This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.
ON ACCURACY, ROBUSTNESS AND RELIABILITY
OF LASER-BASED LOCALIZATION

SIMONA NOBILI

Doctor of Philosophy
Institute of Perception, Action and Behaviour
School of Informatics
The University of Edinburgh

2018
Simona Nobili

On Accuracy, Robustness and Reliability of Laser-based Localization,

Doctor of Philosophy, 2018

SUPERVISORS:

Dr. Maurice Fallon
Prof. Barbara Webb
To my brother, my inspiration, Giacomo.

To my brave horse, my passion, Asia.
Localization is a task in robotics which involves determining where a robot is located with respect to a known reference. The capability of localizing itself is fundamental for an autonomous robot to be able to reach a desired location using mapping and motion planning.

The objective of the research in this thesis is to enable accurate, robust and reliable localization of wheeled and legged robots in real world challenging scenarios, for example during search and rescue missions in disaster zones or inspection of industrial areas. These applications require robots to autonomously navigate across rough terrain, traverse obstacles, detect and manipulate objects, similar to what humans can do.

In Part i a method for accurate and robust laser-based localization was proposed. The method allows operation in real-world situations with noisy sensor feedback, low frequency lasers and the presence of clutter and occlusions in the environment. Furthermore, a multisensor fusion approach was developed, and integrated within the control-loop of a quadruped robot during dynamic locomotion and operation in sensor impoverished situations.

In Part ii a geometry-driven approach to predict unsuccessful laser-based localization and prevent system’s failures was proposed. This is important for the reliability of a system in real-world situations where the success of localization can be compromised due to occlusions or constrictions in the environment.

In Part iii the applicability of deep learning to a localization problem using laser was investigated. The interest of this research lies in exploring whether this approach can be beneficial in robotics applications by providing better generalizability to different environments and sensors.
Legged robots are expected to demonstrate autonomous skills in situations which are not suitable for wheeled platforms, such as cluttered disaster areas and outdoor trails.

State estimation strategies for legged robots are typically based on inertial sensing and leg kinematics, and are affected by continuous drift. In a process called localization, perception sensors such as cameras or laser scanners can be used to compensate for this drift via scene registration.

The objective of this dissertation is to develop laser-based localization techniques which are suitable for the safe and continuous operation of wheeled and legged robots in real-world environments. We explore approaches fusing inertial, kinematics, stereo vision and laser signal sources in various combinations, in order to enable state estimation in the presence of challenges such as limited sensor field-of-view, varying lighting conditions, occlusions and non-uniform spatial overlap during motion.

A number of novel contributions are presented, including: a method for laser-based localization which is robust to large variations in spatial overlap and initial alignment error, a state estimation system which fuses multiple sensor sources in a manner which is suitable for a quadruped’s feedback controller, a module which can predict localization failure. The final contribution, explores the applicability of deep learning techniques to laser odometry.

We carry out extensive experimental evaluations running real-time in closed loop control on the NASA Valkyrie humanoid robot and the IIT quadruped HyQ, while exploring a variety of test environments. We run further experiments using a dataset collected with the Boston Dynamics humanoid Atlas during the DARPA Robotics Challenge finals, as well as on two mobile robots, using ours and publicly available datasets.
The hardest thing is to go to sleep at night, when there are so many urgent things needing to be done.

A huge gap exists between what we know is possible with today’s machines and what we have so far been able to finish.

— Donald E. Knuth

ACKNOWLEDGEMENTS

This thesis is the result of years of passionate studies and research towards a contribution to the world of robotics. The unique opportunity which I have been given to develop this research around real full-sized robots is priceless, and gave me additional motivation to work with deep dedication towards robotic systems which are functional despite some of the challenges occurring in the real-world.

My PhD journey would have not been the same without the people who accompanied me along the way.

I would like to express my deep gratitude to Dr. Maurice Fallon for his supervision and support. He continually and convincingly guided me on the right path forward. Thanks to IPAB and Prof. Barbara Webb for the availability demonstrated and their support. A particular thanks to the ORI members, for being enthusiastic colleagues and friends.

Above everything, I owe an enormous debt of gratitude to my beloved parents Francesco and Lorena, for their unconditional support and encouragement in the toughest moments, to my brother Giacomo for being my brother and friend, to my grandparents who have been waiting for me day by day, and to my closest and faithful friends Erica and Silvia for not forgetting.

Last but not least, I want to give my sincere thanks to Francesco, my travel buddy, first colleague and great friend since many years ago, to Nived, my wisest friend and guide, to Thomas, who has always been so distant and yet so close to me.
DECLARATION

I declare that this thesis was composed by myself, that the work contained herein and the included publications are my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Edinburgh, 2018

Simona Nobili
August 31, 2018
# CONTENTS

1 INTRODUCTION

1.1 Motivation and Requirements ........................................ 2
1.2 Contributions ....................................................... 5
1.3 Outline ............................................................. 6

2 BACKGROUND

2.1 Proprioceptive State Estimation ................................. 9
  2.1.1 Tools Overview .............................................. 9
    2.1.1.1 Inertial Navigation .................................. 9
    2.1.1.2 Wheel/Leg Odometry ................................. 10
  2.1.2 State Estimation for Multi-Legged Robots .................. 11
  2.1.3 State Estimation for Bipedal Robots ....................... 12

2.2 Exteroceptive State Estimation ................................. 12
  2.2.1 Tools Overview .............................................. 12
    2.2.1.1 Visual Odometry ..................................... 13
    2.2.1.2 Iterative Closest Point .............................. 13
  2.2.2 Scene Registration Methods ................................. 14

2.3 Multisensor State Estimation ................................. 17
  2.3.1 Tools Overview .............................................. 17
    2.3.1.1 Filtering ............................................. 17
    2.3.1.2 Smoothing ............................................ 18
  2.3.2 Review on Heterogeneous Sensor Fusion .................... 18

2.4 Reliability during Pose Estimation ............................. 20
  2.4.1 Tools Overview .............................................. 21
    2.4.1.1 Principal Component Analysis ....................... 22
    2.4.1.2 Support Vector Machines ........................... 22
  2.4.2 Methods to Evaluate Reliability ............................ 22

2.5 Thesis Overview .................................................. 23
  2.5.1 Preliminary Tools ........................................... 24
  2.5.2 Experimental Platforms and Sensors ....................... 27
I  ACCURACY AND ROBUSTNESS

3  OVERLAP-BASED ICP TUNING FOR ROBUST LOCALIZATION

3.1  Introduction .................................................. 32
    3.1.1  Leveraging Kinematic-Inertial State Estimation ............... 34
    3.1.2  Integrating a Laser Correction .......................... 35
    3.1.3  Impact of Registration Computation Time .................. 35
    3.1.4  Main Assumptions .................................. 37

3.2  Publication: ICRA 2017 ....................................... 37

3.3  Additional Results ......................................... 47

3.4  Conclusion and Future Work ............................... 48
    3.4.1  Limitations of the FOV-based Overlap Parameter .......... 49
    3.4.2  Limitations of the Sensor .............................. 51

4  HETEROGENEOUS SENSOR FUSION FOR ACCURATE STATE ESTIMATION
    OF DYNAMIC LEGGED ROBOTS

4.1  Introduction ................................................. 52
    4.1.1  Multisensor Fusion .................................... 53
    4.1.2  Laser-based Localization .............................. 54
    4.1.3  Filter State and Error Propagation ...................... 54
    4.1.4  Main Assumptions .................................. 55

4.2  Publication: RSS 2017 ....................................... 55

4.3  Conclusion and Future Work ............................... 66
    4.3.1  Applications: Motion Planning on Challenging Terrain .... 66

II  RELIABILITY

5  PREDICTING ALIGNMENT RISK TO PREVENT LOCALIZATION FAILURE

5.1  Introduction ................................................ 71
    5.1.1  Main Assumptions .................................. 72

5.2  Publication: ICRA 2018 ....................................... 73

5.3  Conclusion and Future Work ............................... 82
    5.3.1  Formulating the Overlap Metric using Octrees ............. 83
    5.3.2  Considerations relating to the Formulation of Alignability .. 84
    5.3.3  Applications: Leveraging Alignment Risk for Loop Closure De-
            tection .............................................. 85
III END-TO-END LEARNING:
TOWARDS GENERALIZABILITY 87

6 LASER ODOMETRY USING A DEEP NEURAL NETWORK 88
6.1 Introduction .................................................. 88
6.2 Related Work ................................................ 90
   6.2.1 Geometry-driven Laser Odometry .................. 90
   6.2.2 Advent of Machine Learning and Application to Robotic Navi-
gation ........................................................... 91
      6.2.2.1 Tools Overview .................................. 91
      6.2.2.2 Literature ........................................ 93
      6.2.2.3 Applications ...................................... 93
   6.2.3 End-to-End Learning Applications .................. 94
   6.2.4 Learning from Laser Point Clouds .................. 95
6.3 Approach ....................................................... 97
   6.3.1 Input Representations .................................. 98
   6.3.2 Output Representations ................................ 99
   6.3.3 Geometric Loss Function ............................ 101
6.4 Experimental Evaluation .................................... 102
   6.4.1 Training and Testing ................................. 104
   6.4.2 Results ................................................. 105
6.5 Conclusion and Future Work .............................. 112

7 CONCLUSIONS 115

A APPENDIX – OTHER CONTRIBUTIONS 118
   A.1 Direct Visual SLAM Fusing Proprioception .......... 118
   A.2 Seeing the Wood for the Trees ....................... 119

B APPENDIX – KALMAN FILTERS 122
   B.1 Kalman Filter ............................................. 122
   B.2 Extended Kalman Filter ................................ 123

C APPENDIX – EVALUATION METRICS 124
   C.1 Drift per Distance Travelled ........................... 124
   C.2 Root-Mean-Square Error ............................... 125

BIBLIOGRAPHY 126
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Experimental platforms and scenarios</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Iterative Closest Point error metrics</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Carnegie Robotics Multisense SL sensor</td>
<td>29</td>
</tr>
<tr>
<td>2.3</td>
<td>Velodyne HDL-64E sensor</td>
<td>29</td>
</tr>
<tr>
<td>2.4</td>
<td>Boston Dynamics Atlas robot</td>
<td>30</td>
</tr>
<tr>
<td>2.5</td>
<td>NASA Valkyrie robot</td>
<td>30</td>
</tr>
<tr>
<td>2.6</td>
<td>IIT HyQ robot</td>
<td>30</td>
</tr>
<tr>
<td>2.7</td>
<td>Clearpath Husky robot</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Kinematic-inertial pose estimator drift</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Scene registration problem</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>World to laser frame kinematics chain on Valkyrie</td>
<td>36</td>
</tr>
<tr>
<td>3.4</td>
<td>Registration computation costs</td>
<td>37</td>
</tr>
<tr>
<td>3.5</td>
<td>Registration performance with varying overlap in urban scenario</td>
<td>48</td>
</tr>
<tr>
<td>3.6</td>
<td>Closed-loop validation of AICP on Valkyrie</td>
<td>49</td>
</tr>
<tr>
<td>3.7</td>
<td>Closed-loop validation of AICP on Valkyrie – Staircase</td>
<td>50</td>
</tr>
<tr>
<td>3.8</td>
<td>FOV-based overlap limitations</td>
<td>51</td>
</tr>
<tr>
<td>4.1</td>
<td>Visual challenges on HyQ</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Closed-loop state estimator validation on HyQ</td>
<td>67</td>
</tr>
<tr>
<td>4.3</td>
<td>Closed-loop state estimation in a realistic scenario</td>
<td>67</td>
</tr>
<tr>
<td>4.4</td>
<td>Estimator performance with confidence intervals</td>
<td>68</td>
</tr>
<tr>
<td>4.5</td>
<td>Application: motion planning using HyQ’s state estimator</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Examples of unconstrained geometry</td>
<td>72</td>
</tr>
<tr>
<td>5.2</td>
<td>Proposed localization system</td>
<td>82</td>
</tr>
<tr>
<td>5.3</td>
<td>Octree-based overlap metric</td>
<td>83</td>
</tr>
<tr>
<td>5.4</td>
<td>Example of underestimated volume of overlap</td>
<td>84</td>
</tr>
<tr>
<td>5.5</td>
<td>Application: loop closure detection using alignment risk</td>
<td>86</td>
</tr>
<tr>
<td>6.1</td>
<td>Comparing laser odometry architectures</td>
<td>89</td>
</tr>
<tr>
<td>6.2</td>
<td>Deep Laser Odometry architecture</td>
<td>98</td>
</tr>
<tr>
<td>6.3</td>
<td>Point cloud top-view representation</td>
<td>100</td>
</tr>
</tbody>
</table>
Figure 6.4 Point cloud front-view representation .................................. 101
Figure 6.5 DLO training and validation losses ........................................... 105
Figure 6.6 DLO hyperparameters tuning ..................................................... 105
Figure 6.7 Odometry performance using different approaches .................. 107
Figure 6.8 Relative $x$ and yaw errors versus displacement ..................... 109
Figure 6.9 Example of failed ICP alignment .............................................. 110
Figure 6.10 Estimated odometry trajectories using different methods .......... 111
Figure 6.11 Estimated odometry trajectories on additional sequences ........ 113
Figure 6.12 Estimated odometry trajectories on challenging sequences ......... 114
Figure A.1 SLAM fusing proprioception on Valkyrie ............................... 119
Figure A.2 Place recognition in natural environments ............................... 121

LIST OF TABLES

Table 1.1 Thesis overview ................................................................. 5
Table 2.1 Summary of point cloud registration approaches ..................... 17
Table 2.2 Summary of state estimation approaches ................................ 21
Table 2.3 Overview of the tools used in the thesis .................................. 24
Table 2.4 Typical proprioceptive pose estimation performance ................ 28
Table 3.1 Characteristics of our sensors ............................................... 47
Table 4.1 Advantages and limitations of estimation approaches ................ 66
Table 6.1 DLO: CNN configuration ....................................................... 97
Table 6.2 Sequences of the KITTI dataset used for our experiments ........... 103
Table 6.3 Summary of odometry estimation accuracy ............................... 106

ACRONYMS

2D two-dimensional
3D three-dimensional

AICP Auto-tuned ICP

BA bundle adjustment

BICP Baseline ICP

CNN convolutional neural network

CoM centre of mass

DARPA Defense Advanced Research Projects Agency

DDT Drift per Distance Traveled

DLO Deep Laser Odometry

DLS Dynamic Legged System

DoF degree-of-freedom

DRC DARPA Robotics Challenge

DRS Dynamic Robot Systems

DVO Deep Visual Odometry

EKF extended Kalman filter

FAST Features from Accelerated Segment Test

FOG Fibre Optic Gyroscope

FOV field-of-view

FOVIS Fast Odometry from VISion

FPGA Field-Programmable Gate Array

GPS Global Positioning System

GICP Generalized-ICP

HyQ Hydraulic Quadruped
ICP  Iterative Closest Point
IF  information filter
IHMC  Institute for Human and Machine Cognition
IIT  Istituto Italiano di Tecnologia
IMU  Inertial Measurement Unit
IPAB  Institute of Perception, Action and Behaviour
KD  $k$-dimensional
KF  Kalman filter
KI  kinematic-inertial
KNN  $k$-nearest-neighbour
LIDAR  Light Detection and Ranging
LO  Laser Odometry
LOAM  LIDAR Odometry and Mapping
LS3  Legged Squad Support System
LSTM  Long Short-Term Memory
MEMS  Micro Electro Mechanical System
MIT  Massachusetts Institute of Technology
MUMC  Minimally Uncertain Maximal Consensus
NASA  National Aeronautics and Space Administration
NDT  Normal Distribution Transform
NICP  Normal ICP
NN  artificial neural network
NSM  Natural Segmentation and Matching
ORI  Oxford Robotics Institute
PCA  Principal Component Analysis
PF  particle filter
ReLU  rectified linear unit
RGB  Red, Green, Blue
RGB-D  RGB-Depth
RMSE  Root-Mean-Square Error
RNN  recurrent neural network
RCNN  recurrent convolutional neural network
SLAM  Simultaneous Localization and Mapping
SURF  Speeded Up Robust Features
SVC  Support Vector Classifier
SVM  Support Vector Machine
UAV  Unmanned Aerial Vehicle
VO  Visual Odometry
NOTATION \(^1\)

Sets

\[
\begin{align*}
\mathbb{R} & \quad \text{set of real numbers} \\
\mathbb{N} & \quad \text{set of natural numbers} \\
\text{SE}(\cdot) & \quad \text{special Euclidean group} \\
\text{SO}(\cdot) & \quad \text{special orthogonal group} \\
|\cdot| & \quad \text{cardinality of a set}
\end{align*}
\]

Vectors and Matrices

\[
\begin{align*}
x, \alpha & \quad \text{scalars} \\
x, \alpha & \quad \text{vectors} \\
\Delta x & \quad \text{relative vector} \\
^a x & \quad \text{vector expressed in frame } a \\
^a T_b & \quad \text{transformation from frame } b \text{ to frame } a \text{ such that } ^a p = ^a T_b \, ^b p \\
A, B & \quad \text{matrices} \\
A^T & \quad \text{matrix transpose} \\
A^{-1} & \quad \text{matrix inverse}
\end{align*}
\]

Other symbols

\[
\begin{align*}
x & \quad \text{state vector} \\
u & \quad \text{input vector} \\
z & \quad \text{measurement vector} \\
\mathcal{N}(\mu, \Sigma) & \quad \text{Gaussian distribution with mean } \mu \text{ and covariance } \Sigma \\
\mathcal{C} & \quad \text{point cloud} \\
\mathcal{O}(\cdot) & \quad \text{big } \mathcal{O} \text{ time complexity}
\end{align*}
\]

\(^1\) Standard notation used in the thesis, unless otherwise specified.
Navigation is a fundamental skill for any autonomous system. The term involves a set of tasks which range from determining where a robot is located with respect to a known reference – localization, to being able to reach a desired location using mapping and motion planning.

**Human Localization** Research in the field of robotic navigation draws inspiration from living beings. The vestibular system of a human measures direction of motion and acceleration continuously and naturally. Moreover, a person can monitor the environment through the sense of vision, touch, hearing, smell and taste. A human’s brain combines each one of these inputs with proprioception throughout the body, and with prior memories to augment his/her knowledge about his/her location and surroundings. Thanks to this capability, humans can move from place to place and interact with the environment safely and efficiently. For example, we can easily plan routes between obstacles and execute specific tasks, such as reaching and opening a door.

**Robot Localization** Our research goal is to develop robotic platforms which can achieve task-level autonomy similar to human-like skills. However, accurate and reliable localization, mapping, planning are challenging tasks for a robot processing noisy sensor feedback.
One can imagine a scenario where a robot has to explore a real-world environment. The robot would process *inertial and encoder measurements* to understand its body’s dynamics and kinematics, and incrementally estimate its own pose (i.e., position and orientation) and its velocity. As a matter of fact, this estimate would accumulate drift over time due to sensor noise, and because the linear position and absolute yaw states are not observable through proprioception. The drift rate would be affected by the dynamics of the robot’s motion, such as its feet striking the ground when walking or trotting (in the case of a legged robot), as well as slippage and balancing reactions due to the terrain’s features.

This drift can be corrected for using perception sensors, such as cameras or laser scanners, via *scene registration*. Scene registration refers to the process of aligning two observations of the same scene, by leveraging the mutually visible elements, i.e., the two observations should be characterized by *spatial overlap*. Similarly to human localization, this procedure adds mutual constraints between prior memories (e.g., a map of the environment) and what the robot perceives from its current location, effectively solving the localization problem.

### 1.1 Motivation and Requirements

The research in this thesis follows the DARPA Robotics Challenge (DRC), where humanoid robots were engaged in solving manipulation tasks, motion planning and walking on uneven terrains and staircases, etc. The Challenge was motivated by the goal to enable human-like robots to operate in challenging scenarios which are not accessible by more traditional wheeled robots, and to ultimately substitute humans during search and rescue operations in disaster zones or exploration of unknown areas of space.

The DRC represented an opportunity to encourage research in the field of legged robotics, as well as to identify the limitations of state-of-the-art control and navigation systems (including localization) during operation in real-world situations.

In this thesis we analyse these limitations, and focus on laser-aided localization techniques which are suitable for the continuous and safe operation of wheeled and legged robots in real-world environments.
1.1 motivation and requirements

The localization system needs to operate with a low failure rate for long periods of time and in a variety of situations (e.g., varying dynamics of motion, different terrains and lighting conditions). It should include auto-tuning capabilities, redundant sensing modalities and fail-safe operation modes (Cadena et al., 2016), so as to adapt to different challenges, and to achieve greater autonomy by maintaining consistent and precise reference of the terrain and objects in the environment.

The requirements are broken down as follows:

1. Localization which is sufficiently accurate to allow the robot to autonomously exploit long-term collision-free navigation tasks.
2. **Robustness** to complications caused by the dynamics of motion, such as noisy measurements affecting the drift rate of the robot’s state estimate\(^1\).

3. **Robustness** to visual challenges, such as limited sensor field-of-view, varying lighting conditions, occlusions and spatial overlap variation during motion.

4. **Reliability**\(^2\) of the system, which should be able to explicitly predict and prevent localization failure.

The reliability of a localization system depends highly on scene registration. A typical cause of failure is the absence of mutually visible geometric features which are necessary to constrain the alignment between consecutive scenes. For example, long corridors are unconstrained because of missing geometric features in the longitudinal dimension. Failures can also occur when passing through doorways, or due to occlusions and limited sensor field-of-view (FOV), which cause large overlap variation in the volume scanned by consecutive sensor measurements.

Throughout this dissertation, we cover these critical requirements and address the inherent challenges of achieving accurate, robust and reliable localization using a 3D laser scanner as the main sensor source.

Our first objective is the introduction of localization algorithms which are suitable for robots characterized by different locomotion skills, from mobile robots moving on wheels, to static walking humanoid robots\(^3\) and dynamic quadrupeds. Fig. 1.1 shows the robot platforms we consider and some of the scenarios we have experimented in.

During the development of the thesis, we extend these algorithms to function in increasingly more complex environments, from lab scenarios to industrial and building inspection.

---

\(^1\) As explained in detail in the next chapter, **state estimation** refers to the task of computing an estimate of a robot’s pose (i.e., position and orientation) and velocity. Forms of state estimation include **localization** and **odometry**.

\(^2\) We distinguish between robustness and reliability. **Robustness** is the design-specific capability of a system to deal with a certain distraction. With **reliability** we refer to the capability to safely operate in nominal conditions, by avoiding unmanaged failures.

\(^3\) Locomotion techniques can be divided into static and dynamic. As opposed to **dynamic** balance, during **static** motion at each instant the static equilibrium condition is satisfied (e.g., wheeled and static walking robots) (Siciliano and Khatib, 2007).
Finally, we investigate the applicability of state-of-the-art deep learning approaches to the laser odometry problem.

Tab. 1.1 overviews the thesis by chapter and topic, and lists the robotic platforms and sensor sources involved in the experimental evaluation. The platforms are described in more detail in Sec. 2.5.2. Most of our robots are equipped with a 2D laser scanner spinning about the forward facing axis. Every few seconds, the laser spins half a revolution and accumulates a 3D observation.

1.2 contributions

Our main contributions are broken down as follows:

**Chapter 3** We analyze the effect of spatial overlap variation on the performance of point cloud alignment and propose a strategy for non-incremental 3D scene registration in real environments. Our strategy increases the basin of convergence of standard approaches in conditions of variable spatial overlap between the input clouds and initial misalignment. This allows accurate and robust estimation of the robot’s pose, in the presence of challenges such as limited sensor FOV and occlusions.
Chapter 4 We present a multisensor system for state estimation of a dynamically walking and trotting quadruped. We investigate the use of multiple sensing modalities to provide the system with the degree of redundancy which is necessary to overcome the complications caused by the dynamic motions of robots, as well as disturbances such as uneven or rough terrain, slips or missteps, varying lighting conditions. The proposed solution is the first to discuss the fusion of kinematics, inertia, stereo vision and laser at very different frequencies and latencies, in a way that is suitable for a quadruped’s feedback controller. A substantial experimental evaluation demonstrates accuracy and robustness of the system, achieving continuous localization and drift per distance traveled below 1 cm/m.

Chapter 5 We propose a novel approach to predict the risk of a failed registration which we learn as a function of the spatial overlap between the input point clouds and the mutually visible constraints available in the region of overlap. This model allows the system to prevent localization failure when the geometry in the scene is unconstrained (due to an absence of geometric features) and where the overlap between the clouds is not uniform, for example due to constrictions, occlusions or a limited sensor FOV.

Chapter 6 We explore the applicability of deep learning techniques to the laser odometry problem. Differently from previous work, we train a neural network to learn from laser point clouds rather than camera images, achieving improved accuracy in the odometry task using laser. Despite the promising results, the approach does not report state-of-the-art accuracy as compared to standard laser odometry.

1.3 outline

The remainder of this document is organized as follows:

Chapter 2 provides a general background overview and introduces the preliminary concepts and tools used within the thesis.
The following chapters have been split into three parts. Part i covers the contributions related to system accuracy and robustness. Part ii focuses on the reliability of the system. Part iii presents preliminary results towards end-to-end learning.

Chapters 3, 4 and 5 present our publications and break down in detail our contributions, with comparisons to the current state-of-the-art, discussion and potential future work. Chapter 6 presents related work, preliminary results and future work in the context of deep learning for laser odometry.

Finally, Chapter 7 presents the overall conclusions.
In this chapter, we provide a background overview of state estimation and scene registration techniques, especially focusing on their application to legged robots. Furthermore, we overview the preliminary concepts and state-of-the-art algorithms used within the thesis, and present the experimental platforms and sensors in detail. In Chapter 6, we review the literature related to the advent of deep learning, and the introduction of learning-based solutions for robotic navigation tasks.

Along with the development of the thesis, we refer to the world frame attached to a fixed origin point on Earth, and to the body frame, which is the frame rigidly attached to a robot’s torso. As a convention, the body frame is oriented with the $x$-axis pointing forward, $y$-axis pointing leftward, and $z$-axis pointing upward.

We make use of the key terms of state estimation, localization and odometry. **State estimation** refers to the task of computing an estimate of a robot’s pose (i.e., position and orientation) and velocity. We distinguish between proprioceptive and exteroceptive state estimation. State estimation based on proprioceptive sensing involves measuring physical quantities describing the internal state of the robot, such as wheel positions, joint angles, acceleration, angular velocity, etc. using dedicated sensors (e.g., encoders, accelerometers and gyroscopes). On the other hand, exteroceptive sensors such as cameras or laser scanners are used to perceive the exterior and to measure mutual relationships between the robot and the environment (e.g., distance to
objects). Despite the extra degree of complexity introduced by exteroceptive sensors, these are fundamental when estimating linear position and absolute yaw orientation. Indeed, proprioceptive estimators are affected by unbounded drift over these states due to non-observability. More accurate state estimation can be achieved by fusing proprioceptive and exteroceptive inputs, as explained throughout this chapter.

Typically, localization suggests the use of exteroceptive sensors to aid proprioceptive pose estimation relative to the environment (e.g., within a prior map or relative to a past observation).

Odometry is a form of localization (Ben-Ari and Mondada, 2018): the task involves estimating the pose of a moving platform (body frame), starting from a known location (such as the world frame) and measuring the displacement incrementally. Examples are wheel odometry, leg odometry, laser or visual odometry, achieved through scene registration or by estimating displacement from camera images, respectively.

Throughout the thesis, unless otherwise specified, we refer to localization by displacement (i.e., by leveraging a prior pose estimate from proprioception), rather than global localization (i.e., place recognition). We specify case-by-case whether the localization is incremental (i.e., frame-to-frame), or non-incremental (i.e., frame-to-reference, where the reference is a past observation). Effectively, frame-to-reference localization also includes a (very low frequency) incremental component, introduced by each reference update.

2.1 proprioceptive state estimation

2.1.1 Tools Overview

State-of-the-art techniques for proprioceptive state estimation include inertial navigation, wheel/leg odometry, and the fusion of these.

2.1.1.1 Inertial Navigation

Inertial navigation tracks position, orientation and velocity of a moving platform using three orthogonal accelerometers and three orthogonal gyroscopes within an
2.1 proprioceptive state estimation

Inertial Measurement Unit (IMU). While the sensor directly measures acceleration $a$ and angular velocity $\omega$, the unobservable states of position and absolute yaw (rotation about the gravity vector $g$) can be estimated via integration of the known ones (Camurri, 2017). However, as discussed in Woodman, (2007) and Grewal and Andrews, (2010), accelerometers and gyroscopes are affected by a variety of errors and biases, which are propagated to the integrated states. For instance, gyroscope errors affect attitude (i.e., roll and pitch) estimation. In turn, attitude errors result in an inaccurate projection of the gravity vector $g$ on the coordinate acceleration $\ddot{x}$, which affects also the estimate of the integrated states of position and velocity. For this reason, inertial navigation only is insufficient for state estimation of legged robots. Better results are achieved in combination with leg odometry measurements.

2.1.1.2 Wheel/Leg Odometry

As mentioned previously, odometry is the process of estimating the current position and orientation of the body frame, starting from a known location in the world frame and measuring the displacement during motion. Specifically, the wheel odometry approach involves incrementally estimating a vehicle’s displacement by considering the number of wheel turns performed over time (measured using dedicated encoders).

While the term leg odometry is in analogy with the more traditional wheel odometry task, the application of odometry techniques to legged robots requires specific modelling of the more complex kinematics of motion. Leg odometry is achieved by means of forward kinematics (computed from joint encoder measurements) applied to the feet in contact with the ground (either sensed or estimated if no contact sensors are available).

The accuracy of proprioceptive state estimation depends on sensor accuracy, external noise, and degrades over time. Improved performance can be achieved by using measurements from multiple sensor sources, for instance fusing inertial sensing and leg odometry information.

1 Note that, as opposed to the coordinate acceleration $\ddot{x}$, $a$ indicates the acceleration vector which includes the effect of gravity.
In the following we overview the historical advances in proprioceptive only state estimation, with a focus on legged platforms and including approaches which fuse inertial measurements and odometry.

We discuss the state-of-the-art on exteroceptive state estimation in Sec. 2.2, and the fusion with proprioception in Sec. 2.3. In particular, an explanation of the fusion process achieved through extended Kalman filtering (EKFs), particle filtering (PFs) or smoothing is given in Sec. 2.3.1.

2.1.2 State Estimation for Multi-Legged Robots

Original advances in the field of state estimation for multi-legged robots date back to the work of Roston and Krotkov, (1992), with the first application of leg odometry techniques to a walking platform, the robot Ambler. Ambler was a hexapod robot weighing 2500 kg and 5 m tall, designed for space exploration. The approach was to determine the pose of the robot’s legs in the body frame, to distinguish between stance and swinging legs, and use the stance legs to estimate the robot’s body position and orientation in the world. The robot was performing only statically stable motions, with one leg moving at a time. The approach was based on the assumption that the feet do not move unless commanded to do so. In reality, positional perturbations on the feet poses are often introduced for example by slippage or sinkage into the terrain.

Lin et al., (2005) used a similar approach for a 50 cm, 7 kg hexapod robot called RHex. The robot lifted three legs at the time and moved only during full support (i.e., when all legs were in contact with the ground). In this case, the body pose could be computed through forward kinematics at the beginning and at the end of the full support phase. The authors made one of the first attempts to fuse leg odometry estimates with other sensor inputs, i.e., from an accelerometer, motivated by the need to extend their system to dynamic gaits (Lin et al., 2006).

More recently, Blösch et al., (2012) presented a state estimator which fused leg kinematics and inertial measurements within an EKF. The state vector included the feet position, and uncertainty estimates to account for disturbances and varying contacts with the ground. The approach was not restricted to specific gaits, leg configurations, or terrains.
More solutions have been proposed combining kinematics and inertial inputs, and using filtering techniques for multisensor fusion such as EKFs or PFs (Chitta et al., 2007; Camurri et al., 2017).

2.1.3 State Estimation for Bipedal Robots

History about state estimation for bipedal robots starts in recent years, following the original advances in the field of multi-legged robotics. Before the DRC, Stephens, (2011) used the Linear Inverted Pendulum model to estimate the position and velocity states of the centre of mass (CoM) of a humanoid robot while executing periodic gaits under external forces. Alternative approaches used extended Kalman filters for inertial-kinematics modelling. For example, the work by Blösch et al., (2012) first introduced an EKF-based state estimator for a quadruped robot, which Rotella et al., (2014) extended for bipedal state estimation.

During the DRC, solutions to state estimation were typically based on proprioceptive sensing only, and were affected by accumulated errors. Xinjilefu et al., (2015) used the Linear Inverted Pendulum model to infer modeling error and/or unexpected external forces during state estimation. Koolen et al., (2016) proposed a model for the elasticity of their robot’s leg joints to better distribute error. The same model was also utilized by Fallon et al., (2014) within an EKF filter to achieve low-drift proprioceptive state estimation with the Boston Dynamics Atlas robot.

2.2 Exteroceptive State Estimation

2.2.1 Tools Overview

Techniques for exteroceptive state estimation include visual odometry and scene registration approaches.
2.2.1.1 Visual Odometry

Visual odometry is the process of estimating incremental motion from a sequence of consecutive camera images. This is typically achieved using feature-based or appearance-based approaches (Scaramuzza and Fraundorfer, 2011; Garcia et al., 2012).

**Feature-based** approaches extract features (such as lines, corners, etc.), match them between consecutive image frames and minimize the reprojection error of one feature set onto the other to estimate the camera’s relative motion.

**Appearance-based** approaches monitor intensity changes between image pixels, instead of extracting features. For example, the camera motion can be estimated using optical flow, which uses the pixel intensity values in a neighbourhood to compute the displacement of brightness patterns from one frame to the next.

2.2.1.2 Iterative Closest Point

We refer to scene registration as the process of finding the rigid transformation $T$ which aligns two point clouds, by leveraging the mutually visible elements.

More formally, let $C^A = \{p_i^A\}$ and $C^B = \{p_j^B\}$, with $i \in [1, |C^A|]$ and $j \in [1, |C^B|]$ be the two set of points. We want to find the transformation $T$ that minimizes the distance between corresponding points in the two observations, such that:

$$T = \arg \min_T \sum_j \|p_i^A - T \cdot p_j^B\|^2$$

$$= \arg \min_T \sum_j \|p_i^A - (R \cdot p_j^B + t)\|^2$$

(2.1)

where $T$ is a transformation matrix restricted to the class of rigid body motions forming the special Euclidean group $\text{SE}(3)$, including a rotation matrix $R \in \text{SO}(3)$ and a translation vector $t = [x, y, z]^T \in \mathbb{R}^3$. Throughout this work, we refer to the inputs to the registration algorithm as a *reference* and a *reading* cloud, the latter to be aligned into the reference, according to the notation from Pomerleau et al., (2013).

The Iterative Closest Point (ICP) algorithm is one of the most commonly used techniques for 3D scene registration. The algorithm estimates the relative alignment between two 3D point clouds iteratively, through four main steps: pre-filtering, data association, outlier rejection and error minimization. Firstly, the input point clouds can be filtered to discard redundant points, extract meaningful patches or compute
2.2 Exteroceptive State Estimation

Figure 2.1: Illustration of point-to-point and point-to-plane ICP distance metrics. The grey boxes highlight the main difference between the two strategies, the dashed pink lines indicate the distance to be minimized.

Descriptors such as normal vectors. Secondly, the algorithm iteratively finds pairwise correspondences between the reference and reading cloud based on Euclidean distance. An outlier rejection policy, is then applied to account for the fact that some points will not have any correspondences in the second cloud. This policy adds a maximum matching threshold $d_{\text{max}}$, for example a quantile from the distribution of distances (Zhang, 1994). Once the set of valid correspondences is defined, the algorithm calculates the rigid transformation $T$ which minimizes the alignment error based on a distance metric. The most common distance metrics are point-to-point (Besl and McKay, 1992), which minimizes the distance between each pair as shown in Eq. (2.1), and point-to-plane (Chen and Medioni, 1992), which minimizes the error along the reference’s surface normal. The algorithm using the point-to-plane distance metric is described in Alg. 1. A graphical comparison between the two strategies is shown in Fig. 2.1, where $d_j$ and $n_j$ are the distance and normal vector for the $j$-th match.

2.2.2 Scene Registration Methods

Several methods for point cloud registration have been explored in the last two decades. They can be classified into two main categories.
Algorithm 1: Point-to-plane Iterative Closest Point

**input**: Reference cloud, reading cloud: $C^A = \{p^A_i\}, C^B = \{p^B_j\}$

Initial transformation: $T_0$

**output**: Transformation $T$ which aligns $C^B$ onto $C^A$

1. $T \leftarrow T_0$
2. while not converged do
3.   for $j \leftarrow 1$ to $|C^B|$ do
4.     $p_i^A \leftarrow \text{GetNearestPointInReference}(T \cdot p_j^B)$
5.     if $\|p_i^A - T \cdot p_j^B\| \leq d_{max}$ then
6.         $w_j \leftarrow 1$
7.     else
8.         $w_j \leftarrow 0$
9.   $T \leftarrow \arg \min_T \left\{ \sum_j w_j \| n_j \cdot (p_i^A - T \cdot p_j^B) \|^2 \right\}$

On the one hand, **sparse methods** find point correspondences based on the local appearance (i.e., features) of meaningful clusters of points. These methods do not require prior information about the relative pose of the clouds, and are typically used for place recognition.

On the other hand, **dense methods** require an initial prior about the relative pose offset between the two clouds, but rely on the points themselves and simpler heuristics to determine correspondences. The Iterative Closest Point is an example of dense method.

In the following, we overview the literature on dense registration methods. For the applications we consider in this work, a prior to initialize the alignment is typically available from proprioception.

The basic implementation of ICP dates back to Besl and McKay, (1992), who first proposed the point-to-point distance metric. Notable improvements to the original algorithm have been introduced by Chen and Medioni, (1992) with the point-to-plane error metric, which exploits the local continuity and smoothness of points in the same neighbourhood, and is better suited to structured environments.
Alternatively, the Normal Distribution Transform (NDT), introduced in Biber and Strasser, (2003) (and extended in Magnusson et al., (2007) to the 3D case), uses a combination of normal distributions defined on a grid-like structure, resulting in a locally smooth representation of the point cloud describing the probability of finding part of the surface in any cell. Standard optimization methods (e.g. Newton’s algorithm) are applied on this representation for alignment. In Magnusson et al., (2009) the authors analyzed the performance of ICP and NDT: although NDT was demonstrated to have a larger basin of convergence, it was found to be less predictable than ICP. For example, in several cases a registration would be successful in spite of large initial misalignment, but would fail from a more accurate initial pose.

Pathak et al., (2010b) presented a third alternative to ICP and NDT, which is a strategy entirely based on the availability of large uniform surfaces in plane-rich environments. Planar patches are extracted, matched minimizing planes-parameter uncertainty, and aligned by maximizing geometric consistency (i.e., Minimally Uncertain Maximal Consensus (MUMC)).

Segal et al., (2009) proposed Generalized-ICP (GICP), which combines the standard point-to-point and point-to-plane error metrics into the same framework. Instead of analyzing the planarity and continuity features of the reference cloud only (see definition in Sec. 2.2.2), it models local planarity for both the input clouds by introducing a plane-to-plane strategy. GICP has been implemented to give the minimization process a probabilistic interpretation, that is the measured points belong to Gaussian distributions centred at the point of perfect correspondence.

Normal ICP (NICP) (Serafin and Grisetti, 2015) uses a scene representation inspired to 3D-NDT. A Gaussian distribution is computed from a point’s neighbourhood and used to define correspondences. Specifically, the method labels each point with the properties of its neighbourhood, i.e., normal and curvature, finds the correspondences based on these features, and determines the clouds’ alignment by minimizing the distance between the corresponding point pair and their normals.

ICP provides a simple tool for registration of point clouds in 3D. However, it makes the implicit assumption of fully overlapping point clouds, which is violated in reality when the scenario and the robot’s point-of-view are not static and the sensor’s FOV is limited. This can be mitigated by defining a criteria to identify outliers in the
correspondence set (e.g. Zhang, (1994)). Nonetheless the robustness of registration algorithms remains an open research problem, as it is highly sensitive to the properties of the input clouds, such as structural features (i.e., the presence of smooth surfaces), initial alignment error and the degree of spatial overlap.

Summary

A summary of the different scene registration approaches cited above is illustrated in Tab. 2.1, as well as the first contribution of this thesis (Nobili et al., 2017b), which we discuss in detail in Chapter 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>Data Association</th>
<th>Outlier Filtering</th>
<th>Error Minimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Besl and McKay, (1992) ICP</td>
<td>point-wise</td>
<td>–</td>
<td>point-to-point</td>
</tr>
<tr>
<td>Chen and Medioni, (1992) ICP</td>
<td>point-wise</td>
<td>–</td>
<td>point-to-plane</td>
</tr>
<tr>
<td>Zhang, (1994) ICP</td>
<td>point-wise</td>
<td>fixed quantile distance</td>
<td>point-to-point</td>
</tr>
<tr>
<td>Biber and Strasser, (2003) NDT</td>
<td></td>
<td></td>
<td>direct solution based on local point distributions</td>
</tr>
<tr>
<td>Segal et al., (2009) GICP</td>
<td>point-wise</td>
<td>fixed max distance</td>
<td>plane-to-plane</td>
</tr>
<tr>
<td>Pathak et al., (2010b) MUMC</td>
<td></td>
<td></td>
<td>direct solution based on planar patches consensus</td>
</tr>
<tr>
<td>Serafin and Grisetti, (2015) NICP</td>
<td>point features</td>
<td>fixed max distance</td>
<td>feature-to-feature</td>
</tr>
<tr>
<td>Nobili et al., (2017b) AICP</td>
<td>point-wise</td>
<td>auto-tuned quantile distance</td>
<td>point-to-plane</td>
</tr>
</tbody>
</table>

Table 2.1: Overview of point cloud registration approaches.

2.3 MULTISENSOR STATE ESTIMATION

2.3.1 Tools Overview

Multisensor state estimation involves filtering or smoothing techniques for sensor data fusion.

2.3.1.1 Filtering

Filtering approaches, rather than storing the latest reported measurement, propagate a probability distribution to represent the estimated robot’s state and uncertainty, given the current inputs and the previous state. By construction, the previous state incorporates all information about the past states and measurements, thus no additional data need to be stored. A common example of such filters are extended Kalman filters (EKFs) (Welch and Bishop, 1995; Thrun et al., 2001), which estimate the state in two steps, prediction (e.g., from the IMU) and update (e.g., from leg odometry.
and/or exteroception). Kalman filtering is explained in more detail in Appendix B. Alternatively, particle filters (PFs) (Gustafsson et al., 2002) model the posterior distribution with a finite set of samples called particles, each one representing a different hypothesis of the real state (Thrun et al., 2005).

2.3.1.2 Smoothing

Smoothing approaches recover the maximum a posteriori estimate for the entire robot’s body trajectory, given all past and current inputs (Kaess et al., 2007). This is achieved via bundle adjustment (BA) optimization, and allows correction of the past estimates as new measurements are processed. However, the approach is typically slower than filtering, as smoothing is performed over a history of stored states, and computational complexity grows over time. This can be mitigated by selecting a reduced number of past frames to process (Strasdat et al., 2012).

In the next section we overview state-of-the-art approaches which fuse proprioceptive and exteroceptive measurements for state estimation.

2.3.2 Review on Heterogeneous Sensor Fusion

Multisensor state estimation leverages different sensing modalities in a manner which is complementary, and provides redundancy to the challenges encountered by each sensor during navigation. The aim is to achieve state estimation which has low drift and is robust, i.e., functional across different scenarios, for example when the lighting conditions vary or the dynamics of the robot’s motion cause noisy sensor measurements. For example, vision techniques require sufficient lighting and features in the scene, and are more sensitive to sensor’s shaking which causes image blur. Instead, laser measurements often assume structured environments, are sometimes available at lower frequency than images (e.g., when using a spinning 2D laser), and are unreliable in areas where there are reflections or transparencies (e.g., water, glass).
Research on multisensor state estimation made considerable advances in the field of flying robotics. It consists of loosely and tightly integrated\(^2\) approaches, based on filtering (Mourikis and Roumeliotis, 2007; Blösch et al., 2015; Faessler et al., 2016) or smoothing techniques (Leutenegger et al., 2015; Forster et al., 2017).

A basic assumption of state estimators based on EKFs is that the input measurements are uncorrelated. This assumption is violated in reality, and leads to suboptimal estimates of the robot’s state and covariance. Lynen et al., (2013) and Shen et al., (2014) developed their systems based on stochastic cloning (Roumeliotis and Burdick, 2002; Mourikis et al., 2007). This technique improves on the system’s accuracy by augmenting the state space of a Kalman filter to account for the correlation between relative measurements during odometry.

With a focus on legged robots and perception in the loop, Chilian et al., (2011) combined inertial, kinematics and stereo sensors on-board of a miniature six-legged robot. Multisensor fusion was achieved using an indirect feedback information filter (IF), which is an inverse formulation of an EKF whose state vector contains the errors of the actual state rather than the state variables themselves.

The work of Ma et al., (2016) is most closely related to the scope of this thesis in terms of scale and dynamism of their robot. The system fused the input from a high quality IMU with stereo visual odometry to produce a pose estimate for navigation tasks such as path planning. Optional GPS and leg odometry were used to update the EKF when visual odometry failed. Their approach was not used within the robot’s closed loop controller.

Successful implementations of visual localization for humanoid robots include Stasse et al., (2006) and Alcantarilla et al., (2013). Hornung et al., (2010) proposed a laser-based localization approach with application to a 0.58 m tall NAO robot while exploring a miniature 3D world model. This work has been extended in Oßwald et al., (2012) to include observations from a monocular camera.

\(^2\) **Loose integration** refers to the case where measurements from different sensor sources are processed separately, and only the resulting state measurement is incorporated. **Tight integration** refers to the case where the measurements are processed jointly (e.g., inertial measurements may be leveraged during feature matching).
During preparation for the DRC, the teams explored the usage of laser measurements to reduce drift on the robot’s pose estimate from proprioception. This drift had a major impact on task planning and execution autonomy, causing any motion plan generated at the start of movement to become unsafe and invalid over time, and requiring the periodic intervention of the human operator for re-planning.

In Fallon et al., (2014) (MIT team), the authors used the depth data from a spinning 2D laser on the Atlas robot to compute position measurements relative to a prior map. Each planar laser scan was incorporated as a sample in a Gaussian particle filter at 40 Hz, with an associated likelihood value result of the comparison with the map. Rather than planar laser measurements, the IHMC team (Koolen et al., 2016) attempted lower frequency ICP-based registration of full 3D point clouds (accumulated from a spinning 2D laser). Both approaches were demonstrated in a laboratory environment. However, neither method could be used during the DRC due to a lack of field testing for scene registration accuracy and robustness. This experience demonstrated the importance of adaptation and tuning of the baseline laser registration algorithms to the challenges which occur in real-world situations, such as the presence of moving objects, occlusions and reduced sensor FOV.

Summary

A summary of the state estimation approaches for legged robots cited in this review is presented in Tab. 2.2, as well as the second contribution of this thesis (Nobili et al., 2017a), which we discuss in detail in Chapter 4.

2.4 Reliability during pose estimation

As mentioned in the previous chapter, the reliability of a localization system is often dependent on preventing registration failure, which might occur due to lack of geometric features or constraints, and occlusions in cluttered regions or through doorways. The optimal registration estimated by ICP is indeed highly affected by the mutual properties of the input clouds. By definition, the point-to-plane distance metric allows the input clouds to slide against each other along continuous regions (e.g., planar, curved surfaces). If all point pairs belong to parallel surfaces, the algorithm
2.4.1 Tools Overview

Traditionally, machine learning approaches are used in computer science to learn from distributions of data. Specifically for our applications, we overview the Principal Component Analysis (PCA) approach and Support Vector Machines (SVMs), which will be used in the thesis as tools to analyze the content of laser point clouds, i.e., whether the set of points is distributed in the 3D Euclidean space in a way which can constrain the alignment. A more detailed explanation of the machine learning process and, in particular, artificial neural networks (NNs) is given in Sec. 6.2.2.
2.4.1.1 **Principal Component Analysis**

Principal Component Analysis (PCA) is a dimensionality reduction procedure which transforms a set of variables into a lower dimensional set, called principal components, that still embed most of the information contained in the data. The first principal component accounts for most variability in the data, and each succeeding component for the remaining highest variability dimensions. The resulting axes, called eigenvectors, are oriented with the directions of maximum variation of the original observations. The associated eigenvalues indicate how much variation in the data set is explained by each eigenvector.

2.4.1.2 **Support Vector Machines**

A Support Vector Machine (SVM) is a supervised\(^3\) learning approach for classification (also known as Support Vector Classifier (SVC)). Given a set of labelled training samples, it defines an optimal separating hyperplane, which can be employed to categorize new samples from a test set. Thus, the SVM’s input is a set of samples \(x_i\) with \(i \in [0, N]\), and the output after training is a set of weights \(w_i\), one for each sample, whose linear combination predicts the output result \(y_i\), which is either a discrete value 0 or 1, or a continuous value \(y_i = [0, 1] \in \mathbb{R}\) computed as the sample’s distance from the hyperplane. The main difference with NNs is that SVMs’ predictions for the test data depend only on the kernel function evaluated at a subset of the training data. Indeed, the optimization process reduces the number of non-zero weights to the samples in proximity of the hyperplane. These samples “support” the hyperplane, and are called support vectors.

2.4.2 **Methods to Evaluate Reliability**

In this section, we overview the literature on the stability of point cloud registration. Previous work (Guehring, 2001; Gelfand et al., 2003; Zhang et al., 2016) has shown that the stability of a particular registration problem can be evaluated after the set of point correspondences has been selected, using PCA on the covariance matrix used for alignment error minimization. If the covariance matrix is not full rank, the

\(^3\) In **supervised** learning scenarios, the ground truth labels (i.e., training labels) are fully observed.
registration is underconstrained. Zhang et al., (2016) used this approach to identify unconstrained directions in the state space, and then solved the problem partially, optimizing only along the well-constrained dimensions. However, this analysis depends on the data association step of the registration algorithm. When the degree of overlap is low, the number of point correspondences available for data association might be insufficient to find stable eigenvectors of the covariance matrix, thus leading to an unreliable measure of the constraints. This is a major problem for point cloud registration algorithms, as they are highly sensitive to the degree of overlap between input clouds. Our work is motivated by a need for failure prediction methods which are reliable in the presence of varying overlap.

Pathak et al., (2010a) recognized the sensitivity of registration to low data overlap and formulated two overlap metrics which can be used to study the cause of alignment failures. However, the metrics were computed using the ground-truth alignment between the clouds.

Zhen et al., (2017) formulated a localizability measure, which is related to the stability of point cloud registration, for a laser with respect to a prior 3D map, based on the analysis of the point clouds’ content, rather than the covariance matrix. It was computed offline by generating synthetic laser data to simulate observations from within the map. They used it to plan trajectories of a UAV so as to stay within areas with high localizability.

2.5 thesis overview

Throughout the thesis we focus on localization methods which satisfy the requirements described in Sec. 1.1. For this, we use a 3D laser scanner (also called Light Detection and Ranging (LIDAR)) as the main sensor. Advantages of using lasers are for instance high measurement accuracy at long ranges and better suitability to function under varying lighting conditions.

In this section, we describe the preliminary tools used within the thesis, as well as the experimental platforms and sensors.
2.5.1 Preliminary Tools

Tab. 2.3 summarizes the standard approaches used in this thesis. In the following sections, we describe in more detail the specific implementations of proprioceptive state estimation, visual odometry and scene registration used within the thesis.

<table>
<thead>
<tr>
<th>Type</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>Approach</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proprioceptive State Estimation</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>KI w/ contact sensors</td>
<td>Koolen et al., (2016)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KI w/o contact sensors</td>
<td>Camurri et al., (2017)</td>
</tr>
<tr>
<td>Visual Odometry</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>FOVIS</td>
<td>Huang et al., (2011)</td>
</tr>
<tr>
<td>Scene Registration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>ICP</td>
<td>Libpointmatcher</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>AICP</td>
<td>Nobili et al., (2017b)</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>PCA, SVM</td>
<td>OpenCV</td>
</tr>
<tr>
<td>Approaches</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>CNN, RNN</td>
<td>TensorFlow</td>
</tr>
</tbody>
</table>

Table 2.3: Overview of the approaches and implementations used in this thesis, with corresponding chapter. *Legend: C3 = Chapter 3, C4 = Chapter 4, C5 = Chapter 5, C6 = Chapter 6.*

Proprioceptive State Estimation

Estimation with contact sensors In Chapter 3 of this thesis we leverage the EKF-based kinematic-inertial state estimator by Koolen et al., (2016) (IHMC team), as a prior for laser-based localization. The estimator was originally developed for the humanoid robot Atlas during the DARPA Robotics Challenge (DRC), and a few years later integrated on the NASA humanoid Valkyrie. The system relies on joint encoders, IMU measurements, and incorporates constraints derived from contact sensors on the robot’s feet to produce estimates of position, orientation and velocity. The contributions of joint encoders and the IMU are weighted differently: at high frequencies, the measurements from the accelerometer are predominant, so as to reduce disturbances coming from the joint backlash, at low frequencies, the use of leg odometry is preferable to avoid cumulative error due to integration of the measured linear acceleration.

Estimation without contact sensors In Chapter 4 of this thesis we leverage the EKF-based kinematic-inertial state estimator proposed by Camurri et al., (2017) for a walking and trotting quadruped robot, the IIT’s HyQ, with no ground contact sensors on the feet. In practice, avoiding the use of contact sensing on highly
dynamic and heavy robots is desirable as it reduces the cost, weight and also the sensitivity of the system, as contact sensors frequently break. However, legged robots make intermittent contact with the ground, and only the legs in contact contribute to the robot motion. This requires a way to identify contacts when computing the estimate.

Their approach extends the system proposed by Fallon et al., (2014) for the Boston Dynamics humanoid Atlas. It identifies the feet which are in reliable contact with the ground by using the joints’ force-torque sensors, instead of contact sensors on the feet. This is achieved by learning a contact threshold on the ground reaction force. A contact is detected when the error between the velocity estimates obtained from the joint encoders and forward kinematics during characteristic motions and the ground truth is minimal. Impact information and mutual agreement between the velocities of the legs in contact are then used to compute the covariance and update the filter. The experimental evaluations carried out online on the robot demonstrated comparable results to platforms with contact sensors.

Extended Kalman filtering is the fusion technique used in Koolen et al., (2016), Fallon et al., (2014), and Camurri et al., (2017). As shown in their papers, this solution demonstrated acceptable real-time performance on real platforms, with drift rates in the order of 1-4 cm/m, as shown in Tab. 2.4.

Visual Odometry

In Chapter 4 we use a feature-based visual odometry approach called Fast Odometry from VlSion (FOVIS) (Huang et al., 2011). The algorithm estimates incremental motion from camera images at 10 Hz. The pose estimates produced are then integrated into the EKF-based state estimator proposed by Camurri et al., (2017) as linear position updates.

**Fast Odometry from Vision**  
**FOVIS** is based on a standard feature-based stereo visual odometry pipeline. Firstly, the input images are pre-processed, and features are extracted at each level of a Gaussian pyramid using the **FAST** feature detector (Rosten and Drummond, 2006). Secondly, the features are matched according to the sum of absolute differences score between their descriptors, and arranged
as the nodes of a graph for outlier rejection using a mutual-consistency check. The final motion estimate is computed by minimizing the Euclidean distance between the matched features.

This approach was originally developed for drones equipped with an IMU and an RGB-D camera. In our work, we use FOVIS with the depth data provided by the stereo camera on the Multisense SL sensor (see Sec. 2.5.2). In contrast to depth sensors such as RGB-D and lasers, the quality of depth data generated with stereo cameras is highly dependent on the presence of texture in the scene, and dark or featureless areas can lead to poor results.

**Scene Registration**

**Iterative closest point** In Chapter 3 of this thesis we discuss the properties of ICP in detail, and propose a solution to overcome its limitations for the localization of wheeled and legged robots in real-world scenarios. We build upon the modular implementation of ICP proposed by Pomerleau et al., (2013), using the point-to-plane formulation of the algorithm. Their software is publicly available under the name of *Libpointmatcher*. Pomerleau et al., (2013) suggested that the point-to-plane variant, achieves better overall performance than point-to-point especially with the presence of structural features in the scene, which is a typical use case for the applications and environments in this work, shown in Fig. 1.1. With this baseline formulation, stable performance can be achieved if: 1) the alignment is initialized within a constrained basin of convergence (i.e., with less than 10 cm and 10° initial error in 3D translation and rotation), and 2) the spatial overlap between the inputs is constantly high.

**Auto-tuned iterative closest point** We propose Auto-tuned ICP (AICP) as a strategy for non-incremental 3D point cloud registration, which extends the standard implementation of ICP to more robustly register point clouds which have reduced overlap. This is achieved by automatically tuning the outlier-rejection filter, i.e., the maximum matching threshold $d_{\text{max}}$, to account for the degree of spatial overlap between the input clouds. This framework allowed for accurate registration with an increased basin of convergence and in conditions of non-uniform spatial overlap. This approach is proposed and described in detail in Chapter 3, and used as the registration framework in Chapter 4 and 5.
Machine Learning Approaches

In Chapter 5 we study the stability of ICP by analyzing the input point clouds’ content prior to registration, using PCA, as well as considering the degree of spatial overlap between the clouds. A SVC is trained on these features and employed to predict alignment failures.

We leave the discussion about convolutional neural networks (CNNs) and recurrent neural networks (RNNs) architectures to Chapter 6, where we investigate the applicability of these deep learning techniques to the laser odometry problem.

2.5.2 Experimental Platforms and Sensors

The robotic platforms we use for experimental evaluation in the thesis are the Boston Dynamics humanoid Atlas (Fig. 2.4), the NASA humanoid Valkyrie (Fig. 2.5), the IIT Hydraulic Quadruped (HyQ) robot (Fig. 2.6), and the Clearpath Husky mobile robot (Fig. 2.7). We run more experiments on publicly available datasets, such as KITTI (Geiger et al., 2013).

In the following, we give an overview of each robot with a focus on the proprioceptive sensors it is equipped with:

atlas The Atlas robot has six joints per leg. The angle of each joint is measured from the motion of its hydraulic actuator using a potentiometer, and then computing a transformation through the leg linkage. As explained in Fallon et al., (2014), the model does not account for flexion of the linkage when loaded, or backlash when a joint changes direction, introducing on average 1° angle measurement error. The IMU
<table>
<thead>
<tr>
<th>Robot</th>
<th>xyz</th>
<th>z</th>
<th>Drift [deg/s]</th>
<th>IMU</th>
<th>Gait</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlas</td>
<td>1.67</td>
<td>0.27</td>
<td>0.010</td>
<td>KVH 1770</td>
<td>static walk</td>
</tr>
<tr>
<td>Valkyrie</td>
<td>0.75</td>
<td>0.51</td>
<td>–</td>
<td>Microstrain GX4-25</td>
<td>static walk</td>
</tr>
<tr>
<td>HyQ</td>
<td>3.63</td>
<td>3.47</td>
<td>0.119</td>
<td>Microstrain GX4-25</td>
<td>trot</td>
</tr>
<tr>
<td>HyQ</td>
<td>3.27</td>
<td>3.08</td>
<td>0.019</td>
<td>KVH 1775</td>
<td>trot</td>
</tr>
</tbody>
</table>

Table 2.4: Summary of proprioceptive pose estimation accuracy during experiments carried out in Fallon et al., (2014), Scona et al., (2017), and Nobili et al., (2017a). The state estimators are: Atlas – Fallon et al., (2014), Valkyrie – Koolen et al., (2016), HyQ – Camurri et al., (2017). More details on Drift per Distance Traveled (DDT) are provided in Appendix C.

Integrated within the robot is a KVH 1750-IMU (*KVH Industries*) comprised of Fibre Optic Gyroscope (*FOG*)\(^4\) and MEMS accelerometers of tactical grade\(^5\).

**Valkyrie** Valkyrie is equipped with six joint encoders per leg (to directly sense joint angles with lower sample noise, as compared to using potentiometers), feet contact sensors, and a MEMS Microstrain GX4-25 IMU (*LORD Microstrain*) of industrial grade. With this IMU, the performance of the gyroscope is lower than a FOG, and requires online bias estimation to correct for the orientation drift.

**HyQ** The Hydraulic Quadruped (*HyQ*) robot has three joints per leg, each equipped with an incremental optical encoder. The robot does not use contact sensors on the feet. Instead, torque and force at the joints can be measured using dedicated loadcells (joint force/torque sensors). During our experiments, *HyQ* was mounted with different IMUs, either the Microstrain GX4-25 or the higher grade KVH 1775.

Quantitative performance for each of the robots’ proprioceptive state estimators are summarized in Tab. 2.4.

\(^4\) The most common *gyroscopes* are Micro Electro Mechanical Systems (*MEMSs*), which use the Coriolis effect to measure angular velocity, and Fibre Optic Gyroscopes (*FOGs*), which use the Sagnac effect (Post, 1967). Typical performance of a *MEMS* are considerably lower than a *FOG* (which however comes at a higher price) (Grewal and Andrews, 2010).

\(^5\) *Accelerometers* performance can be classified into four grades. In descending order: navigation grade, tactical grade, industrial grade, and automotive grade. The most commonly used in robotics applications are tactical and industrial grade devices.
Regarding exteroceptive sensing, Atlas, Valkyrie and HyQ’s main perception device is a Carnegie Robotics Multisense SL (Fig. 2.2).

The sensor is composed of a stereo camera and a Hokuyo UTM-30LX-EW planar laser – 40 scans per second with 30 m range – spinning about the forward-facing axis. Every few seconds it spins half a revolution and a 3D point cloud is accumulated, with a $\sim 220^\circ \times 180^\circ$ FOV$^6$. For Valkyrie, most of the laser returns fall on the robot’s head cover, allowing only a reduced FOV ($180^\circ \times 120^\circ$). The camera provides $1024 \times 1024$ colour image pairs with $80^\circ \times 80^\circ$ FOV, and corresponding disparity images at a rate of 15 Hz. Disparity images are computed by an implementation of Semi Global Matching (Hirschmüller, 2008) running on an on-board FPGA.

We run other experiments on mobile platforms, such as the Clearpath Husky mobile robot equipped with the Multisense SL with full FOV of $200^\circ \times 200^\circ$ (Fig. 2.7). In Chapter 6 we use the KITTI dataset for training and testing. The dataset stores six hours of traffic scenarios captured from a vehicle equipped with four video cameras (two color and two grayscale cameras), a Velodyne HDL-64E 3D laser (Fig. 2.3) and a combined GPS/IMU inertial navigation system for ground truth tracking.

The Velodyne is a rotating 3D laser scanner, which captures $360^\circ \times 26.8^\circ$ FOV point clouds at 10 Hz, resulting in about 1.3 million points per second. Each cloud has 120 m range, 0.09$^\circ$ angular resolution in azimuth, and ($26.8^\circ/64$ beams) $\simeq 0.42^\circ$ angular resolution in elevation.

---

$^6$ Noted as (horizontal × vertical) FOV.
Figure 2.4: Boston Dynamics Atlas (photo credit: MIT team).

Figure 2.5: NASA Valkyrie.

Figure 2.6: IIT Hydraulic Quadruped (HyQ).

Figure 2.7: Clearpath Husky.
Part I

ACCURACY AND ROBUSTNESS
3

OVERLAP-BASED ICP TUNING FOR ROBUST LOCALIZATION

3.1 INTRODUCTION

As discussed in previous chapters, state estimation strategies for legged robots are based on inertial sensing and kinematics, and suffer from continuous drift. For example, Fig. 3.1 shows the Boston Dynamics humanoid Atlas during the DRC finals, and how drift in the robot’s pose estimate is reflected by the misalignment of point clouds from the robot’s laser. Laser point clouds can be used to compensate for this drift via 3D scene registration.

Standard scene registration approaches are sensitive to error in the initial alignment, as well as the degree of spatial overlap between the input point clouds. The latter is a considerable limitation in real world scenarios, for example due to reduced sensor FOV and the presence of occlusions. Moreover, initial misalignment is affected by the quality of IMU and encoders, as well as the dynamics of the robot’s motion.

In this chapter we discuss our first contribution, on scene registration which satisfies the accuracy requirement 1. (described in Chapter 1), and overcomes the robustness challenge 3. caused by spatial overlap variations during motion. We explore a non-incremental registration approach, i.e., frame-to-reference, which improves the accuracy of localization by avoiding errors (alignment imperfections) which are accumulated on a frame-to-frame basis. This is difficult because the registration of the current cloud to a common reference (a past point cloud) becomes more challenging
Figure 3.1: Kinematic-inertial pose estimator drift. Top-view of the DRC finals arena (yellow map) and corresponding robot’s side camera views (right). The robot moves across the arena from pose $i$ to pose $j$. The drift accumulated in the robot’s pose estimate is reflected by the misalignment of the current measurement from the laser (red point cloud) with respect to a reconstructed map.

as the robot moves away from its original pose. Indeed, overlap decreases with the displacement from the pose of the reference point cloud.

We propose a laser-based localization system capable of overlap-based auto-tuning, which we call Auto-tuned ICP (AICP). The strategy focuses on the robustness of registration when point cloud overlap variations occur. As part of the ICP registration module, the outlier rejection filter is automatically tuned at run-time depending on the degree of overlap between the input clouds, to allow alignment to a common reference. We define a novel metric to estimate overlap, which is the one used to tune the filter. We demonstrate extensive experiments on the NASA humanoid robot Valkyrie and the Boston Dynamics Atlas, and show increased basin of convergence of the proposed registration strategy with respect to state-of-the-art approaches, robustness to high overlap variations, and overall accuracy. Finally, AICP will be integrated as a core module within the state estimation strategy for the quadruped robot HyQ in Chapter 4.

In the following sections, we describe the localization problem in terms of the relevant frames and transformations. An illustration is shown in Fig. 3.2, where the read-
3.1.1 Leveraging Kinematic-Inertial State Estimation

Our registration approach leverages the (drifting) kinematic-inertial pose estimate $jT_{ki}^w$ derived from Fallon et al., (2014) for the humanoid Atlas, and Koolen et al., (2016) for Valkyrie, to initialize the alignment.

A major difference between the two robots is the lower quality of the gyroscope sensing in Valkyrie (as discussed in Sec. 2.5.2), which affects the state estimate and requires online gyro bias estimation to reduce orientation drift.
3.1.2 Integrating a Laser Correction

Considering a humanoid’s kinematics chain, depicted in Fig. 3.3 for Valkyrie, the correction $j_c T^{icp}_j$ for the drifting pose prior $j T^{ki}_w$ is calculated using ICP. The corrected pose $j_c T_w$ is then computed from the concatenation of the two, as follows:

$$j_c T_w = (l_{j_c} T^{fk}_{b_{j_c}})^{-1} \cdot j_c T^{icp}_j \cdot l_{j_c} T^{fk}_{b_{j_c}} \cdot j T^{ki}_w$$ \hspace{1cm} (3.1)

The reader should note that for the Multisense SL (which is the primary sensing unit mounted on Atlas and Valkyrie), the laser’s kinematics chain is composed of two frames (as shown in Sec. 2.5.2), one attached to the main device’s body, and a second frame attached to the planar laser scanner itself, which is a rotating frame about the forward facing axis (x-axis). Each planar laser scan is captured relative to this frame. For simplicity, we omit this detail in the illustration, and refer instead to a unique fixed laser frame $l$, assuming that each planar scan is automatically projected relative to $l$.

3.1.3 Impact of Registration Computation Time

A diagram of our system computation information is presented in Fig. 3.4 with a toy example. The speed of rotation of the laser determines the time $t_{acc}$ required to accumulate a full 3D point cloud (2 – 5 s). The registration algorithm runs on a parallel thread and produces a pose correction with a computation time $t_{icp}$ of 0.6 s for approximately 10000 points per cloud in our experiments. A pose correction $T^{icp}$ is integrated to update the pose estimate with latency $t_{icp}$.

The computation time of the ICP algorithm depends on the implementation used to handle the data association step, and increases with the size of the point clouds. In its basic implementation, the algorithm has complexity $O(NM)$, where $N$ and $M$ are the number of points in the reference and reading point cloud respectively. In order to decrease the nearest neighbour search time, we use a $k$-dimensional binary search tree (KD tree) structure. Theoretically, searching in a KD tree with $N$ entries is of complexity $O(\log N)$. Thus, in the case of ICP, the overall complexity becomes
Figure 3.3: Illustration of the world to laser frame kinematics chain. The robots corrected pose $j^c T_w$ is computed considering the base to laser frame transform $l_j T_{b_j}^{fk}$ from forward kinematics. We indicate in blue the pose prior, in magenta the correction, in black the corrected pose estimates, in orange the base to laser frame transform.

Legend: $w =$ world frame, $b =$ base frame, $l =$ laser frame, $i =$ reference frame, $j =$ reading frame, $ki =$ kinematic-inertial, $fk =$ forward kinematics, $icp =$ ICP.

$O(M \log N)$. As a result, the speedup as compared to the basic implementation scales with the number of points.

In practice, a main difficulty of the ICP approach is related to convergence to the correct solution, which is not guaranteed by more points or iterations. It depends instead on the distribution of points across the 3D space. Typically, ICP techniques are sensitive to high initial misalignment between the input clouds and rely on the assumption of full spatial overlap and uniform distribution of points. As discussed in Sec. 2.2.2, the assumption of full overlap is violated in many real-world situations, for example due to occlusions and reduced sensor FOV. In our work, we demonstrate that registration can be achieved also for initial misalignment > 10% of the cloud dimensions, and overlap as low as 10% by updating the outlier rejection filter at run-time.
3.2 Publication: ICRA 2017

Figure 3.4: Registration computation costs. We indicate with $C^A$, $C^B$, $C^C$, $C^D$ the reference and reading point clouds, each accumulated in time $t_{acc}$. The corrections $T^{icp}$ to align the reading clouds into the reference are integrated with latency $t_{icp}$.

3.1.4 Main Assumptions

In this chapter we make three main assumptions which are: 1) a prior estimate of the robot’s pose is available, 2) the drift accumulated between the reference and the reading pose is sufficiently limited to allow alignment and a realistic approximation of overlap (initial misalignment of more than 1 m$^2$ on the $xy$ plane and 30° in yaw was not limiting in our experiments), and 3) the environment is structured. Specifically, the point-to-plane optimization metric chosen for ICP-based alignment is better suited to structured environments.

3.2 Publication: IEEE International Conference on Robotics and Automation 2017

In this section, we present our laser localization approach in detail. We published this work as a contributed paper in the proceedings of the IEEE International Conference on Robotics and Automation (Nobili et al., 2017b).

Personal contributions include:

- Literature review.
- Evaluation of state-of-the-art methods and problem statement.
- Hypothesis development and experimental design.
- Implementation of the proposed method in C++.
- Data collection for Exp A, B, C, in collaboration with Raluca Scona. Data used for Exp D were collected during the DRC finals by the MIT team.
• Experiments after integration of the proposed algorithm in the robot controller (Exp C) in collaboration with Raluca Scona.

• Results analysis on all experiments.

• Primary author on paper content and figures. All authors contributed to update the content of the paper.
Overlap-based ICP Tuning
for Robust Localization of a Humanoid Robot

Simona Nobili, Raluca Scona, Marco Caravagna, Maurice Fallon*

Abstract—State estimation techniques for humanoid robots are typically based on proprioceptive sensing and accumulate drift over time. This drift can be corrected using exteroceptive sensors such as laser scanners via a scene registration procedure. For this procedure the common assumption of high point cloud overlap is violated when the scenario and the robot’s point-of-view are not static and the sensor’s field-of-view (FOV) is limited. In this paper we focus on the localization of a robot with limited FOV in a semi-structured environment. We analyze the effect of overlap variations on registration performance and demonstrate that where overlap varies, outlier filtering needs to be tuned accordingly. We define a novel parameter which gives a measure of this overlap. In this context, we propose a strategy for robust non-incremental registration. The pre-filtering module selects planar macro-features from the input clouds, discarding clutter. Outlier filtering is automatically tuned at run-time to allow registration to a common reference in conditions of non-uniform overlap. An extensive experimental demonstration is presented which characterizes the performance of the algorithm using two humanoids: the NASA Valkyrie, in a laboratory environment, and the Boston Dynamics Atlas, during the DARPA Robotics Challenge Finals.

I. INTRODUCTION

The primary input to a bipedal locomotion control system is a high frequency estimate of the robot’s state — the 6 degrees-of-freedom (DOF) pose of the robot’s pelvis and its joints configuration. The accuracy of the state estimate is critically important to facilitate effective control and to achieve greater autonomy by maintaining consistent and precise reference of the terrain and objects in the environment.

Approaches for state estimation which have been tested on humanoid robots fuse proprioceptive measurements from joint encoders, contact sensors and inertial sensors. Drift in the estimate is reflected by mis-alignment of consecutive point clouds. However, point cloud registration algorithms are highly sensitive to the properties of the input clouds, such as structural features (the presence of planar surfaces), the initial alignment error and the degree of overlap.

Overlap is influenced by multiple factors such as the presence of non-static elements in the scene, the viewpoint of the sensor/robot and its field-of-view (FOV). In addition, while the registration of consecutive point clouds leads to accumulated errors, the registration of the current cloud to a common reference prevents accumulated errors but becomes more challenging as the robot moves away from its original pose. Indeed, overlap decreases with the distance from the pose of the reference point cloud also due to occlusions and non-uniform sampling of the sensor.

In this paper we demonstrate how laser-based localization can be combined with a proprioceptive state estimator for a humanoid robot to fulfill the exacting accuracy and robustness requirements described in Section III.

We analyze the effect of point cloud overlap variation on the performance of Iterative Closest Point (ICP) alignment. In the case of human-like robots, one of the biggest challenges is introduced by the reduced FOV of the sensors available for exteroception (Figure 2). We define a parameter which describes overlap between two point clouds based on the relative positions of the sensor, the maximum range and the sensor FOV, as well as the distribution of points in the clouds.

We propose a strategy for non-incremental 3D scene registration in real environments, called Auto-tuned Iterative Closest Point (AICP). Having first pre-filtered the raw input point clouds to include macro-features such as planes and to implicitly exclude people and clutter, the algorithm automatically tunes the standard ICP outlier filter at run-time using the proposed overlap parameter to define the inlier matches set for the reference and reading clouds. We describe the

*The authors are with the Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, UK. {simona.nobili,raluca.scona, marco.caravagna, maurice.fallon}@ed.ac.uk

†Using the notation from [1], we refer to the ICP inputs as a reference cloud and a reading cloud, the latter to be aligned to the reference.
methods in detail in Section IV.

The localization system is evaluated on two full-sized humanoid robots, in Figure 1: the NASA Valkyrie in our laboratory, and the Boston Dynamics Atlas, using a dataset collected by the MIT team during the DARPA Robotics Challenge (DRC) Finals. We present extensive experimental results in Section V, which demonstrate the advantages of flexible outlier-rejection depending on the proposed overlap parameter.

II. RELATED WORK

A. Localization of Humanoid Robots

Proprioceptive state estimation for bipedal robots followed from original advances in the field of multi-legged robotics such as Roston and Krotkov [2]. For example, Bloesch et al. [3] first introduced an EKF-based state estimator for a quadruped which Rotella et al. [4] extended for bipedal state estimation.

A major focus for humanoids is achieving accurate center of mass (CoM) estimation relative to the supporting feet and accounting for errors in the modeled CoM. Xinjilefu et al. [5] directly estimated this offset using an inverted pendulum model to infer modeling error and/or unexpected external forces. Instead, the approach of Koolen et al. [6] modeled the elasticity of their robot’s leg joints to better distribute error. Our own prior work [7] utilized that elasticity model within an EKF filter to achieve low drift proprioceptive state estimation with the Boston Dynamics Atlas robot.

All of these approaches estimate the pelvis pose at high frequency (∼ 500Hz) by combining legs kinematics with IMU data. However these approaches, by their nature, will accumulate incremental drift over time. External sensing is often used to reduce or avoid this drift. Monocular cameras are perhaps the most commonly used exteroceptive sensors, with successful implementations of visual localization including [8] and [9].

In this paper we will instead focus on localization using scanning laser range finders (also known as LIDAR), as vision systems are not as accurate as lasers at long ranges. Hornung et al. [10] initially proposed a laser-based localization method for a NAO robot in a miniature 3D world model with extensions to include observation from a monocular camera presented in [11].

During preparation for the DRC, teams explored using LIDAR to reduce pelvis pose drift as it had a major impact on task level autonomy. In our previous work [7], we computed position measurements relative to a prior map using a rotating 2D laser scan on the Atlas robot. These measurements were integrated into our state estimate using a Gaussian particle filter at the 40Hz frame rate of the LIDAR. Koolen et al. [6] described their approach which instead used lower frequency ICP registration of full 3D point clouds.

Both approaches were demonstrated in the laboratory but unfortunately neither method could be used in the DRC Finals due to a lack of field testing and because the arena’s layout contained wide-open spaces with crowds of people. We feel that what was missing was the adaption and tuning of the baseline registration algorithms to these kinds of issues, as well as introspection to detect failures of the registration system.

B. Scene Registration

The Iterative Closest Point is one of the most commonly used techniques for point cloud registration. Its basic implementation involves the iterative minimization of the point-to-point distances between two point clouds to estimate the relative alignment [12]. Notable improvements to the original algorithm have been introduced in [13] with a point-to-plane error metric better suited for structured environments, and subsequently in [14] and [15].

Alternatively, the Normal Distributions Transform (NDT), introduced in [16], uses standard optimization methods (e.g. Newton’s algorithm) for the alignment. In [17] the authors analyzed the performances of ICP and NDT: although NDT was demonstrated to have a larger valley of convergence, it was found to be less predictable than ICP.

ICP makes the implicit assumption that the input point clouds are fully overlapping. This is violated in reality and is typically managed by defining a criteria to identify outliers in the correspondence set (e.g. [18]). Nevertheless, tuning this outlier filter is a critical task for the success of the alignment.

III. SYSTEM OVERVIEW AND REQUIREMENTS

The system with the modular configuration of AICP (Section IV) is shown in Figure 3. We present a localization strategy made up of two main components.

Firstly, a kinematic-inertial state estimator is used within the closed-loop locomotion controller (either [6] or [7]) and computes a stable but drifting estimate of the robot’s pelvis pose at high frequency (∼ 500Hz). Typical estimation drift is presented in Figure 9 for the continuous walking experiment. Over 200 secs of walking accumulated drift of 10 cm in translation and 5° in yaw. Error about the pitch and roll axes is negligible due to the IMU.
Secondly, the proposed AICP algorithm leverages the low drift state estimator to initialize the alignment to a common reference and properly filter the current point cloud. It updates the state estimate with a correction computed with respect to the global coordinate frame. This assumes that a reference cloud was first captured at the start of operation with the robot observing most of the scene in which we want to localize.

Both of the robots involved in our experiments use the Carnegie Robotics Multisense SL as their primary sensing unit, composed of a stereo camera and a Hokuyo UTM-30LX-EW planar laser — 40 scans per second with 30 m range — spinning about the forward-facing axis. Every 6 secs the laser spins half a revolution and a 3D point cloud is accumulated. The FOV is however occluded by a protective cover over the robot’s head meaning that a single point cloud consists of approximately 100,000 points from the forward facing hemisphere. The speed of rotation of the device (5RPM) is chosen so as to densely sample the terrain when walking. On a parallel thread, the correction is produced with a computation time of about 1 sec.

The proprioceptive estimator produces a high rate, low latency estimate without discontinuities while the exteroceptive registration can allow discontinuities (at a low rate) but aims to avoid global drift.

The main requirements we identify for such a localization system are (1) accuracy close to 1 cm on average in position and below 1° in orientation, (2) reliability in real semi-structured environments, and (3) registration to a single reference point cloud while supporting large translation offsets of as much as 14 m (∼ half the sensor range) and the resulting decrease in overlap, as in Figure 10.

IV. ROBUST LOCALIZATION

The ICP algorithm has 4 main phases: pre-filtering, data association, outlier filtering and error minimization (Figure 3). Pomerleau et al. [1] proposed a modular implementation of the ICP chain to provide the user with a protocol for the comparison of state of the art ICP variants. Their software is publicly available under the name of libpointmatcher\(^3\) and will be used as the registration framework in this work.

The authors identified two classical ICP variants based on [13] and [19] and use these as their baseline ICP configurations. Their results suggest that the point-to-plane variant, achieves better overall performance than point-to-point. Stable performance can be achieved if the alignment is initialized within a constrained basin of convergence, i.e. 10 cm and \(10^2\) initial error in 3D translation and rotation, and secondly if the overlap is constantly high while the robot moves in a structured environment.

In the following, we discuss the implementation of the AICP algorithm, which overcomes the limitations of the baseline ICP strategies in our real application.

A. Pre-filtering

We observe that the alignment of non-uniform point clouds is mainly influenced by denser regions (usually in proximity of the sensor). However, for the alignment to be successful surfaces at different distances should give a balanced contribution to the optimization process. Our pre-filtering approach is divided into two main phases. First, the two input point clouds are uniformly downsampled using a voxel filter [21] (the leaves size is set to 8 cm in our case). Second, we extract planar macro-features such as walls and large surfaces because:

- Planar surfaces are represented by a locally regular distribution of points, therefore in the case of slightly incorrect matching, the wrongly associated points still have a good chance of behaving like the correct ones.
- People and clutter are implicitly filtered-out.

We adopt a region growing strategy for plane segmentation [21]. A region is accepted only if it satisfies criteria about its planarity and dimensions (e.g. larger than \(0.30 \times 0.30\) m). This makes the filtering suitable for man-made environments at least. Figure 4 shows an input cloud before and after the pre-filtering phases. At this stage the remaining point cloud is uniform and has been filtered of clutter points, people in the environment, as well as small and irrelevant regions of the cloud, which as a result do not contribute to the alignment.

A comparison between our filter chain and the baseline ICP is presented in Table I.

B. Auto-tuned ICP

Including false data association matches is a common cause of ICP registration failure. Where points fall on people moving in the scene, on objects outside of the reading cloud’s FOV or which are occluded by other parts of the scene, no successful correspondence can be found and these points will then generate false matches.

The standard outlier filter is intended to reject false matches according to a criteria such as a maximum allowed distance or a fixed quantile of the distribution of closest points. For example [19] retains 70% of closest points in

\(^3\)https://github.com/ethz-asl/libpointmatcher

---

For operational simplicity we do not consider building a continuously expanding map using SLAM in this work.
Step | Baseline ICP | Description | AICP | Description
--- | --- | --- | --- | ---
Reference pre-filtering | MinDist RandomSampling SurfaceNormal | keep points beyond 1m random down-sampling, keep 10% normals extraction | Down-sampling RegionExtraction SurfaceNormal | uniform down-sampling region growing plane segment, normals extraction
Reading pre-filtering | MinDist RandomSampling | keep points beyond 1m random down-sampling, keep 5% | Down-sampling RegionExtraction OverlapParam | uniform down-sampling region growing plane segment, compute $\Omega$
Data association | KDTree | matching with approximation factor $\epsilon = 3.16$ (from [20], [1]) | KDTree | matching with approximation factor $\epsilon = 3.16$ (from [20], [1])
Outlier filtering | TrimmedDist | keep 10% closest points (fixed ratio $\approx 0.7$) | AutoTrimmedDist | keep auto-tuned percentage of closest points (ratio depends on $\Omega$
Error minimization | PointToPlane | point-to-plane | PointToPlane | point-to-plane

TABLE I: Comparison between the baseline ICP and the proposed AICP configuration.

![Fig. 4: Pre-filtering. Top: raw point cloud from Valkyrie’s dataset, people are circled in red. Bottom: after pre-filtering. People and small irrelevant features have been filtered-out.](image)

its trimmed outlier filter. We believe that these approaches are too general and produce unsatisfactory performance in practice. Using a fixed parameter assumes constant overlap and this assumption is violated in real scenarios. This is a critical limitation of the baseline ICP solution.

Instead we propose to dynamically vary the outlier filter ratio through analysis of the input clouds before registration. Crucially we take advantage of the low drift rate of the kinematic-inertial state estimator. In the following section we define a metric, $\Omega$, to quantitatively represent the overlap between the input clouds.

Intuitively, we envisage that the proportion of true matches after data association can be correlated with this overlap metric. In other words:

- If overlap is high, the proportion of true matches will be high and could reasonably be approximated by $\Omega$.
- If overlap is low, the proportion of true matches will be lower, therefore we need a conservative ratio for the outlier filter, with $\Omega$ again being a reasonable approximation.

C. Overlap Filter

We define $\Omega$ by taking into account the initial estimated alignment, range $r$ and field of view $\theta$ of the sensor. Being:

- $wP$ and $wQ$ the reference and reading clouds respectively, expressed in the world coordinate frame, denoted $w$. Each cloud is a set of points contained within a subspace of $\mathbb{R}^3$ delimited by the sensor range and FOV. We name these subspaces $V_i$ and $V_j$ respectively. Each subspace is a portion of a sphere centered in the sensor pose, with radius $r$, sectioned by two vertical planes defined by the horizontal FOV $\theta$. In the case of Valkyrie in Figure 2, $r = 30$ m and $\theta = 180^\circ$. We neglect the reduction in the vertical FOV.

- $i$ and $j$ the coordinate frames representing the sensor poses from which $wP$ and $wQ$ have been captured respectively. These frames are defined by the transformations $wT_w$, $wT_w$.

Consider the set $jP$ of points belonging to the reference cloud $P$, expressed in the coordinate frame of the reading cloud $j$, as well as the set $iQ$ of points belonging to $Q$ expressed in $i$, as:

$$jP = jT_w^{-1} P \quad iQ = iT_w^{-1} Q$$

With each point cloud represented in the coordinate frame of the counterpart, we can determine the subset of these points which lie within the counterpart sensor’s FOV. $S_j$ and $S_i$ are then defined as the sets of points living in the volume of intersection between $V_i$ and $V_j$:

$$S_j = \left\{ \forall p \in jP : \|p\| \leq r \land \arctan \frac{p_y}{p_x} \leq \frac{\theta}{2} \land p_x > 0 \right\}$$

$$S_i = \left\{ \forall q \in iQ : \|q\| \leq r \land \arctan \frac{q_y}{q_x} \leq \frac{\theta}{2} \land q_x > 0 \right\}$$

where $p = [p_x, p_y, p_z]^T$ and $q = [q_x, q_y, q_z]^T$ represent an individual point from each cloud. We define the overlap parameter as

$$\Omega = \frac{|S_j|}{|P|} \cdot \frac{|S_i|}{|Q|}$$

where $|\cdot|$ indicates the cardinality of a set. This metric is so defined under the assumptions that:

1) The pre-filtering strategy described in Section IV-A has removed points belonging to small elements and people in the original point clouds.

2) The initial alignment is within the basin of convergence. In our case, this assumption is satisfied as the drift rate of the state estimator is $1 \rightarrow 2$ cm per step and the correction is computed regularly (every $6$ secs).

We use $\Omega$ to set the outlier filter ratio, with special care for extreme overlap cases:

- Where $20\% < \Omega < 70\%$, the inlier ratio is set to $\Omega$.
- Where $\Omega < 20\%$, the inlier ratio is set to $0.20$.

Generally, at least $20\%$ closest matches are required for alignment optimization.
Fig. 5: Outlier Filtering. In each image, white points belong to the reference cloud, captured from the blue pose. The red points are accepted inlier matches and the green points are rejected outliers, all belonging to the reading cloud, captured from the yellow pose. Top: ratio = 0.20, Ω = 10%. Bottom: ratio = 0.70, Ω = 10%.

- Where Ω > 70%, the inlier ratio is limited to 0.70. Overlap is very high and 70% of the closest points are sufficient for the alignment.

This simple relationship between the overlap metric and the outlier ratio results in satisfactory performance in our experiments (Section V). As an illustrative example, Figure 5 (top image) shows the matches preserved in case of low overlap, filtered using a ratio of 0.20, given Ω = 10%. The accepted matches are true matches and as a result the alignment is successful. In contrast, in the lower image an inlier ratio of 0.70 was used and as a result many accepted matches are false and the alignment diverges.

V. EXPERIMENTAL RESULTS

So as to demonstrate the proposed approach we carried out a series of experiments with the Valkyrie and Atlas robots which correspond to over 60 mins of operation time in total:

a) Evaluation of how the proposed approach increases the basin of convergence relative to the baseline ICP approach.

b) Exploration of the effect of reducing the overlap between the model and reference point clouds showing that using our prior knowledge of the overlap increases the region of attraction of the core error minimization routine.

c) Demonstration of the algorithm running online on the robot where precise localization is essential to approach a target and to climb a set of stairs.

d) Finally, a demonstration of performance of the algorithm using a dataset collected at the DARPA Robotics Challenge Finals with the Atlas robot. The algorithm is successful in a semi-structured and crowded environment. The failure of the baseline ICP algorithm in this experiment motivated our work.

A view of the operation environments is in Figure 1. The relevant features of each dataset are presented in Table II.

4Additional demonstrations can be viewed at: robotperception.inf.ed.ac.uk/humanoid_estimation

### Table II: Features of the Valkyrie and Atlas datasets. Cells are colored red if the feature reduces the basin of convergence for alignment and green otherwise.

<table>
<thead>
<tr>
<th>Features</th>
<th>Valkyrie Datasets</th>
<th>Atlas Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOV</td>
<td>Reduced (180° x 120°)</td>
<td>Reduced (220° x 180°)</td>
</tr>
<tr>
<td>Dynamism</td>
<td>None</td>
<td>People, left side</td>
</tr>
<tr>
<td>Overlap</td>
<td>Exp.A,C: always ≫ 50%</td>
<td>Exp.D: decreasing to 10%</td>
</tr>
<tr>
<td></td>
<td>Exp.B: from 9% to 100%</td>
<td></td>
</tr>
<tr>
<td>Structure</td>
<td>Structured</td>
<td>Semi-structured, right side</td>
</tr>
<tr>
<td>Duration</td>
<td>Exp.A: 786 s</td>
<td>Exp.D: 1296 s</td>
</tr>
<tr>
<td></td>
<td>Exp.B: 1237 s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exp.C1,C2: 341 s, 50 s</td>
<td></td>
</tr>
<tr>
<td>Scene Area</td>
<td>(5.7 x 13.7 x 3.9) m</td>
<td>~ (14 x 11 x ∞) m</td>
</tr>
<tr>
<td>Start Pose</td>
<td>Shown in Figure 1</td>
<td>Shown in Figure 10</td>
</tr>
<tr>
<td>Vicon</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>

*Kinematic-Inertial State Estimator used in the control loop.

### Evaluation Protocol

The Valkyrie experiments were carried out in a laboratory with a Vicon motion capture system, used to generate ground truth. For a fair validation of our approach we analyze the performance of the AICP algorithm using the evaluation protocol proposed in [1], namely:

1) AICP is compared to a commonly accepted ICP baseline, which we denote BICP.
2) AICP and BICP are compared on large real world datasets from different environments.
3) Robust statistics are used to produce comparative error metrics.

In each case, we compare the estimated pose \( P_e \) to the ground-truth robot pose \( P_g \). Being the error \( \Delta P \) computed as

\[
\Delta P = \begin{bmatrix} \Delta R & \Delta t \\ 0 & 1 \end{bmatrix} = P_e P_g^{-1}
\]

the 3D translation error \( e_t \) is defined as the Euclidean distance given the translation vector \( \Delta t \):

\[
e_t = \| \Delta t \| = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}
\]

and the 3D rotation error \( e_r \) is defined as the Geodesic distance given the rotation matrix \( \Delta R \):

\[
e_r = \arccos \left( \frac{\text{trace}(\Delta R) - 1}{2} \right)
\]

We compare the error distributions using robust statistics (i.e., the quantiles for probabilities 0.50, 0.75, 0.95, which we indicate with Q50, Q75, Q95), which are indicative of accuracy and precision: results are accurate if these quantiles are close to zero, and precise if their difference is small. The choice of error metrics and statistics follows the evaluation convention in [1].

### A. Sensitivity to Initial Perturbations

As mentioned in Section I, the baseline ICP algorithm is sensitive to initial perturbations (errors in the initial alignment). Here we demonstrate that our proposed pre-filtering strategy increases the basin of convergence for AICP, with respect to BICP.
Fig. 6: Basin of convergence. An overhead view of the reference cloud is shown at the reference pose (top). Yellow boxes indicate sparse planes in the y direction. We performed tests in a $2 \times 2$ m basin, with yaw perturbations varying from $0^\circ$ to $90^\circ$ (color scale).

**Exp. 1:** We select two highly overlapping input clouds ($\Omega \gg 70\%$) and initialize their respective poses with a uniform-grid distribution of perturbations over 3 dimensions, i.e. x, y, yaw as shown in Figure 6.

We see that the baseline ICP has a small area of convergence with perturbations of 0.2 m or $10^\circ$ causing the algorithm to fail to properly converge. By comparison, the AICP algorithm is much more robust to perturbations in x and y and rotation in yaw. Successful alignment can be achieved with translation offsets of more than 0.8 m and up to $80^\circ$ in yaw rotation. In particular the pre-filtering strategy enlarges the valley of convergence along the y direction in this case. The main surfaces in this direction are only sampled sparsely due to the range and the axis of rotation of the sensor. In this situation the baseline ICP suffers from a weak contribution to the alignment along this axis.

**Exp. 2:** In their paper Pomerleau et al [1] proposed an experiment in which perturbations are randomly sampled from Gaussian distributions with increasing complexity. In this experiment we present results from such an analysis using 131 samples drawn from each of the following standard deviations:

- easy perturbation, EP - 0.1 m and $10^\circ$
- medium perturbation, MP - 0.5 m and $20^\circ$
- hard perturbation, HP - 1.0 m and $45^\circ$

In this experiment the robot continuously walked forward and back. Each new alignment is initialized with a sampled error and the result compared to ground truth.

In Figure 7 we show the cumulative distribution of errors for Valkyrie. Not only does AICP outperform the baseline ICP, but it produces reliable alignments in the easy and medium difficulty cases for all the runs.

### B. Sensitivity to Point Cloud Overlap

In Section IV we discussed the importance of properly tuning outlier filtering to account for variations in the point cloud overlap — particularly for sensors with limited FOV. Here we evaluate the sensitivity of AICP to variations in the overlap between the inputs.

Valkyrie walks and turns in place by approximately $130^\circ$ degrees in each direction. As a result the reading point clouds captured during this run have a large variation in overlap relative to the reference point cloud — between $9\%$ and $100\%$ (top plot, Figure 8) — making the alignment challenging. As mentioned in Section III, corrections from registration are fed back to the state estimate to initialize the next alignment. As a result the initial perturbation is negligible in each alignment and the result is mainly influenced by the current degree of overlap.

For the baseline ICP we see that when the overlap falls below $50\%$, alignment in both rotation and translation fails. In contrast the proposed AICP algorithm is successful throughout by selectively tuning the outlier filter to match the degree of overlap. Robust statistics are provided to demonstrate the distribution of errors (Figure 8).

The results show that AICP has the capacity to support non-incremental registration in spite of considerable overlap variations. In turn, non-incremental registration allows recovery from the less accurate alignments (e.g. between seconds 600–800).

### C. Online Integration of AICP

In our final Valkyrie experiments we demonstrate integration of the AICP algorithm within our closed loop walking system. These experiments are captured in the video accompanying this paper.

**Exp. 1:** Valkyrie walks repeatedly forward and backward towards a fixed target identified at the beginning of the run. Over the course of the experiment, the median error in translation and rotation estimated by our algorithm are 1.6 cm and 0.4$^\circ$ respectively (Figure 9). This satisfies the requirements about expected localization accuracy (requirement 1, Section III). Thanks to this localization performance, the robot reaches the target and maintains a precise pose estimate during the entire run. In contrast, in the case of
Fig. 8: Sensitivity to varying input clouds overlap. Boxplots on the right show the statistics Q50 (tick red bars), Q25 and Q75 (lower and higher end of blue rectangles), Q95 (top end of dashed lines).

Fig. 9: System integration. The blue line shows the kinematic-inertial typical estimation drift while in red we see the corrected estimate from the AICP algorithm.

A proprioceptive state estimation drift only, the robot fails to reach the target repeatedly due to continuous drift.

Exp. 2: Valkyrie is placed at 1 m distance from a staircase. The task is to walk towards it and climb up the steps. Planning is performed only once from the starting pose. Over the course of this 50 secs experiment, the median errors in translation and rotation are comparable to Exp. 1. This level of accuracy allows the robot to safely perform the task without needing to re-plan. In contrast, during the DRC robots typically took a few steps at a time to climb stairs or transverse uneven terrain, being paused periodically to manually re-localize and re-plan. In this context, our system was demonstrated to enable greater autonomy in task execution.

D. Localization during the DRC Finals

In this final experiment we test our algorithm against a dataset collected during a run by the MIT team at the DRC Finals (Pomona, CA, 2015) with the Boston Dynamics Atlas robot. The environment was a semi-structured area of about 14 × 11 m with walls on the right side of the robot and an open-space populated by a crowd of people (walking and sitting) on the left. The robot walks through the test scenario along a 16 m path while passing over uneven terrain and manipulating objects. The scene from Atlas’s point of view at the beginning of the dataset is shown in Figure 1. The presence of many people on the left side of the scenario is a challenge for registration and made localization difficult during the DRC.

Corrections from registration are used to update the state estimate and to initialize the next alignment — simulating closed loop integration. The performance of AICP, qualitatively evaluated from careful observation of the map after the run (Figure 11), is such that the computed trajectory is close to error free. People have been filtered-out and do not contribute to the alignment (top right), such that the system satisfies requirement 2. The algorithm is stable and robust enough to compute successful non-incremental alignments during the entire run (with more than 14 m displacement and overlap decreasing to just 10% — as shown in Figure 10), satisfying requirement 3. This experiment is captured in the video accompanying the paper.

Figure 12 shows the kinematic-inertial state estimator [7]
and the baseline ICP with errors computed against AICP-based localization. The baseline ICP fails for overlap less than 50% and demonstrates to be unreliable for complex scene registration. Its strategy is unsatisfactory as it adopts no specific measures to deal with the presence of people and to explicitly take advantage of the few structures available in the scene.

VI. Conclusions

In this paper, we proposed an algorithm for robust and accurate scene registration, which we name Auto-tuned ICP. We explored the degree to which the performance of the ICP algorithm is affected by the overlap between the input point clouds, as well as by the magnitude of initial perturbation between them.

We leverage the drifting state estimate derived from our humanoid state estimator [6]-[7] and develop a registration strategy based on careful pre-filtering, adjustment to overlap variation and non-incremental alignment. The proposed approach increases the basin of attraction of the error minimization step of ICP, allowing us to align to a single reference cloud. Consequently our approach avoids incremental error and recovers from failures.

Our algorithm overcomes the weaknesses of the baseline ICP identified in [1] in the context of humanoid localization. AICP satisfies all requirements identified in Section III.

Extensive experiments were demonstrated on two full-sized humanoid robots. Future work will focus on the extension of this solution to a SLAM system with failure recognition.

VII. Acknowledgments

This research was supported by the Engineering and Physical Sciences Research Council (EPSRC) and by the School of Informatics, University of Edinburgh. We would like to thank the Humanoid Team at the University of Edinburgh, as well as our collaborators at NASA, IHMC and MIT.

REFERENCES

3.3 ADDITIONAL RESULTS

In this section, we present additional results using our registration approach in urban settings. This is important to evaluate generalizability of the method to a different scenario than the ones discussed in the paper, such as vehicle driving.

In our previous experiments, we demonstrated that when variations in the overlap between the reference and reading point clouds occur, using this knowledge of the overlap measure increases the robustness of the registration routine. In this section, we show how similar results are achieved in a urban scenario.

For this, we use sequence 05 of the KITTI dataset (Geiger et al., 2013), which contains 2768 3D point clouds, and correspondent ground truth poses from a combined GPS/IMU system. The point clouds are captured using a Velodyne HDL-64E. As described in Sec. 2.5.2, each cloud from the Velodyne is three-dimensional and has 120 m range. The sequence was recorded at 10 Hz from a vehicle driving at speeds up to about 40 km/h.

Main differences between KITTI and the humanoid datasets include faster average robot speed of motion, wider FOV, longer range and higher sensing frequency for the laser, as summarized in Tab. 3.1.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>FOV</th>
<th>Range</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multisense SL</td>
<td>&lt; 220° × 180°</td>
<td>30 m</td>
<td>0.2 – 0.5 Hz</td>
</tr>
<tr>
<td>Velodyne HDL-64E</td>
<td>360° × 26.8°</td>
<td>120 m</td>
<td>10 Hz</td>
</tr>
</tbody>
</table>

Table 3.1: Characteristics of the different sensors in the datasets we consider.

**Performance in a urban scenario** We sampled 2447 point cloud pairs from the sequence at one frame per second. For each reference, we considered nine subsequent point clouds to be aligned into it. As a result the reading point clouds had a large variation in overlap relative to the reference – between 0% and 90%. We initialized the reading poses with random perturbations sampled from a zero-mean Gaussian distribution with 0.20 m and 20° variance, and compared the result of the alignment to ground truth.

Notice that, although the plane segmentation pre-filtering strategy helped point cloud alignment in our previous experiments by discarding clutter (e.g., objects and
Figure 3.5: Registration performance with varying overlap in a urban scenario. Both approaches perform successfully for overlap between 45% and 100%. The boxplots on the right show the median (tick red bars) and the quantiles 25% and 75% (lower and higher end of blue rectangles) of the distribution of errors. The blue lines approximate the distribution of errors for each of the two approaches.

people), in large scale urban environments static macro-features are predominant to clutter. Thus, in this experiment we did not perform plane segmentation. Instead, we found empirically that leveraging additional elements in the scene such as trees and parked cars leads to more robust results in urban environments.

In Fig. 3.5 we compare the performance of AICP to the baseline ICP, which we denoted BICP following the notation used within the paper. The plot illustrates the translation and rotation errors for varying overlap. Both approaches perform successfully for overlap greater than 45%. For the baseline, we see that when the overlap falls below 45%, alignment in both rotation and translation can be inaccurate or fail, resulting in median translation and rotation errors of 0.27 m and 0.20° respectively. AICP outperforms the baseline with median errors of 0.10 m and 0.11°, and is robust to overlap as low as 20%.

3.4 Conclusion and Future Work

In this chapter we proposed a strategy for accurate and robust point cloud registration, which we called AICP. AICP demonstrated increased basin of convergence, accuracy and robustness to severe overlap variations. This is achieved by automatic
3.4 Conclusion and Future Work

Figure 3.6: Closed-loop validation of AICP on Valkyrie (Exp.C1). The robot is commanded to walk towards the white stripe forth and back repeatedly. The robot’s start (and return) pose (a) is about 1.5 m away from the stripe. After several minutes of forward and backward motion, the robot can still reach the target stripe accurately (c).

In addition to the discussion in the experimental section of the paper, in Fig. 3.6 and Fig. 3.7 we show pictures taken during the online experiments on Valkyrie, which demonstrate the accuracy of localization. This experiment is also recorded in the video accompanying the paper.

In the following sections we discuss the limitations of our system, and the future work to overcome those limitations.

3.4.1 Limitations of the FOV-based Overlap Parameter

The overlap metric presented in this work was simply based on the shape of the sensor’s footprint (i.e., delimited by the sensor’s range and FOV), as shown in Fig. 3.8. Because of this characteristic, we call it FOV-based overlap metric. The figure illustrates how the region of overlap (shown in red) between two point clouds (shown in green and violet) is delimited by the intersection of the sensor’s footprint from the
Figure 3.7: Closed-loop validation of AICP on Valkyrie, for a staircase climbing task (Exp.C2). The robot is commanded to walk towards a staircase and climb three steps. A footstep plan is computed only once from the robot’s starting pose (c), using the planner by Deits and Tedrake, (2014). (a) and (b) show the robot’s mesh, current point cloud, and footstep plan visualization (b) in our user interface (Marion et al., 2016). The start pose is about 1 m away from the staircase. The localization performance are sufficiently accurate to allow the robot to execute the task in full, without needing to re-plan, and reaching the last step safely (c).

poses $i$ and $j$. The sensor’s FOV covers a volume $V$ which varies depending on the robot, from $180^\circ \times 120^\circ$ for Valkyrie to $220^\circ \times 180^\circ$ for Atlas.

However, this formulation only depends on the sensor’s FOV, rather than the current visibility, incurring some limitations. For example, the metric does not consider occlusions such as walls.

In Chapter 5, we define a novel overlap metric which takes into account the relative poses from which the clouds are captured, the structural features, as well as the free space information. We demonstrate that this formulation better suits real-world scenarios, particularly when occlusions occur.
3.4.2 Limitations of the Sensor

Our localization system produces pose corrections at variable frequency, depending on the laser spin rate. While the accuracy of registration depends on a minimum point cloud density (i.e., during accumulation, slower spin rates generate denser point clouds), accumulating a cloud over several seconds might be problematic because of the state estimator drift. During our experiments on Valkyrie, we observe about 0.03 m and 1.5° drift per minute, in 3D translation and rotation respectively. With the laser spinning at 5 RPM, a point cloud is available once every 6 seconds, resulting in negligible accumulated drift between corrections. Thus, for Valkyrie at the speed and dynamics of locomotion tested here, this issue has not been limiting.

Although at higher speeds it might be necessary to use a higher frequency laser, in Chapter 4, we explore how a spinning laser can be used for localization of a quadruped robot, during dynamic locomotion at speeds up to 0.5 m/s.
4.1 Introduction

Legged robots are expected to surpass wheeled platforms when it comes to navigating extreme terrains, such as cluttered disaster areas and outdoor trails. Locomotion of legged robots consists of intermittent feet contact with the terrain, allowing the robot to place footsteps individually and pass over obstacles. This comes at the cost of higher mechanical complexity, and additional challenges for state estimation. For example, the robot’s feet striking the ground during motion, or slipping on rough surfaces, cause acceleration spikes. The perception algorithms are challenged by the resulting sensors’ shaking, which causes image blur and cloud distortion.

In this chapter, we consider the state estimation of the quadruped robot HyQ, in the presence of complications caused by the robot’s dynamic motion, and visual challenges such as image blur and darkness, as shown in Fig. 4.1. Specifically, we focus on state estimation which is accurate (requirement 1.), and satisfies the robustness requirements 2. and 3. described in Chapter 1.

As mentioned earlier in the thesis, state estimation aims to produce an estimate of the robot’s trajectory which is continuously accurate and smooth, so as to make it suitable for integration in a robot’s control feedback. The estimate should have low drift, allowing the robot to maintain consistent and precise reference to the environ-
4.1 introduction

Figure 4.1: Example images captured from the HyQ’s left camera during industrial exploration. The images show feature-poor areas, image blur during motion (a), and poor lighting conditions at night (b), (c).

ment, and be functional across different scenarios. This can be achieved by leveraging multiple sensing modalities, such as proprioceptive and exteroceptive sensors, in a manner which is complementary and provides redundancy to the challenges encountered during navigation. For example, laser-based approaches often assume structured environments and are available at lower frequency, whereas vision techniques require sufficient lighting and features in the environment.

With this in mind, in the following sections we propose a multisensor state estimation strategy, which fuses kinematics, inertial, camera and laser measurements. In particular, our contribution consists in a method to fuse these sensor sources in a manner which is suitable for the feedback controller of a quadruped robot during dynamic locomotion.

4.1.1 Multisensor Fusion

The central block of the system is the kinematic-inertial state estimator which was developed in Camurri et al., (2017). The estimator processes joint encoders and IMU measurements into an EKF to produce pose and velocity state predictions at 1 kHz.

Visual odometry and laser localization pipelines run as separate processes to update the EKF state at different frequencies. Visual odometry estimates are computed at 10 Hz using FOVIS (Huang et al., 2011), which tracks image features in a frame-to-reference fashion. The laser localization module, based on AICP, produces pose
corrections at low frequency (0.5 Hz in our experiments) due to the accumulation time of a 3D point cloud.

4.1.2 Laser-based Localization

Earlier in the thesis we introduced AICP. AICP leverages a drifting state prior while being robust to larger initial point cloud misalignment, as compared to standard approaches. This characteristic represents a key advantage to enable localization of a dynamic robot. Indeed, as summarized in Tab. 2.4, the drift rate of a quadruped’s proprioceptive pose estimator is often higher than for less dynamic robots. Furthermore, AICP supports spatial overlap variations between the input clouds, enabling better adaptability to different scenarios. For example, during exploration of disaster zones overlap variations may be caused by structural occlusions (e.g., constrictions, clutter, or limited field-of-view).

For these reasons, we believe that AICP is suitable to help localization of a dynamic quadruped robot, and is the approach we use here for laser localization.

4.1.3 Filter State and Error Propagation

The robot body state vector includes linear position \( w\mathbf{x} \in \mathbb{R}^3 \) relative to the world frame \( w \), linear velocity \( b\dot{\mathbf{x}} \in \mathbb{R}^3 \) relative to the body frame \( b \), and orientation \( w\mathbf{\theta} \in \text{SO}(3) \):

\[
\mathbf{x} = \begin{bmatrix} w\mathbf{x} & b\dot{\mathbf{x}} & w\mathbf{\theta} & b^a & b^\omega \end{bmatrix}^T.
\]

Additionally, the state includes IMU acceleration and angular velocity biases, \( b^a \) and \( b^\omega \) respectively, which are computed when the robot is stationary to initialize the filter, and modelled as white Gaussian noise at run-time.

As discussed in Sec. 2.1.1.1, IMUs are affected by a variety of measurement errors. Effectively, to compromise between simplicity and modelling accuracy, \( b^a \) and \( b^\omega \) model only the constant biases, random walk error on the bias terms, and random walk error on the acceleration and angular velocity measurements.
We estimate the mean and covariance of the Gaussian distribution over the state $x$ using the EKF framework from Camurri et al., (2017). In particular, a state prediction is propagated using as inputs the acceleration and angular velocity estimated by the IMU. Subsequently, a filter state update is calculated via integration of the velocity measurements from the leg odometry.

The filter covariance is tuned to properly balance the fusion of the various signal sources. The contributions of visual odometry and laser localization are weighted differently: being typically more robust in our scenarios, the estimates from AICP are predominant – with the lowest covariance, and also used to reset the visual odometry reference frame. Visual odometry estimates are integrated at higher frequency but lower confidence, so as to reduce the kinematic-inertial drift while waiting for a pose correction from the laser. While for AICP and FOVIS the covariance values are set constant, an adaptive covariance is computed on the leg odometry, by leveraging both the consistency between the velocity contributions of stance legs and the detection of impacts. In this case, a higher uncertainty is set per stance leg in order to avoid conflicts with exteroception.

4.1.4 Main Assumptions

The main assumption we make is that the drift accumulated between the reference and the reading cloud during proprioception is within the basin of convergence of AICP.

4.2 publication: robotics: science and systems 2017

This work was published as a contributed paper in the proceedings of Robotics: Science and Systems (Nobili et al., 2017a), and is the result of a collaboration with the Dynamic Legged System (DLS) lab, led by Dr. Claudio Semini at the Istituto Italiano di Tecnologia (IIT)\(^1\).

Personal contributions include:

\(^1\) https://dls.iit.it/
• Literature review.

• Evaluation of state-of-the-art methods and problem statement.

• Hypothesis development and experimental design.

• Implementation of proposed strategy in C++ and tuning, in collaboration with Maurice Fallon.

• Experiments after integration of the proposed algorithm in the robot controller in collaboration with Marco Camurri.

• Results analysis on all experiments.

• Primary author on paper content and figures. All authors contributed to update the paper content.
Heterogeneous Sensor Fusion for Accurate State Estimation of Dynamic Legged Robots

Simona Nobili1∗, Marco Camurri2∗, Victor Barasuol2, Michele Focchi2, Darwin G. Caldwell2, Claudio Semini2 and Maurice Fallon3

1 School of Informatics, University of Edinburgh, Edinburgh, UK
simona.nobili@ed.ac.uk
2 Advanced Robotics Department, Istituto Italiano di Tecnologia, Genoa, Italy
{firstname.lastname}@iit.it
3 Oxford Robotics Institute, University of Oxford, Oxford, UK
mfallon@robots.ox.ac.uk

Abstract—In this paper we present a system for the state estimation of a dynamically walking and trotting quadruped. The approach fuses four heterogeneous sensor sources (inertial, kinematic, stereo vision and LIDAR) to maintain an accurate and consistent estimate of the robot’s base link velocity and position in the presence of disturbances such as slips and missteps. We demonstrate the performance of our system, which is robust to changes in the structure and lighting of the environment, as well as the terrain over which the robot crosses. Our approach builds upon a modular inertial-driven Extended Kalman Filter which incorporates a rugged, probabilistic leg odometry component with additional inputs from stereo visual odometry and LIDAR registration. The simultaneous use of both stereo vision and LIDAR helps combat operational issues which occur in real applications. To the best of our knowledge, this paper is the first to discuss the complexity of consistent estimation of pose and velocity states, as well as the fusion of multiple exteroceptive signal sources at largely different frequencies and latencies, in a manner which is acceptable for a quadruped’s feedback controller. A substantial experimental evaluation demonstrates the robustness and accuracy of our system, achieving continuously accurate localization and drift per distance traveled below 1 cm/m.

I. INTRODUCTION

For legged robots to be useful and eventually autonomous, they must be able to reliably walk and trot over a variety of terrains and in the presence of disturbances such as slips or pushes. They must also be able to perceive their environment and to avoid collisions with obstacles and people.

Legged robot control systems typically act to regulate the position, the orientation, and the associated velocities of the robot’s base or center of mass. This state vector is used for the planning of body trajectories, balancing and push recovery, as well as local mapping and navigation. Accurate and reliable state estimation is essential to achieve these capabilities, but it is a challenging problem due to the demands of low latency and consistency that high-frequency feedback control place on it. Meanwhile, impulsive ground impacts, aggressive turns and sensor limitations cause many modern exteroceptive navigation algorithms to fail when most needed.

Despite the improvements demonstrated by bipedal systems in the DARPA Robotics Challenge, for example [12], quadruped robots (Boston Dynamics LS3 [14], MIT Cheetah 2 [18], ANYmal [11]) present a more immediate solution to explore the parts of the world that are inaccessible to traditional robots.

∗Both authors contributed equally to this work.

II. RELATED WORK

There is a significant body of literature in state estimation and navigation of legged robots. As Ma et al. [14] described,
performance can be distinguished by multiple factors, such as the quality of the sensors, the dynamics of the robot’s motion, as well as the degree of challenge of the test environments and extensiveness of the testing performed. To that list we would add the quality of velocity estimation and suitability for use in closed loop control.

Exteroceptive and proprioceptive state estimation are often dealt with differently. Exteroceptive state estimate is closely related to Simultaneous Localization and Mapping (SLAM) and Barfoot [2] is an excellent resource in this area.

The motivation for proprioceptive state estimation is somewhat different for legged control system. Notably, Blösch et al. [3] presented a rigorous treatment of the fusion of leg kinematics and IMU information with a particular focus on estimator consistency, which becomes important when fusing very different signal modalities.

The method of sensor fusion we present is similar to that of Chilian et al. [6] which discussed stereo, inertial and kinematic fusion on a six-legged crawling robot measuring just 35 cm across — yet combining all the required sensing on board. It was unclear if computation was carried out on-board. The work of Chitta et al. [7] is novel in that it explored localization against a known terrain model using only contact information derived from kinematics.

With a focus on perception in the loop, the electrically-actuated MIT Cheetah 2 [18] produces impressive jumping gaits which are cued off of a LIDAR obstacle detection system. Because their work focuses on control and planning, the perception system used therein is not intended to be general nor it is used for state estimation.

The work of Ma et al. [14] is most closely related to ours in scale and dynamism of their robot. Their system was designed to function as a modular sensor head fusing a tactical grade inertial measurement unit with stereo visual odometry to produce a pose estimate for navigation tasks such as path planning. Robot’s kinematic sensing was only used when visual odometry failed. Their approach was focused on pose estimation and was not used within the robot’s closed loop controller. Their extensive evaluation (over thousands of meters) achieved 1% error per distance traveled.

For cost and practical reasons we wish to avoid using such high quality inertial sensors where possible. Our approach was developed with a MEMS IMU in mind. In all of our experiments we recorded both MEMS and Fiber Optic IMUs. In Section VII we present some initial results comparing the performance when using either sensor.

Finally, the estimator used in this work is based on a loosely-coupled EKF. This general approach has been applied to micro-aerial flight including Shen et al. [21] and Lynen et al. [13].

### III. Experimental Scenario

Our experimental platform is a torque-controlled Hydraulic Quadruped robot (HyQ, Figure 1) [20]. The system is 1 m long, and weighs approximately 85 kg. Its 12 revolute joints have a rotational range of 120°. A summary of the core sensors on the robot is provided in Table I. The 1 kHz sensors are read by our control computer (using a real-time operating system). All other sensors are connected to a perception computer and are passively synchronized with the real-time sensors [17].

The robot’s main exteroceptive sensor is the Carnegie Robotics Multisense SL which is composed of a stereo camera and a Hokuyo UTM-30LX-EW planar ranging laser (LIDAR). The laser produces 40 line scans per second with 30 m maximum range — while spinning about the forward-facing axis. Every few seconds, it spins half a revolution and a full 3D point cloud is accumulated. The stereo camera was configured to capture 1024 x 1024 images at 10 Hz and has a 0.07 m baseline. Within the unit, a Field Programmable Gate Array (FPGA) carries out Semi-Global Matching (SGM) [9] to produce a depth image from the image pair. The depth image is used to estimate the depth of point features in Section V-B as well as for other terrain mapping tasks. Figure 2 shows an example of a left camera image and a depth image taken during an experiment — indicating the challenging scenarios we target.

### IV. Requirements

The purpose of the state estimator is to produce a low drift estimate of the floating base of the robot model, which is typically the main link of the robot with the IMU rigidly attached to it. The estimate should have low latency (including transduction and data transmission) which is important for the velocity estimates used in the feedback loop of a controller. Low drift or drift-free state estimation is also used in navigation tasks (such as mapping and trajectory planning) as basic building block for many autonomous systems.

Our system is designed such that our core estimator requires only inertial and kinematic measurements to achieve low drift (with varying drift rates for different gaits). The additional sensing modalities of stereo vision and LIDAR can be incorporated in a manner which is complementary and provides

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Sensor Freq</th>
<th>Sensor Latency</th>
<th>Integration Freq</th>
<th>Integration Latency</th>
<th>Variables Measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMU</td>
<td>1000</td>
<td>&lt; 1</td>
<td>1000</td>
<td>n/a</td>
<td>ω̂x, ω̂y, ω̂z</td>
</tr>
<tr>
<td>Joint Encoders</td>
<td>1000</td>
<td>&lt; 1</td>
<td>1000</td>
<td>&lt; 1</td>
<td>ϕ̂x, ϕ̂y, ϕ̂z</td>
</tr>
<tr>
<td>LIDAR</td>
<td>40</td>
<td>10</td>
<td>0.2 - 0.5</td>
<td>600</td>
<td>ϕ̂x, ϕ̂y, ϕ̂z</td>
</tr>
<tr>
<td>Stereo</td>
<td>10</td>
<td>125</td>
<td>10</td>
<td>42</td>
<td>ϕ̂x, ϕ̂y, ϕ̂z</td>
</tr>
</tbody>
</table>

TABLE I: Frequency (Hz) and latency (ms) of the main sensors and for computing corresponding filter measurements.

Fig. 2: Example of left camera image and depth image produced by the robot’s stereo camera. This reflects the difficult lighting conditions and challenging structure of the test arena. The scene is illuminated with the sensor’s on-board lights.
redundancy to mechanical compliance and deformation in the terrain (e.g., mud or loose stones). As the exteroceptive sensors are captured with much lower frequency and higher latency (Table I), care must be taken in how their inputs are incorporated into the estimate.

V. APPROACH

We build upon an inertial-kinematic estimator recently described in [5]. In this section, we overview the core approach and use the same notation introduced therein. The 15 elements of the robot’s base link state vector are defined by:

$$\mathbf{x} = \begin{bmatrix} \dot{x}_b & \dot{y}_b & \dot{z}_b & \dot{\theta}_b & b_a & b_w \end{bmatrix}$$

(1)

where the base velocity $\dot{x}_b$, is expressed in the base frame $b$, while the position $x_b$ and orientation $\theta_b$ are expressed in a fixed world frame $w$ (the list of frames and their location on HyQ is depicted in Figure 1). The orientation is expressed as quaternion, but the attitude uncertainty is tracked by the exponential coordinates of the perturbation rotation vector, as described in [4]. The state vector is completed by IMU acceleration and angular velocity biases $b_a$ and $b_w$, which is updated by an EKF from [8].

Measurements of acceleration and angular velocity are taken from the IMU at 1 kHz. These are transformed into the base frame (subject to the estimated biases) to estimate the base acceleration $\dot{x}_b$ and angular velocity $\dot{\omega}_b$. Then, EKF is propagated using a direct inertial process model.

IMU biases are typically estimated when the robot is stationary and held static thereafter, as they are difficult to infer on a dynamic robot. When operating, the robot drift of the yaw estimate is a significant issue. We have typically used a Microstrain 3DM-GX4-25 IMU but more recently explored using the KVH 1775, a tactical grade IMU equipped with a Fiber Optic Gyroscope (FOG). For this reason, we compare the estimation performance of both IMUs in Section VII.

A. LEG ODOMETRY MODULE

Joint sensing contributes through a Leg Odometry (LO) module [5], which also runs at 1 kHz. During the filter update step, a measure for the base velocity $\dot{x}_b$, is computed as a combination of the individual velocity measurements $\dot{x}_b_f$ from each in-stance foot, as follows:

$$\dot{x}_b = -\dot{x}_f - \omega_b \times x_f,$$

(2)

where $\dot{x}_f$ and $x_f$ are the velocity and position of foot $f$ in the base frame, respectively.

As the robot is not equipped with contact sensors, we use the probabilistic contact classifier described in [5] to infer the combination of feet which are in stable and reliable contact. The velocity measure is then a weighted combination of the individual components, proportional to the probability of a particular foot being in reliable contact. An adaptive covariance associated with the velocity measurement accounts for harsh impact forces (up to 600 N when trotting) and helps ensure a smooth and accurate velocity estimate.

In experiments with trotting and crawling gaits, the proprioceptive estimator achieved drift rates of approximately 3 cm per meter traveled. This (or greater) accuracy is needed to build accurate terrain maps (in motion) and to allow the robot’s rear feet to achieve desired footholds (of 2–3 cm size) when sensed by the robot’s forward facing sensors.

B. VISUAL ODOMETRY MODULE

Visual Odometry (VO), and more broadly Visual SLAM, is becoming more feasible on legged platforms. This is enabled by more rugged sensors which are less susceptible to failure due to the dynamic motion of the robot. Nonetheless, certain types of robot motions (in particular ground impacts and aggressive turns) cause motion blur, especially in low light conditions.

Chilian et al. [6] suggested that leg odometry and visual odometry can be complementary, as difficult terrain often contains texture. In our experience however, locomotion struggles (such as with a mis-timed footstep) it instead induces motion blur and reduces VO performance (Figure 3). Latency is another important issue to consider. As stated in [14], a camera packet is typically received once every 50 inertial measurement packets.

Our visual odometry pipeline uses the open source implementation of FOVIS [10]. While its performance is competitive with more recent approaches, it could be straightforwardly replaced by a more recent VO system such as ORB-SLAM [15]. Its only input is a sequence of left/depth image pairs. It tracks FAST features in a key-frame approach so as to estimate incremental camera motion, from image frame $k-1$ to frame $k$ which we denote $\dot{T}_c(k-1,k)$, where $c$ indicates the camera frame. Using the known camera-to-body frame transformation $uT_c$, this can be expressed at the corresponding estimate of the motion of

Fig. 3: Visual odometry performance during a trotting sequence: the robot first trots forward at 0.3 m/s and then turns in place sharply over a 5 s period. During the initial trotting phase, VO performance is satisfactory. However, image blur causes the number of inliers to fall and mean re-projection error to spike. During this part of the experiment, no VO measurement packets are incorporated into the main motion estimate.
the body frame from \( k - 1 \) to \( k \) as:

\[
\hat{T}_b^{k-1:k} = \hat{T}_c^{k-1:k}(\hat{T}_c^{-1})
\]  

(3)

We have considered a number of ways of incorporating this information into the 1 kHz running estimate. The manner in which it is incorporated can conflict with other signal sources. Due to the accuracy of the gyroscope sensor, we currently incorporate only the translation element and as a result that orientation estimate can drift in yaw.

**Velocity measurement:** Initially we explored using the VO signal as a second velocity source. Operating the camera at its highest frequency (30 Hz), a measure of velocity can be computed by differencing the incremental motion estimate

\[
\dot{x}_b^{k} = \frac{b^k N_b - b^{k-1} N_b}{(t_k - t_{k-1})}
\]  

(4)

where \( t_k \) is the timestamp of image frame \( k \). While this signal does approximate velocity, this is unsatisfactory because of the low frequency and high latency of the camera.

**Frame-to-frame position measurement:** A more straightforward approach is to use this relative motion estimate to infer a position measurement of the robot relative to a previous state of the filter.

Taking the posterior estimate of the EKF filter corresponding to time \( t_{k-1} \), a measurement of the pose of the body at time \( t_k \) can be computed as follows:

\[
w^k \hat{T}_b = w^k T_{b}^{k-1} \oplus w^k \hat{T}_b^{k-1:k}
\]  

(5)

This can be incorporated as an EKF position measurement. Ma et al. [14] used this approach to estimate the robot pose estimate (at 7.5 Hz) and occasionally relied on LO when a failed VO frame occurred. Probabilistic fusion of redundant signal sources was not carried out. Instead, our goal is consistent estimation of position and velocity at high frequency, which makes subtleties of the integration important.

Consider Figure 4, which shows the position estimate of the robot while trotting. Overlaid on the figure are red markers indicating the timestamps of image frames. Any pose estimate computed using VO would be below the Nyquist frequency of the robot’s motion and demand very precise time synchronization.

**Position measurement over several frames:** We choose a less fragile approach which integrates the visual motion estimate over several image frames and to compute a compounded EKF position measurement. Specifically, we integrate the VO estimate for a \( N \)-frame window \( b^k \hat{T}_b^{k-N} \) to form a position measurement in the world frame as follows:

\[
w^k \hat{T}_b = w^k T_{b}^{k-N} \oplus w^k \hat{T}_b^{k-N:k}
\]  

(6)

where \( N \) is the number of frames used for integration (typically \( N \) corresponded to 2–3 s). This is similar to key-frame tracking where tracking for an extended duration can improve accuracy over frame-to-frame tracking. Finally the position portion of this measurement, \( w^k x_b \), is then used to create an EKF correction of the body position.

![Fig. 4: Height (z-dimension) of the robot’s base frame (top) and raw z-axis accelerometer measurements (bottom) while trotting. Indicated with red dots are timestamps of received stereo camera images. The bandwidth of the base motion is much higher than for many wheeled robots, while foot strikes cause acceleration spikes.](image)

**C. LIDAR-based Localization**

To incorporate information from the LIDAR sensor, we use Iterative Closest Point (ICP) registration of 3D point clouds to estimate the robot’s pose. Using the terminology of [19], this involves aligning a reading cloud to a reference cloud so as to infer the relative position of the sensor which captured the clouds. In particular, we want to measure (at time \( k \)) the relative pose \( w^k \hat{T}_b \) between the robot’s base frame \( b \) and the world frame \( w \), and then incorporate it as an observation in the EKF.

Registration of consecutive point clouds is often used to incrementally estimate motion, but it accumulates error over time. On the other hand, repeatedly registering to a common reference cloud is difficult when the robot moves away from its original position, as the overlap between the reference and the current cloud decreases over time.

In [16], we proposed a strategy for non-incremental 3D scene registration, which shows increased robustness to initial alignment error and variation in overlap. That work extended the libpointmatcher ICP implementation of Pomerleau et al. [19] with pre-filtering of the input clouds and automatic tuning of the outlier-rejection filter to account for the degree of point cloud overlap. The approach, called Auto-tuned ICP (AICP), leverages our low drift inertial-kinematic state estimate to initialize the alignment (Section V-A) and to compute an overlap parameter \( \Omega \in [0, 1] \) which can tune the filter. The parameter is a function of the maximum range and the field of view of the LIDAR sensor.

Here, we use the AICP framework to prevent accumulated drift and maintain accurate global localization. In our experiments, we could reliably register point clouds with only 11% overlap, which corresponded to a position offset of approximately 13 m.

**Forced reference update:** When the overlap drops dramatically, a reference point cloud update is required. In this work, we extend the AICP algorithm to trigger a reference point
cloud update when $\Omega$ decreases below the empirical threshold of 11%. When the threshold is crossed, the reference cloud is updated with the most recent reading cloud, whose alignment was successful. We follow three heuristics to determine if an alignment is successful. First, the mean residual point-wise error should be smaller than the threshold $\alpha$:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} r_i < \alpha \quad (7)$$

where $r_1, \ldots, r_n$ are the residual distances between the accepted matching points in the input clouds. Second, the median of the residual distribution, $Q(50)$, should be smaller than the threshold $\alpha$:

$$Q(50) < \alpha \quad (8)$$

Third, the quantile corresponding to the overlap measure should be also smaller than $\alpha$:

$$Q(\Omega) < \alpha \quad (9)$$

The first two conditions are commonly used metrics of robustness, while the third automatically adapts to the degree of point cloud overlap. The parameter $\alpha$ was set to 0.01 m during our experiments.

The limited frequency of the Hokuyo (40 Hz) and the speed of rotation of the sensor define the density of the accumulated point clouds. Increasing the spin rate reduces the density of each cloud. When trotting at 0.5 m/s, a sensor spin rate of 15 RPM corresponds to a new point cloud every 2 s — with the robot traveling about a body length in that time. Running on a parallel thread, the AICP algorithm produces a pose correction with a computation time of approximately 600 ms.

**VI. Implementation Design**

Filtering a heterogeneous set of signals with different frequencies and latencies requires careful consideration. A block diagram of our system is presented in Figure 5, with timing information for acquisition and integration in Table I.

At each iteration of the main 1 kHz inertial-kinematic loop, we calculate the prediction step and then immediately output the predicted state estimate ($\hat{x}^p$) to the control system to minimize latency. Subsequently, a velocity measurement is calculated using the 1 kHz leg odometry. This is applied to the filter as a Kalman update. These two components run in a single thread with no inter-process communication between them.

The visual odometry and the LIDAR registration modules operate at much lower frequencies and higher latencies. The VO pipeline takes no input other than the camera imagery and outputs the relative distance estimate at 10 Hz. The acquisition time for our stereo camera is significant (125 ms) — partially due to the SGM algorithm [9] running on the FPGA and image transport.

The LIDAR scans are received with much lower latency, but are then accumulated into a point cloud before the registration algorithm computes an alignment. The corrected pose estimate is then calculated and transmitted to the core estimator in the same manner as for VO — albeit at much lower frequency. Thus, both modules run as decoupled processes without affecting the core estimator.

**Considerations due to latency**: The implementation of the filter maintains a history of measurements so as to enable asynchronous corrections with significant latency — specifically the VO and LIDAR corrections. In Figure 6, we explain how this filter works with a toy example (for simplicity, leg odometry is left out of this discussion). In blue is the best estimate of the state over the history at that moment in time. In red is the effect of EKF update steps caused by measurements. In green are portions of the filter history which have been overwritten due to a received measurement.

**Event #1**: Before Event #1, the IMU process model will have been predicting the state of the robot until Time A. At this instant, a LIDAR correction is received which is based on LIDAR line scans collected over a period of several seconds stretching from Time B to Time C. This means that the position correction estimate from the LIDAR over that period is significantly delayed when it is finally computed. Also the accumulation is dependent on the accurate IMU+LO state estimate — which creates a coupling between these modules.

**Event #2**: The LIDAR measurement is incorporated as an EKF correction which produces the posterior estimate $T^C$ which causes the mean of the EKF to shift. The remaining portion of the state estimate is recalculated to incorporate the correction (such that the head of the filter becomes $T^A$). The green trajectory is overwritten (this is a crucial step).

**Event #3**: Over the next period of time the filter continues to predict the head of the estimator using the IMU process model. At Time D, a new visual odometry measurement is created which measures the relative transformation of the body frame between Time E and Time F as $h T^F_{E,F}$. This measurement is typically received with about 170 ms of delay.

**Event #4**: We wish to use this information to correct the pose of the robot towards $T^F$, as described in Section V-B. The key step is that this correction to the filter is carried out using the re-filtered trajectory (mentioned in Event #2). After the correction is applied, the head of the filter becomes $T^D$ and the estimator continues as normal.

The final sub-figure (on the right) shows the state of the head of the filter over the course of the example. This is the running
A. Experiment 1: Validation and Repeatability

The robot was commanded to continuously trot forward and backward to reach a fixed target (a particular line in Figure 7, top). Robot position and velocity estimates are used by the controller to stabilize the robot motion while tracking the desired position, as described in [1].

Periodically, the operator updated the target so as to command the robot to trot a further 10 cm forward. The experiment continued for a total duration of 29 min. At the end of the run, the robot had covered a total distance of about 400 m and trotted forward and backward 174 times. The configuration used on-line in the experiment was IMU-LO-AICP.

To measure body-relative drift we compute the average Drift per Distance Traveled (DDT) relative to the ground truth pose. The per-sample DDT is as follows:

$$\text{DDT}(k) = \frac{||\Delta \bar{t}_{k-N:k} - \Delta \bar{t}_{k-N:k}||}{\sum_{j=k-N}^{k} ||\Delta \bar{t}_{j-1:j}||}$$

which is the mean absolute position drift over the period $k-N : k$ (we used $10 \, \text{s}$) divided by the ground truth path integral of motion of the base link (the path integral tends to overstate the distance traveled and understate DDT). For an entire run, we calculate the median of this function, which is relevant because a continuously low DDT is required for accurate footstep execution and terrain mapping. For yaw drift, we use the median absolute yaw drift per second.

In Table II, we show the results for the four configurations using the KVH 1775. One can see that the IMU-LO-VO combination reduces the DDT relative to the baseline – in particular by reducing drift in $z$. IMU-LO-AICP removes global drift and keeps DDT below $1 \, \text{cm/m}$. Using all the sensors (IMU-LO-AICP combination) the drift is further reduced to $0.72 \, \text{cm/m}$. This result is comparable to the measurement noise of the Vicon system and satisfies our requirements.

Comparison between IMUs: We present the results for two different IMU configurations, using the industrial grade Microstrain 3DM-GX4-25 in addition to the KVH 1775. For the IMU-LO baseline, the median absolute rotation drift rate is an order of magnitude greater than for the KVH (0.119 °/s). However, by incorporating VO and AICP, we demonstrate that...
TABLE II: Median Drift per Distance Traveled (DDT) and Median Absolute Yaw Drift from Experiment 1 (see Section VII-A).

<table>
<thead>
<tr>
<th>Sensor Combination</th>
<th>Drift per Dst. Traveled [cm/m]</th>
<th>Median Yaw Drift [deg/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XYZ</td>
<td>X</td>
</tr>
<tr>
<td>KVH 1775 FOG</td>
<td>3.27 0.71 0.42 0.41 3.08</td>
<td>0.019</td>
</tr>
<tr>
<td>IMU-LO</td>
<td>1.67 0.80 0.48 0.43 1.30</td>
<td>0.021</td>
</tr>
<tr>
<td>IMU-LO-AICP</td>
<td>0.89 0.66 0.35 0.41 0.42</td>
<td>0.014</td>
</tr>
<tr>
<td>IMU-LO-VO-AICP</td>
<td>0.72 0.56 0.32 0.30 0.31</td>
<td>0.014</td>
</tr>
<tr>
<td>Microstrain GX4-25 MEMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMU-LO</td>
<td>3.63 0.97 0.70 0.55 3.47</td>
<td>0.119</td>
</tr>
<tr>
<td>IMU-LO-VO-AICP</td>
<td>0.78 0.58 0.35 0.31 0.36</td>
<td>0.016</td>
</tr>
</tbody>
</table>

TABLE III: Summary of the dataset used for Experiment 2, including log duration, size of arena, type of motion (F/B = forward/backward trajectory), laser spin rate, and terrain features.

<table>
<thead>
<tr>
<th>Name</th>
<th>Gait</th>
<th>Duration</th>
<th>Area m²</th>
<th>Laser</th>
<th>Ramp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log 1</td>
<td>crawl</td>
<td>889 s</td>
<td>20×5, F/B</td>
<td>3 RPM</td>
<td>✓</td>
</tr>
<tr>
<td>Log 2</td>
<td>crawl</td>
<td>675 s</td>
<td>20×5, F</td>
<td>3 RPM</td>
<td>✓</td>
</tr>
<tr>
<td>Log 3</td>
<td>trot</td>
<td>313 s</td>
<td>20×5, F/B</td>
<td>15 RPM</td>
<td></td>
</tr>
<tr>
<td>Log 4</td>
<td>trot</td>
<td>330 s</td>
<td>20×5, F/B</td>
<td>10 RPM</td>
<td>X</td>
</tr>
<tr>
<td>Log 5</td>
<td>trot</td>
<td>469 s</td>
<td>7×5, F/B</td>
<td>10 RPM</td>
<td>✓</td>
</tr>
</tbody>
</table>

we can reduce the rotation drift to be comparable with the KVH sensor (0.016°/s). Space limitations preclude a more detailed discussion.

The results presented here show that incorporating VO reduces the drift rate relative to the base line system, while adding AICP achieves localization relative to a fixed map. So as to test performance with uneven terrain and where the reference point cloud must be updated, a second series of experiments was carried out in a larger environment.

B. Experiment 2: Comparing Variants in a Realistic Scenario

The robot explores a 20 × 5 m² industrial area (Figure 7, bottom). It navigates over uneven and rough terrain (ramps and rock beds), crawling and trotting at up to 0.5 m/s. Turning in place (as seen in Figure 3) represents an extra challenge for the state estimation system. Lighting conditions vary dramatically during data recording, from bright light to strong shadows and from day to night-time. In some experiments, on-board lighting was used. The dataset is summarized in Table III and consists of five runs, for a total duration of 44 min and 300 m traveled.

No motion capture system is available in this space: to quantitatively evaluate the state estimation performance on the dataset, we built a prior map made up of a collection of 4 carefully aligned point clouds and we estimated drift relative to it.

Given a trajectory of estimated robot poses from an experiment, for every full laser rotation we align the point cloud to the prior map. To evaluate the accuracy of the estimated pose \( T_r \), we can estimate the correct pose \( T_c \) from this alignment, which we assume will closely match the true ground truth pose. The error \( \Delta T \) is computed as follows:

\[
\Delta T = \begin{bmatrix} \Delta R & \Delta t \end{bmatrix} = T_c T_r^{-1} \tag{11}
\]

with translation error as \(|\Delta t|\) and rotation error as the Geodesic distance given the rotation matrix \( \Delta R \), as in [19].

\[ a) \text{ Experiment 2a - Crawling Gait:} \] In Experiment 1, we have shown (while trotting) that integrating VO reduces the pose drift rate between the lower frequency AICP corrections. Here, we focus on the importance of using VO in addition to AICP.

Figure 9 shows the estimated error over the course of Log 1, recorded in the arena of Figure 8. The robot started from pose A, reached B and returned to A. The robot crawled for 40 m and paused to make 3 sharp turns. The experiment was at night and used the on-board LED lights.

During this run, the reference point cloud was updated 4 times. After 860 s, the state estimation performance had not significantly degraded, despite no specific global loop closure being computed.

In Figure 10, one can see that the median translation error was approximately 3 cm while the median correction made by the EKF was about 3 mm — both with and without VO. Because we do not observe a significant improvement in drift rates, we choose not to recommend using VO while crawling.

This is because of the lower speed of motion and the reduced drift rate of this less dynamic gait.

\[ b) \text{ Experiment 2b - Trotting Gait:} \] As mentioned previously, trotting is a more dynamic gait with a higher proprioceptive drift rate, which means that the VO could better contribute when combined with AICP. Empirically, this can be seen in the inset plot in Figure 8. In this case, the algorithm with VO produces a smoother trajectory (in green) than without (in yellow). This is important because the robot’s base controller uses these estimates to close position and velocity control loops. Discontinuities in the velocity estimate could lead to undesired destabilizing foot forces and controller reactions.

In brief, for the trotting logs (Logs 3, 4, 5) the integration of AICP allowed state estimation with an average 3D median translation error of approximately 4.9 cm (Figure 10, left). The integration of VO reduced the median translation error to 3.2 cm. Similar behavior can be seen for the magnitude of the position correction (Figure 10, right). These results demonstrate that continuous drift has been removed and that incremental drift is minimal.
VIII. DISCUSSION AND LIMITATIONS

As described above, the proposed system is able to overcome a variety of challenges and to support accurate navigation despite the dynamic locomotion gaits. The current system limitations are: a) the incremental error introduced by updates of the reference cloud, b) the frequency of the LIDAR sensor and resulting point cloud accumulation, and c) the susceptibility of the VO system to occasionally fail during short periods of poor lighting and the absence of visual features.

The system cannot recover from a) without a SLAM or loop closure strategy. Because of the overlap analysis, AICP allows us to change reference frame rarely, meaning that the drift in the demonstrated experiments is under one centimeter.

Depending on the LIDAR spin rate, AICP corrections occur at different frequencies, while accuracy is dependent on a minimum point cloud density. Accumulating a cloud over several seconds is problematic because of the state estimator drift. At the speeds of locomotion tested here, this issue has not been limiting, however at higher speeds a higher frequency LIDAR may become necessary.

Concerning the visual odometry module, failures during experiments occurred when there was limited illumination or motion blur (e.g., Figure 2). In these cases, the VO system merely resets until the next suitable frame is received.

IX. CONCLUSION

We have presented algorithms for the sensor fusion of inertial, kinematic, visual sensing and LIDAR to produce a reliable and consistent state estimate of a dynamically locomoting quadruped built upon a modular Extended Kalman Filter.

In particular we indicated how our approach supports dynamic maneuvers and operation in sensor impoverished situations. The reliability of our approach was demonstrated with dynamic gaits and speed up to 0.5 m/s. A particular technical achievement has been reliably closing the loop with this state estimator in dynamic gaits. During experiments lasting over one hour, our system demonstrated to be robust and continuously accurate with drift per distance traveled below 1 cm/m.

As we move forward with our testing, we will leverage the lessons learned here in more challenging experiments. We are interested in exploring more advanced visual mapping to allow the robot to recover visual localization after events such as sharp turns. Our initial testing indicates that many visual mapping systems do not adapt well to our test scenarios.

As mentioned in Section V-B, our current filter marginalizes out previous state variables. In future work we will explore using windowed smoothing to incorporate measurements relative to previous filter states.

REFERENCES


4.3 CONCLUSION AND FUTURE WORK

We presented a multisensor system for the state estimation of a walking and trot-tling quadruped robot. Our results showed how a heterogeneous sensor fusion approach (i.e., integrating high-frequency/drifting and low-frequency/low-drift estimates within the same filter), can generalize well against adverse motion conditions and visual challenges typical of real-world scenarios. Tab. 4.1 lists the system’s modules, together with their individual advantages and limitations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>KI</td>
<td>real-time frequency (250 – 1000 Hz)</td>
<td>drifting about 3 cm/m</td>
</tr>
<tr>
<td>FOVIS</td>
<td>high frequency (10 Hz)</td>
<td>textured scenes required</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sensitive to poor lighting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sensitive to image blur</td>
</tr>
<tr>
<td>AICP</td>
<td>sufficiently accurate</td>
<td>planar features required</td>
</tr>
<tr>
<td></td>
<td>robust to low overlap</td>
<td>low frequency (0.5 Hz)</td>
</tr>
<tr>
<td></td>
<td>robust to drifting pose prior</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Advantages and limitations of our state estimator’s individual modules.

In addition to the discussion in the experimental section of the paper, in Fig. 4.2 and Fig. 4.3 we present images taken during the online experiments on HyQ, showing the accuracy of the proposed localization system during closed-loop control. In particular, Fig. 4.4 shows the estimation error and 3-sigma confidence intervals over the course of the closed-loop validation against Vicon (Fig. 4.2). As described in Sec. 4.1.3, AICP corrections have the lowest covariance, thus every integration causes the confidence intervals to shrink. The results demonstrate continuously accurate localization and drift per distance travelled below 1 cm/m on highly dynamic gaits and speed up to 0.5 m/s. These experiments are also recorded in the video accompanying the paper.

4.3.1 Applications: Motion Planning on Challenging Terrain

In this section, we introduce some subsequent research at IIT which relies on our state estimation system, integrated on the HyQ robot. These include: motion planning for locomotion on rough terrain (Aceituno-Cabezas et al., 2018; Focchi et al., 2018), and
Figure 4.2: Closed-loop validation in a laboratory (Experiment 1). The robot is commanded to trot between one white stripe on the left and one on the right, back and forth several times at increasing speed. At every speed increment, the robot covers a wider interval (the lines are placed 10 cm apart). After several minutes of motion and the speed being increased from 0.2 m/s to 0.6 m/s, the robot can still reach the target line accurately (a) – (c).

Figure 4.3: Closed-loop state estimation in realistic scenario. The robot is commanded to trot from the starting pose (a) to the red stripe placed 5 m away. After several minutes of back and forth motion, the robot covered the total distance about 29 times, and can still reach the target line accurately (c).

reactive footstep placement during trot (Barasuol et al., 2015; Villarreal et al., 2018).

In Barasuol et al., (2015), the authors proposed a reactive control framework for trotting gaits on irregular terrain. The framework modulates the robot’s feet trajectories according to the terrain (for example to avoid stumbling on low obstacles). For this task, accurate terrain mapping is fundamental in order to avoid planning errors. In turn, mapping depends on the accuracy of state estimation. In this work, the authors used a motion capture system to track the robot’s body state with guaranteed accuracy, as no state estimation was available at that time. Recently, Villarreal et al., (2018) extended the framework to rely on online tracking, using our state estimator on HyQ. The estimator demonstrated sufficient accuracy to enable successful experi-
Figure 4.4: Estimated error with 3-sigma confidence intervals of the state estimator, using the configuration IMU-LO-VO-AICP. The run is referred to Experiment 1 and involved the robot trotting forward and backward to reach a fixed target in a laboratory environment.

Aceituno-Cabezas et al., (2018) proposed an approach to simultaneously plan footstep placement, gait transition and motion for the HyQ. Similar approaches optimize for a whole time horizon due to slow computation time (e.g., a few seconds to optimize for a 10 s horizon), and then execute the planned motion in open-loop, rather than optimizing online. In this work, the authors achieved reduced computation time between 0.5 and 1.5 s. However, it was still not possible to optimize online and perform re-planning, meaning that during experimental validation, the system relied heavily on the accuracy of HyQ’s state estimator to place footsteps. In Fig. 4.5, we show some of the trials described in the paper, where the robot successfully traversed slopes and gaps.

While Aceituno-Cabezas et al., (2018) assumed that a prior fixed map was available for footstep planning, Focchi et al., (2018) exploited the state estimator for online 3D mapping.
Future work will perform additional field testing integrating the proposed state estimator on the *ANYbotics* quadruped ANYmal, which recently arrived in our lab. We will carry out more experiments in larger scale challenging environments, such as an incident ground at a firefighting training facility. Furthermore, we are interested in extending our system to include loop closure detection and mapping.
Part II

RELIABILITY
5.1 INTRODUCTION

Reliable localization is fundamental for a robot to navigate over long distances, from a start location to a goal, without the intervention of the human operator.

In this chapter, we discuss our third contribution, which focuses on the reliability (requirement 4. described in Chapter 1) of laser-based localization during navigation in real-world environments. This involves predicting and preventing system’s failures during operation.

A common cause of localization failure is unsuccessful point cloud alignment. As discussed in previous chapters, the quality of ICP-based alignment heavily depends on selecting valid point correspondences between the input clouds during data association. In turn, the number of valid correspondences available for alignment depends on several factors. These include the degree of spatial overlap between the input clouds, as well as the mutually-visible structural features being available to constrain the alignment. For example, one unconstrained degree of freedom along the direction of motion could occur when trying to align within a long corridor. Fig. 5.1 illustrates cases of unconstrained alignments with simplistic geometric shapes.

We propose alignability as a measure of the constraints available for the alignment. If insufficient correspondences from non-parallel regions of the clouds are found, the
registration might converge slowly, converge to the incorrect pose (i.e., a local minimum), or diverge. Convergence to a local minima is challenging to identify a-posteriori.

Alternatively, in this work we focus on explicit analysis of the point clouds content prior to registration, in order to evaluate overlap and alignability. Based on these measures, we learn a model which can predict the risk of a failed alignment. Specifically, the model allows us to prevent unsuccessful point cloud alignment when the geometry is unstable and the overlap is not uniform.

5.1.1 Main Assumptions

As in previous chapters, the three main assumptions we make here are: 1) a prior estimate of the robot’s pose is available, 2) the drift accumulated between the reference and the reading pose is sufficiently limited to allow alignment and a realistic
approximation of the overlap and alignability metrics prior to registration (in our experiments we were not restricted by drift accumulated using the off-the-shelf wheel odometry prior from the Husky robot), and 3) the environment is structured.

Note that for simplicity, we perform our experimental evaluation on a mobile platform. Nevertheless, no assumptions have been made which explicitly limit the application of this approach to legged robots.

5.2 Publication: IEEE International Conference on Robotics and Automation 2018

This work was published as a contributed paper in the proceedings of the IEEE International Conference on Robotics and Automation (Nobili et al., 2018).

Personal contributions include:

• Literature review.

• Evaluation of state-of-the-art methods and problem statement.

• Hypothesis development and experimental design.

• Implementation of proposed strategy in C++.

• Data collection for Exp D, in collaboration with Georgi Tinchev.

• Experiments.

• Results analysis on all experiments.

• Primary author on paper content and figures. All authors contributed to update the content of the paper.
Predicting Alignment Risk to Prevent Localization Failure

Simona Nobili\textsuperscript{1,2}, Georgi Tinchev\textsuperscript{2} and Maurice Fallon\textsuperscript{2}

Abstract—During localization and mapping the success of point cloud registration can be compromised when there is an absence of geometric features or constraints in corridors or across doorways, or when the volumes scanned only partly overlap, due to occlusions or constrictions between subsequent observations. This work proposes a strategy to predict and prevent laser-based localization failure. Our solution relies on a novel measure of corresponding point clouds. We refer to the inputs to registration as a reference and a reading cloud with the latter to be aligned to the former.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) systems need to be able to reliably operate with a low failure rate for long periods of time and in a variety of environments. These systems should include fail-safe operation modes and auto-tuning capabilities so as to adapt to different challenges and achieve the vision of robot perception described in [1].

The work presented in this paper is focused on guaranteeing fail-safe operation of laser-based localization systems during exploration of cluttered man-made environments. State-of-the-art laser-based localization systems have been demonstrated to achieve low drift over long distances but can also be easily induced to fail in many real-world scenarios.

A cause of failure is the absence of geometric features which are necessary to constrain the alignment between two point clouds. For example, long corridors are unconstrained because of missing geometric features in one dimension. Failures can also occur when passing through doorways, or due to occlusions which cause large variations in the volume scanned by consecutive sweeps. We define \textit{alignability} as a measure of the capacity for two point clouds to be aligned given their geometric constraints, e.g., mutually visible planar surfaces.

In many cases sensors have a limited field-of-view (FOV) because they are physically integrated within a vehicle’s chassis. With an obscured FOV, the degree of overlap between consecutive scans made by the sensor is gradually reduced. Occlusions and constrictions in cluttered environments also introduce a large degree of overlap variation. This is a major problem for point cloud registration algorithms, as they are sensitive to the degree of overlap between input clouds. We believe that there is a need for failure prediction which is reliable to varying overlap.

The main contribution of this work is a method which predicts the \textit{risk of failed alignment} between two point clouds, which is learned as a function of the overlap between a reference and a reading cloud\footnote{S. Nobili is with the Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, UK. \textsuperset{2}All authors are with the Oxford Robotics Institute, University of Oxford, UK. \{snobili, gtinchev, mfallon\}@robots.ox.ac.uk} and the constraints available in the region of overlap between them. An example of unconstrained alignment is illustrated in Fig. 1. In contrast to previous works, [3], [4], [5], our method accounts for the degree of overlap between point clouds. It is also independent of point-wise data association and the registration approach used. The pipeline of our approach is shown in Fig. 2, which assumes that an initial (drifting) estimate of the sensor’s pose is available to initialize the alignment.

Our contribution is broken down as follows:

i) we define a novel overlap metric for 3D point clouds which takes into account the relative poses from which the clouds were captured, the structural features of the clouds, as well as the free space information. The metric gives improved performance with respect to previous work, particularly when occlusions occur,

ii) we derive an \textit{alignability} metric which quantifies the
degree to which alignment is constrained by exploiting the planar geometry commonly found in indoor environments. Our data-driven approach achieves a higher accuracy with respect to previous approaches.

iii) we learn a model which can predict alignment risk, based on these overlap and alignability measures. The model allows us to prevent registration failure when the geometry is unstable and the overlap is not uniform.

The remainder of the paper is structured as follows: Sec. II presents related approaches from the literature, Sec. III describes our algorithm in detail, Sec. IV presents an extensive experimental evaluation.

II. RELATED WORK

Iterative Closest Point (ICP) is a very commonly used method for 3D point cloud alignment. Its basic formulation [6] estimates the relative alignment between two point clouds iteratively, through four main steps: data filtering, data association, outlier rejection and minimization of a point-to-point distance function. Several improvements to the algorithm have been proposed over the past years which focus on changing the minimized distance function to suit different environments [7], [8], [9].

However, the optimal registration estimated by ICP is not always a good solution. This can be because of an unconstrained degree of freedom such as when crossing a doorway (Fig. 1), or trying to align within a long corridor.

Previous works [3], [4], [5] have stated that the stability of registration can be evaluated after the set of point matches have been selected, by Principal Component Analysis (PCA) on the covariance matrix used for error minimization. If the covariance matrix is not full rank the registration is underconstrained. However, this analysis depends on the data association step of the registration algorithm. When the overlap is low, the number of point matches available for data association might be insufficient for the eigenvectors of the covariance matrix to stabilize, thus leading to an unreliable measure of the constraints. Motivated by this, in our work we focus on analysis before the point-wise data association step of the ICP algorithm, and we explicitly account for overlap variation.

Zhen et al. [10] formulated a localizability measure for a sensor with respect to a prior 3D map of a structure of interest. It was computed offline by generating synthetic laser data to simulate observations from within the map. They used it to plan trajectories of a UAV so as to stay within areas with high localizability. Having a different application in mind, in our work we extend their formulation to a measure of alignability between two subsequent clouds. This is computed online to account for the current dynamics of the scene.

Pathak et al. [11] recognized the sensitivity of registration to low data overlap and formulated two overlap metrics which can be used to study the reason of alignment failures. The first metric used octrees to model occupied regions and then derive an overlapping surface area between two point clouds. The second measured the number of pixels on range-images mutually visible from both scans. The metrics were computed given the ground-truth alignment between the clouds. Herein, we use octrees to model occupied and free space and then derive an overlapping volume between two point clouds. This makes the estimate more robust to initial misalignment.

Pomerleau et al. [2] formulated an overlap measure using the ratio of points from a first cloud for which there was a corresponding point in the second cloud. This approach is asymmetric, for example, when the reference cloud is larger than the reading cloud. In contrast, we define overlap so as to be resilient to this asymmetry.

Finally, in our previous work [12] we proposed a strategy for non-incremental 3D scene registration, called Auto-tuned ICP (AICP). AICP extended a baseline ICP implementation [2] to more reliably register point clouds with reduced overlap by automatically tuning an outlier-rejection filter to account for the degree of overlap of the sensor’s footprint. This framework allowed accurate registration to a single reference point cloud despite significant motion by the robot. In this work we extend AICP with a more robust overlap parameter which corrects for some issues highlighted in Sec. IV.

In summary, our work differs from the state-of-the-art in that we learn a model for predicting alignment risk which is based on both the overlap and alignability parameters. We demonstrate the utility of our model in a general context, where no prior map is available and overlap between the input point clouds varies dramatically.

In a SLAM context, we address the problem of preventing failures from within the front-end module of the system, whereas other works focus on graph optimization strategies to remove outliers at a back-end level [13], [1].

III. PREDICTING ALIGNMENT RISK

In this section, we derive a continuous variable quantifying the risk of alignment failure when registering two point clouds. We first define measures for overlap and alignability of two point clouds. We then generate a meta-parameter which can be used to predict the risk of alignment failure.

A. Measuring Overlap

We define the overlap, \( \Omega \in [0, 1] \), between two point clouds using the initial estimated alignment between them (from odometry for example), their structural features and information about free space which is directly available given the sensor’s origin.

The reference cloud \( wC_i \) and the reading cloud \( wC_j \) are captured from two sensor poses \( i \) and \( j \), expressed in the world coordinate frame \( w \).
Using an octree structure [14], two corresponding octrees are constructed, \( w_{O_i} \) and \( w_{O_j} \). Each explicitly models both free and occupied space. From \( w_{O_i} \) and \( w_{O_j} \), another octree containing the set of common voxels is constructed, \( O^{ij} \), which defines the volume of overlap between the clouds.

We define the overlap parameter as

\[
\Omega = \min \left( \frac{|O^{ij}|}{|O_i|}, \frac{|O^{ij}|}{|O_j|} \right)
\]

where \(| \cdot |\) indicates the cardinality of voxels in an octree. Our volumetric representation of overlap is shown in Fig. 4 (right).

### B. Measuring Alignability

We define **alignability** \( \alpha \) as a measure of the geometric constraints which can be used to constrain alignment between a reference and a reading cloud. The alignment between a pair of 3D point clouds is well-constrained if the transform aligning them is constrained in all three dimensions. Intuitively, we envisage that at least three mutually visible non-parallel planes should exist.

In the following we describe the two steps of our strategy: (i) firstly, we match planes common to the input clouds and compute a matrix \( N \) as the set of normal directions which defines the volume of overlap between the clouds. We compute \( \alpha \) from PCA on the row vectors of \( N \).

#### Matching Plane Patches:

Having segmented the reference point cloud into a set of plane patches\(^2\), we select only the ones which belong to the volume of overlap between the clouds, \( O^{ij} \). We consider the \( u \)-th patch \( P_u \) from this set. Similarly, we define \( P_v \) from the set of patches extracted from the reading cloud.

For each pair \( P_u \) and \( P_v \), we compute a matching score \( \Omega_p \) which is defined as the degree of spatial overlap between the patches. We consider each patch as a set of points contained within a bounding box in \( \mathbb{R}^3 \). We define the bounding boxes as \( V_u \) and \( V_v \).

\( \Omega_p \) is computed from the set of points \( P_u^{ij} \) and \( P_v^{ij} \) belonging to the patches \( P_u \) and \( P_v \) and living in the volume of intersection between \( V_u \) and \( V_v \), as shown in Fig. 3.

\[
\Omega_p = \frac{|P_u^{ij}|}{|P_u|} \frac{|P_v^{ij}|}{|P_v|}.
\]

In Eq. (2) \(| \cdot |\) indicates the cardinality of a set.

The best match for \( P_v \) between any plane \( P_u \) is the one maximizing the overlap \( \Omega_p \). For a match to be accepted, both a condition on \( \Omega_p \) and on the maximum angle between the normals must hold. The normal directions \( n_k^T = [n_{kx}, n_{ky}, n_{kz}] \) are extracted per point \( k \in [1:M] \) (where

\( ^2\)We refer to plane patches as locally planar distributions of points. We adopt a region growing strategy for plane segmentation [15]. A patch is accepted only if it satisfies criteria about its planarity and dimensions (e.g., larger than 0.30 \( \times \) 0.30 m).

\( M \) is the number of points from all matched patches in the reading cloud, and \( N \in \mathbb{R}^{M \times 3} \) is defined as:

\[
N = \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ \vdots & \vdots & \vdots \\ n_{Mx} & n_{My} & n_{Mz} \end{bmatrix}.
\]

One could argue that plane matching is a variant of data association. The data association step of ICP typically involves a local point search within each pair of patches. Instead, we consider simple geometric models (the bounding boxes) to match the plane patches directly, instead of their points. This global search is easier to solve and does not depend on local density of the points. This approach will be further explored in future work by considering convex hulls as in [11].

#### Computing Alignability:

We consider the constraints imposed on the pose of the sensor by the current measurement with respect to a previously captured point cloud. Using a formulation similar to [10], we consider two point clouds captured sequentially in time and the constraint between a current measurement point \( p_j \) and a measurement point \( p_{i,t} \in \mathbb{R}^3 \) at time \( t \) in the past, both lying on the same plane:

\[
n_p^T(p_j - p_{i,t}) = 0
\]

where \( n_p \) is the plane normal. Given the robot position \( x \in \mathbb{R}^3 \), we can formulate a second constraint:

\[
x + r_j = p_j
\]

where \( r_j \) is the ray vector from the current measurement. Substituting Eq. (5) into Eq. (4) and combining the constraints imposed by all measurements in a sweep, we obtain the system of equations

\[
Nx = c_j
\]

where \( c_j \) is a constant vector \([c_1 \ldots c_k]^T\). The matrix \( N \) represents the constraints which exist between the reference and reading cloud from the current sensor pose. We can identify the unconstrained dimensions of the system in Eq. (6) by
Fig. 4: Left: FOV-based overlap parameter [12]. A reference (blue) and reading (green) cloud have been captured from the sensor poses i and j, the latter after crossing a doorway. The region of overlap (red) is delimited by the sensor footprint. In this case the wall between room A and B occludes the view between i and j and the region of overlap is over-estimated. Right: The proposed octree-based overlap parameter corrects for the issues occurring in previous work, particularly when occlusions occur. Two octree structures are constructed to model free and occupied space from the two clouds. The volume shown in red, identified by our parameter, reflects what we intuitively think of as volume of overlap.

Fig. 5: Alignment risk model learned from overlap and alignability estimates. The classifier has been trained on a set of 1200 binary labelled samples (1:failure/0:success). We show the predictions on the test set, where high risk of alignment failure (red) is expected for low overlap and alignability values, following a polynomial relationship. We observe that using only one parameter (overlap or alignability) is not sufficient, e.g., just using a threshold at 5% on alignability would still accept all samples with overlap below 30%, which would risk a faulty alignment.

The scattering parameter is defined as the probability of a scattering of N samples (1:failure/0:success). As shown in previous work, particularly when occlusions occur. Two octree structures are constructed to model free and occupied space from the two clouds. The volume shown in red, identified by our parameter, reflects what we intuitively think of as volume of overlap.

\[ \rho = f(\Omega, \alpha) \]  

We learned a model (Fig. 5) for \( \rho \) using a third degree polynomial Support Vector Classifier (SVC), which we found empirically to capture the function. Particularly, the prediction \( \rho \) is evaluated as the distance of a sample from the optimal hyperplane. Depending on the navigation application, the choice of \( \rho \) allows for operation at a preferred point on a Receiver Operator Characteristic (ROC) curve.

The classifier is trained on 1200 real-data observations, as described in Sec. IV. Each observation is registered using AICP and each alignment is manually labelled as a success or a failure.

IV. EXPERIMENTAL EVALUATION

So as to validate the study in this paper we carried out a series of experiments using the datasets in Tab. I, as well as a fourth dataset similar to Forum (IF) to train the classifier. The proposed metrics are compared with two baseline terms:

- **Inverse Condition Number (ICN):** the Condition Number is used to determine whether a linear system is ill-conditioned [17]. For the linearized system \( \arg \min_x \|Ax - b\| \), it is computed as the ratio of the minimum and maximum eigenvalues of \( A^T A \). For comparison purposes, we consider its inverse and denote it as ICN.

- **Degeneracy (D):** for the linearized system above, the degeneracy factor measures the stiffness of the solution w.r.t. disturbances to the constraints [5]. It is computed from the eigenvalues of \( A^T A \) as \( D = \lambda_{\min} + 1 \).

Our evaluation consists of four experiments:
A) An example comparison between the proposed point cloud overlap parameter $\Omega$ and the parameter proposed in our prior work [12]. We demonstrate that $\Omega$ is robust in the presence of occlusions.

B) A validation of the *alignability* factor with simulated data. We show that in our experiments *alignability* can predict geometric instability more reliably than *degeneracy*.

C) An evaluation of how the proposed measure of *alignment risk* (AR) outperforms $ICN$ and $D$ using two standardised datasets from [18]. The experiments created 2986 point cloud alignments and demonstrate the accuracy of our solution with respect to overlap variations.

D) A demonstration of the performance of our localization system on a third dataset, where the proposed method is essential when navigating along corridors and through constrictions and environmental clutter. The system is successful when travelling along a $\sim$180 m path. It prevents a total of 21 failures and allows the robot to return to the starting location with a final position error of 0.41% of the trajectory length.

In all our experiments we utilised the thresholds shown in Tab. III when predicting registration failure. We define a failure as 3D pose translational error greater than 0.02 m or rotational error greater than $1^\circ$.

A video to accompany this paper is available online.

### A. Example Illustrating Overlap

To demonstrate how the proposed octree-based formulation of point cloud overlap better suits real-world scenarios we compared it to our previous FOV-based overlap parameter [12], which was based only on the shape of the sensor’s footprint. We now define overlap based on the actual structure of the data rather than the simple sensor’s FOV, as discussed in Sec. III-A. A graphical comparison between the two metrics is shown in Fig. 4.

Consider the situation depicted in Fig. 6 (top). During exploration of an indoor environment the robot moves from one room to another through a doorway. At the start, the robot captures a point cloud (shown in blue) which will be used as the reference cloud. Thereafter the robot moves further away from the starting pose (Cloud 2) and the degree of overlap between the reference and the subsequent point clouds decreases – particularly after passing through a doorway.

The plot in Fig. 6 (bottom) shows that the FOV-based parameter over-estimates what we intuitively think of as overlap. As the robot enters the new room and turns to the left (Cloud 6), the overlap is clearly low. In contrast, the proposed octree-based parameter successfully identifies this decrease in overlap. Our subsequent experiments will solely use the octree-based overlap parameter.

---

![Fig. 6: Exp. A. An example where the basic FOV-based overlap is overconfident while the proposed octree-based parameter correctly estimates overlap. Top: Top-down view of the scenario in Fig. 1. A reference and a reading point cloud (blue and green) are captured from the first pose (black) and the current one (red), after crossing a doorway. Bottom: Estimate of overlap between the reference and a series of subsequent clouds. Due to its formulation, the FOV-based overlap is overconfident in such a scenario.](image_url)
and values below threshold indicate unconstrained geometry. Fig. 7 illustrates the result of the experiment. The average accuracy\(^3\) of the two parameters was 79.8\% for \textit{degeneracy} and 97.7\% for \textit{alignability}. The \textit{degeneracy} factor is computed after the point-wise data association step of ICP, which suffers from higher sensitivity to the initial perturbation. In the case of \textit{alignability} the estimates have lower variance and the results show better separability between constrained and unconstrained events. In our approach we analyse semantic representations (the matched planes), rather than points. This makes it less sensitive to the initial alignment error and more stable overall.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig7}
\caption{Alignability validation on the simulated problem in Exp. B. The green and red backgrounds show constrained and unconstrained geometries respectively. The boxplots show the quantiles 5\% and 95\% (bottom and top end of dashed lines), 25\% and 75\% (lower and higher end of blue rectangles), 50\% (red bars) of the distributions. Note that the boxes for \textit{alignability} are very narrow due to low variance.}
\end{figure}

\section*{C. Performance on Variable Overlap}

In Sec. III we discussed the importance of accounting for variations in spatial overlap when predicting point cloud alignment failures in real-world scenarios. Here we evaluate the sensitivity of failure predictions to variations in the overlap between input clouds. We provide a comparison between our \textit{alignment risk (AR)} against ICN and D.

In this experiment we use two publicly available datasets – Stairs (ST) and Apartment (AP), which consist of a Hokuyo UTM-30LX-EW planar laser mounted on a tilting unit. Ground-truth is available using an external tracking system.

For each dataset we perform a pair-wise alignment between the point clouds. We initialized the reading clouds with random perturbations sampled from a zero-mean Gaussian distribution with 0.10 m and 10\(^\circ\) variance. We compute the overlap and measure ICN, D, AR for each alignment. In order to predict a registration failure, we selected a fixed threshold for AR, which is the optimal one learned by our model. In the case of ICN and D a threshold had to be fine-tuned specifically for each dataset, as detailed in Tab. III. The results are presented in Fig. 9.

Fig. 9a,e illustrate the matrix of overlap estimates using our proposed \(\Omega\) parameter for both datasets. The diagonal elements show high overlap between each cloud with itself. Note that our estimates are symmetric by formulation.

Fig. 9b-d and Fig. 9f-h present the confusion matrices for ICN, D and AR, which are marked as illustrated in Tab. II.

Considering a localization task, we prefer predicting there will be a failure where there is none (false positive – FP) rather than the opposite (false negative – FN). In the regions where overlap is low AR predicts fewer false negatives than ICN and D. For ICN and D the number of point matches is insufficient for the eigenvectors to stabilize. Our approach which measures alignability and overlap, results in a robust prediction.

ICN and D often predict false positives along the diagonal despite high overlap. This suggests sensitivity of the parameters to initial perturbations. On the other hand, AR is accurate both for low and high overlap.

Fig. 8 shows the accuracy of failure prediction as a function of overlap for each of the three parameters. The results indicate that AR has higher accuracy overall. As the overlap reaches 30-40\% we note that the success/failure of the alignment becomes less predictable. Nevertheless, our approach performs competitively.

\section*{D. Online Performance Analysis}

AICP is used as the localization framework for this last experiment, because of the increased robustness to initial alignment error and variation in overlap. The approach leverages a low drift proprioceptive-based state estimate (for example we used the wheel odometry information) to initialize the alignment. In [19], we extended the AICP algorithm to trigger a reference point cloud update when overlap decreases below an empirical threshold. In contrast, in the system presented here, we aim for continuously reliable localization by applying a registration correction to the robot state estimate depending on the predicted risk of alignment failure. When the risk is high, we query a reference point cloud update and rely on proprioception until the next laser measurement becomes available. Furthermore, we replace the original FOV-based overlap parameter with our octree-based one.

We test our localization system on the IF dataset. Our dataset is collected by a Clearpath Husky mobile robot equipped with a Carnegie Robotics MultiSense SL. This sensor is composed of a stereo camera and a Hokuyo UTM-30LX-EW planar laser spinning about the forward-facing axis. Every few seconds it spins half a revolution and a 3D point cloud is accumulated. The speed of rotation of the device is set to 15RPM as a compromise between density of the clouds and accumulation time.

For this experiment we can align in three dimensions (x, y and yaw) and use wheel odometry to estimate roll, pitch and z. We predict registration failures using either D or AR, for comparison purposes. The robot navigates along a \(\sim 180\) m path while exploring the area shown in Fig. 10, which includes two cluttered rooms, two wide atria with high ceilings and a corridor. The exploration involves passing across constrictions such as doorways.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig8}
\caption{Exp. C. Accuracy of failure predictions with respect to overlap variation.}
\end{figure}

\(^3\)Accuracy is given by the number of true positives and true negatives by the total population.
The localization results, qualitatively evaluated from careful observation of the map built without loop closures in Fig. 10, are such that the estimated trajectory is close to error free when using AR. By aligning the last point cloud (red) into the first one (black) we estimate a pose error at the end of the run of 0.73 m in translation, corresponding to 0.41\% of the trajectory length, and 0.71° in rotation. This result demonstrates improved localization reliability when using AR for failure prediction. The parameter could predict and prevent all failures during this experiment, while being robust to geometric instability and overlap variation occurring in a real-world scenario.

In Fig. 11 we show the registration failures identified by D (top plot) and AR (bottom plot). D could not detect the alignment failures at locations 1, 2, 3, 4 which caused the system to lose track of the robot’s trajectory. As shown in the images in Fig. 11 (right) the mis-predictions occurred in situations where the overlap between the input clouds was low, with a consequent instability of the D factor. Due to these missed detections we could not complete the run on this dataset when using D for failure prediction.

V. DISCUSSION

The proposed approach is able to reliably prevent localization failures in real-world scenarios where geometric instability and overlap variations occur frequently and challenge point cloud registration algorithms.

In our work we recognized some fundamental advantages in using a learning based approach: i) during all experiments on different datasets we selected a fixed threshold for AR.

This threshold is the optimal one learned by our model. ii) depending on the application, we might want to be very robust to false negatives, preventing localization failures and hard recovery. The trade off for this scenario comes at the cost of accepting more false positives and reference cloud updates. In turn, each reference cloud update introduces a small incremental error to our estimate. This is useful when a low drift odometry prior is available, allowing us to
lower the AR threshold to accommodate more false positive predictions.

Additionally, we believe that our model could facilitate the search of loop closures in a SLAM system, by sorting the candidates by AR score prior to data association.

VI. CONCLUSIONS

In this work we proposed a strategy for point cloud alignment failure prediction. We explored the degree to which alignment failure is affected by geometric instability of the input point clouds, as well as the spatial overlap between them. We adopted a data-driven approach to evaluate the geometric constraints available for alignment and the volume of spatial overlap between the clouds. We used this data to learn a model to predict the risk of a failed alignment.

This allows us to be independent of the adopted registration strategy and point-wise data association, as well as to easily select the optimal threshold learned by our model in order to predict a failure, which avoids manual fine-tuning.

We evaluated our approach on different datasets and provided comparisons to existing techniques. Our algorithm overcomes the weaknesses of the baseline techniques identified in the context of real-world scenarios, where constrictions and occlusions cause reduced overlap between observations. We demonstrated how our approach can help improve the reliability of laser-based localization systems during exploration of unknown and cluttered man-made environments. In a large indoor exploration demo the system was able to reliably estimate the robot state with a final pose error of 0.41% of the trajectory length, and to build an accurate 3D representation of the environment.

Future work will focus on the extension of this approach to a SLAM system with loop closure detection.

REFERENCES


5.3 CONCLUSION AND FUTURE WORK

In this chapter, we discussed the reliability of laser-based localization, and proposed a geometry-driven approach to predict unsuccessful scene registration and prevent system’s failures. This is achieved by analyzing the content of the input point clouds prior to registration. Our experiments showed that, while evaluating either alignability or overlap individually is not sufficient, the proposed meta-parameter can predict the risk of a failed registration when the geometry is unstable and the overlap is not uniform. Furthermore, the use of a learning approach allowed us to reduce manual parameter tuning. The alignment risk threshold is the optimal one learned by the proposed model, which was functional during operation on different indoor datasets.

An overview of our complete localization system is shown in Fig. 5.2. The system includes AICP and the module for failure prediction which we proposed in the paper.

In the following sections we discuss advantages and limitations of our metrics, as well as the application of our approach to loop closure detection.

Figure 5.2: Proposed localization system. The system is composed of three main submodules. The proprioceptive state estimator produces a drifting pose prior (1). Depending on the robot, state estimation could be based on joint/wheel encoders and an IMU. Before AICP, the failure prediction module (green) evaluates overlap and alignability on the input point clouds $C^A$ and $C^B$, and predicts the risk of a failed alignment (2). If risk is low, AICP registration is enabled, and updates the pose prior (3).
5.3.1 Formulating the Overlap Metric using Octrees

We defined a novel overlap metric for 3D point clouds, which takes into account the relative poses from which the clouds were captured, the structural features of the clouds, as well as the free space information given the sensor’s origin.

An illustration of this is shown in Fig. 5.3, where we consider the toy problem of a rectangular room. A laser is located at pose $i$ inside the room, and captures a reference and a reading point clouds. Although the sensor does not move, the estimated pose prior drifts from $i$ to $j$ over time, resulting in initial misalignment between the clouds. The overlap metric is computed by constructing an octree structure (constituted of occupied and free voxels) for each of the two clouds, and considering the volume of overlap (common voxels). Notice that since the volume of overlap is computed before registration, it accounts for the initial alignment error.

In our work, we recognized a critical advantage in modelling both occupied and free space information.

In the toy scenario discussed above, considering solely the occupied voxels would be insufficient to approximate the volume of overlap, due to the initial alignment error.

Figure 5.3: Octree-based overlap metric. Top-view of a toy scenario. Left: reference (violet) and reading (green) point clouds projected to poses $i$ and $j$ in the world frame $w$, respectively. The accumulated drift between $i$ and $j$ is shown in magenta. Right: corresponding octree structures and volume of overlap (red) estimated by the the proposed octree-based metric. The metric is also illustrated in Fig. 4 of the paper with a real-world example.
5.3 Conclusion and Future Work

We defined a novel alignability metric, which is a measure of the mutually-visible geometric constraints available to constrain alignment between two point clouds.

The first step to compute alignability consists of finding matching planes between the clouds. This step involves extracting planar patches and fitting them inside rectangular bounding boxes $\in \mathbb{R}^3$, to then compute the volume of intersection between every pair. The use of bounding boxes for this task is experimental, with dimensions which are roughly related to the sensor’s noise. A plausible alternative to bounding boxes would be using convex hulls to better represent the distribution of points. Nevertheless, the matching score is computed from the points belonging to the volume

\[ \Omega \]

This effect is mitigated in our formulation, where considering both occupied and free voxels results in a better approximation of the volume of overlap, leading to $\Omega = 60\%$ in Fig. 5.3.

5.3.2 Considerations relating to the Formulation of Alignability

We defined a novel alignability metric, which is a measure of the mutually-visible geometric constraints available to constrain alignment between two point clouds.

Figure 5.4: From the toy scenario in Fig. 5.3, we show an example where, due to initial alignment error, the volume of overlap (red) is underestimated when considering only occupied voxels. In our formulation of overlap this problem is mitigated by considering both occupied and free voxels.

\[ \Omega \ll 5\% \]

The value of overlap $\Omega$ can be computed from Eq. (1) in the paper, given the prior poses $i$ and $j$ and the room size in voxels.
of intersection, rather than being directly related to the volume’s shape itself.

The proposed approach does not explicitly account for rotation instabilities. For example, if we consider a cloud constituted of a set of points distributed on a sphere, the scattering parameter from PCA would be high. However, this point cloud would have three unconstrained rotations, as illustrated in Fig. 5.1 for the sphere.

This fact was not limiting in our experiments, as predominant curved shapes are not luckily to appear in real-world point cloud data.

Furthermore, one should notice that rotation instabilities are implicitly accounted for and filtered out by selecting only planar patches in the strategy to measure alignability. Also, for most applications an IMU can be used to constrain the pitch and roll dimensions with respect to the direction of gravity.

5.3.3 Applications: Leveraging Alignment Risk for Loop Closure Detection

In this section, we introduce preliminary work which has been carried out during a research project I supervised at the Oxford Robotics Institute (ORI), University of Oxford (Nascimento, 2018).

Fig. 5.5 shows a reconstructed map of the IF dataset, which was also used for experimental evaluation in our paper. Mapping was conducted using iSAM for graph optimization (Kaess et al., 2007). During mapping, loop closures were detected using the proposed alignment risk measure.

This solution is a dense method for loop closure detection, i.e., does not require the extraction of dedicated features. An advantage of using this approach it implicitly selects loop closures with sufficient alignability.

Future work will evaluate the proposed failure prediction approach in different scenarios and with different sensors, such as outdoor environments and lasers with a wider FOV. We are interested in integrating our complete localization system (shown in Fig. 5.2), including the module for failure prediction, on a legged platform, for additional field testing and evaluation. Furthermore, we will further explore using the alignment risk measure to detect loop closures and extend the system to perform mapping.
Figure 5.5: Top view of a map reconstructed using the alignment risk measure to detect loop closures. The path taken was 1, 2, 3, 4, 5, 3, 2 and 1. Loop closures were detected in rooms 3, 2, and 1. Photo credit: Nascimento, (2018).
Part III

END-TO-END LEARNING:
TOWARDS GENERALIZABILITY
6.1 INTRODUCTION

Laser-based localization tasks, including scene registration and odometry, are typically formulated as geometric problems, and solved using rigorous engineering techniques. Such solutions often rely on the assumption of prior knowledge about the structure of the environment (e.g., indoor, outdoor). They also depend on sensor features, temporal and spatial vicinity between the point clouds, and require additional human effort for parameter tuning.

Fig. 6.1a illustrates the main steps of a conventional scene registration pipeline for laser odometry, where each module needs to be configured individually to function in specific situations.

A key feature still missing from current localization systems is the capability to gain general understanding about the data, and to learn from previous experience to improve performance. In this sense, neural networks have the potential to narrow the gap between raw sensor inputs and understanding.

In this chapter we present our fourth contribution. Motivated by the work on deep visual odometry presented in Wang et al., (2017a) and Wang et al., (2017b), we investigate whether it can be beneficial to employ a deep learning architecture to tackle the laser odometry problem in an end-to-end fashion, as shown in Fig. 6.1b, effectively
Figure 6.1: Comparison between laser odometry architectures. (a) shows an example of standard geometry-driven laser odometry pipeline. The shaded box indicates the additional feature extraction step performed by sparse methods. (b) illustrates a data-driven architecture, which trains a deep neural network to regress relative poses from a sequence of point clouds. The network itself can model sequential dependencies and account for incremental motion.

reducing engineering effort, and aiming for better generalizability to different environments and sensors. This comes at the cost of a more involved procedure to collect and organize a sufficiently large and broad variety of data to guide the learning. In this sense, the approach is referred to as being data-driven.

The main contributions of this chapter include: 1) we explore how a deep neural network can estimate laser odometry. Differently from related work, we train the network to learn from a representation of laser point clouds, rather than RGB camera images. We provide experimental evaluations using different input representations and loss functions. 2) we present results which in our experiments demonstrate improved accuracy using laser as compared to RGB camera images. 3) we present preliminary results comparing the performance of a baseline geometry-driven laser odometry approach using ICP to the proposed data-driven approach.

The rest of the chapter is organized in four main sections. In Sec. 6.2 we describe the state-of-the-art of traditional laser odometry, as well as learning approaches ap-
plied to robotic navigation tasks. In Sec. 6.3, we present our approach, followed by experimental results in Sec. 6.4. Finally, we conclude in Sec. 6.5.

6.2 RELATED WORK

Laser-aided navigation has been extensively investigated in robotics. In the literature, successful performance have been achieved in odometry and localization using rigorous engineering principles based on geometry.

6.2.1 Geometry-driven Laser Odometry

Odometry is a fundamental problem in robotics which can be dealt with using laser sensors and scene registration techniques. Standard registration methods (Pomerleau et al., 2013; Zhang and Singh, 2014) are based on modular frameworks, and tackle specifically the complications which might arise depending on the scenario and the sensor characteristics.

Let’s consider the data association and outlier rejection steps. The proportion of inlier point correspondences after data association is correlated with the spatial overlap between the point clouds. Thus, point cloud registration is particularly sensitive to overlap variation. If the sensing frequency is relatively low, for example when using the spinning planar laser described in Sec. 2.5.2, overlap variations between accumulated 3D point clouds can be severe, and need to be treated specifically (Nobili et al., 2017b).

In the case of sparse registration methods, the feature extraction module is typically designed to operate on specific sensor data. In order to recover laser odometry, in Dong and Barfoot, (2013) and Anderson and Barfoot, (2013), SURF features are extracted (Bay et al., 2006) from laser intensity images, and matched across subsequent frames. This technique functions under the assumption that sufficiently dense point clouds are available. On the other hand, the systems proposed in Bosse and Zlot, (2009a), and similarly LOAM (Zhang and Singh, 2014), rely on geometric features in Cartesian space, such as edges and corners, thus relaxing the requirement on cloud density.
In the case of dense registration methods, different optimization policies might be more effective depending on the application. Pomerleau et al., (2013) showed that while point-to-plane optimization is better suited to structured environments, and in general is superior to point-to-point, it can be less precise for large initial misalignments and loses its advantages in unstructured environments.

In contrast with traditional approaches, we are interested in training a neural network to regress laser odometry in an end-to-end fashion, rather than modularly, so as to avoid fine-tuning and to relax task-specific assumptions.

In the following section, we overview the machine learning process and in particular the development of artificial neural networks (NNs), from the early stages of digit recognition, to the ImageNet (Deng et al., 2009) challenge on image classification, to recent research which employed neural networks for robotics applications.

6.2.2 Advent of Machine Learning and Application to Robotic Navigation

6.2.2.1 Tools Overview

Machine learning describes the process of fitting a parameterized model to observed data points. A machine learning model typically operates in two phases: during training time the model is presented a set of training data and iterates over it while adjusting its model parameters to match the observed mapping as closely as possible. Afterwards at test time, the trained model predicts labels for a new set of unobserved data. The model’s performance on a fixed test set is typically used as an indicator for its overall performance. The training process can either be supervised or unsupervised. In supervised learning scenarios, the training data consists of pairs \((x, y)\) and the model is supposed to learn a mapping which assigns the correct label \(y\) for each input \(x\) in the training set. In unsupervised learning scenarios, the training data is not fully observed, e.g., the label set may be known, but not every training sample \(x\) carries a label. Depending on the nature of the label space a machine learning problem usually translates either to a classification (discrete label space) or a regression task (continuous label space).
In recent years, one family of machine learning models yielded tremendous successes: artificial neural networks. A NN is an interconnected architecture of neurons (i.e., nodes) inspired by the human brain. In its basic implementation, the network processes input data through a series of hidden layers of nodes, and produces an output which is a non-linear function of the sum of the inputs. Learning is achieved by computing the derivative of a loss function (which models the difference between the network’s prediction and ground truth labels), and back-propagating the errors to update the parameters in the network’s layers iteratively.

A key property of NNs is that they represent an arbitrary non-linear function and can hence learn very complex input-output mappings. Furthermore, NNs are entirely differentiable which allows the computation of gradients with respect to all of their parameters.

**Convolutional Neural Networks** While in standard neural networks the nodes on a single layer are not connected and function independently, convolutional neural networks (CNNs) (one of the first was proposed in (LeCun et al., 1998)) have neurons in each layer arranged in three dimensions (width, height and depth) and are connected to the previous layer in a sliding window fashion (i.e., through convolutions), rather than fully-connected. The architecture is named after the convolution operator, which allows CNNs to leverage spatial correlations between cells in dense grids, such as images.

**Recurrent Neural Networks** The main feature of recurrent neural networks (RNNs) (Schmidhuber, 1993) is the capability to handle sequential data in a principled manner. A RNN iterates over every element of a data sequence in a loop, allowing information to persist such that each output depends on the current input as well as on the previous iterations. The capability of connecting previous information to the present state is fundamental for many applications, for example in a language processing task remembering previous words in a sentence might help the prediction of the next term.

Hochreiter and Schmidhuber, (1997) first introduced Long Short-Term Memory (LSTM) networks, which maintain the recurrent structure of RNNs, but contain four
interacting memory layers. Because of this configuration, LSTMs are capable of learning longer-term dependencies.

6.2.2.2 Literature

Early stages of artificial neural networks date back to the work by Rumelhart et al., (1986), which showed that a neural network with multiple layers could be trained through error back-propagation to learn non-linear functions, effectively overcoming the limitations of previous approaches (Rosenblatt, 1958). The algorithm allowed one of the first CNNs to be trained to recognize handwritten digits (LeCun et al., 1998). However, it did not scale well to larger problems.

Because of this, traditional machine learning approaches such as SVMs (Cortes and Vapnik, 1995) became the preferred method for the next decade.

In recent years, deep neural networks (hence the term deep learning) achieved state-of-the-art performance in computer vision, surpassing traditional approaches. When the ImageNet dataset (Deng et al., 2009) was released, containing millions of labelled images for image classification, remarkable success was achieved on this task using deep neural networks, including AlexNet (Krizhevsky et al., 2012), VGG (Simonyan and Zisserman, 2014), GoogleNet (Szegedy et al., 2015), and ResNet (He et al., 2016).

6.2.2.3 Applications

Although a large part of the research effort in the past has been focused on image classification and object detection tasks in the computer vision domain, deep learning is becoming increasingly attractive to the research community for a variety of other applications, including robotic navigation.

With a focus on navigation using laser, one of the most explored applications is place recognition, where machine learning techniques have been employed to replace individual modules within the standard system architecture, including feature extraction, feature description and matching (Dubé et al., 2017; Elbaz et al., 2017; Tinchev et al., 2018; Dubé et al., 2018).

Dubé et al., (2017) proposed a place recognition approach based on the matching of 3D segments from laser point clouds. Matching is achieved using traditional learning from a Random Forest (Ho, 1995). Tinchev et al., (2018) extended the approach to
function also in unstructured scenarios, such as forested zones. This is achieved by extracting more repeatable features based on oriented keyframes, and by training on a hybrid descriptor defined using PCA and a Gestalt shape (Bosse and Zlot, 2009b).

Elbaz et al., (2017) generated feature descriptors using a deep neural network. The network is trained on point cloud segments extracted using overlapping spheres, and projected onto depth map images. Candidate matches are selected using a $k$-nearest-neighbour (KNN) search in feature space.

Dubé et al., (2018) presented SegMap, which leverages an alternative data-driven descriptor in order to extract meaningful point cloud features. A 3D binary voxel grid of fixed dimension is fed into a convolutional neural network composed of three 3D convolutional layers. The descriptor is obtained by taking the activations of the last fully connected layer. Similarly to Elbaz et al., (2017), candidate matches are identified using a KNN search.

Nevertheless, when a set of matches is identified, these approaches require an additional step to determine the sensor’s pose within the prior map.

6.2.3 End-to-End Learning Applications

Interesting applications have been proposed which use data-driven approaches to replace traditional systems in an end-to-end fashion, including camera re-localization (Kendall et al., 2015; Clark et al., 2017b; Patel et al., 2018), odometry from monocular camera images (Wang et al., 2017a; Wang et al., 2017b), as well as visual-inertial odometry (Clark et al., 2017a).

Kendall et al., (2015) proposed a camera re-localization framework called PoseNet. In their work, re-localization is achieved by training a CNN to regress the camera position and orientation from single RGB images in an end-to-end manner. This work is one of the first attempts to regress six degree-of-freedom camera poses using a convolutional network end-to-end.

Initial work on stereo visual odometry through deep learning has been presented by Konda and Memisevic, (2015). Firstly, depth and motion dynamics are estimated using network layers with multiplicative interactions. Secondly, a CNN extracts discrete values for changes in direction and velocity using the softmax function. The network was capable of learning the mapping between a sequence of images and the
sensor’s motion. However, it is implicitly designed to treat odometry as a classification problem instead of regression.

Although CNNs demonstrated the potential to learn geometric relationships between images, they do not make use of sequential information. This is a limiting factor when learning about odometry, where by definition the system estimates a sensor’s displacement from subsequent measurements incrementally. Similarly, a neural network for visual or laser odometry should leverage sequential information.

Wang et al., (2017a) proposed to tackle the monocular visual odometry problem using an RCNN-based network, which combines the capabilities of CNNs to extract features from high dimensional data, such as images, with LSTMs for sequential learning. In particular, six degree-of-freedom camera poses are regressed from single RGB images using a deep convolutional neural network followed by two LSTMs. The network achieved competitive visual odometry results and often surpassed the traditional monocular approach. This work was further improved in (Wang et al., 2017b) to include uncertainty estimation.

Similar RCNN architectures have been employed for video re-localization in Clark et al., (2017b) and Patel et al., (2018). The networks demonstrated improved performance as compared to previous approaches based only on convolutions.

In this chapter, we draw inspiration from the work by Wang et al., (2017a). Using an equivalent architecture, we investigate whether it can be applied to the laser odometry problem.

In the following section, we overview the common techniques used to process point cloud data before passing a representation to the network.

6.2.4 Learning from Laser Point Clouds

As mentioned before in this chapter, the main feature of CNNs is the capability of leveraging local correlations in dense grids (e.g., images) through convolutions. However, point clouds are irregular and unordered. Thus, before training, they typically need to be converted into structured representations.
A common representation is discrete data structures, like octrees in Wu et al., (2015). The approach involves dividing the space into 3D voxels and extending CNNs to use 3D convolutions, rather than 2D. However, the higher dimensions of the input and of the network’s filters have a negative impact on computation. In order to mitigate for this, Riegler et al., (2017) and Engelcke et al., (2017) used sparse voxel grids. In this case, convolutions are skipped on empty space, but the kernels of the network are still dense and three-dimensional.

Recent work has been proposed which directly deals with sparse points, such as Qi et al., (2017) and Li et al., (2018). As opposed to previous approaches, the networks are designed to be invariant to the order of the input data, allowing both the input and the filters to be kept sparse. The approaches have been successfully employed for segmentation and classification of 3D point clouds. Nevertheless, the application to other tasks is still unexplored.

Chen et al., (2017) use more compact representations, such as front-view and top-view, for 3D object detection in a urban scenario. The front-view representation is obtained by projecting a point cloud into an unfolded cylinder, i.e., azimuth and elevation are discretized into a 2D grid of fixed size. The grid has three channels encoding the range, height and intensity of each point projected at the corresponding cell. In the top-view representation instead, the point clouds are projected by their $x$ and $y$ coordinates onto 2D grids with fixed resolution. Each grid has three channels encoding height, intensity and density values.

In this chapter, we investigate whether a neural network can benefit of the characteristics of laser point clouds to compute odometry. We are motivated to use laser inputs for odometry because, as opposed to images, point clouds directly incorporate geometric information with millimeter accurate depth. Furthermore, they typically provide a wider field-of-view, are less sensitive to shaking and are resilient to light changes.

To our knowledge, laser odometry from the perspective of deep learning has not yet been approached in the literature.
### 6.3 Approach

In this section, we describe the architecture, the input and output representations, and the loss function of the proposed system, which we call Deep Laser Odometry (DLO).

DLO uses the network architecture presented in Wang et al., (2017a), shown in Fig. 6.2. As compared to the original work, in our case the input to the network will be laser point clouds, rather than RGB images, after conversion to a structured representation. Pairs of consecutive measurements are stacked together and fed into a CNN (blue) to extract a $4 \times 4 \times 1024$ feature vector, which is then passed through a series of two LSTMs (green) for sequential learning. At each time step, the network outputs the estimated relative pose $\mathbf{x} = (\mathbf{x}, \mathbf{\theta})^\dagger$, composed of a 3D translation vector $\mathbf{x}$ and a 3D rotation vector $\mathbf{\theta}$.

Although the network is trained to estimate 3D translation and rotation, in practice, when evaluating the results we focus only on the $x, y$ and yaw dimensions for two reasons: 1) we estimate odometry during a vehicle navigation task where the magnitude of $z$, roll and pitch variations is negligible, 2) for both the driving and walking applications considered in this thesis, an IMU is typically available to estimate pitch and roll.

The structure of the CNN is based upon FlowNetS (Dosovitskiy et al., 2015), which is pre-trained to estimate the optical flow between two RGB images stacked together.

---

1 In full, $\mathbf{x}$ is a relative pose indicated as $\Delta \mathbf{x} = (\Delta \mathbf{x}, \Delta \mathbf{\theta})$. For simplicity, throughout this chapter we adopt a minimal notation.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Receptive Field</th>
<th>Padding</th>
<th>Stride</th>
<th>Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv 1</td>
<td>$7 \times 7$</td>
<td>3</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>Conv 2</td>
<td>$5 \times 5$</td>
<td>2</td>
<td>2</td>
<td>128</td>
</tr>
<tr>
<td>Conv 3.1</td>
<td>$5 \times 5$</td>
<td>2</td>
<td>2</td>
<td>256</td>
</tr>
<tr>
<td>Conv 3.2</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>Conv 4.1</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>2</td>
<td>512</td>
</tr>
<tr>
<td>Conv 4.2</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Conv 5.1</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>2</td>
<td>512</td>
</tr>
<tr>
<td>Conv 5.2</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Conv 6</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>2</td>
<td>1024</td>
</tr>
</tbody>
</table>

Table 6.1: CNN configuration.
Figure 6.2: DLO architecture, composed of nine CNN layers (blue), each followed by ReLU activations. The last convolutional feature is passed to a RNN (green). The CNN dimensions are given as an example, considering $256 \times 256$ pixels top-view input images, and the network configuration in Tab. 6.1. The network’s output is a tensor containing the sensor’s relative pose estimate at each time step $k$. Legend: $x = \text{translation, } \theta = \text{rotation}.$

The configuration of the nine convolutional layers is detailed in Tab. 6.1, where, as discussed in Wang et al., (2017a), the size of the receptive fields is progressively reduced so as to capture small features, whereas the number of filters is increased so as to capture different features. Each layer is followed by a ReLU activation except the last one. After deconvolution, the last convolutional feature shows the estimated optical flow image, which represents a left turn in Fig. 6.2.

The aim of the RNN is to implicitly model the geometric relations between the convolutional features extracted from a sequence of measurements. As in Wang et al., (2017a), this is achieved using a deep architecture composed of two LSTMs in series. In order to capture complex dynamics, each LSTM has 1024 hidden states, being the input to the next layer (Wang et al., 2017b). The motion estimate $x$ is regressed by a fully-connected layer appended at the bottom of the RNN structure.

6.3.1 Input Representations

In our experiments, we consider vehicle driving settings from the KITTI dataset (Geiger et al., 2013). The dataset is described in detail in Sec. 6.4. The vehicle is equipped with a Velodyne HDL-64E, which captures 3D point clouds.
Given a point cloud $C$ composed of a set of 3D laser points in Cartesian coordinates $p_i = [x_i, y_i, z_i]$ with $i \in [1, |C|]$, expressed in the sensor’s reference frame, we consider the top-view and front-view representations proposed in Chen et al., (2017):

**TOP-VIEW**  We project each point by its $xy$ Cartesian coordinate into a $256 \times 256$ pixels grid with resolution of 0.2 m. The representation encodes height, intensity and density of the points within three image channels. For each cell, the height value is the maximum between all points in the same cell. Once selected the point with maximum height, its reflectance gives the intensity value within the cell. Finally, the density feature is calculated from the number of points $M$ in the cell, normalized as $\min(1.0, \frac{\log(M+1)}{\log(64)})$. Fig. 6.3 visualizes the three channels of a top-view representation.

**FRONT-VIEW**  We project each point by its azimuth and elevation into an image of $1024 \times 64$ pixels, to obtain a dense front-view representation. The representation encodes range, height and intensity within three image channels, as shown in Fig. 6.4. For each point $p_i$, the transformed image coordinates $(u, v)$ are:

$$
\begin{align*}
    u &= \left\lfloor \frac{\arctan2(y_i, x_i)}{\Delta \mu} \right\rfloor \\
    v &= \left\lfloor \frac{\arctan2(z_i, \sqrt{x_i^2 + y_i^2})}{\Delta \nu} \right\rfloor
\end{align*}
$$

where $\Delta \mu$ and $\Delta \nu$ are the horizontal and vertical image resolution, respectively. This projection results in the image showing azimuth $[-180, 180]^\circ$ with $0^\circ$ being the direction of forward motion. As opposed to the original work, we apply a horizontal shift in order to cover the azimuth $[-270, 90]^\circ$. In this way distant points along the trajectory (shown in red in Fig. 6.4, top) appear central to the image and can be fully covered by the network’s convolutions, having a higher impact on the output. We found empirically that this horizontal shift contributes positively to the estimation of yaw rotations. Finally, in order to smooth the image around self-observations and empty pixels, we interpolate using a median filtering approach.

### 6.3.2 Output Representations

The output of the network is a 6 degrees-of-freedom estimate of the sensor’s relative pose. While the translation component can be represented by 3D Euclidean
coordinates, rotational quantities are more complicated to learn and the network’s performance might depend on the chosen representation. Common representations for rotation include quaternions and Euler angles.

Quaternions are four-dimensional continuous values mapped to angular rotations through normalization to unit length. They are a differentiable representation of rotation, which makes them appropriate for learning. Since quaternions lie on the unit sphere, each orientation is represented by two quaternions, one per hemisphere. This can be easily compensated for by constraining the representation to one of the two hemispheres.
Euler angles are an intuitive representation of rotation in three dimensions. A main limitation is that they suffer from the gimbal lock problem (Altmann, 1986). Also, they are defined in the range \([0, 2\pi]\) and rewind at \(2\pi\), meaning that multiple Euler values represent the same orientation. In general, such a behaviour is not suitable for learning, as the network is expected to model a uni-modal scalar regression.

In our work, experiments have been carried out using either quaternions or Euler angles. Indeed, for our application we consider frame-to-frame displacements, resulting in small relative rotations primarily about a single axis (\(z\)-axis). Because of this, the system is not affected by the limitations of Euler angles (during our experiments relative yaw rotations were typically less than \(5^\circ\)). When using quaternions, we manually constrain the poses to the positive hemisphere.

### 6.3.3 Geometric Loss Function

The optimal parameters \(\phi^*\) of the network are learned by maximizing the conditional probability of a trajectory of \(N\) poses \(\mathcal{X} = \{x_i\}\) given a sequence of point clouds \(\mathcal{C} = \{C_i\}\), with \(i \in [1, N]\):

\[
p(\mathcal{X}|\mathcal{C}) = p(\{x_i\}|\{C_i\}) \tag{6.3}
\]

\[
\phi^* = \arg\max_\phi p(\mathcal{X}|\mathcal{C}; \phi) \tag{6.4}
\]
Thus, using the formulation in Wang et al., (2017b), we train the network to minimize the Euclidean loss computed as a linear weighted sum of the distance between the ground truth pose $x^k = (x^k, \theta^k)$, with $\theta^k$ expressed in Euler angles, and its estimate $\hat{x}^k = (\hat{x}^k, \hat{\theta}^k)$ at time $k$:

$$\phi^* = \arg \min_{\phi} \frac{1}{N} \sum_{k=1}^{N} \left( \|x^k - \hat{x}^k\|_2^2 + \beta \|\theta^k - \hat{\theta}^k\|_2^2 \right) \tag{6.5}$$

where $\phi$ are the network weights and biases, $\|\cdot\|_2$ is the $L_2$ Euclidean norm, and $\beta$ is a hyperparameter introduced to balance the contribution of the two components of the loss function during learning. Indeed, the loss function is composed of two distinct parts for translation and rotation. Each of the two parts is characterized by a different scale and measurement unit. Despite the tuning required for $\beta$, Kendall and Cipolla, (2017) showed that learning translation and rotation together is preferable to training two models separately.

One notices that the rotation loss is formulated in Euclidean space, rather than on the unit sphere using a Geodesic distance metric. As discussed in Kendall and Cipolla, (2017), during training the difference between the ground truth and the estimated pose becomes small enough, such that the distinction between Euclidean and Geodesic distance is negligible.

Finally, we evaluate the performance of our network on an alternative loss function (Kendall et al., 2015; Kendall and Cipolla, 2017) based on quaternions:

$$\phi^* = \arg \min_{\phi} \frac{1}{N} \sum_{k=1}^{N} \left( \|x^k - \hat{x}^k\|_2^2 + \gamma \left\| q^k - \frac{\hat{q}^k}{\| \hat{q}^k \|_2} \right\|_2^2 \right) \tag{6.6}$$

where $q$ is the rotation expressed in quaternions.

### 6.4 EXPERIMENTAL EVALUATION

So as to evaluate the performance of the network on the laser odometry task, we use the KITTI dataset (Geiger et al., 2013). The dataset stores traffic scenarios in city, residential and motorway settings. It includes 22 sequences of RGB images and laser point clouds, of which sequence 00-10 are associated with ground truth poses from a combined GPS/IMU system, whereas sequence 11-21 serve for testing and are
Table 6.2: Sequences of the KITTI dataset (Geiger et al., 2013) used in our experiments. Each sequence is indicated in green if accompanied by ground truth poses, in red otherwise.

provided with raw images and point clouds without ground truth poses. The point clouds are captured using a Velodyne HDL-64E. As described in Sec. 2.5.2, each cloud is three-dimensional and has 120 m range. More details on the sequences involved in our analysis are reported in Tab. 6.2. The dataset was recorded at 10 Hz from a vehicle driving at speed up to about 90 km/h in the motorway scenes. Particularly, sequence 14 records a natural scenario in a park.

Our evaluation consists of four experiments. Experiments A. and B. evaluate the network’s performance when choosing different losses or input representations. Experiments C. and D. compare the network’s performance against other odometry approaches. In particular:

A. An evaluation of two different loss functions and representations of rotation, either using Euler angles or quaternions. Additionally, we compute statistics on translation and rotation errors using a set of balancing factors $\beta$ and $\gamma$.

B. An evaluation of the network performance when learning from the front-view or the top-view input representations. On the one hand, the front-view is a dense representation, which makes it naturally suitable for convolutions. On the
other hand, despite its sparsity, the top-view directly captures the geometries which are more crucial during vehicle driving.

C. A comparison between the accuracy of the estimated odometry when processing RGB camera images or laser point clouds, as proposed in our work. We refer to the two approaches as DVO and DLO, respectively.

D. A comparison between the performance of a baseline laser odometry approach using ICP and the proposed DLO. As the baseline, we consider both the results obtained from frame-to-frame ICP on the original point clouds, as well as on the point clouds after ground removal (indicated as ICPng).

6.4.1 Training and Testing

We train and test the model with supervised learning on sequences 00-10, using the ground truth poses from GPS/IMU. Specifically, during our experiments all sequences comprising ground truth are used for training, except sequence 05 which is used for quantitative evaluation during testing. Additional qualitative results are demonstrated on sequences 11, 12, 14 and 15 where no ground truth is available.

We train the network in two phases. In order to reduce the amount of input data required and the training time, we aid the learning by initializing the CNN to the weights of FlowNetS (Dosovitskiy et al., 2015), which are pre-trained on synthetic RGB images to estimate optical-flow. During the first phase, these weights are kept fixed and only the RNN is trained. Training is performed using the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.001. This value is the one recommended in Kingma and Ba, (2015), and kept unchanged from Wang et al., (2017b). The second training phase consists of fine-tuning the weights of the entire network with a slower learning rate of 0.001/4. The training runs for up to 300 epochs in the first and second phase.

Dropout and early stopping techniques are used to prevent the network from overfitting. In Wang et al., (2017a), the authors give some insights about the effect of model overfitting on the odometry task, and discuss how a gap between the training and validation losses would indicate overfitting. In Fig. 6.5, the DLO training and validation losses show good fit of the model.
6.4 Experimental Evaluation

In this section, we analyze the results from our experiments.

A. Selecting the Loss Function Tab. 6.3 (rows 3 and 4) illustrates the performance of the network using the losses in Eq. (6.5) or Eq. (6.6), based on Euler angles and quaternions respectively. In the experiment, the network performs comparably using Euler angles or quaternions, showing that both these representations are suitable to learn relative poses.

In Fig. 6.6 we show statistics on translation and rotation errors using a range of scale factors \( \beta \) and \( \gamma \) for each loss. Given these results, we set \( \beta = 120 \) and \( \gamma = 120 \) for our experiments. By choosing these values we prioritize lower rotation error over translation. This choice is motivated by the fact that in our results translation errors...
Table 6.3: Summary of relative pose estimation results computed per frame on test sequence 05 for different approaches. Rows 2, 3 and 4 indicate three different variants of the proposed DLO system. Experiments C and D solely compare the approaches on rows 1, 3, 5 and 6, highlighted in grey. Legend: $x$ = position, $\theta$ = rotation in Euler angles, $q$ = rotation in quaternions (indicate the variables of the loss function we used).

vary of a few millimetres for different values of the scale factors. Such errors are negligible as compared to the network accuracy, which is in the order of centimetres.

Our subsequent experiments will solely use the loss in Eq. (6.5) based on Euler angles. Indeed, in our evaluation, Euler angles achieved comparable performance to quaternions and have the upside of being a more intuitive representation.

B. SELECTING THE INPUT REPRESENTATION  We compare the performance of the network when learning from the front-view or top-view input representations. The results obtained for the test sequence 05 are summarized in Tab. 6.3 (rows 2 and 3). Learning from top-view allowed the network to estimate translation with similar performance as using a denser representation such as front-view. Furthermore, the network achieved 31.18% median yaw error as compared to 41.44% using the front-view representation. Although the projection to front-view generates dense images which are suitable for convolutions, we think that top-view is a more crucial representation for ground robots, as most displacement during driving or walking occurs on the $xy$ plane and yaw rotation.

---

2 We compute the RMSE and the median error. The RMSE (also defined in Appendix C) is a quadratic scoring metric that measures the average magnitude of a distribution of errors. Since the errors are squared before they are averaged, the RMSE penalizes large errors. This metric is relevant for our application because large errors are particularly undesirable. On the other hand, the median error gives low importance to outliers, resulting in a value which is more representative of the overall performance.
Figure 6.7: DLO and DVO network performance on test sequence 05. Results from the baseline laser odometry approach using ICP after ground removal (ICPng) are shown in green. In grey (b) top we highlight the regions where hard turns occur. Legend: GT = ground truth.

C. DEEP VISUAL AND LASER ODOMETRY

In Fig. 6.7 we show the estimated relative motion, translation and rotation errors for the proposed approach (DLO), as
as using RGB images (DVO), similarly to previous work by Wang et al., (2017a). The results, also summarized in Tab. 6.3 (rows 1 and 3), are obtained from testing on sequence 05.

We observe that the network estimates yaw rotation and lateral translation $y$ with comparable accuracy both for DVO and DLO. Nevertheless, the 3D point cloud data processed by DLO provide a wider field-of-view and range than the RGB images. We think that this characteristic of laser data contributed to the DLO network estimating more accurate translation especially along the dimension of highest displacement $x$. For instance, a wider field-of-view mitigates the effect of moving objects occluding the sensor view.

The resulting trajectories are shown in Fig. 6.10a and Fig. 6.10b. From a qualitative perspective, the DVO trajectory appears globally more accurate in tracing yaw rotations than DLO, although the networks achieved comparable yaw estimation accuracy, with 31.18% median error for DLO and 34.87% for DVO. The reason for this is clearer in Fig. 6.8. The plots show that for both networks the average relative yaw error increases in proximity of hard turns (i.e., when the relative yaw is greater than about $0.5^\circ$ for positive or negative rotations). While during this experiment DLO showed higher sensitivity to negative rotations, the accuracy of DVO was balanced between positive and negative angles, which resulted in a compensation effect along the trajectory.

Additional results are shown from sequences 11 and 15 in Fig. 6.11. For these sequences the ground truth trajectory was not available, but we manually traced it in the plots for qualitative comparison.

It is worth noting that in our experiments the DVO network estimated less accurate odometry than the original network in Wang et al., (2017a). Nevertheless, in our evaluation we focused on a fair comparison of how the performance of the same network vary when learning from different inputs, i.e., camera or laser data.

**D. Comparison to standard laser odometry** We compare the performance of the proposed DLO network to a baseline laser odometry approach, which performs point cloud registration incrementally using ICP. For a fair comparison, we constrain ICP to align in three dimensions during this experiment ($x$, $y$ and yaw).
Firstly, in Tab. 6.3 (rows 5 and 6) and Fig. 6.10c-6.10d we show quantitative results and estimated trajectories for the baseline in two different cases: 5) registration of the original point clouds (ICP), 6) registration of the point clouds after ground removal (ICPng). The results demonstrate that ICP is sensitive to the predominance of ground points in the point clouds.

In Fig. 6.7 we show the estimated relative motion, translation and rotation errors for the learning approaches, compared to the baseline ICPng. Specifically, the laser odometry results we look at are summarized in Tab. 6.3 (rows 3 and 5).

With a focus on the direction of motion $x$, in this experiment we notice that ICPng is more accurate than DLO in 98% of the cases. However, large translation errors up
to 1 m occur in 2% of the cases for displacements greater than 0.4 m, as depicted in Fig. 6.8. We observe that most failures occur during forward motion, rather than turning (hard turn regions are highlighted in grey in Fig. 6.7b (top)), due to a combination of two factors: high initial misalignment since no motion prior from proprioception is being used, and insufficient geometric constraints. As an example, in Fig. 6.9, we show the alignment between the point clouds at frames 2618 and 2619. In this case, the initial displacement was greater than 1 m, and the geometric constraints perpendicular to \( x \) resulted insufficient to enforce the alignment along that dimension. As compared to ICPng, the accuracy of DLO is more uniform across the range of displacements along \( x \), with the lowest median error of 3.23% and RMSE of 0.054 m among the approaches we analyzed.

As opposed to ICP, DLO demonstrated to be robust to the predominance of ground points in the original point clouds, and implicitly learned to prioritize the geometric features in the data which are relevant to detect displacement (such as walls and buildings).

In spite of these promising results demonstrated by the proposed DLO network, accumulated relative yaw errors lead to large global distortions in the estimated trajectory (Fig. 6.10b). On the other hand, ICPng is affected by outlier errors but is typically more accurate, resulting in a better estimated trajectory in Fig. 6.10d.

Additional results are shown from a motorway driving scenario (sequence 12) and an unseen natural scenario (sequence 14) in Fig. 6.12. For these sequences the ground truth trajectory was not available, but we manually traced it in the plots for qualitative comparison. Such scenarios are particularly challenging for ICP techniques, due
Figure 6.10: Estimated odometry trajectories on test sequence 05. Legend: GT = ground truth.

(a) Deep visual odometry.
(b) Proposed deep laser odometry.
(c) Baseline ICP odometry from original point clouds.
(d) Baseline ICP odometry after ground removal.

to the lack of structural features. The data-driven approaches show some generalizability skills to these environments. Similarly to previous results on urban sequences, accumulated relative yaw errors lead to large global distortions in the estimated trajectories by DLO and DVO. Nevertheless, the approaches demonstrate good reliability in estimating displacement especially along the predominant direction of motion $x$, where ICPng accumulates about 400 m translation error.
In this section, we investigated whether a deep learning approach can be employed to tackle the laser odometry problem in an end-to-end fashion, effectively replacing the standard geometry-driven pipeline. We formulated the problem by leveraging a state-of-the-art deep learning architecture, originally designed for visual odometry, to process 3D laser point clouds in the form of a structured representation. Our experiments showed preliminary results where the network was capable of regressing a vehicle’s trajectory with better estimation performance along the predominant dimension of motion $x$, as compared to DVO. This is achieved with no explicit assumptions about the structure of the environment or the features of the sensor.

Despite these promising results, the DLO approach did not achieve state-of-the-art accuracy as compared to standard laser odometry. Instead, we value this work as a proof-of-concept validation, and a motivation for future research in this direction.

Future work could compare the DLO approach proposed in this work with an alternative architecture which introduces a CNN to directly learn from unordered laser points (e.g., PointCNN – Li et al., (2018)).

Moreover, the network might benefit from training on a larger variety of data, including indoor scenarios and other 3D laser technologies, in order to test generalizability to more environments and sensors.

The capability of the network to learn odometry in the presence of additional challenges such as overlap variations will be tested in future work, for instance considering lower frequency laser sensors and testing on dynamic legged robots.
Figure 6.11: Estimated odometry trajectories on the additional test sequences 11 and 15, which have no ground truth. The ground truth trajectory in black has been manually traced for qualitative evaluation.
Figure 6.12: Estimated odometry trajectories on the challenging test sequences 12 and 14, which have no ground truth. The ground truth trajectory in black has been manually traced for qualitative evaluation. Specifically, sequence 12 captures a motorway driving scenario, whereas sequence 14 represents a natural scenario unseen during training.
CONCLUSIONS

The objective of our research within the Dynamic Robot Systems (DRS) group\footnote{http://ori.ox.ac.uk/labs/drs/} is to enable wheeled and legged robots to operate in real world scenarios, during search and rescue missions in disaster zones or inspection of industrial areas. Such applications require robots to autonomously navigate across rough terrain, traverse obstacles, detect and manipulate objects, similar to what humans can do. These tasks involve research in control, state estimation and planning on multiple scales: from \textbf{low level} control, to motion planning, to \textbf{high level} perception, localization, path planning and navigation.

In less than four years, the team of the DRS was able to contribute to the development of high level skills, including: localization and multisensor state estimation (Nobili et al., 2017b; Nobili et al., 2017a), proprioception-aided visual localization and mapping (Scona et al., 2017), failure prediction (Nobili et al., 2018), visual articulated tracking (Rauch et al., 2018), global localization in urban and natural environments (Tinchev et al., 2018).

Given the complexity of legged robotics, research in the past has been focused on control, state estimation and planning using proprioception, whereas the role of perception was limited to the minimum necessary to achieve specific tasks. This aspect was a limiting factor for autonomy, and motivated the research in this thesis, towards
The objective of this dissertation was to develop laser-based localization techniques which are suitable for the safe and continuous operation of wheeled and legged robots in real-world environments. We explored approaches fusing inertial, kinematics, stereo vision and laser signal sources in various combinations, achieving the following contributions:

– A method for laser-based localization which overcomes the weaknesses of baseline approaches in real-world situations, being robust to large variations in spatial overlap and initial accumulated drift in the robot’s pose estimate. This is especially important for walking and trotting robots, low frequency laser sensors and navigation in cluttered environments. The method has been experimentally validated on two full-sized humanoid robots, and allowed accurate localization during exploration of the DRC arena.

– A state estimation system which fuses multiple sensor sources, including proprioception, stereo cameras and a laser. A particular technical achievement has been to reliably close the control loop with state estimation during dynamic locomotion. During our field experiments, the system supported the dynamic maneuvers of the robot and operation in sensor impoverished situations, achieving continuously accurate localization. Subsequent research at IIT relied on our state estimator for applications such as motion planning and footstep placement on challenging terrain.

– A module which can predict and prevent failures during localization. This is important in common real-world situations where the success of point cloud registration can be compromised due to occlusions or constrictions between subsequent sensor observations.

– A deep learning approach for laser odometry, which explores the applicability of data-driven techniques to this fundamental problem in robotics. Our interest in this research lies in exploring whether this approach can be beneficial in robotics applications and motivate future work in this direction.

Future work will focus on extending our localization system to include loop closure detection and mapping. The Simultaneous Localization and Mapping (SLAM) system would also leverage the alignment risk prediction module in order to prevent
localization failure. Particularly, the alignment risk score itself can be involved in the detection of loop closures.

We are interested in extending the applicability of our failure prediction approach to outdoor scenarios, as well as performing additional field experiments on legged robots, such as the quadruped ANYmal, which recently arrived in our lab.

Given the potential demonstrated by the deep laser odometry approach, future work will carry on research towards deep learning systems which are suitable for application to robotic navigation tasks.
APPENDIX – OTHER CONTRIBUTIONS

In this appendix we briefly cite additional research which I contributed to during the PhD.

A.1 DIRECT VISUAL SLAM FUSING PROPRIOCEPTION FOR A HUMANOID ROBOT

This work, lead by Raluca Scona, was published in the proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (Scona et al., 2017).

Personal contributions are marginal:

• Support during data collection.

• Support during closed-loop experiments (Exp D).

• Contributor on paper content and figures.

SHORTEN ABSTRACT In this paper, we investigated the application of semi-dense visual SLAM to the humanoid robotics domain. Challenges of visual SLAM applied to humanoids include the type of motion executed by the robot, a lack of features in man-made environments and the presence of dynamics in the scene. We studied the application of a modern method called ElasticFusion (Whelan et al., 2015) to obtain
a visually interpretable map which can be used for collision free motion planning. We tackled the challenges by proposing a more robust pose tracking method. This is formulated as an optimization problem over a cost function which combines information from the stereo camera and a low-drift kinematic-inertial motion prior (Koolen et al., 2016). Extensive experimental demonstrations characterized the performance of our method using the NASA Valkyrie humanoid robot in a laboratory environment. The experiments demonstrated pose tracking robustness to challenges such as sudden view change, dynamics and motion blur in the image, change in illumination and tracking through sequences of featureless areas in the environment.

Results of tracking in dynamic scenes and darkness are shown in Fig. A.1.

Figure A.1: SLAM fusing proprioception on Valkyrie. Top: examples of colour-disparity image pairs captured from the Multisense SL camera mounted on the NASA Valkyrie’s head. Bottom: corresponding relative position estimation error (RPE) for the proposed approach (blue) and the baseline (magenta). Photo credit: (Scona et al., 2017).

A.2 SEEING THE WOOD FOR THE TREES: RELIABLE LOCALIZATION IN URBAN AND NATURAL ENVIRONMENTS

This work, lead by Georgi Tinchev was published in the proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (Tinchev et al., 2018).
Personal contributions include:

- Support for architecture development and experimental design.
- Support for results analysis on all experiments.
- Secondary author on paper content and figures.

**Shorten abstract**
In this paper, we introduced Natural Segmentation and Matching (NSM), an algorithm for reliable place recognition, using laser, in both urban and natural environments. Challenges of place recognition in natural areas include clutter and perceptual aliasing, which prevent repeatable extraction of distinctive landmarks between different runs. Indeed, tree trunks are not distinctive, foliage intertwines and there is a lack of planar structure. In order to address these challenges, we proposed a method for place recognition which uses a more involved feature extraction process. First, stable and reliable object-sized clusters are segmented from a point cloud. Second, repeatable oriented key poses are extracted and matched with a Gestalt shape descriptor using a Random Forest. Our experiments show how the approach can achieve place recognition in forested areas while also outperforming baseline approaches in urban scenarios without specific tuning.

Results of place recognition in natural scenes are shown in Fig. A.2.
Figure A.2: Illustration of the estimated localizations (blue) relative to the target map (black), consisting of previously traversed routes within two natural areas, a park and a forest. Example point clouds during successful place recognition are shown in purple. An example of a missed localization (also observed in regions 1, 2, 3) caused by the absence of rigid objects is shown in orange. Photo credit: (Tinchev et al., 2018).
APPENDIX – KALMAN FILTERS

B.1 KALMAN FILTER

The Kalman filter (Kalman, 1960) is a state observer for linear Gaussian systems. The filter models the evolution of a system’s state $x^k \in \mathbb{R}^{n \times 1}$ at time $k \in \mathbb{N}$ from the previous state at time $k - 1$, according to

$$x_k(x_{k-1}, u_k) = A_k x_{k-1} + B_k u_k + w_k$$

(B.1)

given a control input $u_k \in \mathbb{R}^{n \times 1}$ and the process noise $w_k$. At time $k$, the sensor measurement $z_k \in \mathbb{R}^{m \times 1}$ is modelled by the relationship

$$z_k(x_k) = C_k x_k + v_k$$

(B.2)

The process noise $w_k \simeq \mathcal{N}(0, Q_k)$ and the measurement noise $v_k \simeq \mathcal{N}(0, R_k)$ are taken from a zero mean Gaussian distribution with covariance $Q_k$ and $R_k$, respectively. In Eq. (B.1) the system is considered Markovian, meaning that the current state $x_k$ depends only on the previous state $x_{k-1}$ and the current control input $u_k$. The previous state $x_{k-1}$ embeds all information about the past states $(x_0, x_{k-2})$ and the measurements $(k_0, k_{k-1})$. The model parameters $A_k \in \mathbb{R}^{n \times n}$, $B_k \in \mathbb{R}^{n \times n}$ and $C_k \in \mathbb{R}^{m \times n}$ only depend on the current time $k$. 

122
The filter relies on two main steps:

**Prediction**

\[ \hat{x}_k^- = A_k \hat{x}_{k-1} + B_k u_k \quad (B.3) \]

\[ P_k^- = A_k P_{k-1} A_k^T + Q_k \quad (B.4) \]

**Update**

\[ K_k = P_k^- C_k^T (R_k + C_k P_k^- C_k^T)^{-1} \quad (B.5) \]

\[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - C_k \hat{x}_k^-) \quad (B.6) \]

\[ P_k = (I - K_k C_k) P_k^- \quad (B.7) \]

Firstly, a prediction step estimates a prior \( \hat{x}_k^- \) for the current state, from Eq. (B.1), (B.3), while the prior \( P_k^- \) for the covariance is computed in Eq. (B.4). Secondly, once a new measurement \( z_k \) is captured, the state and covariance priors are updated, as shown in Eq. (B.6) and Eq. (B.7), by means of the Kalman gain \( K_k \) computed in Eq. (B.5). Specifically, \( K_k \) expresses how much the new measurement contributes to the filter update. Since the system is modelled as Gaussian, the output of the filter \( \hat{x}_k \) and \( P_k \) at time \( k \) represent a mean \( \mu \) and a covariance matrix \( \Sigma \), respectively.

The Kalman filter relies on the assumption that the system is linear. These assumption is not valid in most real systems.

**B.2 Extended Kalman Filter**

An extended Kalman filter (EKF) has been developed to accommodate for non-linear process and measurement functions, via linearization through a first order Taylor series expansion. Given

\[ x_k(x_{k-1}, u_k) = g(x_{k-1}, u_k, w_k) \quad (B.8) \]

\[ z_k(x_k) = h(x_k, v_k) \quad (B.9) \]

after linearization of \( g(\cdot) \) and \( h(\cdot) \), the filter prediction and update are computed similarly to the linear case, as explained in detail in Welch and Bishop, (1995).
C

APPENDIX – EVALUATION METRICS

C.1 DRIFT PER DISTANCE TRAVELLED

The Drift per Distance Travelled (DDT) is the performance metric we use to evaluate position accuracy in Chapter 4. The metric is expressed in m/m.

Given $N$ samples of a ground truth relative position $\Delta \mathbf{x} = [\Delta x, \Delta y, \Delta z]$, and the corresponding estimated position $\hat{\Delta \mathbf{x}} = [\hat{\Delta x}, \hat{\Delta y}, \hat{\Delta z}]$, the DDT is defined over the time period $k - N$ to $k$ as follows:

$$DDT = \frac{\|\Delta \mathbf{x}_{k-N:k} - \Delta \hat{\mathbf{x}}_{k-N:k}\|}{\sum_{j=k-N}^{k} \|\Delta \mathbf{x}_{j-1:j}\|}$$  \hspace{1cm} (C.1)

where the numerator expresses the absolute position error over the period $k - N : k$, while the denominator is the ground truth trajectory integral (i.e., the distance travelled). For an entire run, statistics of this function can be calculated, such as the median value to obtain the median Drift per Distance Traveled.

One should notice that it can be problematic to estimate the distance travelled. During a robot’s walking, crawling or trotting gait, the ground truth trajectory presents high frequency oscillations. For example, the typical crawl and trot trajectories for the HyQ robot are shown in Camurri et al., (2017). For this reason, the distance travelled can be under-estimated, by approximating the distance between two points on the trajectory to a straight line, or over-estimated, by considering the level of detail in the trajectory produced at high frequency by the motion capture system.
C.2 ROOT-MEAN-SQUARE ERROR

The Root-Mean-Square Error (RMSE) is an error metric which can be used to evaluate the performance of state estimation systems.

Given $N$ samples of a ground truth relative position $\Delta x = [\Delta x, \Delta y, \Delta z]$, and the corresponding estimated position $\Delta \hat{x} = [\Delta \hat{x}, \Delta \hat{y}, \Delta \hat{z}]$, the RMSE is computed as the square root of the mean of the quadratic error $(\Delta x - \Delta \hat{x})^2$:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N}(\Delta x^k - \Delta \hat{x}^k)^2}{N}}$$ (C.2)


of Leg Kinematics and IMU.” In: Robotics: Science and Systems (RSS) (cit. on pp. 11, 12, 21).


