seed to select the blobs from the first mask obtained by the projection of the clustered 3D points. The verified clusters are used for segmentation of the moving object from the images as shown in Figure 4.

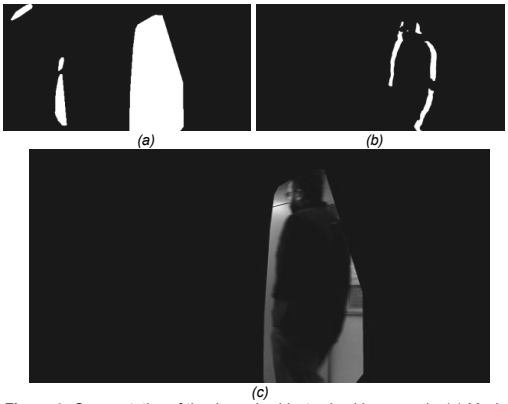


Figure 4. *Segmentation of the dynamic object using binary masks (a) Mask obtained by projecting the clustered 3D dynamic points onto image plane (b) Masked obtained using GMM based motion detection(c) Segmented dynamic object from 2D images*

Experimental Results

This section provides an overview of the experimentation setup adopted for this study and analysis the results obtained using the proposed approach.

In this study, the data was recorded using a commercial stereo camera Zed [17]. The stereo camera follows the Pinhole camera model with a baseline of 12cm between the camera pair. The standard specifications of the camera quote to work both indoor and outdoor with an effective range of 0.5 to 20 meters [17]. However, the accuracy of depth estimation decreases with distance, therefore, we limited our interest to 6.5–7.5 meters during outdoor usage for more consistent and accurate data. The Zed camera was calibrated using the calibration approach provided by Computer Vision System Toolbox™, which is based on the work of Jean-Yves Bouguet [18].

The system is tested on various test scenes of varying dynamic nature to better comprehend the performance of the approach. The scene shown in Figure 5 demonstrates the ability of the system to incorporate dynamic objects in the environment. The scene was recorded at 30fps with the camera being fixed in the environment. The 1st Column of images show the excerpt from the videos sequence; the seconds column shows the updated map/ global cloud and the last column shows the objects segmented objects when in dynamic state. The dynamic points from the moving object are successfully incorporated as part of the map and then effectively updated during the motion.

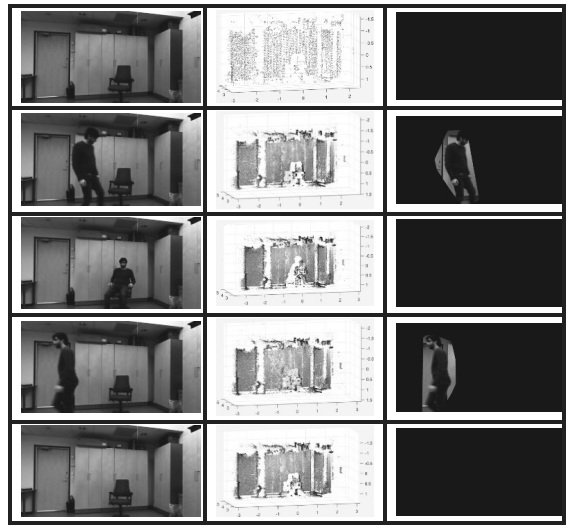


Figure 5. *Test sequence with stationary camera*

The second indoor scene shown in Figure 6 records a dynamic environment where camera motion is introduced as an additional challenge. The test sequence updates the map while the person passes by in the corridor. The segmentation step not only accurately segments the dynamic object in the middle of the scene but also at the far end of the corridor where most of points are unreliable.

The test sequence shown in Figure 7 was recorded in an outdoor environment over a longer time. The scene was recorded on a cloudy winter day. In order to test the robustness of the approach, the video was acquired using a hand-held Zed camera at 10 fps, and as a result, the scene includes sudden erratic motion. It can be observed that the moving objects in the scene are highlighted in the global map and effectively removed once they pass from the scene, however, the static objects such as the tree is retained in the map even though they are exposed for approximately the same period.

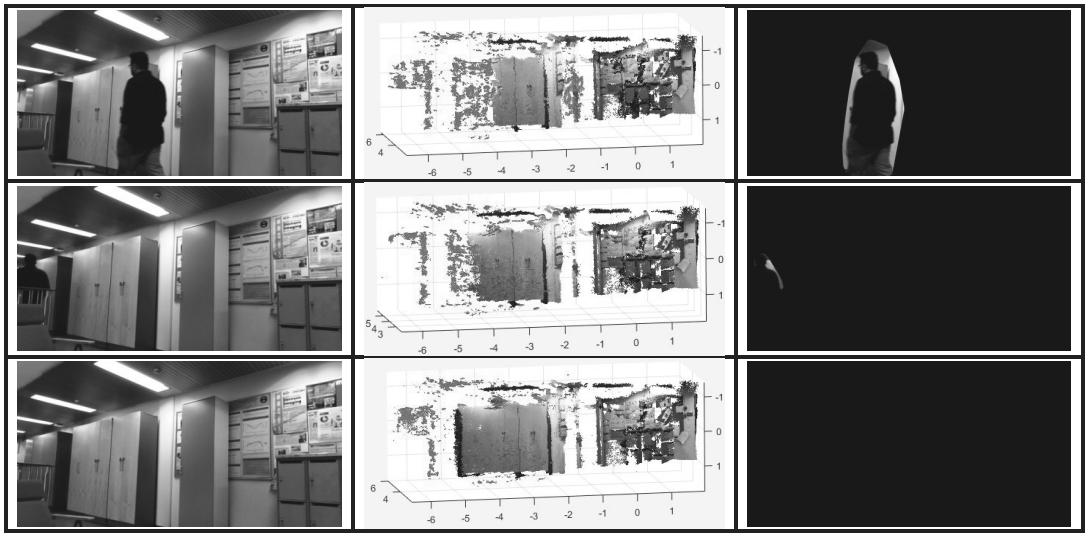


***Figure 6**. Indoor test sequence with moving camera*

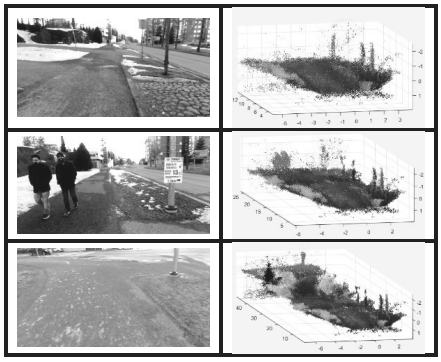


***Figure 7.** Excerpts from outdoor test sequence with dynamic objects and moving camera*

Conclusion

We proposed a scheme to discriminate active dynamic objects present in an environment while localizing and mapping the scene using a stereo camera. The approach is tested on datasets composed of both indoor and outdoor test scenes recorded at various acquisition rates and external challenges such as erratic camera motion, less distinct geometrical structures, and low illumination. The system effectively localizes the observer in the dynamic environment and builds a map irrespective of the relation of motion of camera to the motion of objects in the observed environment. The moving objects are successfully segmented in both the 2D and 3D domains for further extensive analysis.

References

[1] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers and W. Burgard, "An evaluation of the RGB-D SLAM system", in IEEE International Conference on Robotics and Automation, 2012.

[2] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces", in 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007.

[3] K. Litomisky and B. Bhanu, "Removing Moving Objects from Point Cloud Scenes", in Advances in Depth Image Analysis and Applications, Berlin, 2013, Vol. 7854, pp. 50-58.

[4] Bradski, H. Strasdat, J. M. M. Montiel, and Andrew J. Davison. "Scale drift-aware large scale monocular SLAM." Robotics: Science and Systems VI, vol.2, 2010.

[5] Korn M. and Pauli J, "KinFu MOT: KinectFusion with Moving Objects Tracking", in Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Berlin , 2015, Vol. 3, pp. 648-657.

[6] A. Davison, I. Reid, N. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052-1067, 2007.

[7] R. Newcombe and A. Davison, "Live dense reconstruction with a single moving camera", in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[8] M. Meilland and A. Comport, "On unifying key-frame and voxel-based dense visual SLAM at large scales", in International Conference on Intelligent Robots and Systems, Tokyo, 2013.

[9] R. Newcombe, S. Lovegrove and A. Davison, "DTAM: Dense tracking and mapping in real-time", in IEEE International Conference on Computer Vision (ICCV), Germany, 2011, pp. 2320-2327.

[10] C. Kerl, J. Stuckler and D. Cremers, "Dense Continuous-Time Tracking and Mapping with Rolling Shutter RGB-D Cameras", in IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2264-2272.

[11] J. Andrade-cetto, and S. Alberto, “Concurrent map building and localization on indoor dynamic environments”, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 16, no. 3, pp. 361-374, 2002.

[12] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. Seidel and S. Thrun, "Performance capture from sparse multi-view video", ACM Transactions on Graphics, vol. 27, no. 3, p. 1, 2008.

[13] M. Zollhöfer, C. Theobalt, M. Stamminger, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon and C. Loop, "Real-time non-rigid reconstruction using an RGB-D camera", ACM Transactions on Graphics, vol. 33, no. 4, pp. 1-12, 2014.

[14] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich and A. Kolb, "Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion", International Conference on 3D Vision, 2013, pp. 1-8.

[15] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter and R. Siegwart, "Kinect V2 for mobile robot navigation: Evaluation and modeling", International Conference on Advanced Robotics (ICAR), 2015.

[16] P. Kaewtrakulpong, R. Bowden, An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection, In Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01, Video Based Surveillance Systems: Computer Vision and Distributed Processing (September 2001)

[17] ZED Stereo Camera [Internet]. Stereolabs.com. 2017 [cited 10 November, 2017]. Available from: https://www.stereolabs.com/

[18] Bouguet, J. Y. Camera Calibration Toolbox for Matlab. Computational Vision at the California Institute of Technology. Camera Calibration Toolbox for MATLAB.

PUBLICATION IV

FinnForest dataset: A forest landscape for visual SLAM

I. Ali, A. Durmush, O. Suominen, J. Yli-Hietanen, S. Peltonen, J. Collin, and A. Gotchev

Robotics and Autonomous Systems, vol. 132, p. 103 610
DOI: 10.1016/j.robot.2020.103610

Robotics and Autonomous Systems 132 (2020) 103610

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

FinnForest dataset: A forest landscape for visual SLAM

Ihtisham Ali [a,*], Ahmed Durmush [a], Olli Suominen [a], Jari Yli-Hietanen [a], Sari Peltonen [a], Jussi Collin [b], Atanas Gotchev [a]

[a] Faculty of Information Technology and Communication Sciences, Tampere University, Finland
[b] JC Inertial Oy, Finland

ARTICLE INFO

Article history:
Received 6 February 2020
Received in revised form 24 July 2020
Accepted 28 July 2020
Available online 3 August 2020

Keywords:
Forest
Dataset
SLAM
Visual odometry
Navigation
Localization
Mapping
Stereo
Autonomous driving
Mobile robotics
Field robotics
Computer vision

ABSTRACT

This paper presents a novel challenging dataset that offers a new landscape of testing material for mobile robotics, autonomous driving research, and forestry operation. In contrast to common urban structures, we explore an unregulated natural environment to exemplify sub-urban and forest environment. The sequences provide two-natured data where each place is visited in summer and winter conditions. The vehicle used for recording is equipped with a sensor rig that constitutes four RGB cameras, an Inertial Measurement Unit, and a Global Navigation Satellite System receiver. The sensors are synchronized based on non-drifting timestamps. The dataset provides trajectories of varying complexity both for the state of the art visual odometry approaches and visual simultaneous localization and mapping algorithms. The full dataset and toolkits are available for download at: http://urn.fi/urn:nbn:fi:att:9b8157a7-1e0f-47c2-bd4e-a19a7e952c0d. As an alternative, you can browse for the dataset using the article title at: http://etsin.fairdata.fi.

1. Introduction

The intense competition to develop a safe marketable self-driving car has motivated a huge amount of research in the field of autonomous vehicles. Coupled with the growing interest of companies to put self-driving cars on the road, various companies are also interested in introducing other forms of autonomous vehicles to automate various industrial processes such as mining, shipping, agriculture, and forestry. Irrespective of the industry and targeted operations, the autonomy of any machine is highly dependent on advancements in a number of vision technologies, such as object detection [1], reconstruction quality [2], scene perception [3]. However, the base capabilities of an autonomous vehicle that need most attention remain visual odometry, relocalization and mapping [4]. This requires testing and validation on all scenarios that a vehicle/machine can face in a simulated environment. A variety of public datasets are available that provide a good amount of data for testing in various conditions and locations. We will mention some of the well-known datasets in an attempt to measure the expanse of the collections and find a horizon. Most of these datasets focus on urban environments (for example, [5–9]) in order to facilitate testing on public roads in urban areas. The earliest among these datasets that recorded urban environment are Ford Campus [10] and KITTI [11]. Being among the first public datasets in the field, these datasets contributed significantly towards testing and validation. The recent additions to the publicly available datasets are The Oxford RobotCar [12], KAIST Multi-Spectral Day/Night [13], and Complex Urban LiDAR Data Set [14]. All these datasets, when combined, provide a significant amount of testing data for urban environment with short and long trajectories at various speeds [10]. Moreover, they incorporate weather and seasonal changes [12], long term changes in urban structure [12] and gradual/sudden illumination variations [13]. However, all these datasets target indoor or outdoor urban environment.

In contrast, some unique datasets target entirely different environments to assist automation of other form of vehicles. Among these are Aqualoc Underwater [15], Canoe [16] and Underwater Caves SONAR and Vision Dataset [17]. These datasets comprise of data acquired for under water exploration and surface sailing conditions. On the other hand, a few public datasets target more domain specific terrains for their experimentation. In [18], authors recorded data in the Chile's largest underground production-active copper mine. The data was recorded for a length of approximately 2 km using Lidar, radar and stereo cameras fixed on

* Corresponding author.
E-mail addresses: ihtisham.ali@tuni.fi, ihtishamaliktk@gmail.com (I. Ali).

https://doi.org/10.1016/j.robot.2020.103610

Fig. 1. Recording platform. Our vehicle is equipped with four Basler HD cameras, a NovAtel GNSS antenna, and a KVH 1750 IMU with fiber optic gyro.

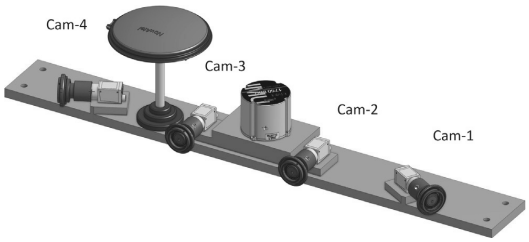


Fig. 2. Rendered 3D model of the sensor rig.

a robotic platform. The Devon Island rover navigation dataset [19] provides a dataset for testing rovers for planetary explorations. The dataset was recorded on Devon Island in the Canadian High Arctic, which is assumed to be analogous to Moon/Mars terrains due to the wide variety of geological features and microbiological attributes of the site.

The dataset that has high relevance to our work is the SFU Mountain dataset [20]. The study used a mobile ground based robot to traverse walking trails in the Burnaby Mountain, British Columbia, Canada. The dataset provides a semi-structured woodland terrain with different illumination and weather conditions and with changing vegetation, infrastructure, and pedestrian traffic. The dataset provides a good amount of data for visual odometry, however, lacks to present opportunities to test loop closure and re-localization.

In [21], the authors carried out brief experiments on the SFU Mountain dataset and their own dataset, Hillwood. The Hillwood dataset consists of photorealistic rendered and real forest video scenes. However, the Hillwood dataset only provide video recordings for testing without any ground truth information. In their conclusive remarks, the authors stressed upon the need and advantage of actual forest dataset with complete synchronized groundtruth poses [21].

In this paper, we present a new dataset that will target a real forest landscape recorded in the outskirts of Tampere, Finland. The goal is to provide testing data in order to facilitate the research towards increasing the autonomy of vehicles traversing rural areas and heavy machines working in the forest. Unlike urban settings, a terrain environment provides fewer discriminate landmarks and more repetitive textures in the scene. Presumably, such a situation strengthens visual odometry to some extent, however, affects adversely relocalization algorithms. This dataset provides semi-structured forest routes under different conditions (i.e. lighting, weather, vegetation, and infrastructure) in a highly self-similar natural environment. Furthermore, the sequences include scenes that best replicate the motions (i.e. stationary, sharp motion, bumps and potholes, slopes, and back-and-forth motion) and environments (i.e. log piles, close-up of trees, off-road routes) involved in actual forestry operations. The dataset includes unique trajectories to test both visual simultaneous localization and mapping (visual-SLAM) and visual odometry algorithms thoroughly. Moreover, each path is traversed in two different condition, namely sunny summer and snowy winter. The dataset provides images from 4 cameras and ground truth poses for each sequence in each condition using Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) data fusion. We provide processed rectified images, calibration data and ground truth at three sampling rates i.e. 40, 13.33 and 8 Hz except for two sequences which are sampled at 20, 10 and 7 Hz. For simpler representation, here onwards, we will approximate 13.33 to 13 Hz in the manuscript. Additionally, we provide raw images (40 Hz) for most of the sequences and the calibration images to the public. For this purpose, we also provide development tools to process raw data and evaluation tools in order to facilitate benchmarking against the state-of-the-art.

We hope this dataset provides a good reference for rural, forest and general terrain environment in order to facilitate the researchers to mitigate the challenges faced in this field of research.

2. Recording platform and sensor configuration

The recording platform and the arrangement are illustrated in Fig. 1. The data was recorded using a sensor rig mounted on a vibration dampening platform affixed to the vehicle. The vibration dampening platform was affixed to the Jeep using strong suction cups. The rig houses all the sensors as shown in Fig. 2.

The sensor and hardware specifications are as follows:

(i) 4 × Basler acA1920-50gc GigE camera with the Sony IMX174 CMOS Color sensor, Resolution (HxV) 1920 × 1200 (2.3 MPx), 84° HFoV, 59° VFoV, 6 mm Focal Length Lens, 20 cm baseline for each stereo pair.
(ii) 1 × KVH 1750 IMU, fiber optic gyro, bias instability $\leq$ 0.05 °/h, 1σ, 200 Hz.
(iii) 1 × NovAtel PwrPak7, OEM7 GNSS, 20 Hz.
(iv) 1 × CC320 Machine Vision Timing Controller, 8 Digital Inputs of 5 V to 24 V at 3 mA to 20 mA, 8 Digital Outputs of 24 V and 20 mA.
(v) 1 × Embedded system with Quad Core Intel Core i7 processor, 2 DDR4 with 64 GB memory, 6 GigE LAN with 4 PoE.

For the sensors and their coordinate systems, we use the following notations,
C_1 Camera 1
C_2 Camera 2 (reference frame)
C_3 Camera 3
C_4 Camera 4
I IMU
G GNSS

All cameras were connected to the embedded computer. The cameras stored data on the computer while the IMU and GNSS data was recorded on the internal memory of the NovAtel Module. To minimize write latency into storage and to prevent losses, we used CAT7 cable and wrote on SSDs using parallel threads for all the cameras.

To obtain high quality images it was essential to control the exposure time of the cameras during the acquisition. To minimize the effect of motion blur, the exposure time was kept below

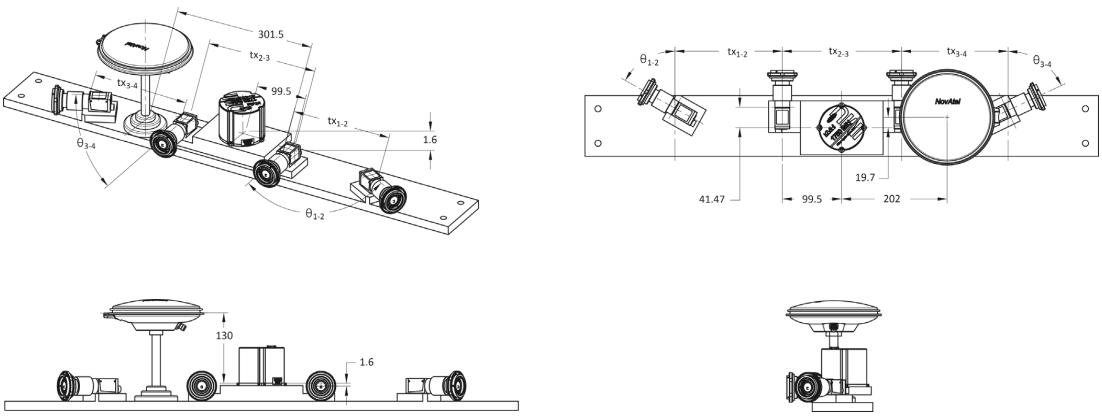


Fig. 3. The figure illustrates the mounting positions of the sensors with respect to each other from various views. The calibrated transformations are provided as part of the dataset.

10 ms. Moreover, to obtain images from four cameras for stereo analysis, it was of the utmost importance to enforce synchronicity. Hence, to acquire synchronized feed from four cameras at 40 fps on a Windows based platform, we utilized a special purpose triggering hardware known as Machine Vision Timing Controller. This timing controller or triggering device sent synchronized signals to all the cameras in order to enforce realtime consistent capture. Additionally, one trigger signal was sent to the NovAtel Module from the triggering device to generate timestamps. The NovAtel module was configured to store a timestamp in GPS time upon receiving the signal from the triggering device. The GPS time is more accurate and does not drift, unlike the clock on the Windows platform. This timestamp signal was sent at a delay of 1.5 ms. Even though this would have had a negligible effect, nonetheless, we compensated for this delay during the ground truth generation. The IMU and GNSS data are pre-synchronized by the NovAtel receiver. Hence, we have a precise synchronization among the cameras, IMU and GNSS data in effect. The raw data of GNSS and IMU are acquired at 20 Hz and 200 Hz, respectively. However, they are not the limits of the system. The maximum acquisition rate of the system is 100 Hz and 1000 Hz, respectively. For this study 20 Hz GNSS is used and interpolated to 50 Hz with the IMU data during post-processing by NovAtel Inertial Explorer software.

The sensor arrangement is illustrated in Fig. 3. It constitutes four cameras, a GNSS antenna and an IMU unit. The sensor rig has the middle cameras (C_2 and C_3) facing forward and houses the IMU unit in between them. The outward facing cameras (C_1 and C_4) are at nearly same angle from the forward direction. The motivation backing this camera arrangement is to test the effects of various camera configurations on the accuracy of joint perception. It is mostly observed in SLAM implementations that during forward motion, the view is dominated by consistently tracked areas of interest that are further away from the camera. This negatively affects the scale estimation for visual odometry. This is more apparent in monocular SLAM algorithms where the SLAM methods fail at certain point because the further points do not exhibit enough disparity change. The methods survive as long as the closest features are not lost due to motion blur. However, if the camera is fixed at an angle, instead of facing the forward direction, then the effective area in which the feature points exhibit disparity change increases as well.

Fig. 4. The GPS trajectory of our recordings in the forest area in the outskirts of Tampere, Finland.

3. Data overview

Our primary contribution through publishing this dataset is to provide publicly accessible data recorded in forest for research towards Advanced driver-assistance systems (ADAS) and autonomous work machines. In general the dataset provides challenges by incorporating sequences that are recorded at various times of day and weather conditions. Moreover, the sequences have been recorded so that they present considerable challenges for both visual odometry and SLAM approaches. The area explored during the course of the recording sessions can be viewed in Fig. 4. The dataset comprises unique trajectories, most of which are recorded in two seasonal conditions. In winter, a part of the route was blocked due to heavy snow and could not be re-recorded in snowy conditions. An overview of the dataset is provided in Table 1. The dataset offers a total of 11 sequences. We provide the dataset at different sampling rates to facilitate testing. The original visual data was recorded at 40 Hz and later subsampled to facilitate testing. The subsampled versions are provided in the form of compressed image packages and Rosbags.

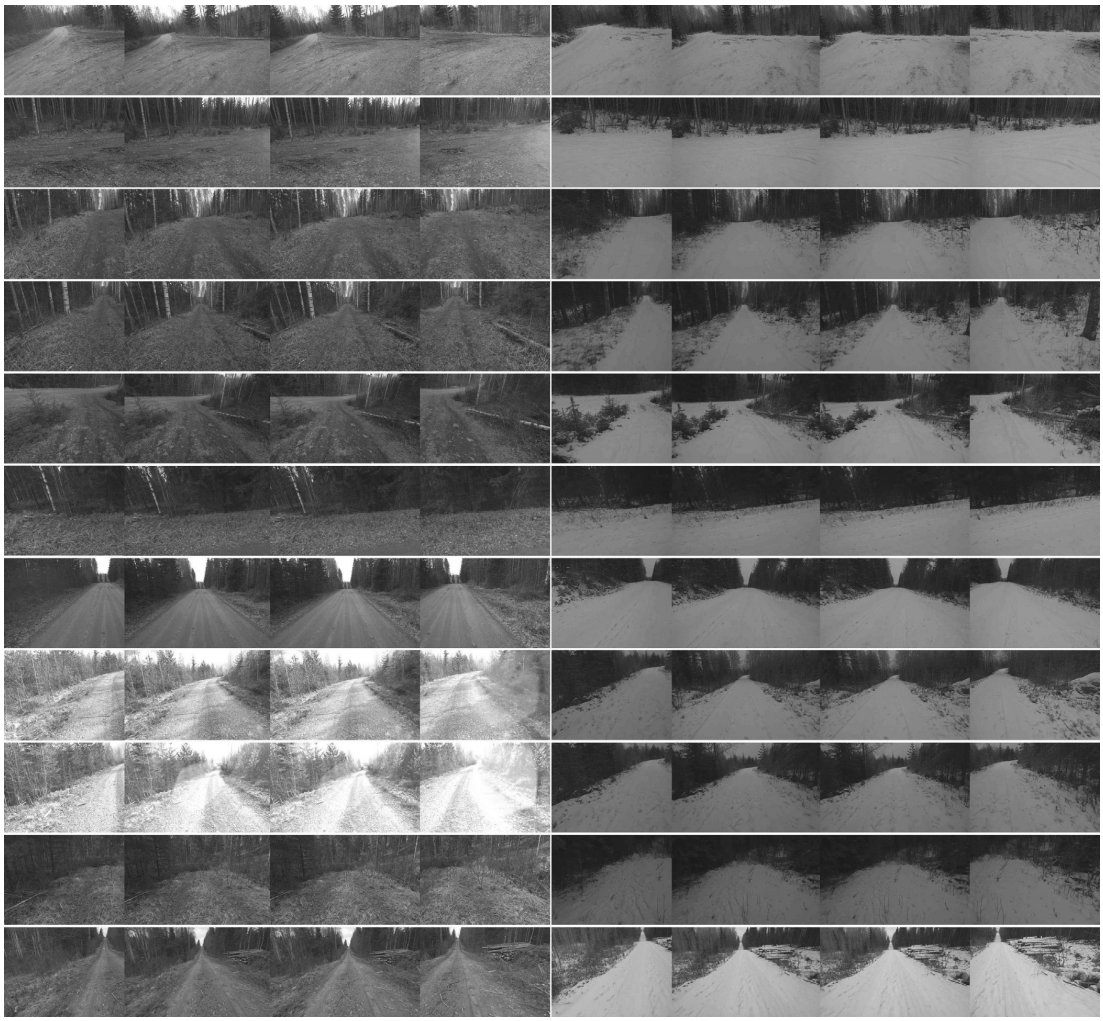


Fig. 5. Montage of images from all cameras arranged left to right illustrating the significant changes in appearance of same scene over seasonal and route changes.

Table 1
An overview of the nature of data in the FinnForest Dataset.

Seq. No	Frames			Distance (km)	Loop	Season	Time of Day
	40/20 Hz	13/10 Hz	8/7 Hz				
*W*01	27 630	9210	5526	1.29	Yes	Winter	Daylight, Overcast
*W*03	23 100	7700	4620	1.69	No	Winter	Daylight, Overcast
*W*04	37 010	12 337	7402	2.35	No	Winter	Daylight, Overcast
*W*05	57 288	19 096	11 458	4.74	No	Winter	Daylight, Overcast
*W*06	20 875	10 438	6959	3.59	No	Winter	Night
*W*07	43 780	21 890	14 594	6.48	No	Winter	Dusk, varying illumination
*S*01	27 960	9320	5592	1.29	Yes	Summer	Daylight, Sunny
*S*02	21 333	7111	4267	1.99	Yes	Summer	Daylight, Sunny
*S*03	15 000	5000	3000	1.69	No	Summer	Daylight, Overcast
*S*04	30 662	10 221	6133	2.32	No	Summer	Daylight, Sunny
*S*05	61 662	20 554	12 333	5.84	No	Summer	Daylight, Overcast

The number of frames at each sampling rate is provided against the sequence name in Table 1. Three of the sequences offer loop closure opportunities while the remaining sequences are aimed at testing visual odometry. We have also tabulated the distance covered while traversing each path. The range of distance traveled varies from 1.3 km to 6.48 km. Information regarding the season

Fig. 6. Illustration of the drastic changes in appearance of the scene produced by different illumination and weather conditions.

and the illumination condition is also provided corresponding to each sequence. The seasonal name is also abbreviated in the name of each sequence for clarity. The dataset covers a variety of conditions with different illumination such as overcast, direct sunlight, dusk and night. However, we would like to state that the dataset does not offer sequences with rain and fog which would have provided further useful information for testing. Further details about the unique challenges of each sequence are provided in Section 7, Discussion.

Fig. 5 presents a montage of selected images illustrating the range of varying appearances of the environment encountered as a result of different season and routes. The left half of the montage constitutes the left to right camera images from the summer dataset while the right half of the montage shows the left to right images from the winter dataset of same scene from almost the same vehicle locations.

Fig. 6 illustrates the changes in appearance of the scene from almost similar camera perspective and location during both seasons and the challenges it brings about. The dark overcast in winter demands longer exposure time and slower vehicle motion to capture the details in the scene accurately. On the other hand, summer season presents challenges like overexposure, rain, puddle and flares in the scenes. The last row of images exhibit the conditions of a night and dusk time with varying illumination.

The high resolution and frame rate of the data recordings make it challenging to store the data on online data repositories. In order to make the usage of the data convenient for users, we have split the dataset into subset sequences. Each sequence can be downloaded and used independently as a .zip package at three sampling rates. Moreover, the most common configuration preferred for stereoscopic analysis is parallel, hence, we only provide the processed images from the forward facing stereo pair i.e C_2 and C_3. Nonetheless, the raw images from all the cameras

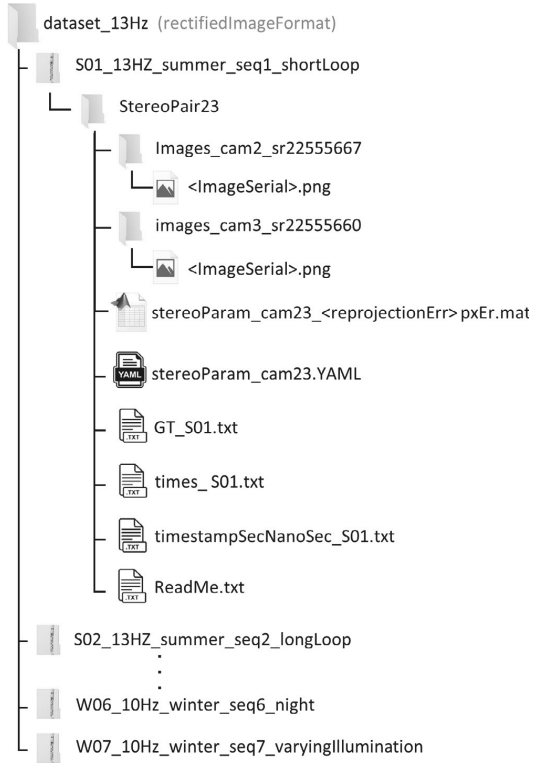


Fig. 7. Directory layout for a sub-sampled subset from the dataset. Extracting them will preserve the folder structure.

are provided in the dataset along with a toolkit to easily extract and process them in ready to use format. The MATLAB toolkit readily extracts the raw images into stereo pairs C_1-C_2, C_2-C_3 and C_3-C_4.

The data structure or format for each sequence is illustrated in Fig. 7. The name of the folder constitutes the nature of the data and the rate at which it is sampled. Each sequence is self-contained and is provided with supporting files inside the compressed file format. The compressed file in turn constitutes sub folders, which correspond to the stereo pairs for forward facing cameras (C_2 and C_3). The Rosbag version contains the Rosbag file instead of the PNG image files for the cameras (C_2 and C_3). Additionally, the calibration files, timestamps and the ground truth poses are provided in the corresponding directories for the rectified cases. The ground truth data already corresponds directly to the images provided and does not need further matching or synchronization. Each row of the ground truth text file corresponds to a new reading of the ground truth pose of 3×4 matrix $[R|t]$ in the row first vectored form as shown below:

$$[R_{11}\ R_{12}\ R_{13}\ t_x\ R_{21}\ R_{22}\ R_{23}\ t_y\ R_{31}\ R_{32}\ R_{33}\ t_z]$$

4. Sensor calibration and ground truth

In this section, we will discuss two forms of calibration that are essential to use the data effectively.

4.1. Cam-to-cam calibration

The first calibration step is the camera-to-camera calibration, which is performed to compute the intrinsic parameters and extrinsic transformations for the cameras. In the dataset, we have included both the processed data (using the calibrations) and the raw data. The processed data from the cameras can be directly used with any SLAM pipeline using the provided calibration parameters. However, for researchers who wish to re-calibrate the cameras and process the raw data themselves, we have included the raw images along with the calibration images in the dataset.

The calibration images are provided as stereo pairs between the nearest two cameras. Special attention was given to calibration by recalibrating the cameras for each recording session. Although, the sensor setup was not altered, some minute numerical differences are possible. It is strongly recommended to use the calibration parameters from the calibration files and not the illustrations. The camera-to-camera calibrations are provided for the nearest camera pairs, namely C_1-C_2, C_2-C_3 and C_3-C_4. These camera pairs are jointly calibrated using MATLAB stereo calibration toolbox for their intrinsic and extrinsic parameters based on the approach presented in [22]. The calibration information is provided in two forms, namely MATLAB stereo-parameters object file and a text file with excerpts of the object file along the dataset.

4.2. Cam-to-IMU calibration

We calibrate the camera and IMU in order to obtain the external transformation between the camera and IMU unit. For this, a sequence was recorded in front of the calibration board, where the motion in all the six degrees of freedom was stimulated by moving along and around each axis. The relation between the camera and the IMU is then analogous to hand-eye calibration problem. For this, we utilize Kalibr toolkit [23] which estimates the spatial and temporal parameters of a camera system with respect to an intrinsically calibrated IMU. Since we have an accurate synchronization between the images acquired from the camera and the data from IMU/GNSS using the timestamps, we are not interested in the temporal relationship provided by the toolkit. However, the spatial parameters or the extrinsic transformation between the camera and the IMU is of interest to this work. We calibrate the IMU unit with the camera C_2. We choose camera C_2 for calibration in order to be consistent with our ground truth coordinate system and the general approach of choosing a forward facing camera.

4.3. Ground truth quality evaluation

Acquiring ground truth information in an enclosed environment is a challenging step. The global accuracy of the ground truth solution is dependent on the availability of GNSS signals. In general, the strength and accuracy of GNSS signals are high in an open area, while poor signals are received in enclosed areas such as indoors, narrow city streets and forests. On the other hand, the local accuracy can be improved by fusing the information acquired through local sensing mechanisms such as IMU, Odometer, Radar, Lidar, Camera, etc. with the GNSS information for better results. As mentioned earlier, we utilize the NovAtel's PwrPak7^{TM} module to acquire a ground truth solution through a tightly coupled pose estimation framework that uses GNSS and IMU information.

To assure the readers of the quality of the ground truth, we provide the estimated position accuracy in the form of standard deviations for the positions at every timestamp for all sequences in Fig. 8. The graphs indicate the standard deviation in the estimated position in the North, East and Elevation/Height directions. The accuracy in North, East, and Elevations directly correspond to the accuracy in the local coordinate frame. The statistics in these figures are provided by the Inertial Explorer application used with the NovAtel's PwrPak7^{TM} module.

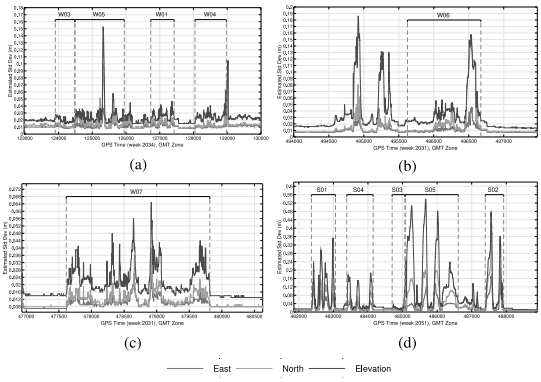


Fig. 8. Estimated position accuracy for ground truth poses (a) Winter sequences W01–W05 (b) Winter sequence W06 (c) Winter sequence W07 (d) Summer sequences S01–S05.

To facilitate readers, we show the range of each sequence in the figures. It can be observed from Fig. 8(a–c) that the average standard deviation for the winter sequences (W01, W03–W07) is lower than 2 cm for East and North with occasional larger deviations. The spikes in deviation are obtained where the vehicles traverse a narrow path with trees densely covering the area around it for a longer period. In all the sequences, the errors in the East axis are the lowest followed by errors in the North. The largest deviations are found in the elevation, which is typical of such a system.

On the other hand, the summer sequences (S01–S05) exhibit slightly larger standard deviations (see Fig. 8(d)). Except for S02, the errors for all the summer sequences in East and North are lower than 15 cm and 20 cm, respectively. As before, the largest deviation is observed in the elevation. This is in the sequence S05 with a value of 0.54 m. The deterioration of the GNSS performance for the summer sequence is logical. In contrast to the winter sequence, which was recorded in December 2018, the summer sequence was recorded near the springtime of May 2019. In the springtime, the GNSS results can be affected by the foliage which can cause 24 to 35% attenuation at L-band [24]. The contributing factor to the attenuation of the signals is the combined effect of signal absorption and scattering from the conglomeration of tree canopies and trunks. In winter, the sparsity of foliage in the tree canopy provides for a larger interval of non-attenuating space, while that advantage is lost in springtime in the presence of dense foliage [25]. In the absence of the GNSS signal, the ground truth pose estimation system relies more on the information provided by the IMU. Nonetheless, considering the task at hand, the results obtained for the summer sequence are good and provide a valid reference for experimentation.

5. Development and evaluation toolkit

The dataset is accompanied by a set of MATLAB scripts that can be used for processing of raw data or evaluating the odometry obtained from user's algorithm against the ground truth poses. The dataset includes ready to be used information for easy access

to the researchers. Nonetheless, we provide a set of MATLAB tools for processing the data. Each data sequence is accompanied by a set of raw images. The raw data is of interest to the researchers who wish to re-calibrate the cameras using the set of calibration images provided with the dataset with their own or different calibration algorithms. The new calibration can then be used to process the raw images of the dataset. MATLAB script *readRaw_Debayer.m* and *readRaw_Rectify.m* can be used to read the raw images from a folder and write the debayered and rectified images onto another directory, respectively. The debayered color images can then be used with the provided calibration data or any newly computed calibration data using the calibration images.

An evaluation script is also provided as part of the toolkit to assess the results. The MATLAB script *mainEvaluate.m* can be used to evaluate the obtained visual odometry poses against the ground truth poses. Prior to using the script, the directories for the text file with the ground truth poses and the self-computed poses should be specified. The evaluation script computes relative pose error (relative translation and rotation errors) and absolute trajectory error (ATE) for each sequence and the overall errors for all sequences. The core reason for selecting these metrics is that relative pose error provides a good analysis of the local accuracy of the trajectory over a fixed distance. Relative comparison over fixed distances can measure the effect of drift more effectively and provide a better response for visual odometry. On the other hand, ATE provides a more coherent and globally consistent comparison using the absolute distances between the corresponding ground truth poses and the poses estimated by the assessed system.

The ATE can be obtained by computing the absolute distance between the estimated and the ground truth trajectory. For global consistency, it is essential that both trajectories are in the same reference coordinate system. If that is the case, then the ATE can be computed directly, otherwise, the alignment can be calculated in the form of a transformation matrix $\mathbf{T}$ in closed-form using Umeyama's method [26]. It is to be noted that ATE only considers the translational errors. The commonly used form of ATE is given as follows [27]

$$ATE_{rmse} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|Tp_i - \hat{p}_i\|^2}, \tag{1}$$

where $T \in \mathrm{SE}(3)$ transforms the trajectory p_i to the coordinate system of the ground truth poses $\hat{p}_i$. Additionally, the mean, standard deviation, minimum, median and maximum errors can be computed to analyze the performance from different perspectives.

As mentioned earlier, relative errors can provide more accurate local information about visual odometry errors. Kümmerle et al. [28] proposed to compute relative error over an interval followed by an average over all these errors. The interval was selected based on fixed distance. This is a good approach, however, the trajectory and orientation errors are amalgamated and form a joint error metric. Geiger et al. [11] took this concept and isolated the rotation and translation part. This enabled them to compute the rotation and translation error independent of each other. The isolated relative translation error (RTE) and relative rotation error (RRE) are defined as follows

$$RTE(\tau) = \frac{1}{|\tau|}\sum_{(i,j)\in\tau} \|(p_j \ominus p_i) \ominus (\hat{p}_j \ominus \hat{p}_i)\|_2 \tag{2}$$

$$RRE(\tau) = \frac{1}{|\tau|}\sum_{(i,j)\in\tau} \angle\,[(p_j \ominus p_i) \ominus (\hat{p}_j \ominus \hat{p}_i)], \tag{3}$$

where the interval τ corresponds to the set of image frames (i, j) that cover a specific length in the trajectory and p_i and $\hat{p}_i$ are the estimated and ground truth poses, respectively. The symbol $\ominus$ denotes the inverse compositional operator explained in [28] and $\angle[\,]$ is the rotation angle for the rotation error.

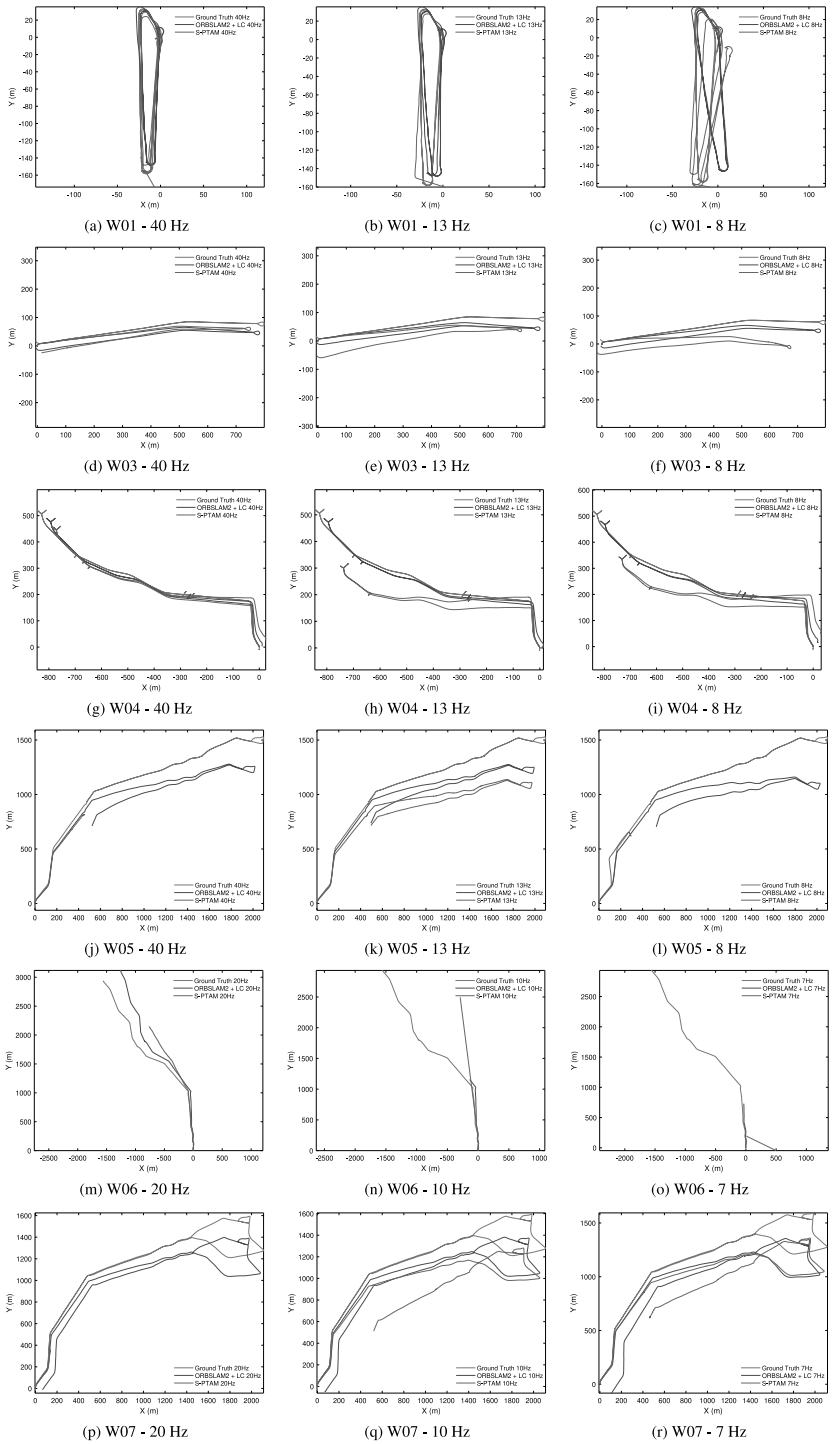
6. Benchmarking

In this section, we discuss the nature of the trajectories planned and traversed during the dataset recording. Furthermore, we provide experimental results of using state-of-the-art visual SLAM methods on the FinnForest dataset.

All sequences start from and end at the same location. Each trajectory has been recorded with an intent to tackle different conditions. The first route (W01 and S01), shown in Figs. 9(a–c) and 10(a–c), comprises a short ellipse shaped trajectory that offers two repeated loop closures while traveling in the same direction and a third loop closure from the opposite direction. The terrain is rather harsh and mimics the uneven ground traversed by work machines. The second sequence (S02), shown in Fig. 10(d–f), offers another loop based trajectory for SLAM approaches. Unlike W01 and S01, this path is traveled only once and therefore forming a single closed loop. Moreover, as mentioned before, no winter recordings are available for this trajectory due to route blockage. The remaining sequences are more visual odometry oriented sequences. These sequences do not offer loop closures by traveling in the same direction. However, the same routes are traversed from the opposite direction, hence, offering an opportunity to explore relocalization possibilities while traveling from the opposite direction. The third, fourth and fifth sequence routes offer short, medium and relatively long trajectories for estimating visual odometry. The third route (W03 and S03), shown in Figs. 9(d–f) and 10(g–i), is the shortest of visual odometry sequences and offers the simplest case for testing. The fourth route (W04 and S04), shown in Figs. 9(g–i) and 10(j–l), offers more of an exploration type of trajectory with back and forth driving to mimic investigative movements of work machines. The fifth route (W05 and S05), shown in Figs. 9(j–l) and 10(m–o), is relatively long and provides more of a challenging odometry test course. Two more visual odometry sequences are provided in the winter condition W06 and W07 (see Fig. 9(m–o) and (p–r)) that are recorded in night and dusk time, respectively. In our opinion, sequence 3–7 are helpful for improving the autonomy of heavy work vehicles in such environments. The sequences mimic the movements of heavy machines that are more fixated on the task at hand in an exploratory manner.

It is noteworthy that the area traversed is deliberately kept limited in terms of displacement from the starting point. Unlike urban infrastructure, forest covered routes provide limited chances to record loop closure over large distances. Recording large distances without loop closure does not suit visual SLAM approaches, therefore, we focused on maintaining short distances with more information in terms of frame rate for improved accuracy. Moreover, at the given framerate, the data recorded is significantly high for the route traversed during the recordings.

Among the state-of-the art visual SLAM implementations that rank high in the KITTI benchmarking suite [29], we chose ORB-SLAM2 [4] and Stereo-Parallel Tracking and Mapping (S-PTAM) [30]. These studies provide open-source implementation of a stereo based visual SLAM method which facilitates the testing phase of our study. It is important to note that both ORBSLAM2 and S-PTAM are used in their standalone mode in order to process all the frames. S-PTAM in specific was not able to process all the incoming frames in its native ROS mode, where it attempts to simulate time-constrained real-time scenario. The implementation was not able to keep up with the incoming frames using the given computational resources. As a consequence, some of the frames were dropped in the ROS mode. To provide a fair and



(a) W01 - 40 Hz (b) W01 - 13 Hz (c) W01 - 8 Hz

(d) W03 - 40 Hz (e) W03 - 13 Hz (f) W03 - 8 Hz

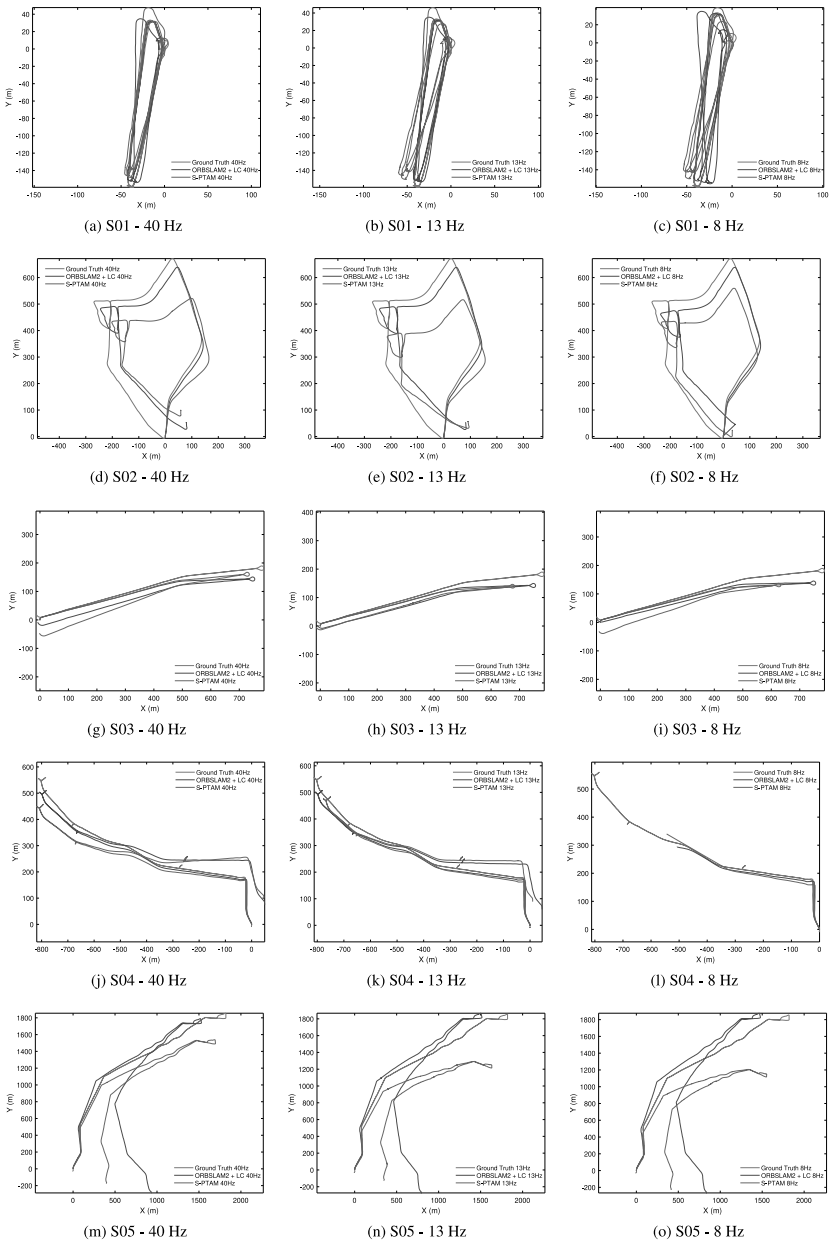
(g) W04 - 40 Hz (h) W04 - 13 Hz (i) W04 - 8 Hz

(j) W05 - 40 Hz (k) W05 - 13 Hz (l) W05 - 8 Hz

(m) W06 - 20 Hz (n) W06 - 10 Hz (o) W06 - 7 Hz

(p) W07 - 20 Hz (q) W07 - 10 Hz (r) W07 - 7 Hz

Fig. 9. Estimated trajectories plotted against the ground truth for the winter sequences in FinnForest dataset.



(a) S01 - 40 Hz (b) S01 - 13 Hz (c) S01 - 8 Hz

(d) S02 - 40 Hz (e) S02 - 13 Hz (f) S02 - 8 Hz

(g) S03 - 40 Hz (h) S03 - 13 Hz (i) S03 - 8 Hz

(j) S04 - 40 Hz (k) S04 - 13 Hz (l) S04 - 8 Hz

(m) S05 - 40 Hz (n) S05 - 13 Hz (o) S05 - 8 Hz

Fig. 10. Estimated trajectories plotted against the ground truth for the summer sequences in FinnForest dataset.

thorough comparison, we provide the results of both methods in their standalone mode with no time constrains for processing. In addition, S-PTAM was used without the loop closure capability due to compatibility issues of the implementation with new versions of dependencies. Except for the sequences with loops, the remaining majority visual odometry sequences should not be affected.

Nonetheless, ORB-SLAM2 and S-PTAM yield excellent results in a typical structured urban environment. These methods have been extensively tested in urban and indoors settings over KITTI, EuRoC, and Level 7 block-set datasets [4,30].

The results obtained with the aforementioned implementations over FinnForest dataset are plotted against the ground truth in Figs. 9 and 10 for all the sequences recorded with the

forward facing stereo pair (C_2-C_3). Thorough quantitative result is tabulated in Tables 2 and 3. For all these experiments, a standard laptop with an Intel Core i7 @ 1.90 GHz processor and 32 GB RAM was used.

The primary aim of testing the dataset with state-of-the-art method is to educate the readers about the challenges provided by the dataset. Large drift and scale errors are observed for the visual odometry sequences, compared to the sequences with loops, in spite of short distances being covered. We will discuss the results obtained from experimentation in more detail in the next section.

7. Discussion

In this section, we discuss the experimental results using the new dataset and state our observations. Our remarks are intended to aid further research and experimentation with the given data.

7.1. Feature tracking in FinnForest

The forest environment provides unique challenges for tracking features. Due to self-similar and repetitive patterns, extracting correct matches and maintaining tracking with a low number of feature points is tricky. To avoid any obvious obstacles towards tracking, we recorded the data at low driving speeds around 25–30 km/h and low exposure time for image acquisition to avoid motion blur. Following the recommendation in [21], we include the skyline in the scene which is expected to be useful for navigation and augments to reliable features for matching. Furthermore, the forest view near the skyline significantly adds to the rotation accuracy (especially yaw and pitch) by providing features that are far away from the camera.

In most of the testing cases, we used 2000 feature points to track with ORB-SLAM2 and 1000 feature points with S-PTAM. The number of feature points selected was a compromise between the image resolution and memory management of the implementation. However, we observed that the selected parameters were suitable for testing most of the sequences and provided sufficient cross over candidates between frames for matching. During experimentation, we observed that S-PTAM required more tuning of the parameters compared to ORB-SLAM2, in which they were kept mostly the same for all experiments. ORB-SLAM2 uses ORB features which are both faster and more robust (due to rotation invariance) compared to features used by S-PTAM. S-PTAM uses the GFIT feature and BRIEF descriptors for matching. The descriptor is not invariant to rotation and as a result, the implementation requires parameter adjustment for various sequences of FinnForest dataset to maintain tracking on the parts of the route with harsh terrain.

As mentioned earlier, unlike urban routes, the path traversed while recording the dataset is a rough terrain. The combined effect of erratic motion, speed and data sampling introduce challenges for testing. It is apparent from the experimental results that the highest errors are observed in the visual odometry sequences while the errors are reduced and distributed in the sequences where loop closure has been achieved.

Effect of data sampling on tracking: The sampling of the dataset at lower rates is intended to facilitate testing and investigate a suitable data rate for real cases. Though lower frames per second (fps) are advantageous for testing purposes, information processing at lower fps can considerably compromise the visual odometry pipeline during real field operation. To exemplify the behavior, we take the experimental results of ORBSLAM2 on S02 at 8 Hz. ORBLSLAM2 fails to continue its tracking of feature points when the vehicle hits a pothole and the scene observes a sharp motion. It is important to note, that ORBSLAM2 successfully completed the same test sequence at higher frame rates (13 and 40 Hz). For further investigation, we significantly changed the parameters by increasing the feature points to 5000 and varying the FAST feature threshold between 4 and 18. However, the result remained the same. Surprisingly, S-PTAM successfully completed the test sequence S02 at 8 Hz when the feature points to detect were set to 1500.

On the contrary, ORBSLAM2 was able to handle a similar situation in W01 at 8 Hz with loosened parameters while S-PTAM failed to continue the tracking. However, none of the implementations were able to successfully complete the sequences W05, W06, and S04. A similar effect was observed in W07 at 7 Hz and the parameters were loosened again. This time ORB-SLAM was able to successfully process the entire sequence while S-PTAM failed.

Effect of motion on tracking: In some cases, the erratic motion due to terrain in combination with the scene is already too much even at a higher frame rate. In the case of W01, we observed that S-PTAM fails to complete the sequence at all sampling rates. The implementation fails while locally adjusting the poses that lie in the range where the sharp movements are observed. On the other hand, ORBSLAM2 was able to process the sequence with relative ease at 40 and 13 Hz without fine-tuning of the parameters. However, at 8 Hz the feature points used for tracking were increased, and the feature threshold lowered to maintain tracking even with ORBSLAM2.

A different cause is expected to be affecting S-PTAM while processing W05 at 8 Hz. The tracking failure occurs when the vehicle slows down to a momentary stationery state and restarts motion. We believe the source of the issue is the predictive feature search that fails to find matches. In both S-PTAM and ORB-SLAM2, a motion model is used to predict the position of the map points on the latest image frame and find matches in the small neighborhood for tracking. In case, if the feature matches are not found in the small predicted neighborhood, ORBSLAM2 expands the search window as a fallback option. On the other hand, we believe, S-PTAM relies only on the decaying velocity model and does not expand its search neighborhood as a fallback option. As a result, a sudden change in velocity at a lower frame rate affects the tracking of feature points. This phenomenon is aggravated by the sub sampling since the same behavior is handled by S-PTAM at sampling of 13 Hz but fails at 8 Hz when change is more abrupt. By requesting more feature points to be detected in the new image frame, we can avoid the tracking failure altogether, however, poor matches with the map then lead to convergence issue in the local bundle adjustment step of S-PTAM.

Effect of illumination on tracking: The dataset includes various opportunities to test the robustness of visual SLAM implementation towards tracking and pose estimation in a scene with varying illumination. The notable opportunities regarding illumination change are provided by W07, S04, and W06. In W07, we observe gradual illumination change as it gets darker. The sequence was recorded at the dusk time and the illumination changes drastically between the start and end of the sequence. ORBSLAM2 did not face any issue in terms of tracking feature points in this sequence, however, S-PTAM faced considerable problems to maintain tracking at all sampling rates. S-PTAM also failed tracking at sampled data of 13 Hz, however, we have included the results since the failure point was close to the end of the sequence.

On the other hand, a more rapid change is observed in illumination due to direct sunlight in the sequence S04. At a sampling rate of 40 and 13 Hz, both ORBSLAM2 and S-PTAM can successfully process the sequence. However, at 8 Hz they fail at different points. The S-PTAM fails directly due to overexposure and flare observed in the scene while ORBSLAM2 fails due to fast erratic motion following the over-exposed scene in the recording.

Table 2
Quantitative results of ORBSLAM2 for the FinnForest dataset at different sampling rates.

Data Sampling	40/20 Hz			13/10 Hz			8/7 Hz		
Seq. No	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)
*W*01	3.35	2.1785	0.00014197	3.6738	2.3016	0.00019584	3.4914	2.4092	0.00021607
*W*03	12.266	9.1805	0.00012107	12.025	9.2299	0.00013344	12.249	9.1962	0.0001253
*W*04	17.421	7.7753	9.7778e−05	17.244	7.8332	9.935e−05	20.666	7.6746	0.00011482
*W*05	55.422	9.2678	0.0001298	56.323	9.4365	0.00013865	75.715	9.7977	0.00022451
*W*06[a]	21.789	32.14	0.00020608	TL	TL	TL	TL	TL	TL
*W*07[a]	37.933	7.2208	0.00011185	34.324	7.2193	0.00013786	48.88	7.2107	0.00016175
*S*01	4.3677	1.9672	0.00022474	3.8189	1.917	0.00019894	6.2793	2.1462	0.00027508
*S*02	26.132	4.2061	0.00017796	26.874	4.2181	0.0001728	TL	TL	TL
*S*03	12.633	5.873	0.00020877	10.986	5.6022	0.00018197	9.8899	5.5459	0.00018258
*S*04	30.053	5.5827	0.0001988	26.825	5.4608	0.00018299	TL	TL	TL
*S*05	228.88	9.4575	0.00025169	191.52	8.8165	0.00020505	200.81	8.9426	0.00021338

[a]Indicates that the data is subsampled at 20/10/7 Hz.
TL: Tracking lost.

Table 3
Quantitative results of S-PTAM for the FinnForest dataset at different sampling rates.

Data Sampling	40/20 Hz			13/10 Hz			8/7 Hz		
Seq. No	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)	ATE (rmse)	RTE (%)	RRE (deg/m)
*W*01	TL	TL	TL	TL	TL	TL	TL	TL	TL
*W*03	19.709	10.166	0.00011828	27.663	12.63	0.0004809	28.369	14.819	0.00063508
*W*04	25.852	9.4934	0.00014839	45.091	14.9	0.00071498	48.944	14.914	0.00073208
*W*05	TL	TL	TL	79.774	11.312	0.00011181	TL	TL	TL
*W*06[a]	TL	TL	TL	TL	TL	TL	TL	TL	TL
*W*07[a]	TL	TL	TL	102.54	8.319	0.00019895	TL	TL	TL
*S*01	7.3247	2.883	0.00018821	9.4022	4.0914	0.00066569	8.652	3.9342	0.00030787
*S*02	34.391	9.2735	0.0005317	44.68	11.63	0.00061402	34.752	9.2786	0.00020271
*S*03	21.779	7.0644	0.00025365	31.418	11.105	0.00025333	47.392	14.82	0.00031883
*S*04	31.891	7.1297	0.00023556	39.749	9.703	0.00019259	TL	TL	TL
*S*05	130.41	10.182	0.00022272	171.55	14.517	0.00032586	201.65	17.9	0.00038022

[a]Indicates that the data is subsampled at 20/10/7 Hz.
TL: Tracking lost.

The Night sequence, W06, is especially challenging for both implementations. Neither of the implementations could process the sequences under normal parameter settings. ORBSLAM2 was able to process the sequence at 40 Hz with relaxed parameters after the FAST feature threshold was reduced to 4 to avoid losing the track of features. S-PTAM is not able to process the W06 sequence at any sampling rate. Even after the feature threshold is reduced, S-PTAM fails to converge at local bundle adjustment. This is expected since the scene in view is limited to a few meters of the snow-covered road. As a result, the poses estimated do not agree over a longer duration and fail to converge at bundle adjustment.

7.2. Loop closure

The dataset provides three sequences with loop closure opportunities. Among these, S01 and W01 repeat the same route twice in one direction and the third time in the opposite direction. This means that ORBSLAM2 can identify the loop closure opportunity at any time of the second lap of the drive. During experimentation we observed that ORBSLAM2 successfully closes the loop and distributes the errors for the aforementioned sequences. In contrast, ORBSLAM2 fails to close the loop for the sequence S02, even though, enough overlap of the start and end scenes is provided. Oddly, ORBSLAM2 can re-localize itself at the end of the sequence S02 that is processed at 8 Hz after losing track of the feature points. A closure can be observed due to re-localization in Fig. 10(f) in the trajectory estimated by ORBSLAM2. We believe that sparser keyframes formed at 8 Hz provided more decisive information compared to the same sequence at higher fps, where the relocalization was not observed.

7.3. Drift

A drift in scale and rotation can be observed in the estimations provided by ORBSLAM2 and S-PTAM for all of the visual odometry sequences. This effect of drift becomes stronger as the sample rate drops down from 40 to 8 Hz. The effect is most apparent in S03 and W03 (see Fig. 9(d–f) and (g–i)).

7.4. Seasonal effect

Seasonal changes have an apparent effect on various aspects of this dataset. As discussed earlier, the ground truth accuracy reduced in the summertime compared to the wintertime due to considerably higher foliage effect in the summer. An added challenge from the perspective of recording was that, while traversing the forest, different parts of the forest provided different levels of shade from the sun due to the density of the trees in that specific part. This created a challenge to avoid over or underexposure of the scenes since we used a fixed aperture. These effects are more obvious in the sequence S04.

The winter sequences, on the other hand, were adequately exposed since most of the recordings are in overcast. In addition, there was enough texture on the ground due to tire tracks in the snow. ORBSLAM2 handled tracking very well with evenly distributed points on the snow-covered ground. S-PTAM focused more on the obvious texture from the trees. Most of the feature points from the snow-covered road are discarded by S-PTAM as false matches.

7.5. Effect of ground truth precision

It is important to note that the experimentation is independent of the precision level of the ground truth position since

the test algorithms did not use the IMU and GNSS information. However, the effect of the ground truth precision indeed has to be considered when comparing the experimental results against the ground truth poses. The ground truth precision for each sequence is shown using Fig. 8 and discussed in Section 4.3. In the context of benchmarking, we can say that we are more confident in the comparison performed in Tables 2 and 3 for the winter sequences (W01–W07) than the summer sequences (S01–S05) since the precision of ground truth position for winter sequences is comparatively higher. Nonetheless, the precision of the ground truth is high enough in both cases for valid analysis of visual odometry/SLAM algorithms.

It is important to remember that the visual odometry/SLAM algorithms may give different responses for the same trajectory recorded under different condition, as we discussed throughout Section 7. Therefore, arguing that one result is better than the other without comparing to the provided ground truth is not an objective conclusion.

8. Summary

In this paper, we have presented a novel dataset that offers a forest-like environment in various light and weather conditions for visual odometry and SLAM systems to process. The dataset provides synchronized and processed image frames from 4 cameras that can be used independently or as stereo pairs. Moreover, raw data is also provided to encourage further examination into the system. We believe this dataset will prove immensely useful towards enlarging the spectrum and diversity of the testing data for autonomous vehicles, especially, autonomous heavy work machines. We hope that this dataset will provide new challenges and inspire exploration of new possibilities for autonomous vehicles/machines.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This Work was partially supported by Business Finland, Finland under the project Spatial Sensing for Machines (SSFM, grant number 1822/31/2016). This work was carried out with the support of Centre for Immersive Visual Technologies (CIVIT) research infrastructure, Tampere University, Finland.

References

[1] R. Benenson, M. Omran, J. Hosang, B. Schiele, Ten years of pedestrian detection, what have we learned? in: European Conference on Computer Vision, Springer, 2014, pp. 613–627.
[2] A. Geiger, M. Roser, R. Urtasun, Efficient large-scale stereo matching, in: Asian Conference on Computer Vision, Springer, 2010, pp. 25–38.
[3] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[4] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. 33 (5) (2017) 1255–1262.
[5] J. Engel, V. Usenko, D. Cremers, A photometrically calibrated benchmark for monocular visual odometry, 2016, arXiv preprint arXiv:1607.02555.
[6] N. Carlevaris-Bianco, A.K. Ushani, R.M. Eustice, University of Michigan North Campus long-term vision and lidar dataset, Int. J. Robot. Res. 35 (9) (2016) 1023–1035.
[7] A.Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, K. Daniilidis, The multivehicle stereo event camera dataset: An event camera dataset for 3D perception, IEEE Robot. Autom. Lett. 3 (3) (2018) 2032–2039.
[8] H. Jung, Y. Oto, O.M. Mozos, Y. Iwashita, R. Kurazume, Multi-modal panoramic 3D outdoor datasets for place categorization, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2016, pp. 4545–4550.
[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.
[10] G. Pandey, J.R. McBride, R.M. Eustice, Ford campus vision and lidar data set, Int. J. Robot. Res. 30 (13) (2011) 1543–1552.
[11] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset, Int. J. Robot. Res. 32 (11) (2013) 1231–1237.
[12] W. Maddern, G. Pascoe, C. Linegar, P. Newman, 1 year, 1000 km: The Oxford RobotCar dataset, Int. J. Robot. Res. 36 (1) (2017) 3–15.
[13] Y. Choi, N. Kim, S. Hwang, K. Park, J.S. Yoon, K. An, I.S. Kweon, KAIST multi-spectral day/night data set for autonomous and assisted driving, IEEE Trans. Intell. Transp. Syst. 19 (3) (2018) 934–948.
[14] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, A. Kim, Complex urban dataset with multi-level sensors from highly diverse urban environments, Int. J. Robot. Res. 38 (6) (2019) 642–657.
[15] M. Ferrera, J. Moras, P. Trouvé-Peloux, V. Creuze, D. Dégez, The aqualoc dataset: Towards real-time underwater localization from a visual-inertial-pressure acquisition system, 2018, arXiv preprint arXiv:1809.07076.
[16] M. Miller, S.-J. Chung, S. Hutchinson, The visual–inertial canoe dataset, Int. J. Robot. Res. 37 (1) (2018) 13–20.
[17] A. Mallios, E. Vidal, R. Campos, M. Carreras, Underwater caves sonar data set, Int. J. Robot. Res. 36 (12) (2017) 1247–1251.
[18] K. Leung, D. Lühr, H. Houshiar, F. Inostroza, D. Borrmann, M. Adams, A. Nüchter, J. Ruiz del Solar, Chilean underground mine dataset, Int. J. Robot. Res. 36 (1) (2017) 16–23.
[19] P. Furgale, P. Carle, J. Enright, T.D. Barfoot, The Devon Island rover navigation dataset, Int. J. Robot. Res. 31 (6) (2012) 707–713.
[20] J. Bruce, J. Wawerla, R. Vaughan, The SFU mountain dataset: Semi-structured woodland trails under changing environmental conditions, in: IEEE Int. Conf. on Robotics and Automation 2015, Workshop on Visual Place Recognition in Changing Environments, 2015.
[21] J. Garforth, B. Webb, Visual appearance analysis of forest scenes for monocular slam, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 1794–1800.
[22] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000).
[23] P. Furgale, J. Rehder, R. Siegwart, Unified temporal and spatial calibration for multi-sensor systems, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 1280–1286.
[24] K.-D. Park, J. Won, The foliage effect on the height time series from permanent GPS stations, Earth Planets Space 62 (11) (2010) 849–856.
[25] J. Goldhirsh, W.J. Vogel, Handbook of Propagation Effects for Vehicular and Personal Mobile Satellite Systems, Vol. 1274, NASA Reference Publication, 1998, pp. 40–67.
[26] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, IEEE Trans. Pattern Anal. Mach. Intell. 13 (4) (1991) 376–380.
[27] M. Salas, Y. Latif, I.D. Reid, J. Montiel, Trajectory alignment and evaluation in SLAM: Horn's method vs alignment on the manifold, in: Robotics: Science and Systems Workshop: The Problem of Mobile Sensors, 2015, pp. 1–3.
[28] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, A. Kleiner, On measuring the accuracy of SLAM algorithms, Auton. Robots 27 (4) (2009) 387.
[29] The KITTI vision benchmark suite 2020, 2020, http://www.cvlibs.net/datasets/kitti/eval_odometry.php, Cvlibs.net.
[30] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, J. Jacobo Berlles, S-PTAM: Stereo parallel tracking and mapping, Robot. Autonom. Syst. (RAS) 93 (2017) 27–42, http://dx.doi.org/10.1016/j.robot.2017.03.019.

Ihtisham Ali received his B.Sc. in Mechatronics Engineering from the University of Engineering and Technology, Pakistan (2014) and his M.Sc. in Automation Engineering from Tampere University, Finland (2017). Currently, he is a doctoral researcher in 3D Media Group at Tampere University. He has worked on several industrial projects pertaining to machine automation using visual cues. His research interest is focused on computer vision and robotics specifically object pose estimation, 3D reconstruction, and visual SLAM.

Ahmed Durmush received the B.Sc. degree in Control Systems Engineering from Istanbul Technical University (2012). He is developing the systems required by the 3D Media group at CIVIT for light field capture and reconstruction and other research areas.

Olli Suominen graduated with both B.Sc. and M.Sc. (Tech) in Information Technology from Tampere University of Technology (2011/2012) with a major in Signal Processing. Since then, he has been a Ph.D. student in 3D Media Group at the Laboratory of Signal Processing at TUT and managing the construction and development of Centre for Immersive Visual Technologies. After starting from the B.Sc. thesis using only one camera (Depth Image Based Rendering) and an M.Sc. thesis using two cameras (Stereo Depth Estimation) he has now scaled up to 40 cameras for the Ph.D. with research interests in multi-camera systems, 3D reconstruction, multimodal sensor fusion, SLAM and light field capture. He currently focuses on applications in heavy mobile work machines, leading several industry driven research projects and developing relations with the industry.

Dr.Tech Jari Yli-Hietanen has background in Signal Processing research. He has worked with various topics including robotic cognition, machine vision and natural language processing. Currently, he is developing IT services for research support at Tampere University.

Sari Peltonen received the M.Sc. degree in mathematics from the University of Tampere, Finland, in 1996. She received the Ph.D. degree in signal processing from the Tampere University of Technology in 2000. She is currently University Lecturer in signal processing at the Unit of Computing Sciences in Tampere University. Her research interests include robust estimation, image processing and tomographic image reconstruction.

Jussi Collin is CEO of Nordic Inertial (JC Inertial Oy) and adjunct professor at Tampere University. He received the M.Sc. and Dr.Tech. degrees from the Tampere University of Technology, Tampere, Finland, in 2001 and 2006, respectively, specializing in inertial navigation algorithms. His research interests are in modern machine learning methods and their industrial applications in the field of inertial sensing.

Atanas Gotchev received his M.Sc. degrees in radio and television engineering (1990) and applied mathematics (1992), his Ph.D. degree in telecommunications (1996) from the Technical University of Sofia, and the D.Sc.(Tech.) degree in information technologies from the Tampere University of Technology (2003). He is a Professor of Signal Processing and Director of the Centre for Immersive Visual Technologies at Tampere University. His recent work concentrates on the algorithms for multi-sensor 3-D scene capture, transform-domain light-field reconstruction, and Fourier analysis of 3-D displays.

PUBLICATION
V

Bi-directional loop closure for visual SLAM

I. Ali, S. Peltonen, and A. Gotchev

arXiv:2204.01524

Bi-directional Loop Closure for Visual SLAM

Ihtisham Ali*, Sari Peltonen, Atanas Gotchev

Faculty of Information Technology and Communication Sciences, Tampere University, Finland.

Abstract

A key functional block of visual navigation system for intelligent autonomous vehicles is Loop Closure detection and subsequent relocalisation. State-of-the-Art methods still approach the problem as uni-directional along the direction of the previous motion. As a result, most of the methods fail in the absence of a significantly similar overlap of perspectives. In this study, we propose an approach for bi-directional loop closure. This will, for the first time, provide users with the capability to relocalise to a location even when traveling in the opposite direction, thus significantly reducing long-term odometry drift in the absence of a direct loop. We devise a technique to select training data from large datasets in order to make them usable for the bi-directional problem. The data is used to train and validate two different machine learning models for loop closure detection and subsequent regression of 6-DOF camera pose between the views in an end-to-end manner. The outcome packs a considerable impact and aids significantly to real-world scenarios that do not offer direct loop closure opportunities. We provide a rigorous empirical comparison against other established approaches and evaluate our method on both outdoor and indoor data from the FinnForest dataset and PennCOSYVIO dataset.

Keywords: Bi-directional, loop closure, relocalisaion, pose regression, deep learning, siamese network, autonomous driving, mobile robotics, field robotics

1. Introduction

Inferring where you are on a map, of your local world, is a core problem of mobile robotics [1], navigation [2], and augmented reality [3]. This problem is widely known as the lost or kidnapped robot [4]. A potential solution is Localisation which refers to the process of recognizing a previously visited place and determining your current pose w.r.t the previous pose from the visual scene representation [5]. Loop closure is a specific case of localisation which is essential to attain robust navigation in any intelligent transportation system as it aids in significantly reducing the accumulated errors during visual navigation [6]. Traditionally, loop closure is detected in an environment that has been previously viewed from a similar perspective e.g., a vehicle traveling toward the north passes by the same location moving in the same direction. Such a configuration maximizes the chances of place recognition. Generally, a binary feature descriptor in conjunction with Bag-of-Words (BoWs), or a deep learning-based approach is used to tackle this problem, with the latter approach exhibiting better performance after the advent of modern Convolutional Neural Network (CNN) architectures. In this paper, we propose an approach to expand the capability of loop closure detection methods towards bi-directional problems. Our proposed approach is able to correctly recognize previously visited places and find the relative pose irrespective of the direction of motion of the vehicle.

Traditionally, the place recognition problem has been approached in a similar manner as the Image Retrieval problem [7, 8]. In general, a query image, whose location needs to be estimated, is compared against a large geo-tagged database of images from previous visits. Each image is represented as an aggregate of numerous local invariant features. The state-of-the-art method still relies on feature detectors and descriptors such as SIFT [9], ORB [10], SURF [11], etc., that are used to extract local information from an image and accumulated into a single feature vector for an entire image using encoded representation through methods such as bag-of-visual-words [12], vector of locally aggregated descriptors (VLAD) [13] or Fisher vector [14]. Fisher Vector adopts the Gaussian mixture model (GMM) to build a visual word dictionary. As a result, Fisher Vector encodes more image information than BoW and at times outperforms BoW in some computer vision tasks. In contrast, VLAD is a simplification of Fisher Vector and provides a trade-off between performance and computational efficiency. In most cases, VLAD performs similarly to Fisher with better efficiency. These methods aid in compressing the image representation and subsequent efficient retrieval of a match from the database [15]. A popular approach based on the aforementioned concepts is FAB-MAP [16] which learns a generative model for the BoW data. The model observes and learns the co-occurrence of appearance words from common objects that are likely to appear or disappear together thus providing valuable probabilistic information. However, FAB-MAP proves to be computationally expensive due to its complex methodology

*Corresponding author

Email address: ihtisham.ali@tuni.fi, ihtishamaliktk@gmail.com (Ihtisham Ali)

[1]Faculty of Information Technology and Communication Sciences, Tampere University, Finland.

for image description and matching.

On the other hand, recent studies have shown that the deployment of CNNs results in significant improvement in accuracy and reduction in complexity for localisation. The models trained on very large datasets significantly outperform the local descriptors such as SIFT in a variety of applications such as object and scene recognition [17]. McManus et al. [18] proposed to learn features from image patches and called them scene signatures. These scene signatures were for matching and retrieving scenes under varying appearance changes. However, the approach required a considerably more careful training phase with data of the test environment under all possible environmental conditions. Some studies directly opt for using the intermediate representations which are learned using object recognition dataset and use them for scene identification [19, 20]. Sunderhauf et al. [20] propose the use of features from intermediate layers to form a descriptor for matching. Features from higher layers of a CNN encode semantic information about the place while features from the lower layer encode more descriptive information about the geometry of the scene. The authors experiment with varying combinations of these descriptions and attempt to find the nearest neighbor based on the cosine distance between the feature vectors of the query and the database.

It is noteworthy that all the aforementioned studies targeted uni-directional or traditional loop closure cases. The closest work done to bi-directional loop closure is [21]. The authors claim to target the problem of bi-directional loop closure in panoramic images. In our humble opinion, the use of panoramic images diminished the complications of the problem by providing roughly similar views to a uni-directional case. The panoramas are captured in an enclosed structural environment with a circular trajectory. As a result, the motion in the reverse direction captures a considerable overlap of the forward motion scenes with some spatial offset in images and only marginal difference in perspectives. This can also be observed from the illustrations in the study which exhibit only spatial changes in the scene. Moreover, the study reports that traditional methods such as FAB-MAP fail to close the loop in practice even in these panoramic images. To the best of our knowledge, this is the only study that targets place recognition and loop closure while moving in the opposite direction and high perspective change.

In our work, we introduce a novel automated technique to leverage the use of existing large datasets for training CNNs towards the task of bi-directional loop closure. This is essential since acquiring new data every time requires considerable time and resources. Then, we present two machine learning models based on CNN that are assigned to first identify potential candidates for loop closure between the query and database and subsequently regress the pose between the matched candidates. The proposed model for place recognition uses a VGG-16 as the base CNN topped with a neural network implementation of VLAD known as NetVLAD [22]. The model learns to recognize places in an end-to-end manner on the training data specifically prepared for the bi-directional loop closure task. Afterwards, the matches are fed to a siamese network with a VGG-16 base model topped with fully connected layers that are regularized with dropout layers. The 6-DoF pose regression in the proposed bi-directional case is significantly more challenging compared to traditional cases and is performed with an independent model. We employ two public datasets namely FinnForest Dataset [23] and PennCOSYVIO dataset [24] to conduct our tests for place recognition and pose regression. The generalization is successfully tested on unseen data thus exhibiting strong comprehension of the visual cues by the model and not just scene memorization.

The article is organized as follows: In Section 2, we give out the system overview and formulate the proposed approaches for place recognition and pose regression tasks. Moreover, we also discuss in detail, the data preparation steps involved in this study for the task at hand. Section 3 provides the experimental results and a comprehensive comparison with other well-established methods on two challenging datasets. Finally, Section 4 concludes the article.

2. System Overview

In this section, we present the overall pipeline of the proposed approach for localisation. The approach constitutes of two modules: a siamese CNN network [25] with triplet structure for maximizing similarity learning and a bi-input siamese model for 6 DoF relative camera pose regression. The overall pipeline is shown in Figure 1. Initially, images are used from the database to train the siamese with a triplet structure for place recognition tasks by learning to identify maximum similarity. Each trained branch of the network is essentially a feature encoder and the extracted feature vectors can be employed to identify matches from the database of images that are nearest neighbors (NN) to the query image in the feature space. Afterward, samples with true positive matches are fed into the independently trained network for pose regression to estimate the relative pose between true matches. The processes are comprehensively explained in the following sections.

2.1. Place Recognition

Feature Encoding We adopt a siamese approach for the task of place recognition, as illustrated in Figure 1. The network constitutes of a base CNN model that takes three inputs, namely query image sample I_q, positive image sample I_p, and negative image sample I_n from the database I_D. These input images are pre-processed based on the prescribed pre-processing technique adopted for the base model. Here, we take VGG-16 as an example which will be mainly used in our work; however, we also provide results with other base models for comparison later in the study. VGG-16 takes an input image of 224x224 pixels and propagates it through five sets of convolution and pooling layers, where the layers are connected through Rectified Linear Unit (ReLU) as an activation function. Each layer in the network learns a further abstraction of the input data with the highest-level abstraction residing towards the last layers. The structure is essential since features from the higher layers of the CNN hierarchy encode abstract semantic information about the scene, while features from intermediate and lower layers

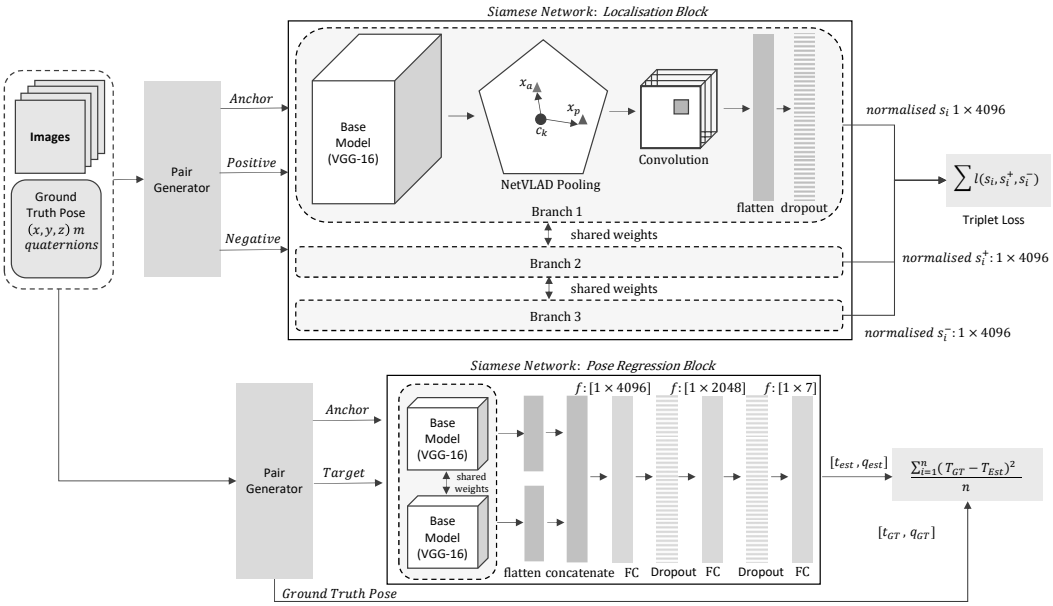


Figure 1: Illustration of the overall system pipeline. A siamese network constituting a VGG-16 base model topped with a VLAD pooling layer is used to learn similarity in the scenes using a triplet loss. Once the training process is completed, we employ the branch as a descriptor to compute feature representations of database and query images for image retrieval towards localisation. The pose regression network (lower) is independently trained to directly regress the 6-DoF relative camera poses between the query and the retrieved match from the database.

encode finer details from the image such as a change in appearance and structure [20].

The outputs of the base model are normalized and fed into the neural network form of VLAD descriptor known as NetVLAD pooling layer [22]. Essentially, VLAD encodes information about the statistics of local descriptors aggregated over an image in the form of feature distance from a cluster centre. For N D-dimensional local image descriptors $\vec{x_i}$ as input, and K cluster centres (visual words) c_k as VLAD parameter, the output VLAD image representation V is $K \times D$-dimensional. The L2-Normalized vector form of V with elements (j, k) is

$$V(j,k) = \sum_{i=1}^{N} a_k(\vec{x_i})(x_i(j) - c_k(j)). \quad (1)$$

Here, $x_i(j)$ and $c_k(j)$ are the j-th dimensions of the i-th descriptor and k-th cluster centre, respectively. $a_k(\vec{x_i})$ indicates whether the descriptor $\vec{x_i}$ belongs to the k-th visual word, i.e. it is 1 if $\vec{x_i}$ belongs to the cluster c_k and 0 otherwise.

To develop a layer reactive to training via backpropagation, it is required that the layer's operation is differentiable with respect to all its parameters and the input. The original relation lacks this differentiation due to the binary nature of $a_k(\vec{x_i})$. To overcome this issue NetVLAD re-writes the original relation as:

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^T \vec{x_i} + b_k}}{\sum_{k'} e^{w_{k'}^T \vec{x_i} + b_{k'}}}((x_i(j) - c_k(j)). \quad (2)$$

where w_k and b_k are sets of trainable parameters for each cluster k which are learned in an end-to-end manner during training. Conceptually, the weight that the descriptor $\vec{x_i}$ is assigned to the cluster c_k proportional to their proximity. Moreover, the relative proximity to other cluster centres also plays a part in the relation.

For our study, we found emperically that 64 clusters and 512-dimensional VGG16 backbone work effectively for the localisation task. The NetVLAD feature vector dimension becomes $512 \times 64 = 32,768$. We further extract principle components through a convolution block and retrieve the encoded description as a normalized feature vector.

Loss Function The similarity in an image is learned by employing a triplet loss over the output of each branch of the triplet siamese. For training, we gather training sample set S such that

$$S = \{(s_i, s_i^+, s_i^-) | (s_i, s_i^+ \in S^+); (s_i, s_i^- \in S^-), i = 1, ..., M\}. \quad (3)$$

Here, S^+ refers to the set of relevant image pairs, S^- refers to negative image pairs, and M indicates the span of the entire triplet set. The triplet loss is then given as

$$\ell(s_i, s_i^+, s_i^-) = max(0, m + \left\| f(s_i) - f(s_i^+) \right\|_2^2 - \left\| f(s_i) - f(s_i^-) \right\|_2^2).$$

Here, margin m is a scaler that defines an offset between positive and negative pairs, and $f(.)$ is an embedding of an image sample. The global loss over all triplet samples is given as

$$L = \sum_{(s_i, s_i^+, s_i^-) \in \boldsymbol{S}} \ell(s_i, s_i^+, s_i^-). \tag{4}$$

Retrieving the nearest neighbours To retrieve a potential match for a query image from the database of images, both images must have a suitable representation before comparison. In the proposed case, we use one branch of the fine-tuned network as a feature extractor to encode the query and database images. This enables us to have the representation in the same embedding space (i.e. 4096-dimensional feature vectors, see Figure 1). In the experimentation section, we will use other methods of feature extraction to encode our images for the sake of comparison. Finally, the top N-ranked database images, $d = (d_n | d_n \in D, n = 1...N)$ are selected as the nearest neighbors to the query image based on the squared Euclidean distance in the embedding space. It is important to note that a query image might have one, many, or no match in the database as it depends on the number of keyframes generated during earlier exploration of the environment.

Neighbour Confidence Sharing Place recognition is a critical task for loop closure in visual SLAM. The problem can become significantly more challenging when the environment contains repetitive textures even for distinct locations such as in forests and large open areas. It is often the case that wrong matches are generated due to similar semantics of different scenes. To overcome this problem, we propose a confidence sharing scheme where the confidence of the previously localized points is propagated to their neighbors in a causal manner. In our case, we consider three neighbors for sharing confidence. We incorporate the traveled distance between the neighbors in order to ascertain the sanity of a potential match for a query point. A new query point has a valid match if (1) it has a considerable match score (in embedding space from the model) with an image from the database and (2) it has nearby localized neighbors that agree in distance traveled with the estimates from odometry. If a new neighbor is found far away from a nearby localized neighbor and the odometry estimates run in favor of the previously localized neighbor, then the new match is discarded as a possible wrong match.

2.2. Pose Regression

The pose estimator block is composed of a VGG-based siamese architecture that takes two monocular images as raw input and predicts a 6-DoF relative transformation between the poses for those specific inputs. The siamese regression block is shown in Figure 1. The shared weights are initialized with a network pre-trained for large-scale place classification task [26] using the Places 365 dataset, and later fine-tuned for the relative pose estimation task as described below. The output of each branch is vectorized and combined into a single encoded description. The relative pose is regressed by passing the feature vector through three fully connected (FC) layers activated through Leaky ReLu functions and intermediate dropout layers for regularization, as shown in Figure 1. The final FC layer gives out the predicted relative pose. Different studies adopt different representations for the angles. In study [27], the authors use deeply learned key points to estimate the Fundamental matrix between the two images using two CNN modules. In a similar study titled UnDeepVO [28], the authors opt for direct image alignment in conjunction with the camera intrinsic to estimate the relative pose. The relative pose is obtained as a decoupled combination of a translation vector and rotation vector in Euler angles.

Euler angles carry briefer representation compared to the fundamental matrix, however, it suffers from discontinuities in the form of gimbal lock. On the other hand, rotation parameterization such as rotation matrices that lie on a manifold, their distance computation requires finding a Euclidean embedding. In our work, we represent the angles using quaternions similar to the work [29]. It is important to note that quaternions lie on a unit sphere, however, during optimization/training the difference becomes so small that the distinction between spherical distance and Euclidean distance becomes insignificant. Therefore, to avoid obstructing the optimization with unnecessary additional constraints, we avoid the use of spherical geometry. Hence, the distance between two quaternions can be measured by the Euclidean l2 norm $\|q_{GT} - q\|$. The authors of the popular study PoseNet [30] and its derivative study [29] advocate using a decoupled approach with a weighted parameterization of the angle, with a scale factor β, to balance the loss function

$$L = \|\Delta t_{GT} - \Delta t\|_2^2 + \beta \|\Delta q_{GT} - \Delta q\|_2^2 . \tag{5}$$

During experimentation, we observed that the approach seemed cumbersome as the vale of β has to be manually adjusted for each dataset. The authors of [30] remark that the value of β can lie anywhere in the range of 120 to 2000 depending on the structure and semantics of the scene [30]. To avoid this issue we propose to discard the scale factor β and independently scale down the entire translation vector and quaternions to the same range during the preprocessing step of dataset preparation. Since we wish to use an adaptation of ReLu activation for the FC layer, it is advisable to rescale both the quaternions and the translation vectors between $[0, 1]$. This makes the relation invariant to any scale factor for the training phase. The scale factors can be extracted from the range of the data in the dataset and applied using the following relation

$$d_{scaled} = \frac{(sc_{max} - sc_{min}) * (d - d_{min})}{d_{max} - d_{min}}. \tag{6}$$

where d denotes the data array and sc indicates the scaler values of the desired range for scaling. The model is trained to predict an arbitrarily scaled version of the pose where the scale is restored in a post-processing step after the prediction. The MSE is then given as

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (T_{scaled\ GT} - T_{Est.})^2. \tag{7}$$

Here, $T_{scaled\ GT}$ is the pose constituting scaled $[t_x, t_y, t_z]$ and

scaled $[q_w, q_x, q_y, q_z]$. $T_{Est.}$ is a similar vector to $T_{scaledGT}$ which is predicted by the model. At test time, a pair of images are fed into the regression model, consisting of two branches, which directly estimates the relative camera pose vector. Finally, the estimated quaternion and translation vectors are scaled up to retrieve real-world values.

2.3. Data Selection and Dataset Preparation

Visual SLAM is a widely researched problem and many datasets exist for testing purposes [31, 32, 33, 34, 35]. However, most of the datasets do not provide bi-directional motion since they are tailored for handling the problem from a uni-directional perspective. For our work, we found that FinnForest dataset [23] and parts of PennCOSYVIO dataset [24] can be used for training and testing purposes.

As indicated by the name, the FinnForest dataset provides data, for visual odometry and SLAM, in a forest landscape. The dataset provides recordings from four RGB cameras that are synchronized with an Inertial Measurement Unit (IMU), and a Global navigation satellite system (GNSS). The dataset contains sequences for odometry that are well suited for this study. Each route is traveled from both directions within the same sequence thus providing all the relevant data for bi-directional loop closure. The dataset is challenging for the problem since it contains repetitive texture, unlike an urban landscape that provides more distinct landmarks over its trajectory. We will only utilize the data recorded in the summer conditions in our study since it offers slightly more landmarks than the winter condition for the localisation block.

The second dataset that we use for training and testing offers indoor data. This was specifically chosen to check the performance of our approach in both indoor and outdoor environments. The dataset records similar sequences with multiple configurations and types of sensors. We found four sequences, recorded with GoPro Hero 4 Black, that are reasonably well suited for the task of bi-directional loop closure. The sequences include slow and fast motions that can represent vehicular motion using a forward-facing camera. Moreover, the data includes ground truth poses that can be used for automatic extraction of training and testing samples in our approach. Other sequences in this dataset are wall-facing and more suited for Structure from Motion applications.

Both of the datasets are passed through a data preparation phase in order to generate sequences that can be used for training our localisation and pose regression models. We generate sub-datasets out of the original datasets and use them for training. Since the localisation and the pose regression are to be performed on the same scenes we can use the data generated for localisation in the pose regression block. For simplification, we split and consider two cases of the bi-directional localisation problem. Assume that a route is traversed in a straight line from both directions then we have images acquired with a camera at somewhat regular intervals from both directions for roughly the same location, given that the camera frame rate is high enough.

Considering the forward motion case, in Figure 2, an anchor sample is acquired at query location (green). For these anchor samples, we can obtain positive sample pairs from nearby locations that share the perspective view. Moreover, it is fairly safe to assume that an image acquired further away or from far back will provide a significantly different view and can be selected as negative samples (red) for the localisation model training. For all the sample pairs we compute the relative ground-truth poses in the form of translation vector and quaternion angles which are later used in the pose regression block. We generate 6 sample triplets for each query location, however, any number of triplets can be generated according to the needs of the task. The straight-line route shown in Figure 2 is a simplified case, however, we should expect irregular movements and changes in angles in real cases. The FinnForest dataset attempts to mimic the conditions that a heavy vehicle might face in a real forest during its operation. We expect close-by samples from a query point that might not share the same visual information due to sharp turns, camera jitter, motion blur, sudden overexposure, or sun flare in the camera view. These samples can deteriorate the learning performance of the model as it expects them to be positive samples however they no longer share the semantic and/or geometric information with the query sample.

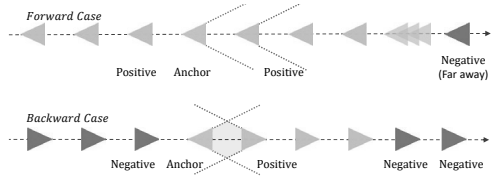


Figure 2: Illustration of training data selection based on distance and direction. The cones represent camera body placed at various locations. The shaded area in the backward case represents potentially overlapping regions in the perspective view

To overcome this, we leverage structure from motion to autonomously generate training triplets by validating the previously created triplet samples. We generate 3D world points from a query stereo pair and track the corresponding key points among all the positive samples for that specific query sample. Similarly, the 3D points are also propagated between the camera frames using the 6 DoF ground-truth poses. The transformed world points are then projected into corresponding image space. Any sample with cumulative reprojection error (for the tracked keypoints) higher than a threshold is discarded. The reprojection error is computed using

$$E_{px} = \|P_{s_i^+} - \Pi(K, [q_{(s_i, s_i^+)}\ ,\ {}_{s_i}t^{s_i^+}]_{HT}, W_{s_i})\|_2^2. \tag{8}$$

This validation step gets rid of the sample pairs with high angular changes in perspective (such as in the case when the vehicle is turning). Here, Π is the perspective projection function that projects the 3D points $W = (X, Y, Z, 1)^T$ from world frame space to image space using the camera intrinsic K. The superscript T indicates the transpose of a vector. The perspective projection yields $\tilde{x} = (\tilde{u}, \tilde{v}, 1)^T$ in the image space of the camera at the pose of interest. The reprojected points $\tilde{x}$ are

compared directly against the observed/tracked 2D points (P) in the corresponding sample image. The symbol $[\,]_{HT}$ indicates the conversion from quaternion q and translation vector t to the homogeneous transformation matrix. We use quaternion angle representation for the sake of coherence.

For the backwards case, we use the same query image samples initially selected and filtered for the forward case and attempt to find pairs for it in the backward motion part of the sequence. In contrast to the forward case, the backward motion case can have positive samples only ahead of the query location (see Figure 2). All the potential samples are expected to be oriented in the direction opposite to the camera orientation at the query point. Moreover, the assumption is that camera poses that are slightly ahead of the query point would share potentially more of the same scene even if from the opposite perspective. The similarity in the scene in this small range is what we want our model to learn and discriminate. A camera pose that is too far ahead or at the back of the query pose would have little and no match with the query perspective, respectively. As before the positive samples are indicated in blue while negative samples are shown in red. The reprojection-based verification is not possible for the backward case since the traditional feature detector cannot detect and track features with such high perspective changes. As a result, there are no reference key point image positions for the reprojected 3D points.

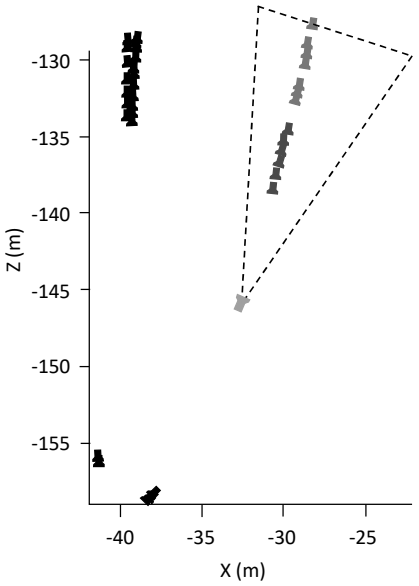


Figure 3: Visualization of the automatic sample selection scheme for the case of backward motion. Valid training samples within the field of view of the anchor/query pose (green) are shown in blue while the negative samples are shown in red. The poses shown in black indicate rejected samples in the vicinity of the query.

A visualization of the automatic sample selection, based on the camera poses, from a sequence of FinnForest dataset is shown in Figure 3. Samples within the field of view of the query pose are considered as potential candidates for triplet grouping. Similarly, we discard very close samples since target samples that are too close will share very little view with the query perspective. Here, the camera poses are shown that pass the constraints set on the field of view, distance from the query pose, and the tolerance of orientation difference from the query pose orientation. The candidates for positive samples are shown in blue while the negative samples are shown in red. Additionally, some camera poses that do not pass the constraints are visualized in black for the sake of understanding.

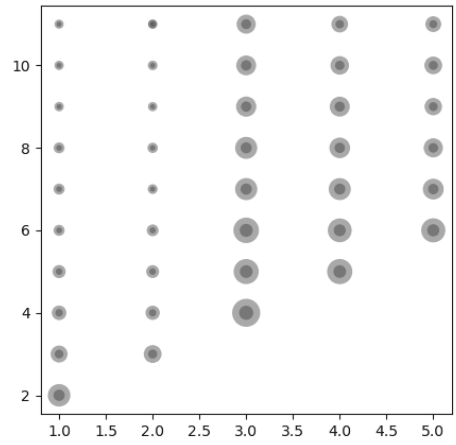


Figure 4: Upto scale visualization of the training loss (purple) and generalization gap (indigo) when the training data is changed for the backward case. The horizontal axis indicates the minimum valid distance (in meters) i.e, the distance from query to the nearest sample while the vertical axis indicates the maximum valid distance (in meters) i.e, the distance from query to the furthest valid sample. We indicate the best result with a green overlay.

It is difficult to conclude from merely visually observing the data as to what should be the minimum and maximum distance between the query and the positive samples. To understand the relationship we follow an empirical approach and train the model with different data distributions. In Figure 4, we explain the effect of data distribution. The figure shows an Upto scale visualization of the training loss (purple) and generalization gap (indigo) for the backward case. The horizontal axis indicates the minimum valid distance i.e, the distance from the query to the nearest sample while the vertical indicates the maximum valid distance i.e, the distance from the query to the furthest valid sample. We observed that the training loss and the generalization gap were minimum for the training data when the nearest positive sample was limited to a distance of 2 meters and the farthest sample was kept to be at 11 meters from the query pose. This indicates that the maximum overlap is found within this range for the given data and that anything out of this range is a potential outlier to what the model attempts to learn. This is obviously specific to the datasets in this study and might vary slightly depending upon the camera and optics used for recordings.

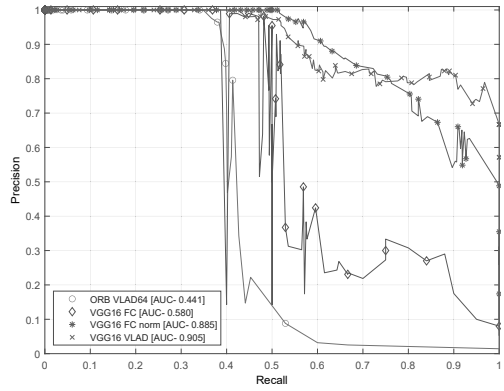
3. Results and Discussion

In this section, we provide our experimental results and quantitatively demonstrate the effectiveness of the proposed system on the FinnForest and PennCOSYVIO datasets. To ascertain the generalization capability of our pipeline, on data previously unseen during training, we hold out one of the scenes in the FinnForest dataset (S5) and PennCOSYVIO dataset (C2-bs) for evaluating and training our model on the remaining scenes. We will discuss the results in the forthcoming subsections and compare our results with other well-established methods. The network models were implemented with the Tensor Flow framework using Keras API. We employ the Adam optimizer to train the network with an early stop. The network setup preferred a small learning rate that started from .0000001 and decreased by one-tenth every epoch.

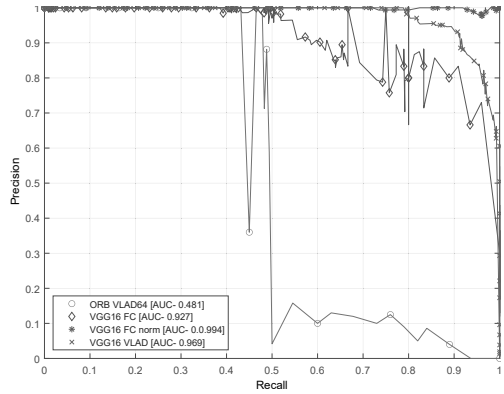
3.1. Place Recognition

For testing place recognition capability, we compare our proposed method with three other approaches to gauge the relative quality of the results. Among these methods, VGG-FC and VGG FC-norm are variants of deep learning approaches where we deploy two fully connected layers with dropouts applied after the VGG-16 network. VGG FC-norm has an additional normalization layer before the feature encoded vectors are extracted from the network. The third approach which we term here as ORB-VLAD uses ORB feature detector and descriptor to encode keypoints from the images and uses VLAD to further re-encode and reduce the dimensionality of the feature vectors. The use of VLAD helps us to have a more direct comparison with our proposed approach since we employ a variant of VLAD known as netVLAD.

For both datasets, the localisation performance is expressed in the form of a Precision-Recall (PR) curve. The results are shown in Figure 5. It can be observed from the results that the proposed approach VGG16-VLAD and VGG16-FCnorm outperform VGG16-FC and ORB-VLAD. The difference between the area under the curve (AUC) for VGG16-VLAD and VGG16-FCnorm in case of both datasets is almost the same. Nonetheless, we remark that VGG16-VLAD is more suited for the task at hand. Our proposition is based on the observation that VGG16-VLAD performs better on the FinnForest dataset which is considerably more challenging compared to the PennCOSYVIO dataset. FinnForest dataset is recorded over a significantly larger spatial area which has repetitive textures and fewer discriminative landmarks. On the other hand, the PennCOSYVIO dataset offers the same indoor scene in all sequences where the route is the same and motion speed is slightly varied. This means we can expect a high correlation in the training and testing data in the case of PennCOSYVIO dataset. In contrast, the route and scene are varied in the FinnForest dataset which will result in a lower correlation between training and testing data, and higher data center distribution (in space that houses encoded data clusters). Hence, we can infer that VGG16-VLAD has better generalization capability compared to the other methods. It is important to mention that we also tested the SURF and SIFT features in combination with VLAD.



(a)



(b)

Figure 5: Precision-recall curves for bi-directional loop closures in the (a) FinnForest dataset and (b) PennCOSYVIO dataset.

However, the results were not included as they were poor and inhibited the readability of the PR curve. These classical feature descriptors work well for the uni-directional cases where the perspective does not change a lot, however, they fail to perform well in the bi-directional cases.

To gain a better understanding of what the network has learned and what it sees in an image, we overlay the activation maps on their corresponding images for visual observation. The activation maps for the forward motion case are shown in Figures 6 and 8 for FinnForest and PennCOSVIO datasets, respectively. Similarly, the activation maps for the motion in the opposite direction (bi-directional case) are shown in Figures 7 and 9. For the forward motion case, it can be observed that almost the same regions are activated in the query and positive test image pairs, which is an intuitive conclusion. In contrast, the activation maps are flipped left-right in the backward motion case for the query image and the positive sample pair. This effect makes sense since the images are acquired from the opposite directions for roughly the same location. This flip effect is

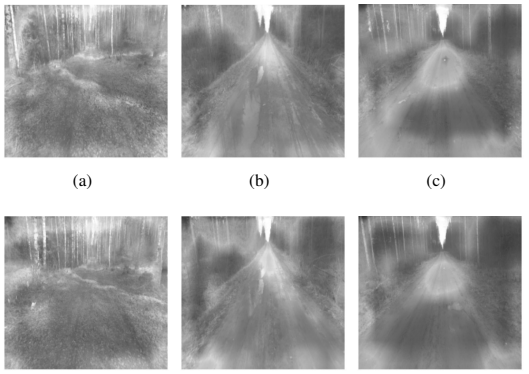


Figure 6: Activation maps over sample image triplet used for testing from FinnForest Dataset for forward/uni-directional case, where maps in (a-c) are for query images, (d-f) are for corresponding (column wise) positive samples.

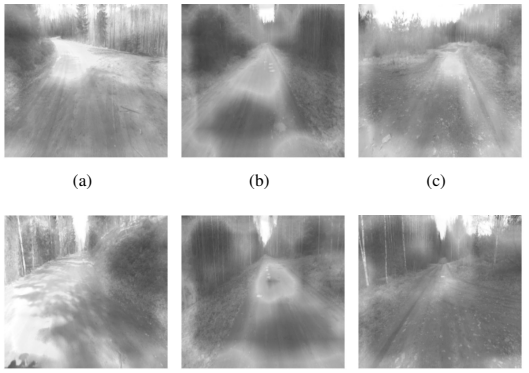


Figure 7: Activation maps over sample image triplet used for testing from FinnForest Dataset for bi-directional motion case, where maps in (a-c) are for query images, (d-f) are for corresponding (column wise) positive samples

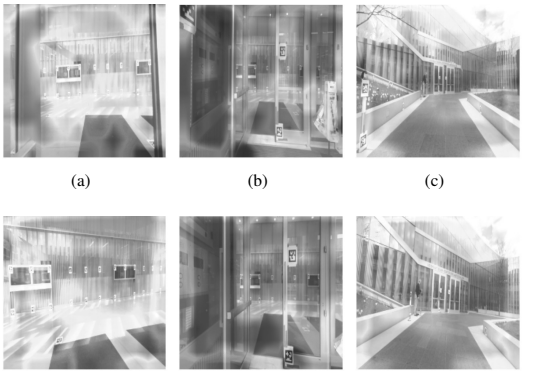


Figure 8: Activation maps over sample image triplet used for testing from PennCOSYVIO Dataset for forward/uni-directional motion case, where maps in (a-c) are for query images, (d-f) are for corresponding (column wise) positive samples.

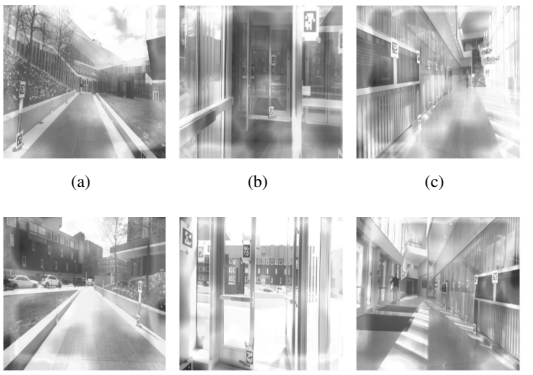


Figure 9: Activation maps over sample image triplet used for testing from PennCOSYVIO Dataset for bi-directional motion case, where maps in (a-c) are for query images, (d-f) are for corresponding (column wise) positive samples.

dominant and easily observed from image pairs in Figure 7.a,d and 7.b,e. Moreover, the regions closer to the camera exhibit stronger activation compared to the regions that are far away. All these observations are in agreement with our hypothesis formulated in Section 2.3 that motivated the study.

PR curves are a good metric for binary classification and to understand the overall performance of a system. However, to fully prove that our method is capable of accurately closing loops in practice, we perform loop closure offline in a causal manner. For this experiment, we simulate keyframe selection for loop closure at regular intervals based on the distance traveled. A loop closure candidate is detected, if the score of the query and a keyframe from the database is above a priori threshold τ and if it has passed the confidence sharing criterion among its neighboring localized keyframes. We eliminate the most recent images from the search space and wait until the database is large enough to start loop detection. The values of τ for the experiments shown in Figures 10 and 11 were selected from Figure 5 such that the recall rate is maximized with good precision returns. A slightly higher threshold was selected to illustrate all the possible outcomes in the experiments. As illustrated, a search can result in no match, a single unique match, multiple valid matches, a valid match with one or many false positives, or an invalid match. Multiple matches can be observed when the query images are acquired in open spaces and the scene does not change much among subsequent keyframes. In some cases, a keyframe can find a true and false positive at the same time due to visual similarity between multiple keyframes. This is more apparent in the FinnForest case where we observe quite frequent repetitive textures in the trees and on the road. Nonetheless, these false positives and invalid

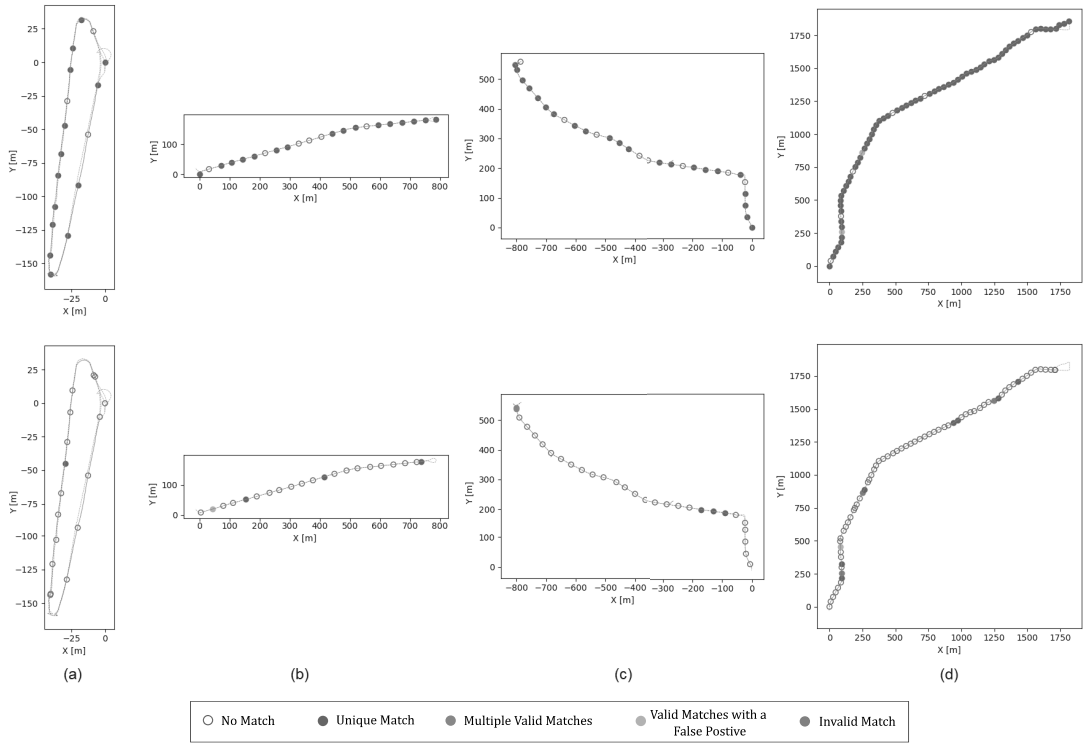


Figure 10: illustration of localisation results for FinnForest dataset. The top row shows the results for detections from the forward pass while the lower row shows the results for localisation from the backward pass.

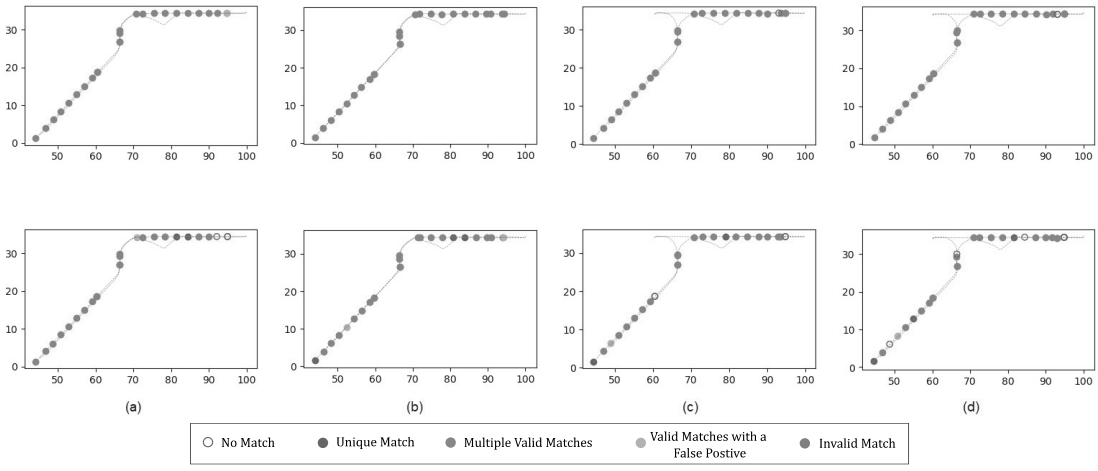


Figure 11: illustration of localisation results for PennCOSYVIO dataset. The top row shows the results for detections from the forward pass while the lower row shows the results for localisation from the backward pass.

matches can be removed with the confidence-sharing approach between the neighboring keyframes.

It can be observed that the matches for the forward motion cases are significantly higher than the reverse case for the FinnForest. This makes sense since the forward motion provides more opportunities to observe similar scenes. In contrast, it is

difficult to localize in the case of motion in the opposite direction for the FinnForest dataset as most of the scenes do not provide distinct landmarks within that short spatial window of observation. Nonetheless, the model was able to successfully identify the absence of matches and found enough matches that could be used to significantly improve odometry results. Even one valid match is enough to drastically improve odometry results and mitigate the errors due to drift that are accumulated in the odometry results.

On the other hand, the model works very well on the PennCOSYVIO dataset since the indoor constrained environment provides distinct visual landmarks that are specific to their corresponding locations. Extrapolating on this observation, we can postulate that the approach would be effective in outdoor scene recorded within a city environment that provides distinct landmarks for bi-directional localisation. As mentioned earlier, we were not able to use the existing visual odometry datasets that were recorded in an urban environment as they are tailored for uni-directional odometry and SLAM purposes.

3.2. Pose Regression

Earlier in Section 2.2, we discussed the pose regression block in detail and stressed the necessity of pose estimation through end-to-end learning. Here, we provide the experimental results of the proposed approach and compare the results against alternative approaches. The experimental results are tabulated in Tables 1 and 2 for FinnForest and PennCOSYVIO datasets, respectively. For comparison, we test the network by replacing the base model (VGG16) with Resnet50. Moreover, the relative impact of weight initialization is also studied by initializing the base models with weights acquired from models previously trained on ImageNet and Places 1365 (an extension of the Places 365 dataset) for classification tasks.

For both datasets, the pose regression performance is measured as the absolute difference between the predicted and ground truth values for the location (in meters) and orientation (in degrees). Similar to localisation, we test the performance of the pose regression network and state the results on individual sequences and the combined case. The combined testing results are effectively the average of the individual results. The number of test image samples and the spatial extent of the area where each sequence was recorded are also mentioned.

It can be observed that the network that has VGG16 as the base model and initialized with the weights of Places 1365 yields the best results followed by VGG16 initialized with ImageNet. This improvement was observed since Places 1365 incorporates visual scenes for scene classification that are quite relevant to our localisation task. In contrast, ImageNet is a more diverse dataset that is tailored for object classification. As a result, initialization with Places 1365 aids our network to generalize better to landscapes. On the other hand, Resnet50 performed poorly for both datasets. It is important to remember that the task at hand is localisation and not visual odometry. The pose regression is aimed at finding the relative pose between the query and a potential match for localisation. Traditional methods fail when we consider bi-directional cases of localisation. The results obtained for bi-directional pose regression in this study match the performance of other state-of-the-art approaches that are reported in studies conducted for uni-directional loop closure [30, 29].

Table 1: Comparison of pose estimation results from the regressor model trained on FinnForest dataset.

Sequence	Test Samples	Spatial Extent (m)	Resnet50 Imagenet	VGG-Imagenet	VGG-Places 1365
S1	2044	47 x 193	5.38m, 1.02°	2.42m, 0.352°	2.26m, 0.3°
S3	2706	800 x 190	5.26m, 1.00°	2.38m, 0.33°	2.31m, 0.29°
S4	3566	812 x 568	5.68m, 1.06°	2.54m, 0.39°	2.36m, 0.32°
S5	8866	1826 x 1883	7.35m, 0.84°	3.35m, 0.44°	3.23m, 0.53°
Combined	17182	2633 x 2014	5.92m, 0.98°	2.67m, 0.38°	2.54m, 0.36°

Table 2: Comparison of pose estimation results from the regressor model trained on PennCOSYVIO dataset.

Sequence	Test Samples	Spatial Extent (m)	Resnet50 Imagenet	VGG-Imagenet	VGG-Places 1365
C2-af	3361	144 x 36	3.79m, 0.71°	1.51m, 0.21°	1.35m, 0.22°
C2-bs	3330	144 x 36	3.85m, 0.72°	1.51m, 0.20°	1.33m, 0.21°
C2-bf	3090	144 x 36	3.81m, 0.73°	1.49m, 0.19°	1.36m, 0.22°
C2-bs	3375	144 x 36	5.75m, 0.80°	2.20m, 0.40°	1.81m, 0.26°
Combined	13156	144 x 36	4.3m, 0.74°	1.68m, 0.25°	1.46m, 0.22°

4. Conclusion

The article presents a learning-based approach to solve the problem of bi-directional loop closure in monocular images. We segregate the tasks of localisation into place identification and pose regression and solve them in two end-to-end deep learning steps. We demonstrate that it is indeed possible to achieve bi-directional loop closure on monocular images by carefully posing the problem and leveraging the training data for the networks. Moreover, we demonstrate that the networks generalize well and aim for learning the geometric and spatial relations in images rather than memorize the scenes/locations. This is validated by the performance of the model on unseen data. We compare the proposed approach with other deep learning methods and classical approaches and demonstrate superior performance for localisation. We provide both qualitative and quantitative results to corroborate the claim. A natural extension of the work would be to extend the case scenarios and test the approach with more datasets.

References

[1] H. Zhang, M. Niu, X. Chen, J. Wu, Y. Zhang, C. Liu, Target recognition and localization based on lightweight single-shot multibox detector network for robotics, Journal of Electronic Imaging 31 (6) (2022) 061803.

[2] X. Meng, C. Fan, Y. Ming, Y. Shen, H. Yu, Un-vdnet: unsupervised network for visual odometry and depth estimation, Journal of Electronic Imaging 28 (6) (2019) 063015.

[3] H. Liu, G. Zhang, H. Bao, Robust keyframe-based monocular slam for augmented reality, in: 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, 2016, pp. 1–10.

[4] I. Bukhori, Z. H. Ismail, Detection of kidnapped robot problem in monte carlo localization based on the natural displacement of the robot, International Journal of Advanced Robotic Systems 14 (4) (2017) 1729881417717469.

[5] L. G. Camara, L. Přeučil, Visual place recognition by spatial matching of high-level cnn features, Robotics and Autonomous Systems 133 (2020) 103625.

[6] D.-W. Shin, Y.-S. Ho, E.-S. Kim, Loop closure detection in simultaneous localization and mapping using descriptor from generative adversarial network, Journal of Electronic Imaging 28 (1) (2019) 013014.

[7] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, in: European conference on computer vision, Springer, 2014, pp. 584–599.

[8] A. Gordo, J. Almazán, J. Revaud, D. Larlus, Deep image retrieval: Learning global representations for image search, in: European conference on computer vision, Springer, 2016, pp. 241–257.

[9] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[10] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International conference on computer vision, Ieee, 2011, pp. 2564–2571.

[11] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Computer vision and image understanding 110 (3) (2008) 346–359.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: 2007 IEEE conference on computer vision and pattern recognition, IEEE, 2007, pp. 1–8.

[13] R. Arandjelovic, A. Zisserman, All about vlad, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2013, pp. 1578–1585.

[14] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE transactions on pattern analysis and machine intelligence 34 (9) (2011) 1704–1716.

[15] H. Jegou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE transactions on pattern analysis and machine intelligence 33 (1) (2010) 117–128.

[16] M. Cummins, P. Newman, Fab-map: Probabilistic localization and mapping in the space of appearance, The International Journal of Robotics Research 27 (6) (2008) 647–665.

[17] A. R. Memon, H. Wang, A. Hussain, Loop closure detection using supervised and unsupervised deep neural networks for monocular slam systems, Robotics and Autonomous Systems 126 (2020) 103470.

[18] C. McManus, B. Upcroft, P. Newman, Scene signatures: Localised and point-less features for localisation, Robotics: Science and Systems X (2014) 1–9.

[19] Z. Chen, O. Lam, A. Jacobson, M. Milford, Convolutional neural network-based place recognition, arXiv preprint arXiv:1411.1509 (2014).

[20] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, M. Milford, On the performance of convnet features for place recognition, in: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2015, pp. 4297–4304.

[21] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, S. Gámez, Bidirectional loop closure detection on panoramas for visual navigation, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings, IEEE, 2014, pp. 1378–1383.

[22] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, Netvlad: Cnn architecture for weakly supervised place recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5297–5307.

[23] I. Ali, A. Durmush, O. Suominen, J. Yli-Hietanen, S. Peltonen, J. Collin, A. Gotchev, Finnforest dataset: A forest landscape for visual slam, Robotics and Autonomous Systems 132 (2020) 103610.

[24] B. Pfrommer, N. Sanket, K. Daniilidis, J. Cleveland, Penncosyvio: A challenging visual inertial odometry benchmark, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 3847–3854.

[25] S. K. Roy, M. Harandi, R. Nock, R. Hartley, Siamese networks: The tale of two manifolds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3046–3055.

[26] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE transactions on pattern analysis and machine intelligence 40 (6) (2017) 1452–1464.

[27] Y.-Y. Jau, R. Zhu, H. Su, M. Chandraker, Deep keypoint-based camera pose estimation with geometric constraints, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 4950–4957.

[28] R. Li, S. Wang, Z. Long, D. Gu, Undeepvo: Monocular visual odometry through unsupervised deep learning, in: 2018 IEEE international conference on robotics and automation (ICRA), IEEE, 2018, pp. 7286–7291.

[29] Z. Laskar, I. Melekhov, S. Kalia, J. Kannala, Camera relocalization by computing pairwise relative poses using convolutional neural network, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 929–938.

[30] A. Kendall, M. Grimes, R. Cipolla, Posenet: A convolutional network for real-time 6-dof camera relocalization, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2938–2946.

[31] J. Engel, V. Usenko, D. Cremers, A photometrically calibrated benchmark for monocular visual odometry, arXiv preprint arXiv:1607.02555 (2016).

[32] N. Carlevaris-Bianco, A. K. Ushani, R. M. Eustice, University of michigan north campus long-term vision and lidar dataset, The International Journal of Robotics Research 35 (9) (2016) 1023–1035.

[33] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, K. Daniilidis, The multivehicle stereo event camera dataset: An event camera dataset for 3d perception, IEEE Robotics and Automation Letters 3 (3) (2018) 2032–2039.

[34] H. Jung, Y. Oto, O. M. Mozos, Y. Iwashita, R. Kurazume, Multi-modal panoramic 3d outdoor datasets for place categorization, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, pp. 4545–4550.

[35] W. Maddern, G. Pascoe, C. Linegar, P. Newman, 1 year, 1000 km: The oxford robotcar dataset, The International Journal of Robotics Research 36 (1) (2017) 3–15.