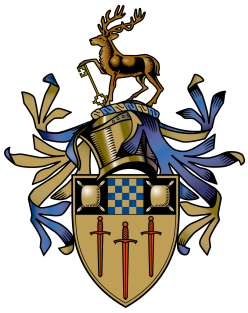
Collaborative Strategies for Autonomous Localisation, 3D Reconstruction and Pathplanning

O. Mendez

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey



Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

September 2017

Abstract

Autonomous 3D reconstruction, the process whereby an agent can produce its own representation of the world, is an extremely challenging area in both vision and robotics. However, 3D reconstructions have the ability to grant robots the understanding of the world necessary for collaboration and high-level goal execution. Therefore, this thesis aims to explore methods that will enable modern robotic systems to autonomously and collaboratively achieve an understanding of the world.

In the real world, reconstructing a 3D scene requires nuanced understanding of the environment. Additionally, it is not enough to simply "understand" the world, autonomous agents must be capable of actively acquiring this understanding. Achieving all of this using simple monocular sensors is extremely challenging. Agents must be able to understand what areas of the world are navigable, how egomotion affects reconstruction and how other agents may be leveraged to provide an advantage. All of this must be considered in addition to the traditional 3D reconstruction issues of correspondence estimation, triangulation and data association.

Simultaneous Localisation and Mapping (SLAM) solutions are not particularly well suited to autonomous multi-agent reconstruction. They typically require the sensors to be in constant communication, do not scale well with the number of agents (or map size) and require expensive optimisations. Instead, this thesis attempts to develop more pro-active techniques from the ground up.

First, an autonomous agent must have the ability to actively select *what* it is going to reconstruct. Known as view-selection, or Next-Best View (NBV), this has recently become an active topic in autonomous robotics and will form the first contribution of this thesis. Second, once a view is selected, an autonomous agent must be able to plan a trajectory to arrive at that view. This problem, known as path-planning, can be considered a core topic in the robotics field and will form the second contribution of this thesis. Finally, the 3D reconstruction must be anchored to a globally consistent map that co-relates to the real world. This will be addressed as a floorplan localisation problem, an emerging field for the vision community, and will be the third contribution of this thesis.

To give autonomous agents the ability to actively select what data to process, this thesis discusses the NBV problem in the context of Multi-View Stereo (MVS). The proposed approach has the ability to massively reduce the amount of computing resources required for any given 3D reconstruction. More importantly, it autonomously selects the views that improve the reconstruction the most. All of this is done exclusively on the sensor pose; the images are *not* used for view-selection and only loaded into memory once they have been selected for reconstruction. Experimental evaluation shows that NBV applied to this problem can achieve results comparable to state-of-the-art using as little as 3.8% of the views.

To provide the ability to execute an autonomous 3D reconstruction, this thesis proposes a novel computer-vision based goal-estimation and path-planning approach. The method proposed in the previous chapter is extended into a continuous pose-space. The resulting view then becomes the goal of a *Scenic Pathplanner* that plans a trajectory between the current robot pose and the NBV. This is done using an NBV-based pose-space that biases the paths towards areas of high information gain. Experimental evaluation shows that the Scenic Planning enables similar performance to state-of-the-art batch approaches using less than 3% of the views, which

corresponds to $2.7 \times 10^{-4}\%$ of the possible stereo pairs (using a naive interpretation of plausible stereo pairs). Comparison against length-based path-planning approaches show that the Scenic Pathplanner produces more complete and more accurate maps with fewer frames. Finally, the ability of the Scenic Pathplanner to generalise to live scenarios is demonstrated using low-cost robotic platforms.

Finally, to allow global consistency and provide a basis for indoor robot-human interaction, this thesis proposes a novel human-inspired floorplan localisation approach. This method uses the intuition that humans use semantic cues, such as doors and windows, to localise within a floorplan. These semantic cues are extracted from an RGB image, presented as a novel sensor modality called Semantic Detection and Ranging (SeDAR) and used as observations within a Monte-Carlo Localisation (MCL) framework. Experimental evaluation shows that SeDAR-based MCL has the ability to outperform state-of-the-art MCL when using range measurements. It is also demonstrated that the semantic cues are sufficient for localisation, as this approach achieves results comparable to state-of-the-art without range measurements.

When combined, these contributions provide solutions to some of the most fundamental issues facing autonomous and collaborative robots. They advance the fields of 3D Reconstruction, Path-planning and Localisation by allowing autonomous agents to reconstruct complex scenes. The field of 3D reconstruction is advanced by demonstrating that intelligent view selection is capable of drastically improving performance of established methods. The field of Path-planning is advanced by establishing that pro-active behaviours can be encoded into low-cost robotics, such that high-level goals result in emergent strategies for collaboration. Finally, the field of Localisation is advanced by validating that human-inspired localisation based on distinctive semantic landmarks is an effective alternative to traditional scan-matching. The experiments in this thesis demonstrate that autonomous agents can navigate unknown complex scenes using simple monocular cameras. This thesis lays the foundation for autonomous, collaborative 3D reconstruction that goes beyond simple SLAM-based solutions and enables high-level collaboration towards a common goal.

Key words: Reconstruction, Path-planning, Localisation, Robotics

Email: o.mendez@surrey.ac.uk

WWW: www.oscarmendez.co.uk

Acknowledgements

I would first like to thank my supervisors. Prof. Richard Bowden, who consistently pulled me out of rabbit holes and pushed me to accomplish more than I thought I could. Dr. Simon Hadfield, whose patience in correcting this thesis (and explaining particle filters) has been invaluable. Finally, Dr. Nicolas Pugeault, whose patience and dedication (especially on tight deadlines) helped me through the darker times of my PhD.

I would also like to thank the members of CVSSP, who have embraced (and tolerated) me throughout my time at Surrey.

Finally, I would like to thank my family. My parents, Oscar and Ana, who pushed me to do this PhD and consistently believed I could accomplish it. My siblings, Rodrigo, Paola and Bruno, who undoubtedly get tired of hearing me talk about my work when I go home. And my girlfriend Liz, whose unwavering emotional support has kept me feeling sane and loved.

Contents

Nomenclature

NBV	Next-Best View
NBS	Next-Best Stereo
ATE	Absolute Trajectory Error
RMSE	Root Mean Square Error
DoF	Degrees of Freedom
SE(2)	Special Euclidean Space
SE(3)	Special Euclidean Space
SO(3)	Special Orthogonal Space
CNN	Convolutional Neural Network
LUT	Lookup Table
IF	Information Filter
L-LS	Linear Least-Squares
ILLS	Iterative Linear-Least-Squares
MVS	Multi-View Stereo
MoCap	Motion Capture
PID	Proportional Integral Controller
VP	Vanishing Point
BA	Bundle Adjustment
SfM	Structure from Motion
CSfM	Collaborative Structure from Motion
VO	Visual Odometry
KF	Kalman Filter

EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
ICP	Iterative Closest Point
PnP	Perspective N-Points
RANSAC	RANdom SAmple and Consensus
VSFM	Visual Structure from Motion
SLAM	Simultaneous Localisation and Mapping
PTAM	Parallel Tracking and Mapping
DTAM	Dense Tracking and Mapping
PF	Particle Filter
RBPF	Rao-Blackwellised Particle Filter
SMC	Sequential Monte-Carlo
MCL	Monte-Carlo Localisation
AMCL	Adaptive Monte Carlo Localisation
VMCL	Vision-Based Monte-Carlo Localisation
RMCL	Range-Based Monte-Carlo Localisation
PRM	Probabilistic Road Map
RRT	Rapidly-exploring Random Tree
RRT*	Rapidly-exploring Random Tree
ROS	Robot Operating System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
RGB	Red, Green and Blue
RGB-D	RGB and Depth
SeDAR	Semantic Detection and Ranging
SoNAR	Sound Navigation And Ranging
UAV	Unmanned Aerial Vehicle

Symbols

Introduced in Chapter 3

$\dot{\mathbb{X}}$	A set of SE(3) poses.
$\dot{\mathbf{x}}_{NBV}$	The Next-Best View pose.
$\dot{\mathbb{M}}$	The set of reconstructed 3D points.
$\boldsymbol{\eta}\left(\dot{\mathbf{x}}, \dot{\mathbb{M}}\right)$	The cost-function for NBV.
$\dot{\mathbf{x}}$	An SE(3) pose.
$\mathbf{I}_{NBV}$	The image at the current NBV.
$\mathbf{I}_{PRV}$	The image at the last NBV.
$\mathbf{F}_{PRV}$	The flow from $\mathbf{I}_{PRV}$ to $\mathbf{I}_{NBV}$.
$\mathbf{m}$	The 2D coordinates of a point.
$\mathbf{m}'$	The corresponding point of $\mathbf{m}$ in another reference frame.
$\mathbf{F}_{NBV}$	The flow from $\mathbf{I}_{NBV}$ to $\mathbf{I}_{PRV}$.
τ_f	Threshold on the bi-directional optical flow.
$\dot{\mathbf{m}}$	The 3D coordinates of a point.
$\mathbf{P}_{PRV}$	The projection matrix that produces $\mathbf{I}_{PRV}$.
$\mathbf{P}_{NBV}$	The projection matrix that produces $\mathbf{I}_{NBV}$.
$\mathbf{A}$	The coefficients for 3D point estimation.
$\mathbf{b}$	The residuals for 3D point estimation.
$\dot{\mathbf{\Lambda}}_{\dot{\mathbf{m}}}$	The covariance matrix (3×3) of $\dot{\mathbf{m}}$.
$\mathbf{\Lambda}_{\mathbf{m}}$	The covariance matrix (2×2) of $\mathbf{m}$.
$\mathbf{\Lambda}_{\mathbf{m}'}$	The covariance matrix (2×2) of $\mathbf{m}'$.
$\bar{\mathbf{\Lambda}}$	The diagonal matrix of pixel ($\mathbf{m}$) covariances

$\mathbf{B}$	The Jacobian of $\left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b}$
$\dot{v}^e$	An empty voxel.
$\dot{\mathbb{V}}^e$	A set of empty voxels.
$\dot{v}^o$	An occupied voxel.
$\dot{\mathbb{V}}^o$	A set of occupied voxels.
$\dot{v}^u$	An unobserved voxel.
$\dot{\mathbb{V}}^u$	A set of unobserved voxels.
$\dot{\mathbb{V}}$	The set of voxels that defines the octree.
$\dot{v}$	A voxel in the octree.
$\dot{\mathbf{m}}_{\dot{v}}$	A 3D point in voxel $\dot{v}$.
$\dot{\mathbb{M}}_{\dot{v}}$	The set of all 3D points in voxel $\dot{v}$.
$\dot{\mathbf{\Lambda}}_{\dot{v}}$	The covariance matrix of $\dot{\mathbf{m}}_{\dot{v}}$.
$\dot{\mathbb{\Lambda}}_{\dot{v}}$	The set of covariance matrices for the points $\dot{\mathbb{M}}_{\dot{v}}$.
$\dot{\mathbf{m}}^\star$	The putative match for $\dot{\mathbf{m}}'$.
$\dot{\mathbf{m}}'$	A new 3D point being added to the reconstruction.
$\dot{\mathbb{M}}'$	The set of all new 3D points $\dot{\mathbf{m}}'$.
$\dot{\mathbf{\Lambda}}'$	The covariance of the new 3D point $\dot{\mathbf{m}}'$.
$\dot{\mathbb{\Lambda}}'$	The set of covariance matrices for the points $\dot{\mathbb{M}}'$.
$\dot{\mathbf{m}}^\star_{\dot{v}}$	The putative match for $\dot{\mathbf{m}}$ in $\dot{v}$.
$\dot{\mathbf{\Lambda}}^\star_{\dot{v}}$	The covariance of the putative match $\dot{\mathbf{m}}^\star_{\dot{v}}$.
$\mathbf{K}_g$	The estimated Kalman Gain for a point update.
N	The number of euclidean nearest-neighbour candidates.
$\dot{\mathbb{\Lambda}}$	The set of covariance matrices for each reconstructed point in $\dot{\mathbb{M}}$.
$\mathbb{R}_{\dot{\mathbf{x}}}$	The set of rays cast from $\dot{\mathbf{x}}$.
$\mathbf{r}_{\dot{\mathbf{x}}}$	A ray cast from $\dot{\mathbf{x}}$.
$\lambda_{\dot{\mathbf{m}}}$	The eigenvalue of the covariance of point $\dot{\mathbf{m}}$.
$\nu_{\dot{\mathbf{m}}}$	The eigenvector of the covariance of point $\dot{\mathbf{m}}$.

$\boldsymbol{\rho}\left(\mathbf{r}_{\dot{\mathbf{x}}}, \dot{\mathbf{m}}_{\dot{v}}\right)$	The NBV cost-function for each point $\dot{\mathbf{m}}_{\dot{v}} \in \dot{\mathbb{M}}_{\dot{v}}$, in voxel $\dot{v}^o$ intersected by ray $\mathbf{r}_{\dot{\mathbf{x}}}$.
$\boldsymbol{\upsilon}\left(\mathbf{r}_{\dot{\mathbf{x}}}, \dot{\mathbb{M}}_{\dot{v}}\right)$	The NBV cost-function for each ray $\mathbf{r}_{\dot{\mathbf{x}}} \in \mathbb{R}_{\dot{\mathbf{x}}}$ that intersects a voxel $\dot{v}^o$.
$\boldsymbol{\eta}\left(\dot{\mathbf{x}}, \dot{\mathbb{M}}\right)$	The total NBV cost-function for pose $\dot{\mathbf{x}}$.
$\dot{v}_{\mathbf{r}}$	The voxel intersected by ray $\mathbf{r}_{\dot{\mathbf{x}}}$.
γ	The parameter that controls exploration vs. refinement.
δ_B	The baseline distance.
$\dot{\mathbf{x}}_{NBV}$	The Next-Best View (NBV) pose.
$\dot{\mathbf{x}}_{NBS}$	The Next-Best Stereo (NBS) pose.
$\mathbf{r}_V$	The principal ray of the NBV camera.
$\mathbf{r}_S$	The principal ray of the NBS camera.
α	The parameters that controls baseline and vergence.
$\dot{\mathbf{m}}_I$	The intersection point of $\mathbf{r}_V$ and $\mathbf{r}_S$.
δ_{VI}	The distance between the NBV camera centre and $\dot{\mathbf{m}}_I$.
δ_{SI}	The distance between the NBS camera centre and $\dot{\mathbf{m}}_I$.
C_B	The baseline component of the NBS cost.
β	The expected vergence angle.
C_T	The view triangulation component of the NBS cost.
$\mathbf{r}_{VI}$	The ray between the NBV camera centre and $\dot{\mathbf{m}}_I$.
$\mathbf{r}_{SI}$	The ray between the NBS camera centre and $\dot{\mathbf{m}}_I$.
$\dot{v}_G$	The occupied voxel closest to $\dot{\mathbf{m}}_I$.
C_G	The scene structure component of the NBS cost.
$\mathbf{r}_{VG}$	The ray between the NBV camera centre and $\dot{v}_G$.
$\mathbf{r}_{SG}$	The ray between the NBS camera centre and $\dot{v}_G$.
$\mathbf{g}_v$	The gravity vector.
$\mathbf{R}_V$	The rotational component of the NBV camera pose.
$\mathbf{R}_S$	The rotational component of the Next-Best Stereo (NBS) camera pose.
C_R	The rotational component of the NBS cost.

$\boldsymbol{\sigma}\left(\dot{\mathbf{x}}_{NBV}, \dot{\mathbb{X}}_s, \dot{\mathbb{M}}\right)$	The final NBS cost-function.
$w_{_G}$	The scene structure component weight.
$w_{_R}$	The rotational component weight.
$w_{_T}$	The view triangulation component weight.
$\dot{\mathbb{X}}_s$	Set of all possible NBS poses.
$\dot{\mathbf{x}}_{NBS}$	The Next-Best Stereo pose.
$\boldsymbol{\delta}_{nn}\left(\dot{\mathbf{m}}, \dot{\mathbb{M}}\right)$	Nearest-Neighbour function between a point $\dot{\mathbf{m}}$ and a point cloud $\dot{\mathbb{M}}$.
e_{nn}	Average Nearest-Neighbour Error
$\dot{\mathbb{M}}_I$	The nearest-neighbour inliers point cloud.
$\dot{\mathbb{M}}_R$	The reconstructed point cloud.
$\dot{\mathbb{M}}_{GT}$	The ground-truth point cloud.
τ_{nn}	The nearest-neighbour inlier distance threshold.
o_{nn}	Outlier Ratio
c_{nn}	Coverage Ratio

Introduced in Chapter 4

$\mathfrak{T}$	The path-planning trajectory.
$\mathbb{P}$	The State Space.
$\dot{\mathbf{x}}_{goal}$	The goal pose of a robot.
$\dot{\mathbf{x}}_{start}$	The start pose of a robot.
$\mathbb{C}$	Configuration Space.
$\mathbb{C}_{free}$	The free Configuration Space
$\boldsymbol{\zeta}$	The collision detector function.
$\boldsymbol{\pi}(\mathfrak{T})$	The path-planning cost function.
$\dot{v}_{\dot{\mathbf{x}}}$	The voxel where the spatial component of $\dot{\mathbf{x}}$ is contained.
$\mathbb{S}_t$	The set of NBV candidate particles.
s_t^i	The i^{th} NBV candidate particle.
$\boldsymbol{\eta}\left(s_t^i, z_t\right)$	The NBV cost-function for a particle.

$\Pr\left(z_t \middle\vert s_t^i\right)$	The likelihood of the current reconstruction ($z_t = \dot{\mathbb{M}}_t$), given the NBV candidate.
z_t	The observation: the current reconstruction $\dot{\mathbb{M}}_t$.
$\dot{\mathbb{M}}_t$	The reconstruction at time t.
w_t^i	The weight of particle i at time t.
$\mathbb{S}_t^r$	The set of resampled particles.
$\mathbb{S}_t^u$	The set of uniformly sampled particles.
$\mathbb{S}_t^p$	The set of propagated particles.
$\Pr\left(s_t \middle\vert z_t\right)$	The NBV estimation posterior.
s_t	An NBV candidate particle.
$\mathbb{S}$	The set of particles acting as a state-space.
q	A node in the RRT* tree.
$\mathbb{Q}$	The RRT* tree.
q_{start}	The RRT* start node.
q_{goal}	The RRT* goal node.
q_{rand}	The RRT* random node.
q_{near}	The RRT* near node.
q_{new}	The RRT* new node.
$\mathbb{Q}_{nn}$	The set of nearest-neighbour node candidates.
q_{nn}	The nearest-neighbour node.
$\boldsymbol{\pi}_{sfm}(\mathfrak{T})$	The SfM path cost function.
$\boldsymbol{\pi}_{col}(\mathfrak{T})$	The collaborative stereo path cost function.
$\dot{\mathbf{x}}_o$	The coordinates of the "other" robot in the collaborative path cost estimation.

Introduced in Chapter 5

$\mathbb{V}$	The set of all map cells.
$\mathbb{Z}_t$	The set of all historical robot sensor observations at time t.
$\mathbb{U}_t$	The set of all historical robot odometry measurements at time t.
$\mathbb{X}_t$	The set of all possible robot poses.

$\mathbf{x}_t$	The current pose to be estimated.
$\mathbb{S}_t$	The set of pose candidate particles.
$\Pr\left(s_{t-1}^i \middle\vert \mathbb{Z}_{t-1}, \mathbb{U}_{t-1}\right)$	The MCL prior.
$\Pr\left(s_t^i \middle\vert \mathbb{Z}_t, \mathbb{U}_t\right)$	The MCL posterior.
$\Pr\left(s_t^{i\prime} \middle\vert u_t, s_{t-1}^i\right)$	The MCL motion model.
$\Pr\left(z_t \middle\vert s_t^{i\prime}, \mathbb{V}\right)$	The MCL sensor model.
z_t	An observation reported by the sensor at time t.
u_t	The odometry reported by the robot at time t.
$\mathbb{S}_{t-1}$	The set of particles at time $t-1$.
$\mathbb{S}_t'$	The set of Particles at time t, before resampling.
$s_t^{i\prime}$	The i^{th} particle with the motion model applied.
s_{t-1}^i	The i^{th} particle at time $t-1$.
w_t^i	The weight of the i^{th} particle at time t.
s_t	An NBV candidate particle.
$\boldsymbol{\Upsilon}_t$	The covariance of the odometry.
$\Pr\left(z_t^k \middle\vert s_t^{i\prime}, \mathbb{V}\right)$	The likelihood of a single observation tuple.
θ_t^k	The k^{th} bearing angle of an observation z_t.
r_t^k	The k^{th} range of an observation z_t.
r_t^{k*}	The range travelled by the k^{th} raycasting operation in the beam model.
$v_{\mathbf{m}}$	A cell at location $\mathbf{m}$ in the map.
$v_{\mathbf{m}}^o$	An *occupied* cell at location $\mathbf{m}$ in the map.
$v_{\mathbf{m}}^d$	A cell with label *door* at location $\mathbf{m}$ in the map.
$v_{\mathbf{m}}^a$	A cell with label *wall* at location $\mathbf{m}$ in the map.
$v_{\mathbf{m}}^w$	A cell with label *window* at location $\mathbf{m}$ in the map.
$v_{\mathbf{m}}^\ell$	A labelled cell at location $\mathbf{m}$ in the map.
d_t^k	The current depth measurement at pixel k.
ℓ_t^k	The k^{th} semantic label of an observation z_t.
$\Pr\left(s_t^{i\prime} \middle\vert u_t, s_{t-1}^i, \mathbb{V}\right)$	The real motion model that includes the map ($\mathbb{V}$).
$\Pr\left(s_{t-1}^i \middle\vert \mathbb{V}\right)$	The motion model prior.

$v^{o}_{s_{t-1}}$	The occupancy likelihood of the cell that contains s^{i}_{t-1}.
δ_{d}	The distance to the nearest door.
ϵ_{G}	The *Ghost Factor*.
δ_{o}	The distance to the nearest occupied cell.
$\mathbf{m}$	A 2D point (on the map).
$\Pr_{\text{RNG}}\left(z^{k}_{t} \middle\vert s^{i\prime}_{t}, \mathbb{V}\right)$	The observation likelihood for range-based measurements.
σ_{o}	The expected noise of the range measurement.
$v^{\ell}_{\mathbf{m}'}$	A cell with label $\ell \in \{a, d, w\}$, at $\mathbf{m}'$.
δ_{ℓ}	The distance to the nearest cell with label $\ell \in \{a, d, w\}$.
δ_{w}	The distance to the nearest window.
δ_{a}	The distance to the nearest wall.
$\Pr_{\text{LBL}}\left(z^{k}_{t} \middle\vert s^{i\prime}_{t}, \mathbb{V}\right)$	The observation likelihood for label-based measurements.
σ_{ℓ}	The semantically adaptive standard deviation.
ϵ_{ℓ}	The weight for the semantic label sensor model.
ϵ_{o}	The weight for the range-based sensor model.
$\Pr\left(\ell\right)$	The label prior.
$\mathbb{G}_{t}$	The set of ground truth poses.
${}^{\mathbb{G}}\mathbf{T}_{\mathbb{X}}$	The transform between the ground truth trajectory and the estimated trajectory.
$\mathbf{g}_{t}$	The current ground truth pose.

List of Figures

List of Tables

Declaration

The work presented in this thesis is also present in the following manuscripts:

- Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. Next-Best Stereo: Extending Next-Best View Optimisation For Collaborative Sensors. In *British Machine Vision Conference (BMVC)*, York, UK, 2016. BMVA Press. (Oral).
- Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. Taking the Scenic Route to 3D : Optimising Reconstruction from Moving Cameras. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017. IEEE.
- Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. SeDAR - Semantic Detection and Ranging: Humans can localise without LiDAR, can robots? *arXiv*, cs.RO, 2017.
- Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. SeDAR - Semantic Detection and Ranging: Humans can localise without LiDAR, can robots? In *International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018. IEEE. Under Review.

Chapter 1

Introduction

Traditional mechanical automation focused on constrained scenarios, such as production lines performing repetitive tasks. The scope of these scenarios was limited by the understanding afforded by simple sensory input. Nonetheless, manual labour was quickly replaced by this type of early robotics, significantly increasing the production of food, goods and services. However, there are fundamental limitations to automating processes in this basic way. Mechanical automation typically uses robots equipped with very simple sensors, such as potentiometers and photo-diodes, which are not capable of generating a nuanced understanding of the world. Due to their simplistic nature, this type of robot relies on an *a priori* knowledge of the world. Fundamentally, this means that the robots have no understanding of the world they interact with and cannot react appropriately to changes in their environment. This lack of understanding also precludes these robots from any kind of collaborative behaviour (human-to-robot or robot-to-robot).

Modern *reactive* automation, in the form of intelligent robotic or robot-like systems, has been taking over more traditional automation technologies. Whether these agents are robotic manipulators, home automation technologies or self-driving cars, they are increasing productivity and reducing human risk. This level of automation has been made possible, in large part, by an increase in the complexity of their sensors. Modern sensors, such as cameras and Light Detection And Ranging (LiDAR), allowed robots to tackle tasks far more complex than anything attempted by their predecessors. Unfortunately, these powerful technologies are normally limited to scenarios that can be understood *passively*. This implies that robots take no action to improve

their sensory input, limiting their understanding of the world. This also limits the ability of the robots to collaborate with other intelligent agents, such as robots and humans, to hard-coded robot-to-robot interactions. It is clear that while current *reactive* automation is powerful, it still presents important limitations.

Pro-active automation, in the form of unconstrained, real-world, collaborative, autonomous agents, is the goal of current robotics research. In the same way that humans can predict and adapt to the actions of a co-worker, robots should be able to do this with both people and other robots. Enabling this level of collaboration requires an understanding of the world that is shared between all agents. In order to share their representation of the world with humans, robots need to be capable of the high-level semantic understanding that humans use everyday. The ability to pro-actively generate this shared semantic understanding of the world, in its full 3D structure, is something that humans take for granted and robots desperately need. It will allow robotic agents to perform autonomous "world-building", where the robot actively seeks the information that enables high-level goal completion. A shared understanding of the world also allows collaborative behaviours to emerge naturally from the high-level goals, rather than being hard-coded to the problem domain. Semantic understanding will enable true human-to-robot collaboration, allowing humans to interact with robots as they do with other people.

One way of increasing the world understanding ability of robots is by allowing them to pro-actively reconstruct the world they operate in. Reconstructions are capable of providing autonomous agents with the information they need to interact with the world, each other, and humans. Historically, reconstruction has played an important role in the field of robotics [29]. More recently, reconstruction has become one of the most active topics in computer vision. This is because having a complete 3D Reconstruction of an object and/or scene is useful for nearly all vision-based tasks. It can aid in Tracking [79], Segmentation [65], Localisation [14], Detection [39] and Navigation [136] (to name a few). This makes reliable, all-purpose reconstruction algorithms extremely important to the development of the field. More importantly, these "vision-based" tasks are also key abilities that robots require in order to interact with humans.

It is clear that increasing the world understanding ability of a robot is one of the key steps in bridging the gap between *reactive* and *pro-active* robotics. However, it is not enough to simply

(a) Dyson Eye (b) Baxter

Figure 1.1: Commercial robotic platforms.

"understand" the world: autonomous agents must be able to actively and collaboratively acquire this understanding. Therefore, the aim of this thesis is to explore methods that will enable modern robotic systems to autonomously and collaboratively generate an understanding of the world. In the following section, this chapter will present the motivation of this aim, and will also provide a breakdown of the objectives necessary to achieve it.

1.1 Motivation

Currently, state-of-the-art robots have begun to enter the mainstream. Products such as autonomous vacuum cleaners, UAVs, and accessible pick-and-place robots have become ubiquitous in everyday life. Whether they are cleaning a room, aiding search-and-rescue or even filming, robots help people perform difficult and/or tedious tasks. Unfortunately, most of these robots are not inherently capable of understanding the world they operate in. This makes these robots reactive, as they are not capable of being pro-active in their actions, goals or sensing. Therefore, the primary motivation of this thesis is to develop techniques that can be used by mainstream robots to become more pro-active in achieving their goals.

Encouragingly, some robotic agents have already developed techniques to address some of these shortcomings. Platforms such as the Dyson Eye (in Figure 1.1a) use localisation approaches

based on Simultaneous Localisation and Mapping (SLAM) to navigate a house for cleaning. On the other hand, static robots such as Baxter by Rethink Robotics (in Figure 1.1b) use depth sensors to intelligently grasp objects. However, both these cases present important limitations. Firstly, sparse SLAM-based techniques are ideal for localisation but provide very little information about the world to the platform. Secondly, depth sensors are expensive, resource hungry, large and (in the case of Baxter) must be registered to the robot. Thirdly, these robots are not capable of collaborating with other similar agents. Finally, and most importantly, these robots do not have a method of using their sensors *pro-actively*. Their understanding of the world is dependant on the observations obtained while performing the tasks they were designed for.

Allowing robots like those in Figure 1.1 to use cheap, low-cost sensors to autonomously and collaboratively generate an understanding of the world would significantly increase their capabilities. However, it is important to note there are several challenges. Creating a 3D reconstruction normally requires large amounts of data, memory and processing power. Autonomously creating a reconstruction additionally requires complex decision-making on the part of the robot (*e.g.* where to go and how to get there). Collaborating with another agent makes the 3D reconstruction more expensive and the decision-making more complex. Lastly, any reconstruction is useless if it does not accurately reflect the underlying geometry and semantics of the world.

These challenges provide a natural set of objectives which set a path to the overall aim of this thesis. These objectives are:

1. To provide quick and efficient 3D reconstruction methods.
2. To develop techniques for autonomous decision-making and exploration.
3. To explore emergent behaviours for collaboration.
4. To explore the utility of semantic information to the reconstruction process.

1.2 Contributions

In order to address these objectives, this thesis presents several distinct but interrelated pieces of work.

Figure 1.2: Sample MVS Reconstruction [46].

To provide context to this work, Chapter 2 will discuss the current state of the art in the fields of 3D Reconstruction, Autonomous Exploration & Navigation and Localisation. Recent, high-impact and seminal work will be discussed in the context of this thesis. Specific focus will be given to the differences in the literature from both the computer vision and robotics perspectives.

To address the first objective, Chapter 3 uses the concept of the Next-Best View (NBV) in order to guide an MVS-based 3D reconstruction. The field of MVS has traditionally been capable of producing impressive reconstructions from unordered image collections [1]. The fidelity of these methods is normally extremely high, approaching photo-realistic results, as shown in Figure 1.2. However, a common draw-back of even the most high-quality reconstructions is the high number of images needed and the associated computational cost of processing & integrating information from the data. In recent years, pro-active view selection has become a much more tractable method for processing large datasets. NBV, as it is known in the literature, is the process of selecting the view that will be most beneficial to the reconstruction. Performing

iterative NBV selection has the ultimate goal of producing a reconstruction comparable to processing the whole dataset, in a fraction of the computing time and memory. This thesis proposes an approach to intelligently filter large amounts of data for 3D reconstructions of unknown scenes using monocular cameras. The contributions are two-fold. Firstly, an approach is presented that efficiently optimises the NBV in terms of accuracy and coverage using partial scene geometry. Secondly, the NBV is extended to intelligently select stereo pairs by jointly optimising the baseline and vergence to find the best stereo pair for the NBV, the Next-Best Stereo (NBS). The NBV and NBS approaches are evaluated on both the Middlebury MVS dataset and a dataset obtained from an Unmanned Aerial Vehicle (UAV) (see Figure 1.3a). The work in this chapter is an extended version of the approach published by the author as an oral presentation within [86].

To address the second and third objectives, Chapter 4 extends the NBV selection of Chapter 3 to a continuous setting, and introduces a path-planning and collaboration method which enables its application to robotics. Path-planning is the problem of travelling from the current position of the robot to a given goal, and usually focuses on obstacle avoidance whilst minimising path length. This approach is ill-suited to reconstruction applications, where learning about the environment is more valuable than speed of traversal. The same can be said about travelling from the current view to the NBV. Furthermore using the NBV selection from Chapter 3 to estimate the goal of a live robot is a non-trivial problem. Chapter 4 addresses these problems through three main contributions. Firstly, a Sequential Monte-Carlo (SMC) approach is used to select the NBV in a continuous pose-space, while simultaneously estimating a cost-space of informative views. Secondly, a *Scenic Pathplanner* is introduced that plans a trajectory (to the NBV) which will benefit the reconstruction, both in terms of total map coverage and accuracy. Finally, an innovative *Opportunistic Collaboration* method is introduced to enable multi-robot reconstruction by allowing sensors to switch between acting as independent Structure from Motion (SfM) agents or as a variable-baseline stereo pair. These techniques are validated in a 6 Degrees of Freedom (DoF) dataset obtained using a low-cost UAV (Figure 1.3a), as well as a live ground-based robotic platform (Figure 1.3c). The work in this chapter was published by the author within [88].

To address the final objective, Chapter 5 presents a global localisation approach that relies solely on the semantic labels extracted from RGB images. This semantic-level understanding

of the world is used to lay the foundation for globally-consistent reconstructions *without* expensive optimisations or depth sensors. Floorplan localisation has the ability to provide globally consistent pose estimates that are, unlike SLAM and MVS, also consistent with the real-world. However, most approaches explicitly measure depths to every visible surface and try to match them against different pose estimates in the floorplan. Known as MCL, this approach is diametrically opposite to what humans do. Humans rely on high-level semantic cues to localise themselves within the floorplan. Evidence of this is that many of the floorplans used in everyday life are not accurate, opting instead for high levels of discriminative landmarks. Unfortunately, in robotics this high-level information is normally discarded in favour of estimated depth measurements, even by vision-based approaches. Instead of discarding this valuable information, semantic segmentation is used in order to augment traditional MCL-based approaches with high-level semantic labels. Chapter 5 introduces a sensing modality that can present bearing, range and semantic information in a way that is usable by more traditional MCL approaches. The complete sensing and localisation framework, called Semantic Detection and Ranging (SeDAR), presents 3 main contributions. Firstly, a motion model is defined using the semantic information present in the floorplan. Secondly, a sensor model that combines the bearing, range and semantic information in SeDAR is introduced. Finally, an additional sensor model is presented that only depends on label and bearing information (an RGB image). The contributions of this chapter are evaluated on a live Turtlebot (Figure 1.3b), where it is shown that SeDAR can outperform state-of-the-art MCL approaches. The work in this chapter was published by the author within [87] and [89].

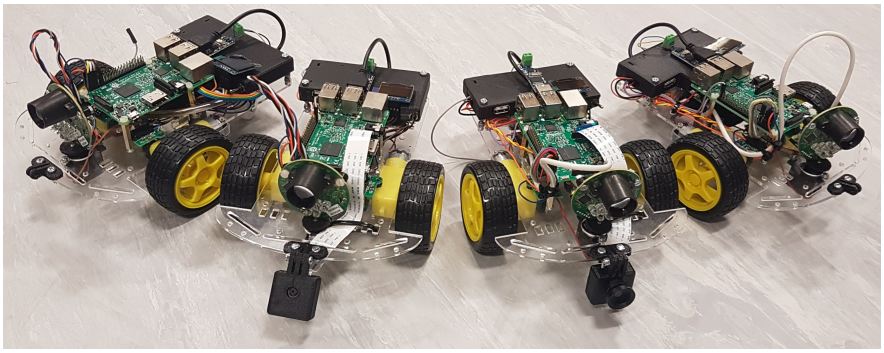
Finally, Chapter 6 concludes this thesis. It presents a summary of the contributions and a discussion of the failure cases. The failure cases and limitations will then be used to inform a series of possible future work directions.

1.3 Hardware Contributions

As stated before, the aim of this thesis is to explore methods that will enable modern robotic systems to autonomously and collaboratively generate an understanding of the world. A key part of this aim is to make the contributions of this thesis directly applicable to live robotic systems. Therefore, robotic hardware is necessary for development, testing and validation of

(a) Parrot ARDrone 2.0 [11].

(b) Willow Garage Turtlebot 2 [49].



(c) CVSSP Roaches.

Figure 1.3: Platforms used in this thesis.

the contributions. However, an extensive explanation of the platforms used is beyond the scope of this thesis. Therefore, this section will aim to present and document the main contributions made in the development of robotic hardware.

This thesis used three main hardware platforms: the Parrot AR Drone 2.0 [11] (Figure 1.3a), the Willow Garage TurtleBot 2 [49] (Figure 1.3b) and the CVSSP Roaches (Figure 1.3c). The Parrot AR Drone 2.0 and the Willow Garage TurtleBot 2 are commercial platforms that were acquired for use in this thesis. In both cases, modifications were required to make them viable for research. On the other hand, the CVSSP Roaches were designed and built specifically for this thesis, and present a more substantial contribution. In all cases, the robots were controlled using the Robot Operating System (ROS) [107].

The Parrot AR Drone 2.0 was used as part of Chapters 3 and 4. The AR Drone is a low-cost UAV designed for the consumer market. It features on-board stabilisation and a forward facing camera. In order to make this platform viable for research, it was necessary to implement a feature-based SLAM and control system [35]. Modifications to the approach of Engel *et al*. [35] allowed the UAV to be controlled directly from client code and safety features to be implemented. Once the UAV was operational and safe, it was used to acquire data. An effort was made to operate the UAV live, but hardware limitations prevented their extended use.

The Willow Garage TurtleBot 2 was used to develop and validate the contributions of Chapter 5. TurtleBot 2 is a ground-based robotic platform specially designed around ROS. It features an RGB-D camera, accurate odometry (Inertial Measurement Unit (IMU) and wheel), and a robust navigation stack. Minor modifications to the hardware were necessary to add a 50amp-hour battery and a mini-computer. Once the TurtleBot was operational, it was used to acquire live data and perform experiments.

The CVSSP Roaches were used to develop and validate the contributions of Chapter 4 in a live scenario. They are ground-based platforms developed by the author, with a total of four robots built and tested. The Roaches feature wheel odometry, an RGB camera, an IMU and a Raspberry Pi for on-board computation. They are an extremely low-cost platform, made entirely out of consumer-level parts. Once the robots were built, a robust localisation and navigation framework was developed for them. The localisation framework was based on a feature-based SLAM [72] system combined with a custom coordinate-frame registration framework designed specifically for the Roaches. The navigation framework was based on the well-tested ROS navigation stack. In both cases, the software required significant changes to work on the low-cost hardware the Roaches are built from. The fact that robust localisation, navigation and 3D reconstruction are feasible on such low-cost hardware present an important contribution to the state-of-the-art for robotic platforms.

1.4 Summary

To summarise, the contributions of this thesis focus on establishing a baseline for autonomous 3D reconstruction that aims to *actively* and *collaboratively* reconstruct a scene. The NBV and

NBS contributions of Chapter 3 satisfy the first goal of this thesis. Chapter 4 presents the Scenic Pathplanner and Opportunistic collaboration, which satisfy the second and third goals. Finally, Chapter 5 addresses the final goal by introducing the floorplan localisation framework, SeDAR. The evaluation of these contributions is performed on real robotic hardware, ensuring that the contributions are directly applicable to live systems.

Chapter 2

State-of-the-Art

During the formative years of Robotics and Computer Vision, localisation and reconstruction were considered independent goals. Simultaneous Localisation and Mapping (SLAM) emerged when it was realised that there was a conditional dependence between these two goals, making joint estimation easier. Over the years, SLAM has been extremely successful in creating dense [100], fast [95] and accurate [75] 3D reconstructions. More recently, Visual Odometry (VO) and Sensor Fusion have also become important parts of the SLAM literature, allowing for efficient localisation approaches. As such, the first part of this literature review will focus on SLAM-based approaches.

As the field of SLAM progressed, Autonomous Navigation began to emerge as a field in its own right, using SLAM as an underlying framework to understand the world. The Autonomous Navigation literature is concerned with enabling high-level decision making and the execution of those decisions. Goal Estimation quickly emerged as a way of allowing robots to decide where they should go next. On the other hand, Path-planning was concerned with enabling robots to reach their goal. Finally, the field of Multi-Robot control emerged out of a need to expand these abilities to multi-agent scenarios. The second part of this literature review will focus on techniques that enable Autonomous Navigation and Exploration.

As SLAM systems became more efficient, and autonomous agents acquired more capabilities, the fields of independent Reconstruction and Localisation began to once again gain traction within the literature.

In terms of reconstruction, SLAM is still consistently outperformed by state-of-the-art offline approaches. The detailed reconstructions provided by batch techniques, such as Multi-View Stereo (MVS), can be used by robots to enhance their understanding of the world. More importantly, the high computational complexity of these approaches gave rise to the field of Next-Best View (NBV). In robotics, and this thesis, NBV is used as a proxy for goal-estimation which allows robots to pro-actively reconstruct an environment. As such, these techniques will be explored in the third part of this literature review.

In terms of Localisation, the recent rise in 3D sensors and scanners has made localising within a pre-built map extremely robust. This type of localisation has the ability to provide robots with a globally consistent pose that can be shared with other agents to enable collaboration. More importantly, recent advances in Deep Learning have allowed the use of high-level semantic information to perform localisation. This level of semantic localisation is the first step towards a shared human-robot understanding of the world. Therefore, the current state-of-the-art for pre-existing map localisation will be discussed as the last part of this literature review.

2.1 Simultaneous Localisation and Mapping (SLAM)

SLAM is the problem of jointly estimating the pose of a sensor and the geometry of the world it is in. It is considered one of the cornerstones of truly autonomous systems. From a theoretical point of view, the SLAM problem can be considered as solved [4]. However, its practical implementation is still an active area of research, with numerous publications at the major robotics and vision conferences [21, 75, 84, 95].

2.1.1 SLAM Paradigms

There are three main SLAM paradigms [133]: filter-based, particle-based and graph-based. Filter-based approaches usually take the form of an Extended Kalman Filter (EKF), such as EKF-SLAM [118]. Particle-based approaches typically employ Rao-Blackwellised Particle Filters (RBPFs), as used in FastSLAM [91]). Graph-based approaches follow some form of optimisation, such as Bundle Adjustment (BA) (*e.g.* [72]). These approaches can be applied to

any kind of data, including non-visual sensors such as LiDAR or visual sensors such as stereo, RGB-D and monocular cameras.

Filter Based Approaches

All three approaches share the same standard Bayesian formulation, which came about during the foundational years of the field. This Bayesian formulation established the correlation of all landmarks with each other via the robot pose [29, 117]. The combined localisation and mapping problem was shown to be convergent [4, 129]: as the number of landmark observations increases, the determinant of the covariance matrix decreases monotonically [129]. This led to the very early filter-based solutions to the SLAM problem [118], which usually relied on an EKF. There are publications using Unscented Kalman Filters (UKFs) and Information Filters (IFs), but the EKF quickly became the dominant approach in the field and efforts concentrated on efficient implementations [4]. As the field evolved, work such as MonoSLAM [22] applied EKFs to vision only, real-time approaches. However, as with all Kalman Filter (KF) based approaches, these algorithms all suffered with a monotonically growing covariance matrix due to the fact that each landmark needs to be added to the state vector and its covariances maintained. EKF-based approaches also suffered from an $O(N^2)$ (where N is the number of features) computation time for measurement updates, since updating one landmark implies updating the whole covariance matrix.

This meant that, while systems that relied on a controlled number of reliable landmarks worked well, autonomous map growth was limited to small or medium size environments using sparse landmarks [22]. In order to solve this, Eade and Drummond [31] introduced a system that maintained a graph of local coordinate frame (nodes) joined by similarity transforms (edges). Coordinate frames were chosen to minimize non-linearity and updates were only local, minimising the computational complexity. KF based systems also suffered from vulnerability to incorrect data association (mistaking which measurements belonged to which landmark) and could quickly become corrupted. For this reason, EKF approaches usually incorporate robust maximum-likelihood heuristics [133]. It should be mentioned that this sensitivity to data association is also one of its main benefits, a correct loop closure. Loop closure is data association that happens when the robot revisits a known location after travelling, and accumulating uncertainty.

Correctly identifying a loop closure tightens the estimate of the re-observed landmarks and propagates the uncertainty reduction back across all landmarks via the covariance matrix update. However, due to its limitations, KF based approaches were soon replaced by other methods.

Particle-Based Approaches

The Particle Filter (PF) solved the inability of EKF-based approaches to operate on large environments. Murphy [97] introduced the RBPF and applied it to grid mapping, Doucet *et al*. [27] proved that the RBPF leads to more accurate results, but it was FastSLAM by Montemerlo *et al*. [91] that first applied a RBPF to online SLAM. RBPFs exploit the idea that each particle can represent the full path of the robot. Having each particle represent a possible path implies that the path is known, and the map can be conditioned upon it [97]. This causes the landmarks to be decoupled from each other, allowing the filter to only have to maintain an independent Gaussian (2×2 EKF) for each landmark. Furthermore, the approach of RBPFs to online SLAM only requires the current pose (and PFs never revisit past estimates) so it is possible to drop the historical pose from particles to reduce the complexity to $O(KlogN)$ (where K is the number of particles and N is the number of landmarks). FastSLAM assumed known data association, it was not until later that Montemerlo and Thrun [90] incorporated ways of dealing with unknown data association.

A common problem with PF-based approaches is particle deprivation [131]. In SLAM, it is caused when the motion model of the robot is noisy compared to its motion. A noisy motion model causes particles to fall into areas with low measurement likelihood, causing them to be terminated. This means that as the robot moves, the particles that represent distinct paths in the past start to converge. This causes problems during loop-closure, as the converged path is not necessarily the correct one. Montemerlo *et al*. [92] introduced a second version of the FastSLAM algorithm where the particles are sampled according to the motion *and the observation*. The motivation being that if the observation is included in the particle propagation, the re-sampling will result in a more diverse pose history that responded better to loop closure. The improvements were significant, but came at the cost of added complexity.

Graph-Bases Approaches

Graph-based approaches can be considered the current state of the art for the field. First introduced by Lu and Milos [81], this approach establishes that landmarks and poses can be thought of as nodes in a graph. The edges of said graph can be thought of as constraints, where poses are constrained by odometry readings and landmarks are constrained by observations [50]. Relaxing these constraints yields the best estimate for the map. The reason this approach works so well is that the graph is sparse and linear in time and number of nodes [133]. This means that the graph allows for constant update time and linear memory requirements. The graph is usually optimised using Gauss-Newton or Levenberg-Marquardt, exploiting the sparse nature of the problem. Furthermore, these graph optimisers linearise the problem at each iteration, as opposed to relying on an initial linearisation like an EKF [50] [135]. Their main weakness is that they assume sparseness of the underlying matrix. This assumption breaks in the case of obtaining many observations in the same vicinity (such as a static robot) because the observations are necessarily correlated to each other. Most approaches solve this by using a keyframe approach, where a heuristic metric is used to determine if a specific frame is to be added to the map. Finally, the graph optimisation is usually an expensive process. It is linear in the number of edges (poses N plus landmarks M), but these can grow quadratically and in the case of a BA the complexity is $O((M+N)^3)$ per iteration. This has prompted most approaches to split into a sensor dependant front-end that captures data and performs localised SLAM and a sensor agnostic back-end that performs global SLAM via optimisation on abstract graph data [50].

The problem has been approached in several different ways by the robotics community, Frese *et al*. [44] apply relaxation at different resolutions, GraphSLAM [134] uses variable elimination techniques to reduce the dimensionality of the problem and Estrada *et al*. [36] use Hierarchical SLAM to maintain independent local maps in an adjacency graph where edges are relative locations of the maps.

From the computer vision literature, the graph method of Eade and Drummond [31] coalesces all observations into locally linear nodes to be optimised by a BA with one view and a prior based on the information matrix of the node. The graph edges correspond to similarity transforms and are later optimised in a global framework. Parallel Tracking and Mapping (PTAM), by Klein and Murray [72], makes use of a FAST [109] corner detector along with a BA in order to obtain

a pose estimate and map a small workspace. Their key contribution was to schedule tracking and mapping in parallel, which gave their algorithm the chance to run expensive global Bundle Adjustments separately from tracking.

Graph-based approaches are particularly suited to solving large-scale mapping problems. Their sparsity allows for fast updates and the ability to apply optimisation techniques in real-time makes them invaluable for accuracy in the long-term. This is demonstrated by Strasdat *et al*. [121] who make a comparison between PTAM by Klein and Murray [72] and the method by Eade and Drummond [31]. The comparison is done in order to make a case against filtering, and the fact that [31] maintains local information matrices essentially classifies the approach as filtering with a graph-based wrapper. They compare them based on accuracy and cost and make the case that in order to increase the accuracy of monocular SLAM, it is better to have more features than more frames. Since filter based approaches are notorious for their inability to maintain large numbers of features, they conclude that while filtering might work on computationally limited frameworks, a BA (and therefore the graph-based approach) is generally superior.

2.1.2 Visual SLAM

Recently, vision-based SLAM has become the approach of choice in both computer vision and robotics. This is because monocular cameras are typically cheaper sensors with lower power requirements than traditional LiDAR systems. This is also true for more complex cameras, such as RGB-D and stereo. Generally speaking, vision-based SLAM can be split into sparse and dense methods.

Feature-Based SLAM

The comparison by Strasdat *et al*. [121], along with the robust nature of the approach, caused PTAM [72] to become the de-facto standard for sparse visual SLAM. As such, PTAM has paved the way for more BA-based approaches. Strasdat *et al*. [122] use the PTAM idea of splitting tracking and mapping to create large scale maps that are optimised on a separate thread. However, instead of a BA, they propose a method of using pose-only graphs (landmarks

are used to create constraints) optimised in a 7-DoF framework that solves for a rigid body transform and scale. Kümmerle *et al*. [74] later generalised BA and pose-graph approaches into a general framework for graph optimisation. Other work extending from PTAM is from Castle, Klein and Murray [15] who add to it the ability to create multiple local maps that a single camera can explore. It does not create any links between the map, but rather uses the reinitialisation procedure to identify previously mapped areas. Harmat *et al*. [52] extend PTAM into an approach capable of tracking multiple rigidly attached cameras in a single map. The quantity and quality of features has a significant impact in the performance of feature-based SLAM. Recently, approaches such as Cavestany *et al*. [16] addressed this issue by curating the feature bundles that correspond to a 3D point, thereby increasing the accuracy of the BA. Finally, Mur *et al*. [95] combine the main ideas of PTAM, the scale awareness of Strasdat *et al*. [122] and other important techniques into a robust framework for SLAM. This framework operates in real-time, uses ORB features [110] for all tasks (localisation, relocalisation, mapping, loop closure, etc.), introduces a new Essential Graph method of performing loop closure and generally presents a state-of-the-art version of sparse SLAM.

Sparse systems are good for pose estimation and stabilisation, but are generally not dense enough to provide scene understanding and detailed reconstructions. In this thesis, sparse SLAM systems are used to provide pose estimates. However, their reconstructions will be replaced with more dense methods.

Direct SLAM

Once the field established that sparse, feature-based SLAM achieved accurate localisation, the focus shifted towards direct, dense (or semi-dense) approaches. Direct approaches typically avoid expensive feature extraction, and instead rely on photometric (per-pixel) methods. This allows these methods to densely track pixels across multiple images, and use this information to both triangulate the pixels and estimate the pose of the camera.

Newcombe *et al*. [100] introduced Dense Tracking and Mapping (DTAM), an approach which uses a discretised cost volume in a global minimisation framework that minimises the sum of projective photometric errors over a set of frames. More explicitly, each keyframe in this approach defines a photometric projective cost volume that can be optimised into a dense depth

map. Tracking is then simply a minimisation of the photometric error on the current frame compared to the projected dense depth map. While the approach of Newcombe *et al*. [100] performed remarkably well, it required a GPU to be optimised reliably. Ondruska *et al*. [101] extended this work to function on mobile phones, being able to reconstruct smaller scenes using a similar volumetric depth map fusion approach.

With the rise of affordable RGB-D sensors, such as the Microsoft Kinect, dense reconstructions became significantly more tractable. Paton and Kosecka [102] use a combination of feature matching and Iterative Closest Point (ICP) to perform per-frame alignment and reconstruct a scene. Kinect Fusion, by Newcombe *et al*. [99] uses a similar volumetric approach to [100], but instead relies on a signed-distance function to define the cost space and ICP-based pose estimation. Keller *et al*. [70] moved away from the volumetric approach opting instead for a surfel-based approach, but keeping the ICP-based pose estimation. Whelan *et al*. [139] extend the work of Keller, but add a time window to mark surfels as inactive which allows them to create larger maps. McCormac *et al*. [84] extend the work of Whelan *et al*. by adding semantic labels extracted from a Convolutional Neural Network (CNN) to the reconstruction. Dai *et al*. [21] and Labbe and Michaud [75] combine multiple techniques from feature-based and depth-based approaches to provide an extremely robust framework that can perform loop closure, visual & depth-based reconstructions and on-the-fly optimisation.

Dense approaches have become extremely robust, unfortunately they are still largely intractable for small mobile robots. They are either computationally demanding, or require expensive RGB-D sensors. While approaches such as that of Labbe and Michaud [75] have seen some use in higher-end robots, they are not ideal for mainstream robotics.

Due to the high computational cost of fully dense approaches, their reliance on GPU hardware and/or expensive sensors, more recent work has focused on semi-dense reconstructions. Engel *et al*. [33] uses direct, semi-dense, gradient-based depth maps, using the back-end optimisation approach of Kümmerle *et al*. [74]. Tateno *et al*. [126] extend the work of Engel *et al*. a CNN-based depth estimation and the semantic label predictions of Laina *et al*. [76]. Similar work by Pizzoli *et al*. [104] uses semi-direct measurements for tracking, and a probabilistic epipolar constraint to achieve dense reconstructions.

While semi-dense approaches are ideal for online reconstruction, they suffer from two important

limitations. Firstly, they are not suitable for navigation. This is because the discrete nature of their reconstructions usually preclude robust path-planning and collision avoidance. Secondly, dense/semi-dense approaches tend to have a slower response time than feature-based approaches and are therefore not widely used in real-time robot pose estimation. Mur-Artal *et al*. [96] attempted to address this by applying a similar technique to Pizzoli *et al*. [104], but over a feature-based framework [110]. However, reconstructing a semi-dense scene is an expensive process that normally slows down these approaches.

2.1.3 Visual Odometry

While SLAM aims to jointly estimate the pose and reconstruct the map, VO aims for pose estimation only. Unfortunately, the process of pose estimation usually necessitates some level of reconstruction to maintain a consistent scale. Therefore Visual Odometry (VO) can be thought of as SLAM with a very constrained local scope. As such, the PTAM [72] paradigm seems to have become prevalent in most VO work, where the practice of running a local BA has become commonplace. Engel *et al*. [34] use direct measurements that rely on gradients to estimate depth. Forster *et al*. [42] use semi-direct methods to avoid feature extraction. The pose estimates from these algorithms are normally used as the "online" part of SLAM systems [41, 33, 37], or as part of sensor fusion frameworks [9, 12].

2.1.4 Sensor Fusion

In robotics, sensor fusion often uses the pose obtained from SLAM or VO, as well as information from other sensors (IMU, Wheel Odometry, Altimeter, etc.) in a framework that merges these sources of pose information using their uncertainty (covariances) in order to create a more robust estimate of the location of the sensor.

Early approaches used external sensors within a SLAM framework to aid pose estimation. One such approach is Kneip *et al*. [73] who constrain the estimates of a Perspective N-Points (PnP) algorithm by using relative IMU readings of the current frame and the nearest keyframe. However, the field quickly moved away from these approaches in favour of KF-based frameworks. Sensor fusion is usually performed using a either an Extended Kalman Filter (EKF) or an Unscented

Kalman Filter (UKF), to overcome the non-linearities of 6-DoF pose space. Engel *et al*. [35] use the pose estimate out of PTAM and UAV odometry (IMU plus altimeter) in order to predict the pose of the robot reliably using an EKF. A Proportional Integral Controller (PID) is then implemented to steer the UAV to the required location. The paper also includes a closed loop solution to finding the scale of a map using the altimeter and judicious (up-down) movement of the UAV. Blosch *et al*. [9] base their work on a more control-theoretic approach of integrating visual and inertial odometry, but essentially modify PTAM to facilitate long term tracking (fewer keyframes, points tracked) and apply it in a control theory framework that yields similar results to Engel *et al*. [35]. Similar work, by Brockers *et al*. [12], expands the functionality to self-calibration and includes dense reconstruction of landing zones.

However, by far the most successful approaches perform completely modular sensor fusion and operate independently from any SLAM systems. For example, approaches such as that of Lynen *et al*. [82], who use an EKF to perform generic multi-sensor fusion able to handle lost and/or delayed sensor readings. More recently, these techniques have been generalised into robust systems such as the approach by Moore and Stouch [93] who fuse an arbitrary number of sensors and provide a state vector containing full pose plus velocity and acceleration.

The approaches presented so far have been *passive*, they do not possess the ability to reason about the environment in order to control the agent the sensor is mounted on. This is a clear indication that SLAM alone does not address the autonomy aspects of this thesis. Chapter 3 will propose a system that is capable of dense reconstruction, while Chapter 4 will add the ability to estimate its own goal poses (to improve the reconstruction) and planning a path to reach it.

2.2 Autonomous Navigation & Exploration

While SLAM provides robust localisation and reconstruction, it is not capable of *pro-active* scene exploration. Approaches that take a more active role in the motion of the sensor require higher-level decision-making capabilities. Unfortunately, autonomous agents cannot rely on reconstructions produced by SLAM, as they are either too slow (dense SLAM) or too sparse (feature-based SLAM) to be useful. However, the pose estimates that a VO and sensor fusion framework provide are sufficient to localise the robot. Once the robust position of an agent has

been reliably estimated, the next step is to enable autonomy. Autonomy can be defined as the ability to know *where to go* and *how to get there*. The process of deciding where to go next is called Goal Estimation. The problem of figuring out how to get there is called Path-planning.

2.2.1 Goal Estimation

Visual Odometry (VO), sensor fusion and goal estimation have seen a lot of overlap, especially in online scenarios. For example, Forster *et al*. [41] use their own VO [42] approach, but add a goal-estimation technique that maximises information gain on the reconstruction. VonStumberg *et al*. [136] take the approach of Engel *et al*. [33] and add an exploration approach based on flying a star pattern around a previously selected goal pose. While these approaches are an important part of autonomous agents, the idea of vision-based goal estimation will be explored further in Section 2.3.2.

Bridging the gap between goal estimation and path-planning, Paull *et al*. [103] use a cell-based approach to maximise coverage in underwater scenes. Mostegel *et al*. [94] plan feature-rich paths for a PTAM-based system which is also capable of selecting local (*i.e*. close) goal poses. Vision-based approaches such as these perform rudimentary path-planning heuristics to achieve their goal poses. However, more generic and robust algorithms have traditionally been presented in the robotics literature.

2.2.2 Path-planning

In robotics, path-planning is not limited to moving a robot from a start to goal pose. Path-planning, motion estimation and/or the piano-movers problem is more generally defined as the steps required to reach a given configuration. In this context, the configuration of a robot is defined as an N-dimensional vector that may contain position, rotation, joint angles, velocities, accelerations, etc. Therefore, pathplanners in the robotics community generally deal with a configuration-space that includes kinematic and holonomic constraints, and is much higher-dimensional than a pose-space.

The high-dimensionality of the problem has made sampling-based approaches the state-of-the-art for path-planning. Generally speaking, sampling-based algorithms work by drawing

samples from the configuration space in order to create a graph-like structure. The graph-like structure can then be used to estimate a path from the start to end configuration. How the graph is constructed, and how it is used, draws an important distinction between state-of-the-art approaches: multi-query and single query. Sampling-based approaches have become ubiquitous in robotics, spawning open-source [125] implementations of the most popular approaches.

Multi-Query

Multi-query approaches create a robust graph once, and use it for multiple queries. Therefore, they are suited to situations where the configuration space does not change significantly. Probabilistic Road Map (PRM) [68] split the path-planning problem into a learning and a query phase. In the learning phase, the approach draws samples from the configuration space and connects them using a local planner (normally interpolation with collision checking). In the query phase, the path from a start to goal can be estimated using standard graph-based techniques. While this approach did not provide any optimality guarantees, later work by Karaman *et al*. [67] provided asymptotic optimality. Dobston *et al*. [26] later introduced an approach that expanded on PRM [68, 67] by creating two graphs in parallel. The first is a PRM-like dense graph, the second is a sparse graph which adaptively samples from the dense graph. This approach provided asymptotic near-optimality, which means the solution is within a constant factor of the optimal solution, but crucially provides a much higher convergence rate than PRM [68, 67].

Single-Query

Single-query approaches build a new graph for every query. Therefore, they are well suited to rapidly changing scenarios. While there are many single query approaches [62, 124, 120], the most successful approach is Rapidly-exploring Random Tree (RRT) and its derivatives. The original RRT approach by LaValle *et al*. [77] build a tree for every query. The tree is grown in such a manner that it is biased towards large Voronoi regions in the configuration space. This makes RRT-based approaches particularly well suited to exploring high-dimensional spaces. Like PRM, RRT did not originally contain any optimality guarantees. Work by Karaman *et al*. introduced RRT* [67], which guarantees asymptotic optimality. Since then, many approaches

have improved on this formula. Gammell *et al*. [48] introduce a hyper-spherical bound on the region Rapidly-exploring Random Tree (RRT*) should sample from in order to improve the estimated path, which significantly improved convergence rates. Most similar to the work presented in Chapter 4, Sadat *et al*. [111] use RRT* to plan a PTAM-friendly path through areas high in visual features. Sadat *et al*. present an important shift in the way path-planning is performed. Traditionally, it has often been assumed that the cost of an individual state in the configuration space is intrinsically linked to the pose alone. Alternatively, it could be tangentially linked to the geometry by the amount of clearance it afforded the robot. Sadat *et al*. , and the work presented in Chapter 4 break this assumption by relating the cost of a state not only to the pose, but also to the geometry of the scene.

In addition to planning vision-friendly paths, the work presented in this thesis is capable of leveraging this path-planning technique to enable multiple agents to collaborate in the reconstruction.

2.2.3 Multi-Robot Control

Collaborative reconstruction, by two or more agents, has recently become popular in the literature. Approaches have mostly been aimed at integrating information obtained independently by different agents.

Rasche *et al*. [108] use potential fields to obtain collision free motion planning (using A*) and a gradient based approach for exploration. Their approach is only evaluated in software, where they assume the pose of the robot is known. More common are approaches that follow the SLAM paradigm of low-level tracking and high-level mapping. Zou *et al*. [143] fuse multiple passive cameras into one coherent SLAM reconstruction, while estimating inter-camera poses to deal with dynamic elements. In terms of active sensing, Forster *et al*. [40] take their VO [73] algorithm and apply it to a Collaborative Structure from Motion (CSfM) algorithm that separates each UAV into a separate thread. Their algorithm appears to use the VO purely as a sensor and does all the processing that an algorithm such as PTAM would do (keyframe selection, triangulation, local BA). More novel is that they use a BRIEF-based overlap detector to know when maps should be merged and confirm it via a RANSAC-based PnP algorithm between point clouds that returns a similarity transform. Loop closure, and the resulting optimisation,

is done via the method detailed in [122]. They also introduce a scale-difference estimation to resolve inconsistencies between the VO and the CSfM. Lazaro and Paz [78] similarly use [122] to create a pose-graph, when two robots meet the pose-graph is condensed to relevant components and shared. Finally, Cunningham *et al*. [20] use local maps, neighbouring robot information and robust data association to provide a decentralised multi-UAV approach.

The crucial limitation in all of these approaches is that they rely on independent agents fusing their maps when convenient. This thesis will present a different approach capable of "opportunistic collaboration" between multiple agents. This is a higher level form of collaborative behaviour where paths are agreed between the agents during the planning stage. The robots choose to either act as a variable baseline stereo pair, or to explore independently, depending on the scene.

2.3 3D Reconstruction

SLAM, sensor fusion and path-planning are capable of creating autonomous exploratory systems that can navigate simple open spaces. However, the sparse SLAM approaches that have traditionally been used in these frameworks are simply not sufficient for meaningful interaction with complex environments. On the other hand, the MVS techniques used in offline approaches are too computationally expensive to be used in real-time systems. However, as will be shown in this thesis, when combined with NBV/goal-estimation these offline techniques have the capability of creating dense reconstructions on-the-fly.

2.3.1 Multi-View Stereo (MVS)

Offline approaches, commonly referred to as MVS, typically find pairwise stereo correspondences and use large optimisations to estimate dense and accurate reconstructions, such as the work by Snavely *et al*. [119]. Denser reconstructions were achieved by Furukawa and Ponce [46] who use sparse feature matching and patch growing, along with photometric and visibility constraints [45] to produce dense reconstructions. Jancosek *et al*. [64] extend [46] by attempting to actively select views in an NBV-like approach to make large datasets feasible by estimating feasible stereo pairs, but provide no results on partial-image reconstruction. Hornung *et al*. [59]

use an octree-like hierarchical volumetric reconstruction along with graph cut minimisation. More recently, Galliani *et al*. [47] expand the patch-matching idea by [8] to use more than two views. Seminal work by Seitz *et al*. [114] established the Middlebury benchmark to compare MVS approaches by providing a calibrated dataset of camera poses and ground truth.

However, the computational cost for dense reconstruction of large structures can be prohibitive, preventing their use online, and lack the ability to choose views dynamically during data capture. Chapter 3 will present an offline reconstruction approach that is capable of reconstructing the scene without the use of expensive optimisations, limiting the computational cost. Furthermore, to make these approaches compatible with online reconstruction systems a view selection mechanism will be employed.

2.3.2 Next-Best View (NBV) Estimation

Intelligent view selection has the capability of removing redundant information which is actively beneficial to the reconstruction [121][45]. It can also be used to define goals for online robotic agents. Therefore, NBV selection can be thought of as a class of goal selection. NBV estimates a new pose in order to improve the existing reconstruction, and can be divided into two main categories: *exploration* and *refinement*.

Exploratory NBV estimation aims at generating the most complete model of the (unknown) scene. Early approaches, focused on small "table-top" scenes. Banta *et al*. [5] assumed the object was in the centre of a sphere and tried to detect occluded surface data using a voxel-based representation. Potthast and Sukhatme [106] focus on a more complex table-top scene, but similarly used a voxel representation to estimate unobserved regions. The NBV is then estimated as a maximisation of predicted information gain from each candidate pose, where the gain is related to the observability of currently reconstructed voxels. More recent approaches attempt to perform more large-scale pose estimation, generally based on the concept of a *frontier*. For example, the work by Heng *et al*. [55] uses a precomputed lattice and defines frontier locations as edges between cells where structure has been observed, and unobserved cells. Frontier pose configurations are then selected based on the information gain they provide and the cost to reach that configuration. In an approach that resembles the work presented here, Bircher *et al*. [7] use path-planning techniques to define a receding horizon. This is done by growing a RRT-based

tree where the gain of each node is the sum of visible unmapped volume (from that node). While this approach resembles the contributions of Chapter 4, Bircher *et al*. do not actually estimate an NBV (which causes their approach to have generally longer paths). More importantly, these systems rely on depth sensors to perform the reconstruction and thus make no attempt to reduce the noise in the scene.

In contrast, refinement NBV estimation aims at selecting poses that improve the 3D model accuracy. Dunn and Frahm [28] use an iterative technique that takes a partial input 3D model, uses patch-based eigenvalues to estimate the best view and performs an offline batch reconstruction method. Similarly, Hoppe *et al*. [57] take a previously constructed partial model and create a full network of poses for a UAV to follow and/or be reconstructed by an offline batch method. Mauro *et al*. [83] also optimise the views for an offline method, and define the NBV as the camera that maximises a view importance metric of aggregate quality features (Density, Uncertainty and Saliency). Finally, Hornung *et al*. [60] build a partial voxel-based proxy model and select the views based on maximising the number of visible low quality voxels. They then refine regions with poor photo-consistency by adding more views of these areas. More recently, Schönberger *et al*. [113] presented an approach for offline view selection that combined photometric and geometric priors. These approaches have similar limitations: Mauro *et al*. [83] and Hornung *et al*. [60] require expensive point cloud reprojection that scales with the size of the scene. Mauro *et al*. [83] and Schönberger *et al*. [113] require actual image information, preventing its use in live scenarios. Furthermore, all of these approaches are aimed at offline optimisation.

Contrary to these techniques, Chapter 3 proposes an approach capable of actively choosing the best locations to improve the reconstruction for both online and offline scenarios. The proposed approach can also balance the two competing objectives of exploration and refinement by probing the current estimate of geometry using raycasting and a voxel based representation. The use of image-based metrics is explicitly avoided in favour of geometric-based costs. Instead sparse, fast, calculations are used for NBV that scale well with map size and extend to collaborative scenarios.

2.4 Localisation

The field of SLAM is predicated on the simple idea that the pose of a sensor and the reconstructed landmarks are conditioned on each other [29, 117]. However, if one of them is known *a priori*, it is possible to marginalise the other [97]. In the same way that independent reconstruction algorithms (as discussed in 2.3) can provide more robust representations of the world, independent localisation algorithms can also provide more robust and consistent pose estimates.

Recent work by Sattler *et al*. [112] demonstrates that large-scale 3D models are not strictly necessary for accurate vision-based localisation. This work motivates Chapter 5, where the aim is to localise within a simple 2D floorplan *without* making assumptions about the 3D structure of the building.

2.4.1 Monte-Carlo Localisation

MCL can be considered the state-of-the-art for mobile robot localisation today. Introduced by Dallaert *et al*. [24], MCL is a form of PF where each particle is a pose estimate (and the map is known). It uses a motion model to propagate particles which in turn causes the weights to become the observation likelihood given the pose [130]. Re-sampling based on the weights then focuses computation in areas with more probable pose estimates.

Monte-Carlo Localisation (MCL) was made possible by the arrival of accurate range-based sensors such as Sound Navigation And Ranging (SoNAR) and LiDAR. These approaches, called Range-Based Monte-Carlo Localisation (RMCL), are robust and reliable and still considered state-of-the-art in many robotic applications. As such, they will be discussed first below.

Recent advances in computer vision have made similar vision-based approaches possible. These approaches, called Vision-Based Monte-Carlo Localisation (VMCL), typically use RGB cameras to avoid expensive sensors and will be discussed second.

Range-Based Monte-Carlo Localisation (RMCL)

RMCL was first introduced by Fox *et al*. [43] and Dellaert *et al*. [24]. RMCL improved the Kalman Filter based state-of-the-art by allowing multi-modal distributions to be represented.

It also solved the computational complexity of grid-based Markov approaches. More recent approaches, such as Kanai *et al*. [66], have moved the focus of RMCL into 3D. Kanai *et al*. focus on a pre-existing 3D reconstruction and simulate 3D depth readings at each particle. In what is probably the closest approach to the one presented in Chapter 5, Bedkowski *et al*. [6] use a 3D LiDAR scanner, extract normals and use them to segment floors, walls, doors and edges between labels. They then use an approach based on ICP, with added label constraints, to estimate the observation likelihood. While this seems like a very promising approach, Bedowski *et al*. use very simple heuristics to classify their points (surface normals, point height, etc.). Techniques based on deep learning provide better estimates of semantic labels, and are therefore used in Chapter 5 to provide a robust observation likelihood.

Vision-Based Monte-Carlo Localisation (VMCL)

RMCL-based approaches require expensive LiDAR and/or SoNAR sensors to operate reliably. Instead, Dellaert *et al*. [23] extended their approach to operate using vision-based sensor models.

VMCL allowed the use of rich visual features and low-cost sensors, but had limited performance compared to the more robust LiDAR-based systems. However, with the rising popularity of RGB-D sensors, more robust vision-based MCL approaches became possible. Fallon *et al*. [38] presented a robust MCL approach that used a low fidelity *a priori* map to localise in, but required the space to be traversed by a depth sensor beforehand. Brubaker *et al*. [13] removed the need to traverse a map with a sensor, and instead used visual odometry, pre-existing roadmaps and a joint MCL/closed-form approach in order to localise a moving car. More recently, approaches began to resemble traditional MCL by localising in an extruded floorplan. Winterhalter *et al*. [140] performed MCL using an RGB-D camera, basing the observation likelihood on the normals of an extruded floorplan. Chu *et al*. [18] removed the RGB-D requirement, by creating piecemeal reconstructions and basing the observation likelihood on direct ICP between these reconstructions and the extruded floorplan. Similar work by Neurbert *et al*. [98] also removed the RGB-D requirement, using synthesised depth images from the floorplan and comparing the gradient information against an RGB image, allowing purely monocular localisation. However, these approaches all rely on geometric information to provide an observation likelihood.

Advances in Deep Learning, such as the approaches of Badrinarayanan *et al*. [3], Kendal *et al*.

[71] and Long *et al*. [115], have recently enabled the use of semantic information for indoor localisation. More importantly, approaches like that of Holder *et al*. [56] have begun to take these approaches outdoors. Poschmann *et al*. [105], and the work presented in Chapter 5, attempt to use semantic information in an MCL context. Poschmann *et al*. follow a very similar approach to Neurbert *et al*. but synthesise semantic images (rather than depth ones) and base the observation likelihood on photometric consistency with a CNN-based segmentation method (on an RGB image). The work presented in Chapter 5 does not synthesise semantic images but rather uses the semantic segmentation of the real observation to augment traditional LiDAR-like sensors.

2.4.2 Closed-Form Localisation Approaches

While the field of MCL evolved in the robotics community, non-MCL-based approaches became more popular in the vision community. Shotton *et al*. [116] used regression forests to predict the correspondences of every pixel in the image to a known 3D scene, they then combined this in a RANdom SAmple and Consensus (RANSAC) approach in order to solve the camera pose. Melbouci *et al*. [85] used extruded floorplans, but performed local bundle adjustments instead of MCL. Caselitz *et al*. [14] use a local SLAM system to create reconstructions that are then aligned using ICP to a LiDAR-built 3D map. However, instead of MCL they optimise the correspondences with a non-linear least squares approach.

More recent approaches have begun to also look at semantic information. Wang *et al*. [137] use text detection from shop fronts as semantic cues to localise in the floorplan of a shopping centre. Liu *et al*. [80] who use floorplans as a source of geometric and semantic information, combined with vanishing points, to localise monocular cameras. These vision-based approaches tend to use more of the non-geometric information present in the floorplan. However, a common trend is that assumptions must be made about geometry not present in the floorplan (*e.g*. ceiling height). The floorplan is then extruded out into the 3$^{\text{rd}}$ dimension to allow approaches to use the information present in the image.

Chapter 5 differs from the approach of Poschmann *et al*. [105], Wang *et al*. [137] and Liu *et al*. [80] in two important ways. Firstly, it does not require an extruded floorplan, opting instead to project the sensory information down to 2D and localise there. This makes the approach of

Chapter 5 be able to run in real time. Secondly, it has the capability of augmenting traditional LiDAR sensors making it a more generic solution.

2.5 Summary

This chapter has presented an overview of the current state-of-the-art approaches in the fields of SLAM, Navigation and Exploration, 3D Reconstruction and Localisation. The remainder of this thesis will attempt to combine these fields into an autonomous, collaborative robotic system. It is therefore important to present a summary of the capabilities, and limitations, of each field.

SLAM is the problem of estimating the position of the robot in the world, while simultaneously reconstructing the geometry. In principle, SLAM should obviate the need for explicit 3D reconstruction and localisation. In practice, the limitations in the field of SLAM preclude it from being a complete solution. Sparse SLAM, while good for localisation, is not dense enough for reconstruction or navigation. Dense SLAM provides excellent reconstructions, but is too computationally demanding for live robots. Semi-dense SLAM presents an interesting compromise, however, these approaches are not yet capable of dealing with the rapid motion required in robotic applications. A further limitation of SLAM is that it is inherently passive, as it does not attempt to estimate a goal pose or navigate to it. Enabling autonomy has become an important goal for robotic systems, such that SLAM is no longer considered a complete solution. This thesis will address this limitation by utilising SLAM as an underlying localisation, with higher-level navigation, exploration and reconstruction.

State-of-the-art navigation and exploration are performed by path-planning and goal estimation, respectively. Path-planning has traditionally been confined within the field of Robotics, where the cost of a path is inherent in the pose of the robot. This is an important limitation, and more recent vision-based approaches have begun to explore the idea of cost-spaces that extend beyond the pose-space. Similarly, goal estimation has begun to adopt ideas from vision-based techniques in order to enable autonomy. This thesis will continue along this trend and explore vision-based path-planning and goal estimation. This will be done with the explicit purpose of improving reconstruction and enabling collaboration.

In order for a robot to navigate and explore its environment, it must also be able to understand it.

3D reconstruction, the process of estimating scene geometry, has usually been strictly offline and/or linked to pose estimation in SLAM. However, more recent approaches have begun to combine ideas from goal estimation, such as NBV, in order to limit the computational cost of a reconstruction. NBV presents an important advancement in the way 3D reconstruction is understood. It enables intelligent view selection, shifting the focus from *how* to reconstruct to *what* to reconstruct. This thesis will aim to use NBV, along with state-of-the-art reconstruction techniques, to autonomously reconstruct a scene by selecting a small subset of views.

Localisation is the task of finding the pose of a robot within its environment. In the absence of SLAM, this requires a map to be created for the robot *a priori*. This presents an important limitation: the robot must traverse the space before localisation can be performed. More importantly, the robot must be equipped with an extremely accurate sensor to enable mapping. Given an accurate map, MCL is widely considered to be the state-of-the-art. Recent advances in vision-based approaches have allowed MCL to be performed on low-cost monocular cameras or RGB-D sensors. However, standard scan-matching MCL approaches are fundamentally limited to the accuracy of the sensor used to obtain the scan and the accuracy of the map. In order to overcome this limitation, more recent work has allowed semantic segmentation to play an important role in improving the performance of MCL. This has enabled the use of less accurate maps, such as floorplans designed for human use. This thesis aims to fundamentally change the way MCL is performed, in favour of a human-inspired localisation approach that leverages semantic information.

Each of the fields mentioned here have important limitations that prevent them from driving a fully autonomous, collaborative robotic system. This thesis will present a series of contributions that will overcome these limitations. This will be done by combining autonomous navigation and exploration techniques with discrete 3D reconstruction and localisation approaches. Novel contributions to the fields of 3D reconstruction and goal estimation will be presented in Chapter 3. Chapter 4 will present a novel path-planning, goal estimation and collaboration framework and combine it with a state-of-the-art sparse SLAM localisation system. Together, these contributions will drive an autonomous and collaborative robotic system. Finally, Chapter 5 will introduce a human-inspired localisation framework that addresses the limitations of scan-matching MCL.

Chapter 3

Next-Best View Estimation

In this chapter, the MVS problem is addressed using an innovative view selection criterion. Instead of only selecting the NBV, the proposed approach is capable of actively selecting stereo pairs directly. This stereo pair selection is done in two steps. Firstly, a novel approach is presented which efficiently optimises the NBV in terms of accuracy and coverage using partial scene geometry. Secondly, an intelligent stereo pair selection process jointly optimises the baseline and vergence to find the best stereo pair for the NBV, the Next-Best Stereo (NBS). In both cases, the aim is to maximise the final reconstruction quality, while reducing the amount of data used.

Experimental evaluation shows that the proposed methods allow efficient selection of stereo pairs for reconstruction. As such, a dense model can be obtained with only a small number of images. Once a complete model is obtained, the remaining computational budget is used to intelligently refine areas of uncertainty. This approach achieves results comparable to state-of-the-art batch approaches on the Middlebury dataset, using as little as 3.8% of the views.

This chapter will first present a brief introduction to, and formalisation of, the NBV problem. This is followed by a description of the 3D reconstruction process that provides both point estimation and uncertainty. The remainder of this chapter will present the novel NBV/NBS framework that was published in [86].

3.1 Problem Definition

At their core, most 3D reconstruction approaches iteratively add views to a pre-existing model. In the case of SLAM, the iterative process takes the form of successive frames in a video. For MVS, the process of selecting putative stereo pairs and progressively adding them to a large BA can also be thought as an iteration over the entirety of the data. Implicit in this interpretation, is making a decision about what image to process next.

SLAM systems normally use a keyframe selection heuristic, as a proxy for "view selection". On the other hand MVS approaches tend to use the full set of images in their reconstruction. While there exist approaches to perform view selection for MVS, such as that of Furukawa *et al*. [45], these normally rely on pre-built sparse Structure from Motion (SfM) models and/or use information in the image (forcing the algorithm to "select" a view, process it and decide whether it is the NBV). In either case, it is understood in the literature that removing redundant information is beneficial to the result of the reconstruction [121][45]. Removing redundant information prevents baselines that are too small from adding noise to the reconstruction.

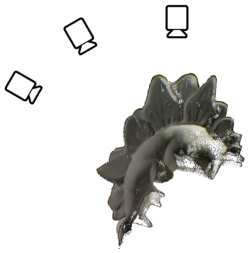
The problem of NBV simply takes this logic a step further. Instead of relying on heuristics to perform view selection, NBV actively tries to find the most informative view. Figure 3.1 shows a basic example of the problem. In Figure 3.1a, there is a partially reconstructed world, along with the source cameras. Figure 3.1b shows a set of putative candidates, and finally Figure 3.1c shows the selected NBV.

In principle, the NBV can be defined as the pose the maximises the information gain. However, as will be shown in Chapter 4, a bounded cost is more easily manipulated than an indeterminately large information gain. Given the set of all poses $\dot{\mathbb{X}}$ and all currently reconstructed points $\dot{\mathbb{M}}$, the NBV is defined as the camera pose $\dot{\mathbf{x}}_{NBV}$ that satisfies

$$\dot{\mathbf{x}}_{NBV} = \underset{\dot{\mathbf{x}} \in \dot{\mathbb{X}}}{\arg\min} \ \boldsymbol{\eta}\left(\dot{\mathbf{x}}, \dot{\mathbb{M}}\right) \tag{3.1}$$

where $\dot{\mathbf{x}}_{NBV} \in \dot{\mathbb{X}}$ is defined as the most informative view in $\dot{\mathbb{X}}$, and $\boldsymbol{\eta}\left(\dot{\mathbf{x}}, \dot{\mathbb{M}}\right)$ is a cost function that is inversely proportional to the information gain. This is a generic formalisation that leaves the implementation of the cost function, and the set of all poses, undefined.

This chapter will focus on an MVS-based implementation of NBV. In this case, the pose-space



(a) Partial reconstruction with used views.

(b) NBV candidates (grey).

(c) Selected NBV (red).

Figure 3.1: Sample NBV problem.

is fully defined by the set $\dot{\mathbb{X}}$ which contains all poses in the dataset or all poses for an active sensor. This naturally raises the question of how to identify the NBV in an intractable pose-space that cannot be searched by brute-force. Chapter 4 will address this question, where $\dot{\mathbb{X}} = \text{SE}(3)$.

The cost-function used to estimate the NBV will be fully defined in this chapter. This cost is dependant on both the *uncertainty* and the *coverage* of the partially reconstructed geometry, $\dot{\mathbb{M}}$. This means that the cost-function is an optimisation between minimising uncertainty and increasing coverage. In order to define these two terms precisely, it is first necessary to discuss the reconstruction strategy.

3.2 Reconstruction of Dense 3D Structure

This section describes a method which creates a dense and accurate scene reconstruction from a new NBV observation. Since the end-goal of this thesis is to implement a live robotic system, expensive optimisations are avoided. Instead, this section focuses on the online aspect of

the reconstruction. This means that only the previous view and NBV are used in updating the reconstruction. In principle, the work described here could easily be applied to a BA, however, experimental evidence shows that state-of-the-art performance can be achieved without expensive optimisations.

The method described in this section makes two fundamental assumptions. Firstly, it assumes that the 6-DoF pose of the camera is known. Secondly, it assumes the intrinsic parameters of the camera are known. These are valid assumption in the case of MVS scenario, since knowledge of the views (pose and calibration) is normally assumed. This is also valid in a live robotic scenario, as the robot needs to know its pose for robust navigation and the intrinsic parameters of the camera can be estimated *a priori*.

Unfortunately, these assumptions also present a limitation when it comes to robustness. The reconstruction method presented in this Section, and used for the remainder of the thesis, explicitly avoids expensive optimisations. This implies that the approach is not capable of coping with large amounts of noise in the intrinsic parameters, as they are not optimised. Furthermore, this also implies the system is liable to reconstruct faulty geometry if there are noisy pose estimates. In practice, this approach is robust against an occasional incorrect and/or noisy pose estimate due to the octree-based representation [61] used to store the geometry.

Under the assumption that the pose and calibration of the cameras are correct and known, the high-density map of the scene is created by iterative reconstruction from two images in a wide baseline stereo arrangement. This is achieved in three steps: dense matching, triangulation and data association. Firstly, dense correspondences between the two images are estimated using a deep learning-based approach [138]. Secondly, a Linear Least-Squares (L-LS) triangulation method is used to estimate 3D points and covariance matrices from the dense matches. Finally, the new 3D scene information is associated with the existing reconstruction using the covariance matrices and an octree representation.

3.2.1 Estimating Dense Correspondences

In order to estimate dense correspondences on a wide-baseline stereo pair, an optical flow algorithm is used [138]. Wide-baseline stereo is required to ensure that the NBV is relatively independent of the current views. This algorithm consists of a deep learning-inspired sparse

correspondence algorithm which relies on a hierarchical, multi-layer, correlational architecture inspired by deep convolutional neural networks [138]. This approach does not actually perform any learning, instead directly computes correlations between the patches of one image and the whole second image by using the patches as convolutional filters. The resulting correlation maps are passed through an aggregation consisting of max-pooling, subsampling, shifted average and non-linear rectification layers. Once the top level of the correlation pyramid is reached, a top-down "backtracking" step is performed that undoes the aggregation and results in atomic patch correspondences. Using this technique, this approach is capable of matching 4×4 pixel patches within a scaling factor of $[0.5, 1.5]$ and a rotation of $[-30°, 30°]$. These putative matches are then placed in a coarse-to-fine energy minimisation framework that estimates dense optical flow and penalises divergence from the putative matches.

The robustness of both the reconstruction and the NBV estimation fundamentally depends on the accuracy of this optical flow. This is enforced by performing bi-directional optical flow.

Assuming an iterative NBV scenario, $\mathbf{I}_{NBV}$ is the image of the NBV estimation at time t. Similarly, $\mathbf{I}_{PRV}$ is the NBV image at time $t-1$. Therefore the bi-directional optical flow estimates the flow field from $\mathbf{I}_{PRV}$ to $\mathbf{I}_{NBV}$ (and vice-versa) and enforces a constraint on divergence between the flow fields.

In more detail, the dense optical flow field

$$\mathbf{F}_{PRV} = \boldsymbol{flow}\left(\mathbf{I}_{PRV}, \mathbf{I}_{NBV}\right) \tag{3.2}$$

defines the motion of every pixel, $\mathbf{m}$, on $\mathbf{I}_{PRV}$ to a corresponding pixel $\mathbf{m}'$ on $\mathbf{I}_{NBV}$ such that

$$\mathbf{I}_{PRV}\left(\mathbf{m}\right) = \mathbf{I}_{NBV}\left(\mathbf{m}'\right) . \tag{3.3}$$

The relationship between the flow field and images is thus defined as

$$\mathbf{m}' = \mathbf{m} + \mathbf{F}_{PRV}\left(\mathbf{m}\right). \tag{3.4}$$

The inverse of this process can also be estimated, where

$$\mathbf{F}_{NBV} = \boldsymbol{flow}\left(\mathbf{I}_{NBV}, \mathbf{I}_{PRV}\right) \tag{3.5}$$

defines the motion of every pixel from $\mathbf{I}_{NBV}$ to $\mathbf{I}_{PRV}$. Finally, the bi-directional constraint on the optical flow becomes

$$\left|\mathbf{F}_{PRV}\left(\mathbf{m}\right) + \mathbf{F}_{NBV}\left(\mathbf{m}'\right)\right| < \tau_f \tag{3.6}$$

where τ_f is a simple threshold in pixels.

This bi-directional constraint has two important implications. Firstly, it allows the optical flow process to discard inconsistent correspondences using a robust error metric. This makes the optical flow estimation more reliable and accurate. Secondly, it makes both flow estimations independent of each other. This allows each optical flow to run concurrently, effectively halving the processing time.

Having defined a robust and dense correspondence estimator, it is now necessary to use these correspondences to triangulate 3D positions.

3.2.2 Triangulation from Correspondences

While there exist many ways to perform triangulation from point correspondences, one of the most widely used approaches is Iterative Linear-Least-Squares (ILLS). In this approach, a dense 3D point cloud $\dot{\mathbb{M}}$ containing points $\dot{\mathbf{m}} \in \dot{\mathbb{M}}$ is reconstructed from classical 3D reconstruction equations (*e.g.* [54]).

Given a set of known 2D correspondences, such as those estimated in Section 3.2.1, the unknown 3D point ($\dot{\mathbf{m}}$) can be described in terms of the projection as

$$\mathbf{m} = \mathbf{P}_{PRV}\,\dot{\mathbf{m}} \tag{3.7}$$

where $\mathbf{P}_{PRV}$ is the projection matrix that produces $\mathbf{I}_{PRV}$ and

$$\mathbf{m}' = \mathbf{P}_{NBV}\,\dot{\mathbf{m}} \tag{3.8}$$

where $\mathbf{P}_{NBV}$ is similarly defined. The matrices $\mathbf{P}_{PRV}$ and $\mathbf{P}_{NBV}$ can be estimated directly from the known pose and intrinsic parameters. It is then possible to use these two equations to define the linear system of equations

$$\mathbf{A}\,\dot{\mathbf{m}} = \mathbf{b} \tag{3.9}$$

where $\mathbf{A}$ is composed of the combined pixel locations and projection matrices. Assuming that $\boldsymbol{rank}\,(\mathbf{A}) = 3$, the location of the 3D point can be estimated as

$$\dot{\mathbf{m}} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b} \tag{3.10}$$

which holds as long as the point is not on the plane at infinity [53]. The resulting reconstructed point can be used to weight equation 3.9, effectively allowing the reprojection error to be estimated. Iteratively weighting the equation and estimating the reconstructed point allows ILLS to estimate a point that minimises the reprojection error for both images $\mathbf{I}_{PRV}$ and $\mathbf{I}_{NBV}$.

The uncertainty of each reconstructed point $\dot{\mathbf{m}}$ is represented as a 3×3 covariance matrix ($\dot{\mathbf{\Lambda}}_{\dot{\mathbf{m}}}$). In order to estimate it, is necessary to define the covariance matrix of the pixels as $\mathbf{\Lambda}_{\mathbf{m}}$ and $\mathbf{\Lambda}_{\mathbf{m}'}$. The covariance of the 3D triangulated point then becomes

$$\dot{\mathbf{\Lambda}}_{\dot{\mathbf{m}}} = \mathbf{B}\,\bar{\mathbf{\Lambda}}\,\mathbf{B}^{\top} \tag{3.11}$$

where $\mathbf{B}$ is the Jacobian of equation 3.10, and

$$\bar{\mathbf{\Lambda}} = \boldsymbol{diag}\left(\mathbf{\Lambda}_{\mathbf{m}}, \mathbf{\Lambda}_{\mathbf{m}'}\right) \tag{3.12}$$

is the diagonal matrix of pixel covariances. This process can be repeated for every point in $\dot{\mathbb{M}}$, allowing covariance matrices to be estimated for every point in the point cloud. These covariance matrices, along with the reconstructed 3D points, will be used to estimate future NBVs. However, before detailing the NBV it is necessary to describe how these points are added to a coherent map that can associate the data from iterative NBV.

3.2.3 NBV Integration

Once the NBV observation has been triangulated, it is necessary to integrate it into the current reconstruction. Fundamentally, this involves associating data from multiple stereo-pairs, which is not a simple task. Indeed, the problem grows exponentially with the number of views and naïve and/or brute force methods are guaranteed to fail. More importantly, while the triangulated 3D points provide a detailed representation of the scene, such a large point cloud is very inefficient for the purpose of reasoning about scene geometry. This is because the point cloud is a discrete set embedded in a continuous space, making it too sparse for meaningful geometric calculations (such as ray casting). Instead, this section proposes a robust data association method that encodes each triangulated point cloud into the leaf nodes of an octree. The structure of the octree, along with the covariance information, are exploited in order to efficiently associate the data and perform geometric calculations.

Octree Encoding of Scene Structure

This work uses a modified version of OctoMap [61]. OctoMap is an octree structure that is widely used in the robotics literature. One of its main advantages is that it keeps track of voxel occupancy. Each voxel is classified as either *occupied* ($\dot{v}^o \in \dot{\mathbb{V}}^o$), *empty* ($\dot{v}^e \in \dot{\mathbb{V}}^e$) or *unobserved* ($\dot{v}^u \in \dot{\mathbb{V}}^u$). This allows the tree to be defined as the set of all voxels

$$\dot{\mathbb{V}} = \dot{\mathbb{V}}^o \cup \dot{\mathbb{V}}^e \cup \dot{\mathbb{V}}^u \tag{3.13}$$

where *occupied* voxels represent areas with reliable reconstructed geometry, *empty* voxels are unoccupied space with no (or unreliable) geometry and *unobserved* voxels are areas of the scene with no observations indicating membership of ($\dot{\mathbb{V}}^o$) or ($\dot{\mathbb{V}}^e$). As such

$$\dot{\mathbb{V}}^o \cap \dot{\mathbb{V}}^u = \dot{\mathbb{V}}^e \cap \dot{\mathbb{V}}^u = \dot{\mathbb{V}}^o \cap \dot{\mathbb{V}}^e = \emptyset\,. \tag{3.14}$$

This voxel structure will be exploited to perform multiple tasks such as raycasting for NBV, sampling for path-planning and goal estimation in the next chapter. More explicitly, $\dot{\mathbb{V}}^e$ is navigable space that rays can propagate through, $\dot{\mathbb{V}}^o$ is directly related to the refinement objective and $\dot{\mathbb{V}}^u$ is related to the exploration objective.

However, the implementation of Hornung *et al*. [61] has certain drawbacks. The main limiting factor is that the octree uses a point cloud to define its structure and then discards it. Instead, the octree proposed here stores the points and covariances at the leaf nodes. Each voxel $\dot{v} \in \dot{\mathbb{V}}$ has a set of points (with covariances) stored in it. The set of points is defined as $\dot{\mathbb{M}}_{\dot{v}}$ and the individual points as $\dot{\mathbf{m}}_{\dot{v}} \in \dot{\mathbb{M}}_{\dot{v}}$. Similarly, the set of corresponding covariances $\dot{\mathbf{\Lambda}}_{\dot{v}} \in \dot{\mathbb{\Lambda}}_{\dot{v}}$ is stored in the voxel. In the following Section, this octree structure will be used to allow efficient data association across different stereo-pairs.

Data Association

It is necessary to decide whether each point in the NBV point cloud is actually new or if it is a new observation of an existing point. This process is known as Data Association, and it is non-trivial even when robust point-to-point matching is possible. There are approaches that can perform this kind of matching based on 2D features/descriptors (such as [72]), as well as "tracking" on the image (such as [33]). However, the former becomes both intractable and

inaccurate in the case of a dense reconstruction. The latter is only possible with online tracking, and thus unsuitable for wide-baseline stereo or MVS.

Instead, this work proposes to use the 3D reconstructed point and the covariance to perform the data association. Broadly speaking, the data association is described by finding the nearest neighbour in Mahalanobis space. Formally, the putative match $\dot{\mathbf{m}}^\star$ is defined as

$$\dot{\mathbf{m}}^\star = \underset{\dot{\mathbf{m}} \in \dot{\mathbb{M}}}{\arg\min} \; \boldsymbol{\delta}_h \left(\dot{\mathbf{m}}, \dot{\mathbf{m}}' \right) \tag{3.15}$$

where $\delta_h$is the Mahalanobis distance. Unfortunately, a naïve exhaustive match of all points from both point clouds using any distance metric would be intractable.

Instead, the proposed data association method uses the octree to constrain the problem. Every new pair of frames produces a point cloud $\dot{\mathbb{M}}'$ that must be fused into the octree. $\dot{\mathbb{M}}'$ contains points $\dot{\mathbf{m}}' \in \dot{\mathbb{M}}'$ and covariance matrices $\dot{\mathbf{\Lambda}}' \in \dot{\mathbb{\Lambda}}'$. Each point in this new point cloud must be either associated with an existing point or marked as a new observation. In order to do this, the first step is to propagate the point cloud through the octree. This can be done in parallel extremely efficiently.

After propagation, each point $\dot{\mathbf{m}}'$ obtains a putative set $\dot{\mathbb{M}}_{\dot{v}}$ of matches corresponding to the voxel that point lands in. This results in a much more constrained and tractable problem. For each point, the data association can be performed as follows. If the leaf node is empty $\left(\dot{\mathbb{M}}_{\dot{v}} = \emptyset \right)$, the point and its covariance matrix are added to $\dot{\mathbb{M}}$. If the voxel is occupied, calculating the putative match becomes

$$\dot{\mathbf{m}}^\star_{\dot{v}} = \underset{\dot{\mathbf{m}}_{\dot{v}} \in \dot{\mathbb{M}}_{\dot{v}}}{\arg\min} \; \boldsymbol{\delta}_h \left(\dot{\mathbf{m}}_{\dot{v}}, \dot{\mathbf{m}}' \right) \tag{3.16}$$

where the minimisation happens over the points in the voxel, not the whole reconstructed geometry. Finally, if the closest Mahalanobis distance falls within a 95% confidence ($\chi^2 < 7.81$), $\dot{\mathbf{m}}'$ is used to update the reconstruction.

Integrating an NBV observation into the reconstruction is performed as a Kalman update on the nearest neighbour point $\dot{\mathbf{m}}^\star_{\dot{v}}$ and its covariance matrix $\dot{\mathbf{\Lambda}}^\star_{\dot{v}}$. During this update, it is assumed that the observation model is identity (as both the state and the observation represent 3D points in the same coordinate frame). In order to compute the update, it is first necessary to estimate a

Kalman gain

$$\mathbf{K}_g = \dot{\mathbf{\Lambda}}^{\star}_{\dot{v}} \left(\dot{\mathbf{\Lambda}}^{\star}_{\dot{v}} + \dot{\mathbf{\Lambda}}' \right)^{-1}. \tag{3.17}$$

This gain can then be used to update the point

$$\dot{\mathbf{m}}_{\dot{v}} = \dot{\mathbf{m}}^{\star}_{\dot{v}} + \mathbf{K}_g \left(\dot{\mathbf{m}}' - \dot{\mathbf{m}}^{\star}_{\dot{v}} \right) \tag{3.18}$$

finally, the covariance is updated as

$$\dot{\mathbf{\Lambda}}_{\dot{v}} = \left(\mathbf{I} - \mathbf{K}_g \right) \dot{\mathbf{\Lambda}}^{\star}_{\dot{v}} \tag{3.19}$$

where $\mathbf{I}$ is a 3×3 identity matrix. In the event where the number of points in the voxel is too large, the top N euclidean nearest-neighbours of $\dot{\mathbf{m}}^{\star}_{\dot{v}}$ are found and used as $\dot{\mathbb{M}}_{\dot{v}}$ in equation 3.15.

Now that a robust structure for geometric operations and data association has been defined, it is possible to discuss the NBV strategy in detail.

3.3 Next-Best View Optimisation

This section proposes a novel criterion for NBV optimisation based on a compromise between the competing objectives of coverage and accuracy. The coverage objective will drive the system to collect views of previously unobserved parts of the scene (e.g., due to restrictions on the field of view or occlusion) defined by $\dot{\mathbb{V}}^{u}$. The accuracy objective will drive the system to choose the next pose to reduce the uncertainty of the point cloud $\dot{\mathbb{\Lambda}}$ which normally implies observing $\dot{\mathbb{V}}^{o}$ from a different vantage point.

These two criteria are optimised jointly, making use of the octree structure, the dense point cloud and the covariances of point cloud. The octree allows for quick and efficient calculations on scene geometry. The dense cloud and covariances allow for more detailed calculations about scene noise and viewing angle. This effectively creates a coarse-to-fine strategy for finding the NBV.

3.3.1 Monocular View Cost

Given a set of sensor poses ($\dot{\mathbb{X}}$), the cost of each pose ($\dot{\mathbf{x}}$) can be estimated by casting a set $\mathbb{R}_{\dot{\mathbf{x}}}$ of random rays from the camera centre through the image plane. Each ray will traverse the

octree until it intersects either an occupied ($\dot{v}^o$) or unobserved ($\dot{v}^u$) voxel, ignoring empty ($\dot{v}^e$) voxels. When a ray $\mathbf{r}_{\dot{\mathbf{x}}} \in \mathbb{R}_{\dot{\mathbf{x}}}$ intersects with an occupied voxel $\dot{v}^o \in \dot{\mathbb{V}}^o$, the cost for each point in the voxel $\dot{\mathbf{m}}_{\dot{v}} \in \dot{\mathbb{M}}_{\dot{v}}$ is estimated as

$$\boldsymbol{\rho}\Big(\mathbf{r}_{\dot{\mathbf{x}}}, \dot{\mathbf{m}}_{\dot{v}}\Big) = e^{-\Big\| \lambda_{\dot{\mathbf{m}}} \nu_{\dot{\mathbf{m}}} \times \mathbf{r}_{\dot{\mathbf{x}}} \Big\|}, \tag{3.20}$$

where $\lambda_{\dot{\mathbf{m}}}$ and $\nu_{\dot{\mathbf{m}}}$ are the largest eigenvalue and eigenvector, respectively, of the covariance $\dot{\boldsymbol{\Lambda}}_{\dot{\mathbf{m}}}$ of $\dot{\mathbf{m}}_{\dot{v}}$. The magnitude of the cross product will change depending on the angle between the vectors $\nu_{\dot{\mathbf{m}}}$ and $\mathbf{r}_{\dot{\mathbf{x}}}$. When they are parallel the magnitude of the cross product is 0, which makes the whole equation evaluate to 1 (the highest cost). When they are perpendicular the cross product evaluates to 1 (since both vectors are unit), which makes the equation exponentially decay based on the eigenvalue $\lambda_{\dot{\mathbf{m}}}$. Basically, this equation favours observing points with large uncertainties from views perpendicular to the largest eigenvector of the covariance. This is because observing a covariance from such an angle will have the best impact on decreasing the uncertainty. The lower this cost, the better the view.

Consequently, the cost of the intersected voxel is defined as the average point cost

$$\boldsymbol{v}\Big(\mathbf{r}_{\dot{\mathbf{x}}}, \dot{\mathbb{M}}_{\dot{v}}\Big) = \frac{1}{\Big|\dot{\mathbb{M}}_{\dot{v}}\Big|} \sum_{\dot{\mathbf{m}}_{\dot{v}} \in \dot{\mathbb{M}}_{\dot{v}}} \boldsymbol{\rho}\Big(\mathbf{r}_{\dot{\mathbf{x}}}, \dot{\mathbf{m}}_{\dot{v}}\Big). \tag{3.21}$$

Finally, the NBV cost of a particular pose $\dot{\mathbf{x}}$ is defined as

$$\boldsymbol{\eta}\Big(\dot{\mathbf{x}}, \dot{\mathbb{M}}\Big) = \frac{1}{\Big|\mathbb{R}_{\dot{\mathbf{x}}}\Big|} \sum_{\mathbf{r}_{\dot{\mathbf{x}}} \in \mathbb{R}_{\dot{\mathbf{x}}}} \begin{cases} \boldsymbol{v}\Big(\mathbf{r}_{\dot{\mathbf{x}}}, \dot{\mathbb{M}}_{\dot{v}}\Big) & \text{if } \dot{v}_{\mathbf{r}} \in \dot{\mathbb{V}}^o \\ \gamma \in [0, 1] & \text{if } \dot{v}_{\mathbf{r}} \in \dot{\mathbb{V}}^u. \end{cases} \tag{3.22}$$

where $\dot{v}_{\mathbf{r}}$ is the voxel intersected by the ray $\mathbf{r}_{\dot{\mathbf{x}}}$. In this equation, γ is a parameter that can encourage or discourage exploration. A γ of 1 will always give the highest cost to unobserved voxels, preferring to reduce the uncertainty of observed voxels. A γ of 0 will give unobserved voxels the lowest cost, giving them preference.

The NBV is calculated as the cost minimisation

$$\dot{\mathbf{x}}_{NBV} = \underset{\dot{\mathbf{x}} \in \dot{\mathbb{X}}}{\arg\min} \left(\frac{1}{\Big|\mathbb{R}_{\dot{\mathbf{x}}}\Big|} \sum_{\mathbf{r}_{\dot{\mathbf{x}}} \in \mathbb{R}_{\dot{\mathbf{x}}}} \begin{cases} \boldsymbol{v}\Big(\mathbf{r}_{\dot{\mathbf{x}}}, \dot{\mathbb{M}}_{\dot{v}}\Big) & \text{if } \dot{v}_{\mathbf{r}} \in \dot{\mathbb{V}}^o \\ \gamma \in [0, 1] & \text{if } \dot{v}_{\mathbf{r}} \in \dot{\mathbb{V}}^u. \end{cases} \right) \tag{3.23}$$

where $\dot{\mathbf{x}}_{NBV}$ is the pose that will provide the most benefit to the existing map.

This section has shown how the NBV can be estimated from a set of candidate poses and existing geometry. This method of estimating the NBV has the advantage of allowing a trade-off between exploration and refinement. It is also a quick and efficient method that does not need the images to make intelligent decisions about what area to observe next. However, there is an important aspect missing from this equation: the views used to generate the prior reconstruction. There is no guarantee that the selected NBV has any overlap with these views. As such, there may be nothing for this view to triangulate against. While this type of approach will work well for RGB-D sensors, a monocular camera requires a stereo pair.

3.4 Next-Best Stereo Optimisation

A naïve approach to solving the triangulation issue would be to make equation 3.23 dependant on the views that have been used to generate the reconstruction. While this might seem like an obvious solution, it has some important drawbacks that make it undesirable. Firstly, excellent views of the environment may be discarded as they do not have any overlap with existing views. This is especially likely for a system that has a high bias towards exploration ($\gamma = 0$). More importantly, such an approach would make NBV intrinsically incompatible with collaborative sensors. This is because it limits each sensor to only look for the NBV in its own vicinity, limiting collaboration to integrating their observations into the same reconstruction. In this thesis, a more meaningful definition of collaboration is demonstrated in Chapter 4 where sensors act as active stereo pairs.

This section proposes an alternative to traditional NBV that does not require knowledge of the views previously used in the reconstruction. This novel approach, called NBS, is capable of optimising the stereo configuration of sensors directly from a partial reconstruction and putative poses. Actively selecting the NBS has a few advantages over naïvely including previous views in the NBV selection. In the MVS scenario, NBS allows independent stereo pairs to be selected in the NBV region. In a collaborative sensor scenario, NBS can dynamically plan views for sensor pairs (as will be shown in Chapter 4). More importantly, the geometry of the sensors can be optimised such that it is ideally suited to the partially reconstructed geometry.

(a) Selected NBV (red).

(b) NBS candidates (blue).

(c) Selected NBV and NBS

Figure 3.2: Sample NBS problem.

3.4.1 Definition

The NBS is a minimisation on a cost function over a set of putative stereo pairs. Assuming that the previously used views are ignored, and therefore the NBV is not constrained to them, stereo-pair constraints could be added to equation 3.23. Fundamentally, this would evaluate every combination of poses in $\dot{\mathbb{X}}$ for the best possible stereo pair. However, this has disadvantages that cannot be ignored. Firstly, using all possible pose pairs means the problem scales quadratically with the size of $\dot{\mathbb{X}}$ ($O(N^2)$). This would make the problem quickly become intractable for large datasets (or continuous pose spaces). Secondly, adding stereo-pair constraints to equation 3.23 causes it to compromise the NBV in favour of sensor-specific constraints (baseline, vergence, etc.).

Instead, this work proposes to select an NBS relative to the previously selected NBV. This approach has several key advantages. Firstly, it massively reduces the pose-space to a more tractable set of candidates ($O(N)$). Secondly, the separation of NBV and NBS prevents them from competing against each other. This separation will become key in applying this approach

to a live system, in Chapter 4.

Figure 3.2 shows how this selection process is performed. In Figure 3.2a, the NBV shares no overlap with the views used for the reconstruction. Instead of discarding this view as "wrong", in Figure 3.2b NBS considers all possible stereo pairs to augment the NBV. Finally, in Figure 3.2c the best possible pair for the NBV is selected. This allows the reconstruction to be updated, and used to select a new NBV/NBS pair.

In order for the NBS to perform robust 3D reconstruction, it should be optimised for depth estimation and dense matching. Depth estimation requires a wide baseline, since as $\delta_B \rightarrow 0$ the conditioning of $\mathbf{A}$ in equation 3.9 decreases. Basically, the error is inversely proportional to the baseline. Dense matching requires a small vergence angle, since the matching difficulty increases as the views become more different. While these are intrinsically competing objectives, a good stereo pair should be able to strike a balance between them. Therefore, this section proposes to define a measure of an "ideal" stereo pair that can be used to score all putative stereo pairs. The further away a pair of views is from the ideal stereo pair, the higher the cost.

3.4.2 Stereo Pair Cost

The quality of a stereo configuration always depends on the same parameters. Namely, the stereo camera baseline and vergence angle and the distance to the nearest geometry. This section describes how these parameters are used to penalise deviation from the ideal stereo pair. Note that similar to the NBV, the NBS depends on the sensor pose, *not* the image content.

Figure 3.3 shows a sample stereo-pair, which will be used to describe the geometry involved in this section. In this figure $\dot{\mathbf{x}}_{NBV}$ and $\dot{\mathbf{x}}_{NBS}$ are the 6-Degrees of Freedom (DoF) poses of each camera. The rays $\mathbf{r}_V$ and $\mathbf{r}_S$ are vectors from each camera centre, respectively, through the principal point (dashed black lines). These rays represent the viewing direction of each camera. The intersection ($\dot{\mathbf{m}}_I$) of these two rays is the point of vergence. Unfortunately, in 6-DoF space, there is no guarantee that the rays will intersect. Instead, the intersection can be calculated using a triangulation similar to the one in Section 3.2. Finally, $\dot{v}_G$ is the centre of the occupied voxel $\dot{v}_G \in \dot{\mathbb{V}}^O$ that is closest to the intersection point $\dot{\mathbf{m}}_I$. In Figure 3.3, the ideal stereo pair would have all the rays perfectly intersect at $\dot{v}_G$.

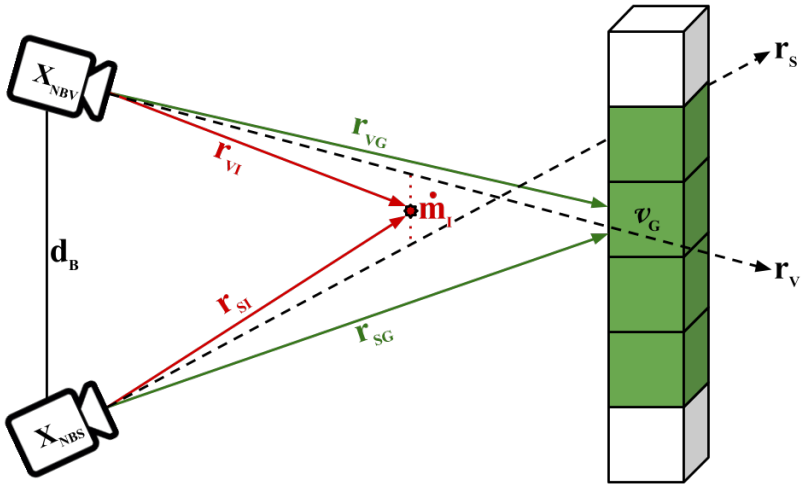


Figure 3.3: Sample stereo pair geometry.

Baseline and Vergence Angle Optimisation

Perhaps the most important aspect of a stereo pair is its baseline. It must be short enough to allow for robust correspondence estimation, while being large enough to provide good depth estimates. However, it makes little sense to enforce a particular baseline. This is because the baseline must be scaled relative to the mean scene depth in order to avoid issues with matching and triangulation.

Instead, the baseline can be parametrised as a fraction of the distance to $\dot{\mathbf{m}}_I$. Therefore, the ideal stereo pair would have the ratios

$$\delta_{VI} = \delta_{SI} = \alpha\,\delta_B \tag{3.24}$$

where δ_{VI} and δ_{SI} are the distances from the cameras to the intersection $\dot{\mathbf{m}}_I$, δ_B is the baseline and α is the desired ratio between the baseline and the intersection point. For a candidate, this can be enforced as a soft constraint using the cost function

$$C_B = \frac{|\delta_{VI} - \alpha\,\delta_B|}{\alpha\,\delta_B} + \frac{|\delta_{SI} - \alpha\,\delta_B|}{\alpha\,\delta_B} + \frac{|\delta_{VI} - \delta_{SI}|}{\delta_B} \tag{3.25}$$

Figure 3.3 shows a sample camera configuration, where this soft constraint is formed by the red lines and the baseline.

This enforces an “ideal” triangular structure defined by alpha (α), where the ratio defines the

expected angle of vergence, β, which can be shown to be

$$\beta = \arccos\left(1 - \frac{1}{2\,\alpha^2}\right) \tag{3.26}$$

where β is the vergence angle. However, this ideal structure is only enforced in a 3-DoF space.

View Triangulation Optimisation

In a 6-DoF space, the principal rays $\mathbf{r}_V$ and $\mathbf{r}_S$ rarely have an exact point of intersection. Instead the triangulation finds the point that is closest to both rays. As shown in Section 3.2.2, there will always be a distance between the actual principal point and the reprojection of $\dot{\mathbf{m}}_I$ (the reprojection error). If this error is too large there might be no overlap in the images, which would make triangulation impossible. While it is possible to measure this error, it is a relatively expensive operation.

Instead, to handle the case where the principal rays do not intersect, the cost

$$C_T = \arccos\left(\frac{|\mathbf{r}_V \cdot \mathbf{r}_{VI}|}{\|\mathbf{r}_V\|\,\|\mathbf{r}_{VI}\|}\right) + \arccos\left(\frac{|\mathbf{r}_S \cdot \mathbf{r}_{SI}|}{\|\mathbf{r}_S\|\,\|\mathbf{r}_{SI}\|}\right) \tag{3.27}$$

is estimated, where $(\mathbf{r}_{VI}, \mathbf{r}_{SI})$ are rays from the camera centres to the intersection point. These are the red rays in Figure 3.3. This effectively penalises large angles between the principle rays and the rays to the intersection point. An angle of zero $(C_T = 0)$, would also mean the reprojection error is zero.

These costs enforce a good stereo arrangement for anything near the intersection point $\dot{\mathbf{m}}_I$. However, having a good configuration is useless if the geometry being imaged is not taken into account.

Optimising Vergence on Scene Structure

The stereo pair defined so far has not yet been coupled with the existing geometry. This should be done in order to avoid situations where the sensors have a vergence point that is far behind, or in front of, the geometry. To enforce this, large angles between the rays to $\dot{\mathbf{m}}_I$ and $\dot{v}_G$ are penalised

$$C_G = \arccos\left(\frac{|\mathbf{r}_{VI} \cdot \mathbf{r}_{VG}|}{\|\mathbf{r}_{VI}\|\,\|\mathbf{r}_{VG}\|}\right) + \arccos\left(\frac{|\mathbf{r}_{SI} \cdot \mathbf{r}_{SG}|}{\|\mathbf{r}_{SI}\|\,\|\mathbf{r}_{SG}\|}\right) \tag{3.28}$$

where $\mathbf{r}_{VG}$ and $\mathbf{r}_{SG}$ can be defined as the rays from $\dot{\mathbf{x}}_{NBV}$ and $\dot{\mathbf{x}}_{NBS}$ to $\dot{v}_G$. These are the green rays in Figure 3.3, where an angle of zero between these rays is desirable.

The costs presented so far would be enough to perfectly align a stereo pair, except for one parameter. If one of the sensors undergoes rotation around the principal ray ($\mathbf{r}_V$ or $\mathbf{r}_S$), the costs presented so far would remain unchanged. This is a problem for approaches that are not invariant to image rotation.

Rotational Optimisation

Since dense, per-pixel, matches are being estimated, it cannot be assumed that the process is rotationally invariant. In order to increase the performance of any matching algorithm, large differences in the orientation of the image are penalised. If the gravity vector is defined as $\mathbf{g}_v = [0, 0, 1]$, the roll is then penalised as the difference in the angle between the gravity vector transformed into the coordinate frame of each camera

$$C_R = \arccos\left((\mathbf{R}_V\,\mathbf{g}_v) \cdot (\mathbf{R}_S\,\mathbf{g}_v)\right) \tag{3.29}$$

where $\mathbf{R}_V$ and $\mathbf{R}_S$ are the rotation matrices of each sensor.

The final cost function can then be defined as

$$\boldsymbol{\sigma}\left(\dot{\mathbf{x}}_{NBV}, \dot{\mathbb{X}}_s, \dot{\mathbb{M}}\right) = C_B + w_T C_T + w_R C_R + w_G C_G \tag{3.30}$$

where $\dot{\mathbb{X}}_s$ is the set of possible stereo pairs for $\dot{\mathbf{x}}_{NBV}$. In practice, it was not necessary to weight each cost differently, so the different weights were defined as $w_T = w_R = w_G = 1$.

This cost can be efficiently computed for thousands of candidate pairs. The optimum configuration can then be selected as

$$\dot{\mathbf{x}}_{NBS} = \arg\min_{\dot{\mathbf{x}}} \; \boldsymbol{\sigma}\left(\dot{\mathbf{x}}_{NBV}, \dot{\mathbb{X}}_s, \dot{\mathbb{M}}\right) \tag{3.31}$$

where $\dot{\mathbf{x}}_{NBS}$ is the final selected stereo pair for the NBV.

This section has presented a fully online method for estimating not only the NBV, but also a stereo pair that allows it to be completely independent from overlap with the previous views. The NBS approach presented here has several properties that make it ideal for stereo-based

reconstruction. Firstly, the baseline of the cameras is scaled depending on the distance to the observed geometry. This enforces a sensible baseline, while implicitly preventing aggressive vergence angles. Secondly, the reprojection error of the intersection point is minimised. This prevents scenarios where the centre rays do not intersect. Thirdly, the distance between the vergence point and the nearest geometry is minimised. This ensures that the sensors are focused on actual scene geometry. Finally, rotation in the image plane is minimised. This ensures that non-rotationally invariant matching methods do not fail.

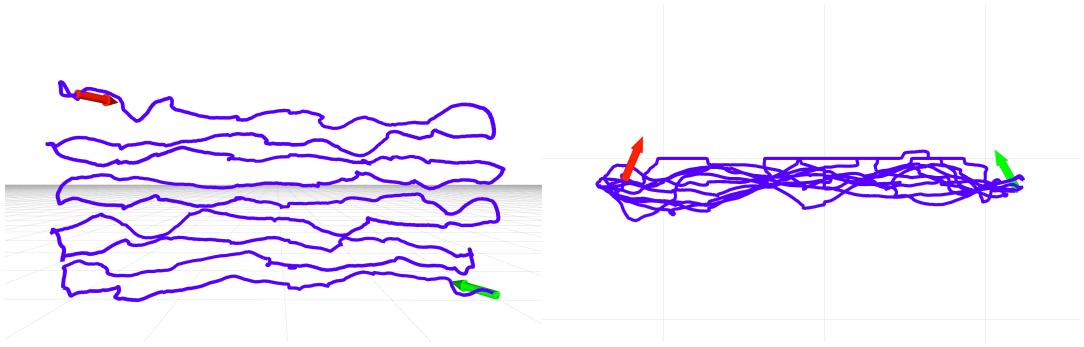
In the following Section, NBV and NBS will be validated in an MVS scenario. It will be shown that iterative NBV/NBS can outperform BA-based methods without the use of an optimisation framework. More importantly, it will be shown that NBV/NBS can outperform both state-of-the-art batch and view selection methods with only a fraction of the views and computational cost.

3.5 Evaluation

The evaluation is done by applying iterative NBV/NBS to two reconstruction datasets, an in-house Unmanned Aerial Vehicle (UAV) dataset and the well-known Middlebury MVS dataset [114]. The UAV dataset is evaluated first as a proof-of-concept. The Middlebury dataset is then analysed to compare against state-of-the-art.

In both cases, the approach is the same. It is first bootstrapped using a manually selected pair of frames and poses. After the initial pair is processed (optical flow, triangulation and octree) the algorithm will pick an NBV from the set of poses using the method detailed in Section 3.3. Once this pose is selected, an NBS pose that satisfies equation 3.31 is selected. This is performed iteratively until the desired number of frame pairs is processed. Upon completion of the reconstruction, the point cloud is extracted from the octree.

In both cases, the analysis is the same. Firstly, the effects of the two main parameters (α, γ) are demonstrated. These parameters encourage different behaviours and can therefore be tailored to different applications. Secondly, a quantitative comparison against other NBV approaches is performed. The comparison shows that iterative NBV/NBS can outperform the state-of-the-art using fewer frames. Lastly, a qualitative analysis is presented. This analysis



(a) Side View. (b) Top View.

Figure 3.4: Different views of the pattern flown by the UAV. The green arrow marks the start of the sequence, while red is the end.

Figure 3.5: Sample dataset images.

visualises the reconstructed point clouds against a reference model computed by a state-of-the-art reconstruction method [46] using all possible views.

3.5.1 Unmanned Aerial Vehicle (UAV) Dataset

In order to provide a proof-of-concept demonstration, a dataset consisting of frames from a UAV was obtained. In this dataset, the UAV was flown in an upwards square wave pattern while looking at a scene (as shown in Figure 3.4). This was done by flying the UAV with an on-board SLAM-based stabilisation system. The dataset consists of 160 images (over 25000 possible image pairs). Figure 3.5 shows sample images.

In order to provide a good estimate of the performance of the system, Kinect Fusion[99] ground truth of the same scene was captured. Three different metrics are used to analyse the approach:

Average Nearest-Neighbour Error (e_{nn}), *Outlier Ratio (o_{nn})* and *Coverage Ratio (c_{nn})*. These error metrics can all be defined in terms of the nearest-neighbour distance

$$\boldsymbol{\delta}_{nn}\Big(\dot{\mathbf{m}}, \dot{\mathbb{M}}\Big) = \min_{\dot{\mathbf{m}}' \in \dot{\mathbb{M}}} \boldsymbol{\delta}\left(\dot{\mathbf{m}}, \dot{\mathbf{m}}'\right) \tag{3.32}$$

where $\dot{\mathbf{m}}$ is an arbitrary point and $\dot{\mathbb{M}}$ is the point cloud where the nearest-neighbour is being looked for.

The *Average Nearest-Neighbour Error (e_{nn})* is computed by efficiently estimating the average nearest-neighbour distance for the set of nearest-neighbour inliers

$$\dot{\mathbb{M}}_I = \left\{ \dot{\mathbf{m}} \in \dot{\mathbb{M}}_R \middle| \boldsymbol{\delta}_{nn}\Big(\dot{\mathbf{m}}, \dot{\mathbb{M}}_{GT}\Big) < \tau_{nn} \right\} \tag{3.33}$$

where $\dot{\mathbb{M}}_R$ is the reconstructed cloud, $\dot{\mathbb{M}}_{GT}$ is the ground truth cloud and τ_{nn} is a threshold distance which is established to avoid skewing the data because of outliers. The error is then calculated as

$$e_{nn} = \frac{1}{\left|\dot{\mathbb{M}}_I\right|} \sum_{\dot{\mathbf{m}} \in \dot{\mathbb{M}}_I} \boldsymbol{\delta}_{nn}\Big(\dot{\mathbf{m}}, \dot{\mathbb{M}}_{GT}\Big), \tag{3.34}$$

which corresponds to the average inlier error.

The *Outlier Ratio (o_{nn})* is estimated as

$$o_{nn} = 1 - \frac{\left|\dot{\mathbb{M}}_I\right|}{\left|\dot{\mathbb{M}}_R\right|} \tag{3.35}$$

which corresponds to the ratio of outliers to total number of reconstructed points.

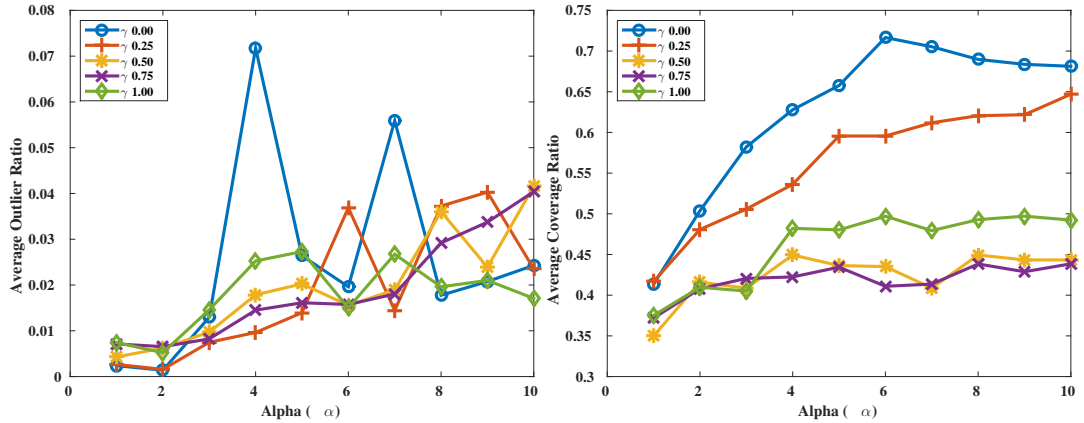
Finally, the *Coverage Ratio (c_{nn})* is estimated by efficiently finding the "inverse" nearest-neighbour inliers,

$$\dot{\mathbb{M}}'_I = \left\{ \dot{\mathbf{m}} \in \dot{\mathbb{M}}_{GT} \middle| \boldsymbol{\delta}_{nn}\Big(\dot{\mathbf{m}}, \dot{\mathbb{M}}_R\Big) < \tau_{nn} \right\} \tag{3.36}$$

and calculating the ratio

$$c_{nn} = \frac{\left|\dot{\mathbb{M}}'_I\right|}{\left|\dot{\mathbb{M}}_{GT}\right|} \tag{3.37}$$

which corresponds to the ratio of inliers to all points in the ground truth.



(a) Low α reduces the number of outliers by decreasing depth error.

(b) High α helps with coverage by making matching easier.

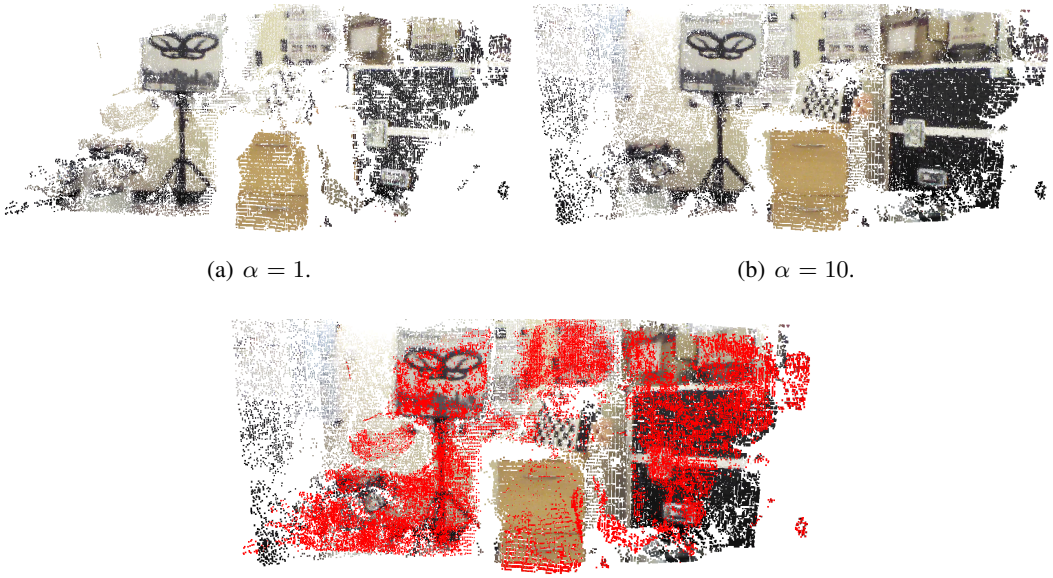
Figure 3.6: Effects of α on coverage and outliers.

Parameter Exploration: α

As a first step, the effects of increasing α are shown. Alpha controls the ratio between the distance to the intersection point and the length of the baseline. Therefore, alpha has a two-fold effect: it controls the baseline and the vergence angle. Lower α makes depth estimation more accurate (wider baseline), but it also makes dense matching harder (large vergence angle). Higher α leads to less sparsity, as the dense matching performs better with low vergence angles, but decreases the accuracy of the depth estimation.

Figure 3.6a shows that a low α is required to keep the number of outliers from increasing. Values of alpha from $1-2$ ensure that the depth error does not accumulate in the reconstruction and skew the results. This motivates choosing a low alpha. However, as mentioned in Section 3.4 there is a trade-off between depth error and dense matching.

Figure 3.6b shows that increasing alpha improves coverage. This happens because as α increases, the vergence angle becomes shallower - making dense matching easier and therefore increasing the density of the point cloud. Qualitatively, Figure 3.7 confirms this: it can be seen that a reconstruction of $\alpha = 1$ is significantly sparser than $\alpha = 10$. This is further shown in 3.7c, where the red dots represent the parts of $\alpha = 1$ not present in $\alpha = 10$.



(a) $\alpha = 1$.

(b) $\alpha = 10$.

(c) Difference: red points are those in $\alpha = 10$ but not $\alpha = 1$.

Figure 3.7: Reconstructions under different values of α.

Parameter Exploration: γ

The effects of γ are now explored. Figure 3.8 shows that as γ decreases, the growth of coverage is dramatically accelerated. As mentioned in Section 3.3, this parameter is able to either encourage or discourage exploration. This is because, in equation 3.1, the cost assigned to a ray that hits a voxel $\dot{v} \in \dot{v}^o$ is controlled by γ. This parameters allows the user to decide what should take priority.

Quantitative Analysis

Having explored the parameter space, it is now possible to show the effect of increasing the number of pairs. The parameters are set to $\alpha = 3$ and $\gamma = 0.25$, The performance of the system is measured by plotting all three metrics as the number of used pairs increases. Figure 3.9 shows that as the number of pairs increases, the Average Nearest-Neighbour Error (e_{nn}) and Outlier Ratio (o_{nn}) decrease. Note that the slight increase at $pairs = 6$ is because the UAV has added new areas into the map. Even though the error is increasing given the new observations, the number of outliers decreases as previously observed areas are refined. Finally,

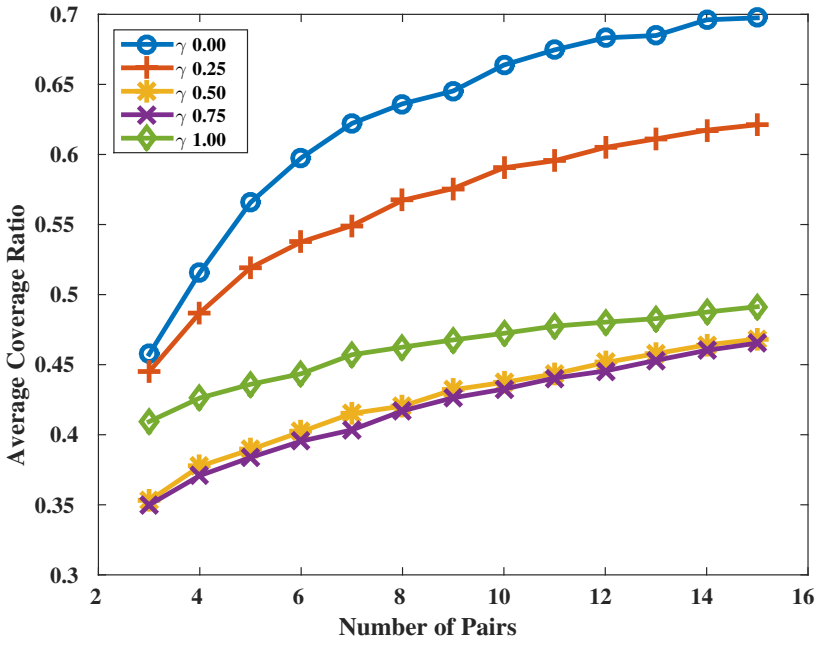


Figure 3.8: Increasing the value of γ encourages the UAV to explore, resulting in a higher coverage.

note that coverage increases quickly at first and settles. This is because the ground truth covers a significantly wider area.

Qualitative Analysis

Figure 3.10 shows sample reconstructions for a full-blown MVS system using all 160 frames, compared to the presented approach using only 14 images. It is clearly shown that the presented approach provides a much more detailed reconstruction, especially in parts that have low texture. More importantly, the approach only uses 8.75% of the views, and actively selects 0.056% of the possible pairs. Figure 3.11 shows pairs of frames that were selected by the proposed system. Note how the images form robust stereo-pairs; they have a similar vantage point and a reasonable baseline.

3.5.2 Multi-View Stereo Evaluation

The Middlebury dataset consists of 2 figurines, Dino and Temple, imaged in a dome-like pattern. For each figurine there are 3 modalities of the images: full ($\sim$300 images), ring ($\sim$50 images)

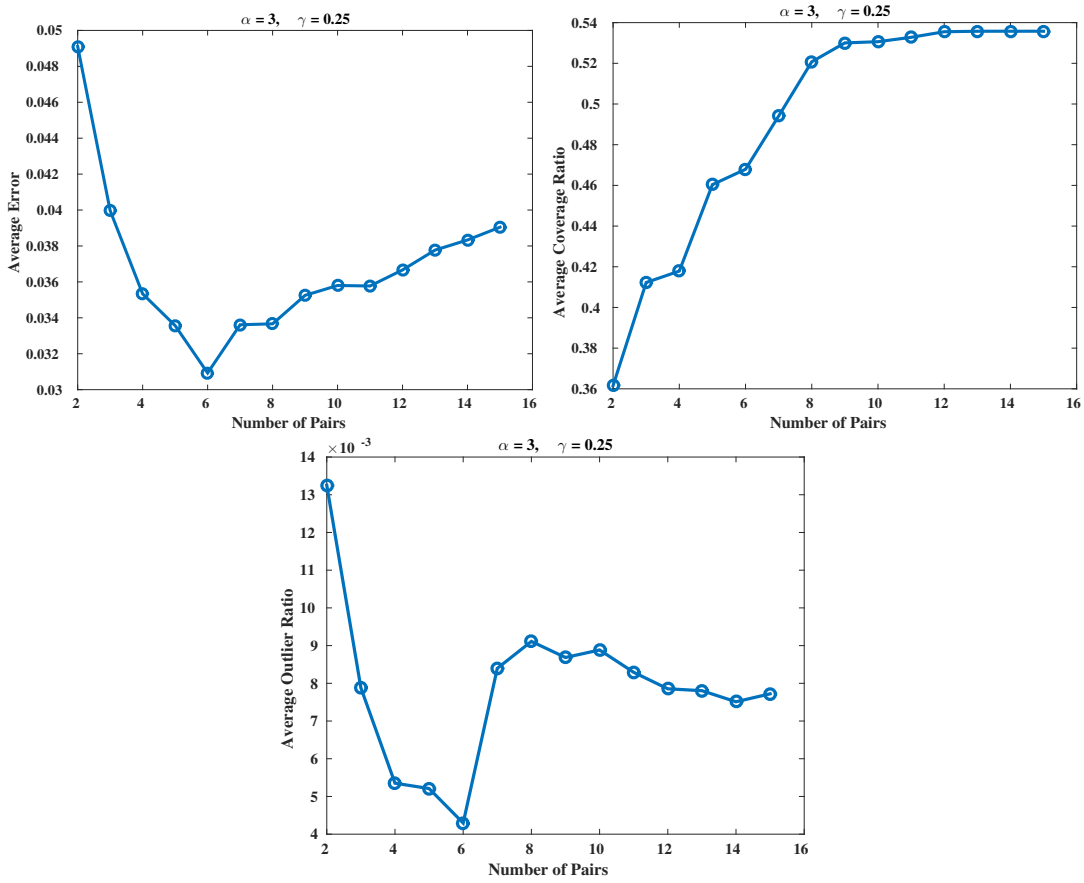


Figure 3.9: Different metrics for a sample reconstruction.

(a) VisualSFM+CMVS using 160 frames [17][45][141].

(b) Proposed Method using 14 frames ($\alpha = 6$, $\gamma = 0.5$).

(c) Ground Truth from Kinect Fusion [99].

Figure 3.10: Qualitative comparison of Reconstructions.

Figure 3.11: Samples of autonomously selected image pairs.

and sparse ring (∼15 images). In terms of NBV selection, the full datasets are more challenging because they present more possible stereo pairs ($O(300^2)$).

In order to perform evaluation, it is necessary to measure the performance of reconstruction as the number of stereo pairs increases. The error metrics used in this evaluation are the same as those used in the Middlebury benchmark by Seitz *et al*. [114]. Seitz measures the nearest-neighbour distance in the same way as Section 3.5.1. The difference is they then estimate the threshold distance τ_{nn} such that a certain percentage of the points are within τ_{nn}. The coverage is similarly estimated by measuring the "inverse" nearest-neighbour. Several thresholds (τ_{nn}) are selected, and the percentage of points in the reference cloud that contain a neighbour within that distance is reported. The difference between these metrics and those in the previous section is that Seitz *et al*. do not directly report the outliers. Instead, they are reported through the percentile.

In this evaluation, these metrics are used to explore the parameter space and measure the effects of α and γ. The Middlebury dataset has no publicly available ground truth, therefore, a reference model was created to aid in parameter exploration. This reference model was created

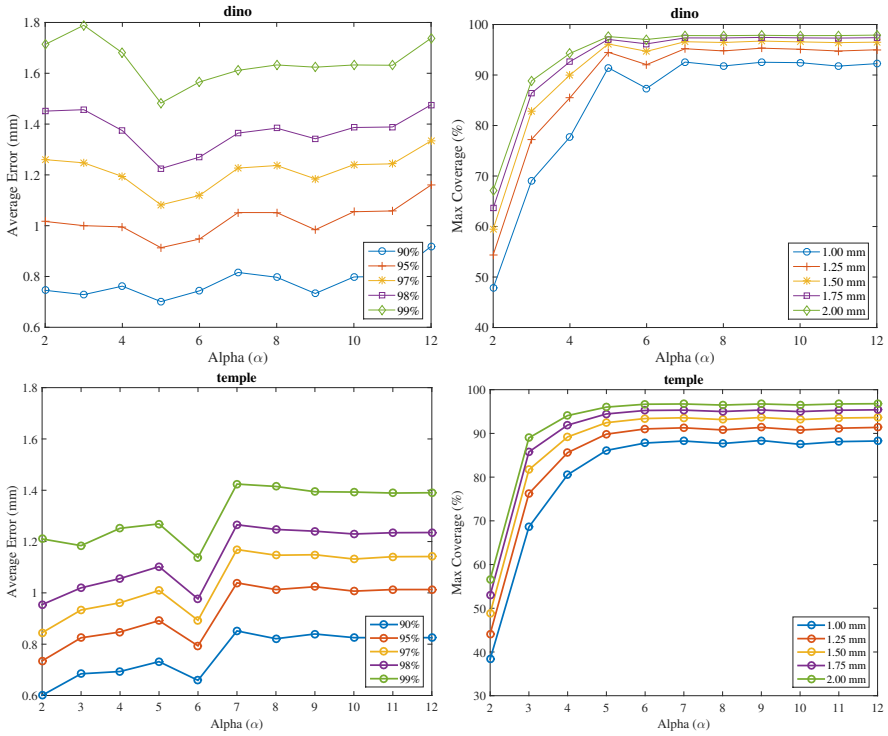


Figure 3.12: Avg. Error (Left) and Max Coverage (Right) with increasing values of α.

from all the images in each dataset using the state-of-the-art MVS reconstruction algorithm from Furukawa and Ponce [46].

Parameter Exploration: α

The evaluation is performed as follows. The γ parameter is disabled (*i.e.* there is only one case in equation 3.22), to allow the effect of α to be observed without other confounding variables. This corresponds to estimating the cost of equation 3.21 using only the rays that hit occupied voxels. The approach then selects 40 pairs of frames. The average error and the maximum achieved coverage are then estimated. This is done for the entire sequence, with 5 different error thresholds.

Figure 3.12 shows error and coverage curves for different values of α. It can be seen that very low values of α have very low coverage since the wide vergence angles make dense matching difficult. On the other hand, very large values start to suffer from increasing depth error due to the relatively narrow baseline. Choosing values of $\alpha \in [5, 7]$, corresponding to a vergence angle

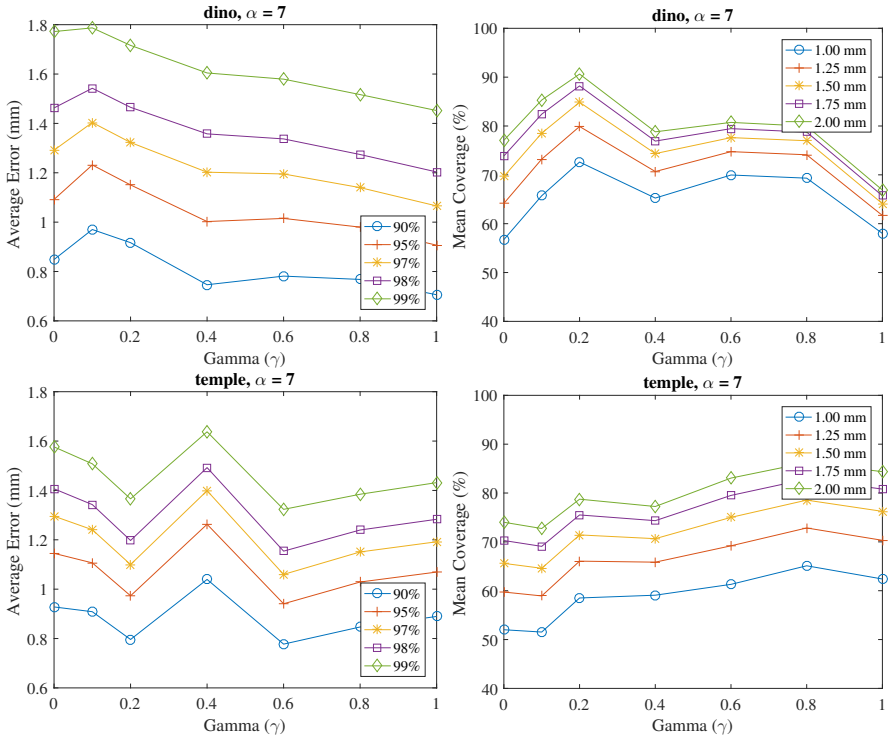


Figure 3.13: Avg. Error (Left) and Mean Coverage (Right) with different values of γ.

of around 9°, achieves high coverage while minimising the average error. It is important to note that these parameters are *not* dependent on the absolute values of depth, baseline or vergence. Rather, they scale with the scene to provide good stereo configurations. More importantly, α can be tailored to other matching approaches. High accuracy, sparse feature matching can have low values of alpha that allow good depth estimation. Denser, per-pixel methods can use high alphas to encourage easier matching.

Parameter Exploration: γ

In order to explore the effects of γ, a value of $\alpha = 7$ is selected. This is done because γ only applies to the NBV so a narrow baseline is chosen to reduce the effects of mismatches during the optical flow stage. Furthermore, the proposed method uses a dense matching approach, which makes the small increase in error justified by the larger coverage and easier matching.

Gamma (γ) provides the ability to either encourage or discourage exploration. As shown in equation 3.22, γ controls how favourable it is for the camera to look at unobserved voxels

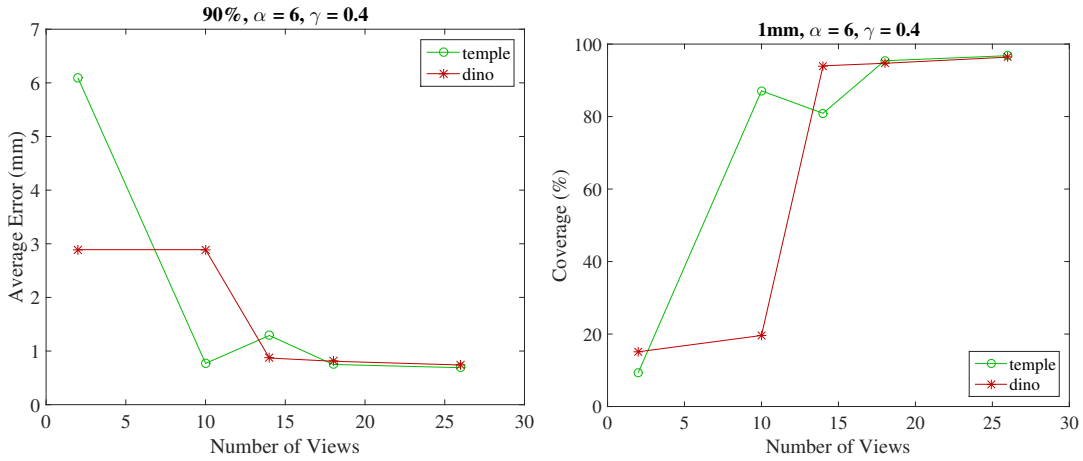


Figure 3.14: Middlebury Benchmark as number of views go up with $\alpha = 6$ and $\gamma = 0.4$

($\dot{v} \in \dot{v}^u$). Setting $\gamma = 0$ assigns the lowest score to unobserved voxels, while $\gamma = 1$ assigns the highest. In Figure 3.13, the coverage curves show the *mean* for the first 5 frames only, since otherwise all values of γ converge to high coverage.

It can be seen that as the value of γ goes up, the average error starts to decrease. This happens when the value of γ prefers refinement over exploration. In Dino, the coverage also decreases as the approach prioritises different views of the same geometry. Note that, despite the general downward trend, the values of $\gamma < 0.1$ are unstable because the NBV concentrates on areas of the scene where there is no geometry, therefore making the stereo pair selection ill-posed. The same is true with the coverage, where extremely low values of γ encourage looking at the narrowest profiles of the object (since they will include the most unobserved voxels). However, it is important to note that all values achieve high levels of coverage. In practice, a value of $\gamma = 0.4$, is used since this allows slight bias towards exploration.

Temple presents a fail-case, where the coverage increases with the value of gamma. As the NBV tries to find a view that is perpendicular to the covariances in the observed voxels, it obtains a view with novel information. This is probably due to the nature of the dataset, as all poses are in a dome around the object. Nevertheless, it should be noted that this parameter performs as expected in all other evaluations in this thesis.

	Thresholds	Uniform [60]	NBV [60]	NBS [64]	NBS Proposed
Num. Frames	-	41	41	unknown	**26**
Error (mm)	80%	0.64	0.59	0.64	**0.53**
	90%	1.00	0.88	0.91	**0.74**
	99%	2.86	2.08	1.89	**1.68**
Coverage (%)	0.75mm	79.5	82.9	72.9	**87.3**
	1.25mm	90.2	93.0	73.8	**96.4**
	1.75 mm	94.3	96.9	73.9	**98.4**

Table 3.1: Middlebury Evaluation for different NBV and MVS approaches.

Quantitative Analysis

Now that good values for α and γ have been established, the comparison against other NBS methods can be performed. In order to evaluate against the online ground truth for the Middlebury benchmark, the point clouds produced by the presented approach are turned into a mesh using Poisson Surface Reconstruction [69]. Two different approaches were considered, Hornung *et al*. [60] and Jancosek *et al*. [64]. These approaches are from 2008 and 2009, respectively, which makes them significantly dated. The reason these approaches were chosen is two-fold. Firstly, as was shown in Section 2.3.2, most approaches either use depth sensors [5, 106, 55, 7], require *a priori* models [28, 57] or extensively use image-based information [83, 113] (making them unsuitable to live scenarios). Secondly, comparison against the full Middlebury benchmark would be unfair due to the fact that most approaches are exhaustive MVS optimisations, rather than online NBV or NBS selection approaches. It is, however, important to note that these were the best methods that performed a comparable NBV selection. Finally, it should be noted that the results are reproduced directly from the publications (*i.e*. the approaches were not re-implemented or experiments re-run). Table 3.1 shows, from left to right, a comparison against the voxel-based MVS approach of Hornung *et al*. [60] with 41 uniformly selected views, the top performing NBV approach and the image-based "NBS" approach of Jancosek *et al*. [64]. It can be seen that the presented NBS approach consistently outperforms Hornung *et al*. [60] using both uniform and selected views, while simultaneously using fewer frames. Furthermore,

the approach outperforms the NBS approach of [64].

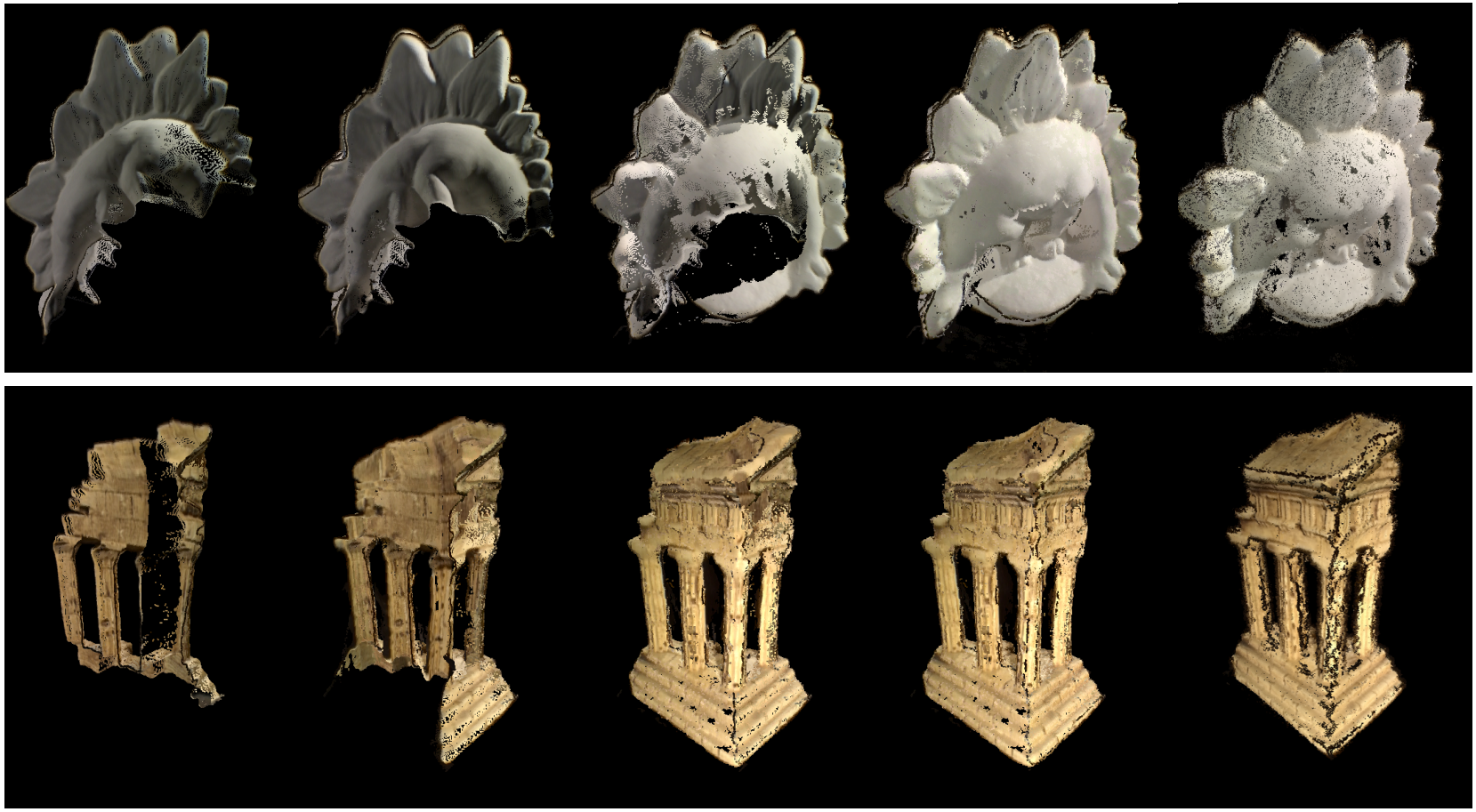
Partial results are also compared against the ground truth from Middlebury. Figure 3.14 shows how the error and coverage change as the number of views increases. As expected, the presented approach can improve coverage whilst simultaneously reducing error. Note that the slight instability at a small number of views is due to problems with the Poisson Reconstruction, not the point clouds. The point clouds produced by this approach are clean and accurate, as shown in Figure 3.15.

Qualitative Analysis

Figure 3.15 shows how, as the number of views increases, so does the quality of the reconstructed cloud. By the time 7 pairs have been selected, the point cloud produced is fundamentally complete. Note that both the front and back of the models have been successfully reconstructed from a maximum of 14 images. This corresponds to 3.8% and 4.5% of the images for Dino and Temple datasets, respectively. Under a naive definition of stereo pair, where the dataset has not been filtered by the relative pose of the cameras, these percentages correspond to actively selecting the best 0.0106% and 0.0144% of the stereo pairs.

3.6 Conclusion

In conclusion, an approach has been presented that is capable of creating a dense reconstruction of a scene by autonomously selecting images that will provide the largest gain to the reconstruction. This approach consists of three key steps. Firstly, a novel deep learning-based method for optical flow, triangulation and reconstruction was presented. Secondly, an NBV selection criterion that can autonomously optimise for either exploration or refinement. Finally, an NBS criterion that can select the best possible stereo pair for the NBV was presented. These three main contributions have been shown to be effective at reducing the number of images required for a high quality reconstruction when compared to brute-force approaches. The result, is a system for pose and view selection that is capable of achieving state-of-the-art results using only 3.8% of the views. More importantly, the approach is able of actively selecting the best 0.0106% of stereo-pairs in the Middlebury dataset.



1 pair 3 pairs 5 pairs 7 pairs Reference (GT)

Figure 3.15: Results for Middlebury Dino (top) and Temple (bottom) Datasets, with varying numbers of stereo pairs. The final column shows the reference model.

However, there is a clear limitation to this approach. As was mentioned in Section 3.1, the methods presented in this chapter only evaluate a finite set of poses. However, outside of MVS, this is rarely the case. If this approach is to be applied to online robotic systems, then there is a need to expand the definition of NBV and NBS to be pro-active, rather than re-active. This means that the NBV and NBS should be actively directing where new observations should be obtained from, rather than relying on passively selecting the best view from a pre-defined set.

Chapter 4

Scenic Path-planning for Multiple Collaborative Agents

In the previous chapter, a series of techniques were presented that shifted the focus from *how to* reconstruct, to *what to* reconstruct. This reduced computation time by pre-selecting the most informative views of the scene. However, the assumption was made that a discrete set of possible observations already existed. The approach then passively selected the most informative views. Unfortunately, this cannot be readily applied to live systems - let alone collaborative ones.

In a live system, it is even more important to take an active role in the data collection process. This implies two requirements. Firstly, the Next-Best View (NBV) estimation must be performed on a much larger continuous pose-space. Secondly, the live agent must be able to navigate through the environment in order to obtain the NBV. These are non-trivial problems that should be solved in a robust and tractable manner.

To address these issues, an active approach to visual reconstruction is introduced in this chapter. The presented approach considers not only the NBV, but the best path (or sequence of viewpoints) to the NBV. This is achieved by adapting techniques from the robotic path-planning literature to use the cost-space defined in Section 3.3. Applying path-planning to this cost space, rather than the traditional Euclidean world, allows the robots to take a "scenic" route to the NBV. It is termed as "scenic" as the robot is choosing a path to the NBV that will provide the best visual information for reconstruction, rather than the shortest path. This approach is called

a “Scenic Pathplanner”.

This chapter also proposes a multi-robot extension to traditional single-robot path-planning. This extension is based on the Next-Best Stereo (NBS) cost defined in Section 3.4. It allows multiple monocular cameras to operate in a *collaborative* manner. Collaborative building of the map, by two or more cameras, has the potential to dramatically increase reliability, while reducing the time needed to perform the reconstruction.

Evaluation on an offline dataset shows that these techniques are able to outperform Bundle Adjustment (BA) based optimisation approaches, while using a fraction of the views. Offline evaluation also shows that the *scenic pathplanner* outperforms competing path-planning approaches. Finally, a live multi-robot evaluation shows that the path-planning approach is deployable on real-world robotic sensors.

This chapter will begin with an introduction to the generalised path-planning problem. The remainder of the chapter will present work published in [88]. This work consists of a NBV-based goal estimation, the scenic pathplanner and finally the collaborative framework. Together, these approaches allow multiple agents to *opportunistically collaborate* in reconstructing an environment.

4.1 Problem Definition

In order to move between any two points in space, there needs to be a continuous, collision-free path between them. Assuming there are no obstacles, the path between two points in a simple 2-Degrees of Freedom (DoF) space is merely the direction vector between them. This could be considered the simplest form of the path-planning problem.

In higher dimensional spaces, such as the 3-DoF Special Euclidean Space (SE(2)), the problem of path-planning becomes more complicated. It is no longer enough to simply move between the start and end position, now the orientation must be considered as well. It is easy to see that as the DoF increase, so does the complexity of the problem. In the presence of obstacles, the problem escalates even further. As can be seen in Figure 4.1, the path must now deal with position, orientation and collisions. The problem escalates even further when non-holonomic constraints are considered, but those are beyond the scope of this work.

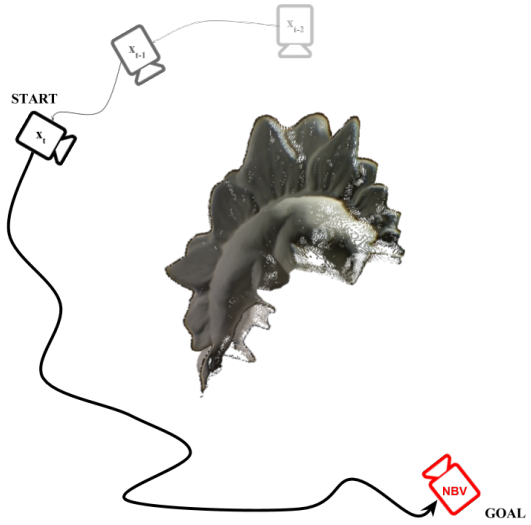


Figure 4.1: Sample stereo pair geometry.

It should be clear that this is a non-trivial problem that has several important requirements. Firstly, the problem must be embedded within a space that is suitable for the task. This implies path-planning should be performed in a space that encapsulates all possible states of the agent. Secondly, the estimated path must be continuous. This means the estimated path should have a value for every infinitesimally small step between the start and goal. Finally, the path must be collision-free. This means that every step along the path must satisfy the global constraints that define "collision".

More formally, path-planning is the task of estimating a continuous trajectory

$$\mathfrak{T} : [0, 1] \rightarrow \mathbb{P} \tag{4.1}$$

from an initial state, $\mathfrak{T}(0) = \dot{\mathbf{x}}_{start}$, to a goal state, $\mathfrak{T}(1) = \dot{\mathbf{x}}_{goal}$. The co-domain of $\mathfrak{T}$ is defined as an n-dimensional *state-space*, $\mathbb{P}$. This is also where $\dot{\mathbf{x}}_{start}$ and $\dot{\mathbf{x}}_{goal}$ are defined.

In principle, the state-space $\mathbb{P}$ can be composed of multiple sub-spaces. Traditional rigid-body path-planning has a state-space ($\mathbb{P} = \mathbb{C}$), where the Configuration Space ($\mathbb{C}$) is normally defined as the set of all possible poses achievable by the rigid-body. In kinodynamic planning, the state-space can consist of both a configuration space, velocities and accelerations. While there are many other alternatives for the space in which $\mathbb{P}$ is defined, they are beyond the scope of this work.

In this thesis, path-planning will be discussed in the rigid-body context

$$\mathbb{P} = \mathbb{C} = \text{SE(3)} \tag{4.2}$$

where SE(3) is the Special Euclidean group consisting of 3-DoF position and 3-DoF orientation. SE(3) fully encompasses the possible states that a rigid-body can assume. This leaves the final requirement: that the path is *collision-free*.

Generally, *collision-free* can be defined as satisfying global constraints imposed on $\mathbb{C}$. Formally, this can be defined as the subset $\mathbb{C}_{free}$ for which the collision-detector

$$\boldsymbol{\zeta} : \mathbb{C} \rightarrow \{true, false\} \tag{4.3}$$

always returns true. Fundamentally, this means that the path $\boldsymbol{\mathcal{T}}$ should exist entirely within $\mathbb{C}_{free}$. It is also possible for $\boldsymbol{\zeta}$ to be a real-valued function (instead of binary). However, for the purposes of this work, it can be assumed that $\boldsymbol{\zeta}$ decides whether any given state $\dot{\mathbf{x}}$ is in collision with the reconstructed geometry.

In practice, it is not enough to simply return a *valid* path, the solution path should also be a *good* one. This is accomplished by optimising

$$\boldsymbol{\mathcal{T}} = \underset{\boldsymbol{\mathcal{T}}'}{\arg\min} \ \boldsymbol{\pi}(\boldsymbol{\mathcal{T}}') \tag{4.4}$$

where $\boldsymbol{\pi}(\boldsymbol{\mathcal{T}}')$ is a cost-function defined on the path. Traditionally, this cost-function has been defined as the Euclidean path-length. Other popular approaches include minimising a real-valued collision function ($\boldsymbol{\zeta}$). These two options can be combined into a cost-integral as will be defined in Section 4.3.2.

The fundamental problem with this approach is that the cost-function, $\boldsymbol{\pi}(\boldsymbol{\mathcal{T}})$, only depends on the Configuration Space ($\mathbb{C}$). The work presented in this chapter will aim to estimate a path that also depends on the partial reconstruction. This path will aim to maximise the reconstruction quality, while reducing the path length. As such, it can be called a "scenic route" from $\dot{\mathbf{x}}_{start}$ to $\dot{\mathbf{x}}_{goal}$. This path will be estimated by including the NBV cost-function (3.22) presented in the previous chapter.

Another important limitation of this approach is that it only considers estimating a path for a single agent. This chapter will present a novel method of path-planning for multiple agents. The

method will estimate collaborative and independent candidate paths. The best path will then be selected. This makes the method an "opportunistically collaborative" approach, where the agents only collaborate when it is convenient and to their combined benefit. This will be done using the NBS cost-function (3.30) from the previous chapter.

It should be obvious that $\dot{\mathbf{x}}_{start}$ is the current pose of each sensor and/or robot. However, it is unclear how $\dot{\mathbf{x}}_{goal}$ is estimated. The following section will show how a Sequential Monte-Carlo (SMC) based approach is used, in conjunction with the NBV cost, to estimate the ideal goal-pose.

4.2 Next-Best View (NBV) Goal Estimation

Traditionally, the goal of a path-planning problem is a high-level directive issued by a human. However, autonomous agents necessitate a method for automatically selecting their goal-state. It is clear that this goal should optimise a particular criteria. How this criteria is defined will be extremely application dependant.

In this chapter, the goal-state is defined as the pose in Special Euclidean Space (SE(3)) space that maximises the potential information gain of the map (*i.e.* the NBV). In Chapter 3, a method was introduced to estimate the NBV. However, it was assumed that the set of putative poses $\dot{\mathbb{X}}$ was known. In order to estimate the NBV for a live-sensor, it is necessary to expand this set. This can be done by extending $\dot{\mathbb{X}}$ to be equal to the $\mathbb{C}$. Formally, this is defined in the same way as the NBV cost from Chapter 3,

$$\dot{\mathbf{x}}_{goal} = \underset{\dot{\mathbf{x}} \in \mathbb{C}}{\arg\min} \ \boldsymbol{\eta}\Big(\dot{\mathbf{x}}, \dot{\mathbb{M}}\Big) \tag{4.5}$$

where $\dot{\mathbf{x}} \in \mathbb{C}$ is a rigid-body pose. However, this would be completely intractable. Instead, this section proposes an SMC-based approach to NBV estimation in the configuration space which approximates equation 4.5 in real time.

4.2.1 Sequential Monte-Carlo Next-Best View (NBV)

Previously, an octree-based space discretisation method was introduced in Section 3.2.3. Using this octree, it was possible to discretise the $\mathbb{R}^3$ component of SE(3) into a more manageable

state. Naïvely, the empty voxels centres ($\dot{v}^e \in \dot{\mathbb{V}}^e$) define a grid of NBV candidates. However, this is a fundamentally flawed approach that suffers from two main limitations. Firstly, the rotational component is completely ignored and adding these extra dimensions would cause the problem to increase exponentially. Secondly, the granularity of the octree would heavily influence the accuracy of the NBV. Increasing the resolution of the octree, in order to increase NBV accuracy, would also cause the complexity of the problem to increase dramatically.

It is clear it would be intractable to attempt an exhaustive search for the NBV in SE(3), even if this is done using a grid-based approach. Instead, this section proposes a SMC sampling method that uses the octree to approximate the distribution of NBV costs across the scene. This approach overcomes the limitations of the grid-based method in both ways. For the first limitation, the SMC method allows rotation-based sampling to be applied. For the second, the particles are not constrained to the grid of the octree. Instead, computational resources are focused in promising areas.

The SMC-based approach is introduced in three steps. Firstly, a proposal distribution is presented. This distribution samples directly from the octree to reduce the likelihood of collisions. Secondly, a NBV-based cost-function is used to define the weights of each particle. Finally, the weighted resampling and propagation process is defined.

Proposal Distribution

It would be intractable to attempt an exhaustive search for the NBV in the whole configuration space, $\mathbb{C}$. Even when done in an SMC framework, the configuration space cannot be directly used as a proposal distribution. This is because SE(3) is simply too large to sample reliably. Instead, the proposed SMC method uses the octree to define the proposal distribution,

$$\Pr\left(\dot{\mathbf{x}}\right) = \begin{cases} 1 & \text{if } \dot{v}_{\dot{\mathbf{x}}} \in \dot{\mathbb{V}}^e \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where $\dot{v}_{\dot{\mathbf{x}}}$ is the voxel that contains the spatial component of $\dot{\mathbf{x}}$. Fundamentally, this implies that the NBV cannot cannot lie within an occupied voxel. This is because, as will be seen in the following Section, occupied voxels are not part of $\mathbb{C}_{free}$ (*i.e.* they are in collision).

Uniformly sampling from this proposal distribution yields a set of particles

$$\mathbb{S}_t = \left\{ s_t^i; i = 0..N \right\} \tag{4.7}$$

where N is the number of particles in the filter. The orientation of each particle is also uniformly sampled. Orientation sampling is application dependant, and Section 4.4 discusses the specific techniques used. In order to use them to model the NBV distribution, these particles must now receive a weight.

NBV Importance Weighting

An observation likelihood is defined using the NBV cost defined in equation 3.22

$$\Pr\left(z_t \middle| s_t^i\right) = 1 - \boldsymbol{\eta}\left(s_t^i, z_t\right) \tag{4.8}$$

where $z_t = \dot{\mathbb{M}}_t$ is the current reconstruction. This implies the normalised weight of each particle is

$$w_t^i = \frac{\Pr\left(z_t \middle| s_t^i\right)}{\sum_j^N \Pr\left(z_t \middle| s_t^j\right)} \tag{4.9}$$

where $\Pr\left(z_t \middle| s_t^i\right)$ is defined as the NBV likelihood for the pose of particle s_t^i. The advantage of this approach is that it is agnostic to the underlying NBV cost, with the exception that it must be bounded and positive (*i.e.* be normalisable).

Once the particles have been weighted by their NBV likelihood, these weights can be used to resample the population.

Resampling

This resampling process is a simple weighted sample with replacement. While there are many techniques to perform this operation, they are beyond the scope of this work. It is sufficient to state

$$\mathbb{S}_{t+1} \sim \Pr\left(z_t | \mathbb{S}_t\right) \tag{4.10}$$

where the symbol $\sim$ implies the particles $\mathbb{S}_{t+1}$ are distributed as $\Pr\left(z_t | \mathbb{S}_t\right)$. This implies samples with higher observation likelihoods are more likely to be drawn in the next particle set,

$\mathbb{S}_{t+1}$. More importantly, the weight in the next iteration

$$w_{t+1}^{i} = \frac{1}{N} \tag{4.11}$$

returns to a flat weight for all particles.

Propagation

A more interesting aspect of this SMC approach is the propagation method. While resampling allows particles with higher likelihoods to be carried forward to the next iteration, they need to be propagated to avoid duplication. Fundamentally, propagation can be defined as adding Gaussian noise to each of the 6-DoF of SE(3). Formally,

$$s_t^r \sim \mathcal{N}\left(s_t, \mathbf{\Sigma}_{s_t}\right) \tag{4.12}$$

where $s_t^r \in \mathbb{S}_t^r$ is a resampled particle, and $\mathbf{\Sigma}_{s_t}$ is the covariance of a Gaussian distribution centred around s_t. Propagating particles in this manner overcomes the two main limitations of the grid-based method. In terms of granularity, Gaussian noise added to the position component moves particles away from voxel centres. For the rotational component, Gaussian noise added to the orientation allows a more robust sampling of the orientation space. As mentioned before, the mechanics of applying Gaussian noise to Special Orthogonal Space (SO(3)) are beyond the scope of this work, and application specific implementations will be discussed in Section 4.4.

In most SMC applications this would be enough to make the proposal distribution match the target over time. However, in this case, the location of the NBV can change drastically as observations are added. More importantly, propagating the particles in the manner described has been known in the literature to lead to particle deprivation. To overcome this limitation, a small percentage of particles ($\mathbb{S}_t^u$) are uniformly sampled directly from the proposal distribution.

Another limitation is that the goal-state ($\dot{\mathbf{x}}_{goal}$) being estimated requires the peak of the target distribution to be found reliably. Propagating particles with Gaussian noise can make this estimate unstable. To avoid this, a small percentage of the best particles ($\mathbb{S}_t^p$) are propagated without diffusion.

The complete set of propagated particles $\mathbb{S}_t$ is then

$$\mathbb{S}_t = \mathbb{S}_t^r \cup \mathbb{S}_t^p \cup \mathbb{S}_t^u \,. \tag{4.13}$$

Note that during resampling and propagation, the particles should *not* converge on a single location. This is because, as will be explained in the next Section, these particles will be used as approximation of the full cost-space. This approximated cost-space will be key to planning the scenic route.

Goal Estimation

While in principle the goal should be estimated as a maximisation on the posterior, $\Pr\left(s_t \middle| z_t\right)$, it is much more efficient to use the observation likelihood, $\Pr\left(z_t \middle| s_t^i\right)$. Therefore, the NBV goal can be estimated as

$$\dot{\mathbf{x}}_{goal} = \underset{s_t \in \mathbb{S}_t}{\arg\max}\ \Pr\left(z_t \middle| s_t^i\right), \tag{4.14}$$

where $\dot{\mathbf{x}}_{goal}$ is the particle that maximises the observation likelihood, and is defined as the NBV.

4.3 Scenic Path-planning

Truly autonomous agents should be capable of negotiating a trajectory to the goal-state. This implies a pathplanner that can provide smooth trajectories, collision avoidance and a cost-minimisation framework.

Traditional robotics-based approaches would normally minimise the path length. Another common operation would be to optimise between path length and distance to collisions. Indeed, most planners perform this kind of operation. The fundamental limitation with these approaches is that they only consider the configuration space in the cost-function. Even when minimising the distance to a collision, the environment is only considered in-so-far as it affects the configuration space.

However, if the goal is reconstruction, then taking the shortest path might result in unfavourable poses for both localization and reconstruction. The sensor will miss good views along the way to its goal and is more likely to get lost. Naïvely, iterative NBV estimation (with increasing radii) could be treated as a path. However, this would have no guarantees about the path length or optimality. Instead, this section presents a novel approach that allows the estimation of a "scenic" route.

4.3.1 Next-Best View Path-planning

The scenic route is defined as the shortest path to $\dot{\mathbf{x}}_{goal}$ that will maximise the potential information gain in the map, both in terms of accuracy and coverage. It follows that the *Scenic Pathplanner* is an approach that can estimate this path. This implies several requirements. Firstly, the path length should be minimised. Secondly, there should be some optimality guarantees[1] within this path estimation framework. Thirdly, the path should aim to maximise the information gain from the environment. Finally, the path should be extensible to multiple collaborative agents.

While there are many traditional robotics approaches that satisfy these requirements, tree-based approaches are particularly well suited. More explicitly, a Rapidly-exploring Random Tree (RRT*) can be used to explore high-dimensional states, optimize path length and guarantee asymptotic optimality. A standard RRT* implementation such as [48] could be used to minimise the NBV cost-function, by adding the NBV cost to the nodes and estimating a cost-integral. However, this would be expensive and inefficient because RRT*-based methods are designed to explore large Voronoi regions of the configuration space with no regard to the cost of that area. They do this by directly drawing samples from the configuration space ($\mathbb{C}$) in order to build the tree. This makes standard RRT* ill-defined to solve a problem when the cost is not just a function of the pose, but rather is a function of the pose and the reconstructed geometry. Instead, a method is required that biases the search towards areas rich in views that will benefit map reconstruction.

To address these issues, this section presents a novel method that combines the high-dimensional exploration of RRT* with a bias towards pre-computed areas of high information gain. Instead of $\mathbb{C}$, the proposed approach samples from the prior distribution of good NBV candidates estimated in Section 4.2.1. In essence, this redefines the configuration space of this problem as $\mathbb{S} \subset \mathrm{SE}(3)$. Stochastically sampling from this distribution biases the growth of the tree towards areas with good NBV cost. More explicitly, this means that the space-exploring properties of RRT* will be aimed at the NBV cost-space. The tree will be biased towards large Voronoi

[1] Guaranteeing optimality means the path-planning approach will always find the best path for a given cost-function. For example, RRT has no optimality guarantees, on the other hand, RRT* has asymptotic optimality guarantees [67] (implying an optimal solution when $t \to \infty$).

regions in the posterior, $\Pr(s_t|z_t)$, rather than SE(3).

A Scenic RRT* Pathplanner can be defined in the configuration space, $\mathbb{S}$, as a collection of nodes $q \in \mathbb{Q}$. The root node is defined as the current position of the robot,

$$q_{start} = \dot{\mathbf{x}}_{start}\,, \tag{4.15}$$

the goal node is similarly defined as the current peak of the NBV distribution (as estimated in Section 4.2)

$$q_{goal} = \dot{\mathbf{x}}_{NBV} \tag{4.16}$$

where $\left\{ q_{start}, q_{goal} \right\} \in \mathbb{Q}$.

Assuming these definitions, the task of growing a scenic RRT* to get from start to goal would usually be done as follows. Firstly, a sample q_{rand} is drawn from the configuration space ($\mathbb{S}$). It is important to stress that sampling from $\mathbb{S}$ fundamentally biases the RRT* towards areas with high concentration of particles (and therefore a good NBV cost). This can be thought of as defining a set of optional "intermediary goals" that the path-planning attempts to visit.

Secondly, the nearest-neighbour to q_{rand} is found in the tree as

$$q_{near} = \underset{q \in \mathbb{Q}}{\arg\min}\ \boldsymbol{\delta}(q, q_{rand}) \tag{4.17}$$

where $\boldsymbol{\delta}(\cdot)$ denotes the Euclidean distance.

Thirdly, a new vertex

$$q_{new} = q_{near} + \Delta_q \frac{(q_{rand} - q_{near})}{\| q_{rand} - q_{near} \|} \tag{4.18}$$

is added at a predefined step Δ_q in direction q_{near} to q_{rand}, shown in Figure 4.2a.

A standard RRT implementation would add q_{new} with q_{near} as a parent. However, this solution does not have any optimality guarantees. To ensure asymptotic optimality, it is necessary to perform a "rewiring" of the tree when a new node is added. This is done by first estimating a set of nearest-neighbour nodes shown in Figure 4.2b,

$$\mathbb{Q}_{nn} = \{ q \in \mathbb{Q} | \boldsymbol{\delta}(q, q_{rand}) < r_{nn} \} \tag{4.19}$$

where r_{nn} is the neighbourhood range being considered.

The first step of the rewiring process, known as re-parenting, replaces q_{near} with any node $q_{nn} \in \mathbb{Q}_{nn}$ that creates a shorter path to q_{new}. This is shown in Figure 4.2c. In the second step, every node q_{nn} is checked for a shorter path that goes through q_{new}. If there is a shorter path, the tree is rewired to reflect this, shown in Figure 4.2d.

The RRT* algorithm terminates when

$$\boldsymbol{\delta}\left(q_{goal}, \mathbb{Q}\right) < \Delta_q\,, \tag{4.20}$$

at that point the tree is considered complete. Algorithm 1 shows a more intuitive example of how the scenic path-planning tree is built in an RRT* context. It should be noted that, in principle, the algorithm can run indefinitely. This continues to optimise the path, guaranteeing asymptotic optimality.

The final trajectory is then simply propagating back up the tree to find the path

$$\boldsymbol{\mathcal{T}} = \left\{ q^i; i = 0..Q \right\} \tag{4.21}$$

where Q is the number of steps, $q^0 = q_{start}$, $q^Q = q_{goal}$ and q^{i-1} is the parent node of q^i.

Finally, the trajectory cost can be calculated as

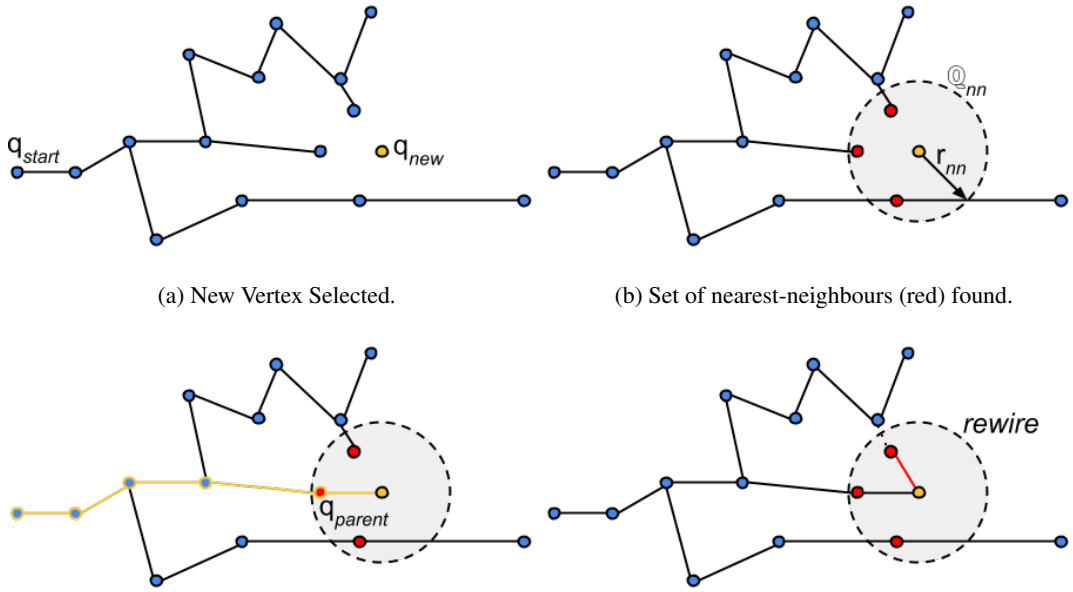
$$\boldsymbol{\pi}(\boldsymbol{\mathcal{T}}) = \sum_{i=1}^{Q} \boldsymbol{\delta}\left(q^{i-1}, q^i\right) \tag{4.22}$$

which implies a Euclidean distance minimisation.

This novel formulation allows high information paths to be estimated. However, as was seen in Chapter 3, there is nothing to guarantee that the NBV at the end of the path will be a sensible stereo-pair to the current view. More importantly, this implementation does not consider multiple agents. Both of these concerns can be addressed by defining a cost-function to replace the Euclidean distance of the graph edges.

4.3.2 Opportunistic Collaboration

Until now, only a single camera performing guided reconstruction of its environment has been considered. However, if there are multiple cameras, the proposed techniques can be extended to perform joint path-planning of all cameras simultaneously.



(a) New Vertex Selected.

(b) Set of nearest-neighbours (red) found.

(c) Select parent with shortest path (yellow).

(d) Rewiring nearest-neighbour nodes.

Figure 4.2: Sample RRT* construction (from [32]).

A naive approach would be to simply estimate independent paths for each agent. This would allow the robots to add their observations to the same reconstruction. However, this is a very superficial level of collaboration.

A second alternative would be to estimate a single path for all agents. In this method, a stereo cost similar to that of Section 3.4 could be used to constrain the path to keep agents acting as stereo pairs. This is a more effective collaboration approach, but constraining the cameras to act collaboratively may not always be optimal.

What is required is a method that allows the sensors to decide whether to act independently or collaboratively. To achieve this, the method grows separate scenic path trees for each mode of operation. This allows the sensors to automatically select the best path from all trees and become *opportunistically collaborative*.

The case of two monocular sensors is described here, however, it should be noted that this method is easily extensible to more than two sensors. The robots are treated as being completely independent from each other. The agents report their position in the global map, rather than the explicit relative position to each other, and pathplanning is performed on the map coordinate

1: **function** BUILDSCENICRRT($\dot{\mathbf{x}}_{start}$,$\dot{\mathbf{x}}_{goal}$)
2: $\mathbb{Q} \leftarrow \emptyset$
3: $q_{start} \leftarrow \dot{\mathbf{x}}_{start}$
4: $q_{goal} \leftarrow \dot{\mathbf{x}}_{goal}$
5: $\mathbb{Q}$.ADDVERTEX($\emptyset$, q_{start})
6: **while** $\boldsymbol{\delta}\left(q_{goal}, \mathbb{Q}\right) \geq \Delta_q$ **do**
7: $q_{rand} \leftarrow$ SAMPLENBVSPACE($\mathbb{S}$)
8: $q_{near} \leftarrow$ NEARESTVERTEX(q_{rand}, $\mathbb{Q}$)
9: $q_{new} \leftarrow$ NEWVERTEX(q_{near}, q_{rand}, Δ_q)
10: $\mathbb{Q}_{nn} \leftarrow$ FINDNEARESTNEIGBOURS(q_{near}, r_{nn}, $\mathbb{Q}$)
11: $\mathbb{Q}$.ADDVERTEX(q_{near}, q_{new})
12: $\mathbb{Q}$.REPARENT(q_{new}, $\mathbb{Q}_{nn}$)
13: $\mathbb{Q}$.REWIRE($q_{new}\,\mathbb{Q}_{nn}$)
14: **end while**
15: **return** $\mathbb{Q}$
16: **end function**

Algorithm 1: RRT* version of Scenic Pathplanner.

frame. More explicitly, it assumed that each sensor only knows the current position of the other robot in the map coordinate frame. In practice, this pose estimate is obtained from a live SLAM-based localisation and sensor fusion framework, and assumed to be correct. The noise characteristics of the pose estimate could be used to aid in pathplanning, however, that is beyond the scope of this thesis. This area of future work is discussed in Chapter 6.

Using the pose information, it is possible for each agent to independently grow two different trees and extract two paths for each camera. Both trees are already biased towards areas of good NBV cost, so their "stereo" costs are optimised instead. Given that a "scenic" path is being estimated, it is important for the estimation to have some notion of path length. This is enforced by estimating the path cost integral, which will be explained in the context of each tree.

The first RRT* is an SfM tree. This tree attempts to optimise the stereo configuration of each successive pair of nodes along the path of a single camera. That is, the cost the sensor incurs by moving from q^{i-1} to q^i is computed from equation 3.30. Formally, the cost of the trajectory is

defined as

$$\boldsymbol{\pi}_{sfm}(\mathfrak{T}) = \sum_{i=1}^{Q} \boldsymbol{\sigma}\left(q^{i-1}, q^{i}, \dot{\mathbb{M}}\right) \boldsymbol{\delta}\left(q^{i-1}, q^{i}\right) \tag{4.23}$$

where the Euclidean distance parameter fundamentally changes this sum into a state-cost integral.

The second RRT* is a collaborative stereo tree. This tree attempts to optimise the stereo configuration of each node along the path with the last known position of the other agent. Formally, the cost is estimated as

$$\boldsymbol{\pi}_{col}(\mathfrak{T}) = \sum_{i=1}^{Q} \frac{\boldsymbol{\sigma}\left(\dot{\mathbf{x}}_o, q^{i-1}, \dot{\mathbb{M}}\right) + \boldsymbol{\sigma}\left(\dot{\mathbf{x}}_o, q^{i}, \dot{\mathbb{M}}\right)}{2} \boldsymbol{\delta}\left(q^{i-1}, q^{i}\right) \tag{4.24}$$

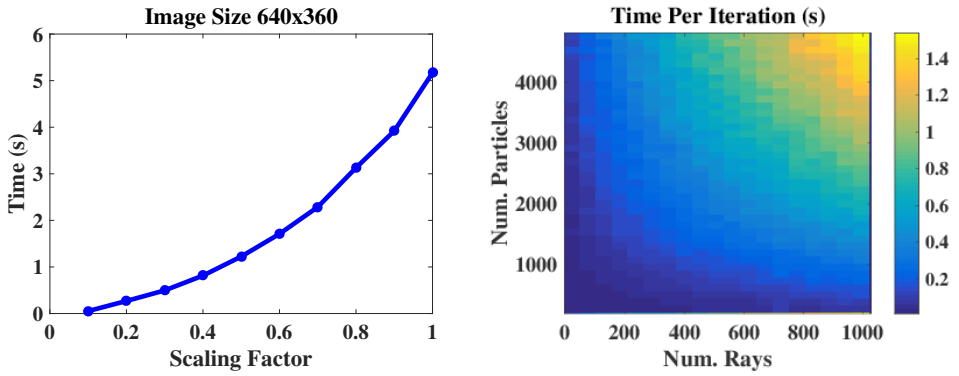
where the average of the cost of two nodes is taken in order to estimate the state-cost integral.

This process is repeated for all agents. Finally, once all paths have been estimated, the agents make an autonomous decision about what the best course of action is. They each share their path costs and the path with the minimum cost will dictate how the sensors operate. There are two possible scenarios. In the first, they both move towards independent goals while performing Structure from Motion (SfM). This path guarantees a good stereo path between successive poses, so SfM is guaranteed to produce reasonable results. In the second, one agent will remain static while the other moves to a position of vantage to collect more data. The collaborative path guarantees that the sensors will be a good stereo pair, allowing wide-baseline stereo techniques to operate robustly.

It is important to note that once the observations are obtained, a new goal (Section 4.2) and paths are estimated. This is because adding observations to the map naturally changes the NBV and therefore the scenic route. While this might seem wasteful, the whole pipeline is extremely efficient and can be performed in parallel. This will be shown in the following Section, along with a quantitative and qualitative evaluation of the work presented here.

4.4 Evaluation

The contributions of this chapter have focused on allowing a pair of mobile cameras to opportunistically and collaboratively explore an unknown area and rapidly create a 3D reconstruction



(a) Optical flow

(b) Cost-space approximation

Figure 4.3: Time-cost distributions showing the effects of different parameters on a) optical flow and b) SMC cost-space approximation.

of the scene. An effective system should be able to plan a path which can rapidly explore and refine the map using a small number of maximally informative views. To demonstrate this, a qualitative and quantitative evaluation on an online dataset is presented. This is followed by an evaluation on a live system that can autonomously reconstruct a scene. However, before showing the performance of the presented approach, an evaluation of the speed is presented.

4.4.1 Timing Information

This section presents an analysis of the time required to perform a successful iteration of the approach. This was done on a machine equipped with an Intel Xeon X5550 (2.67GHz) and 96GB of RAM, the approach does not use a GPU but rather relies on OpenMP for CPU multi-threading. The duration of each step in the pipeline will be discussed, followed by a detailed description of what incurs these costs. The reconstruction will be discussed first, followed by the NBV-based goal estimation and finally the scenic path-planning.

The reconstruction step is usually the most time-consuming part of the pipeline. The reconstruction consists of a bi-directional optical flow, followed by an iterative linear triangulation step and data association. A full-sized image takes around 5 seconds, iterative triangulation takes 0.5 ± 0.4 seconds and data association 2 ± 1.8 seconds. Much faster results can be achieved by downsampling the image, as shown in Figure 4.3a. As it can be seen, most of the time is

used by the optical flow. However, this approach is completely agnostic to the source of the correspondences. It would be relatively trivial to implement a GPU-based approach that runs at framerate.

Estimating the goal-state, and performing the cost-space approximation depends mainly on two factors: the number of particles and the number of rays cast per particle. Figure 4.3b shows a heatmap of the per-iteration time-cost for varying these parameters. In practice, the filter normally consists of 5000 particles casting 100 rays. This corresponds to 0.2 seconds per iteration; usually no more than 2 iterations are necessary per NBV.

Finally, the path-planning is designed to perform the best plan possible in an allocated amount of time. Normally a good path can be found within $1-2$ seconds, and improved upon with increasing amounts of time. In practice, all path-planning tasks are performed in parallel, allowing the best path to be found quickly and effectively.

In total, this approach can be performed in 10 seconds. This is a reasonable amount of time, since a robot/agent needs to move to the next position while most of the processing is happening. Ignoring the reconstruction process which could be replaced by a more efficient approach, path-planning and goal estimation can be done within $3-4$ seconds. More importantly, experimental evidence shows that the number of rays, scale of the image and octree resolution can be reduced dramatically without much loss in reconstruction quality.

Regardless, this approach is capable of operating on both offline and live systems in its current state. In the following Sections, it will be shown that state-of-the-art performance can be achieved on an offline dataset. It will also be shown that a live-system can use these planning methods to autonomously reconstruct a room.

4.4.2 Offline Dataset Reconstruction

In order to evaluate offline performance, a dataset is collected which consists of several minutes of a UAV moving around a room. This footage is extremely dense in the pose space, as the camera is moved multiple times over the same area but with different orientation. This footage is then used to extract 8500 images from the camera. The images are processed using a state-of-the-art batch reconstruction algorithm [141, 17]. This full reconstruction takes

several days to be completed, requires large amounts of memory and contains large amounts of outliers. Nevertheless, it can be used as a "baseline" for comparison of the scenic pathplanner performance.

Apart from the reconstruction, the algorithm provides the set of images with their respective pose in SE(3). The SE(3) poses are assumed to be correct, and are used by the Scenic Pathplanner to reconstruct the scene. As mentioned in Chapter 3, the approach presented here assumes the estimated poses are correct. This cannot be guaranteed by a batch reconstruction approach on such a large dataset. However, the Scenic Pathplanner is robust to a small amount erroneous poses due to the nature of the octree structure. This allows a meaningful evaluation to be performed, while leaving the problem of noisy pose estimates as the remit of future work (discussed in Chapter 6).

In order to obtain a ground truth reconstruction, a depth sensor running Kinect Fusion[99] is used. This provides an accurate reconstruction of the environment that can be registered to the different evaluated methods.

Experimental Setup

Since the objective of these experiments is to map an unknown environment, the reconstruction process starts with absolutely no knowledge of the scene. The algorithm is only provided with a pair of images from the start position which are used to initialise the reconstruction (and octree). After that, the approach is entirely autonomous. In principle, it would be trivial to estimate an ideal "best stereo pair" to initialise from. However, this is unnecessary to prove the contributions presented here. It also would not emulate the behaviour of a live system. More importantly, this demonstrates that the approach is capable of recovering from less-than-ideal initialisation.

Each iteration of the scenic path-planning and reconstruction is then performed as follows. Firstly, a reconstruction from the selected stereo pair is created. This can be either a collaborative stereo-pair, or an independent SfM pair. Secondly, the SMC-based NBV goal estimation is performed. This provides the scenic pathplanner with a goal and a set of samples. Finally, the scenic pathplanner estimates all relevant paths and optimises the behaviour of each sensor. The selected paths dictate whether the sensors will act as a collaborative stereo-pair or independent SfM agents. It will also determine whether the agents have a preference for exploration or

refinement. Once path choice has been made, the first step of the path is performed. In the SfM case, this adds two stereo pairs to the reconstruction (one per agent). In the collaborative stereo case, this adds one pair including both agents.

The goal estimation is performed on a 4-DoF manifold of SE(3). There are various reasons for this approach. Firstly, this allows the sampling of SO(3) to be collapsed into sampling the yaw angle of the camera. Secondly, and most importantly, UAVs *cannot* reliably attain non-zero angles in pitch and roll as this would cause the UAV to move.

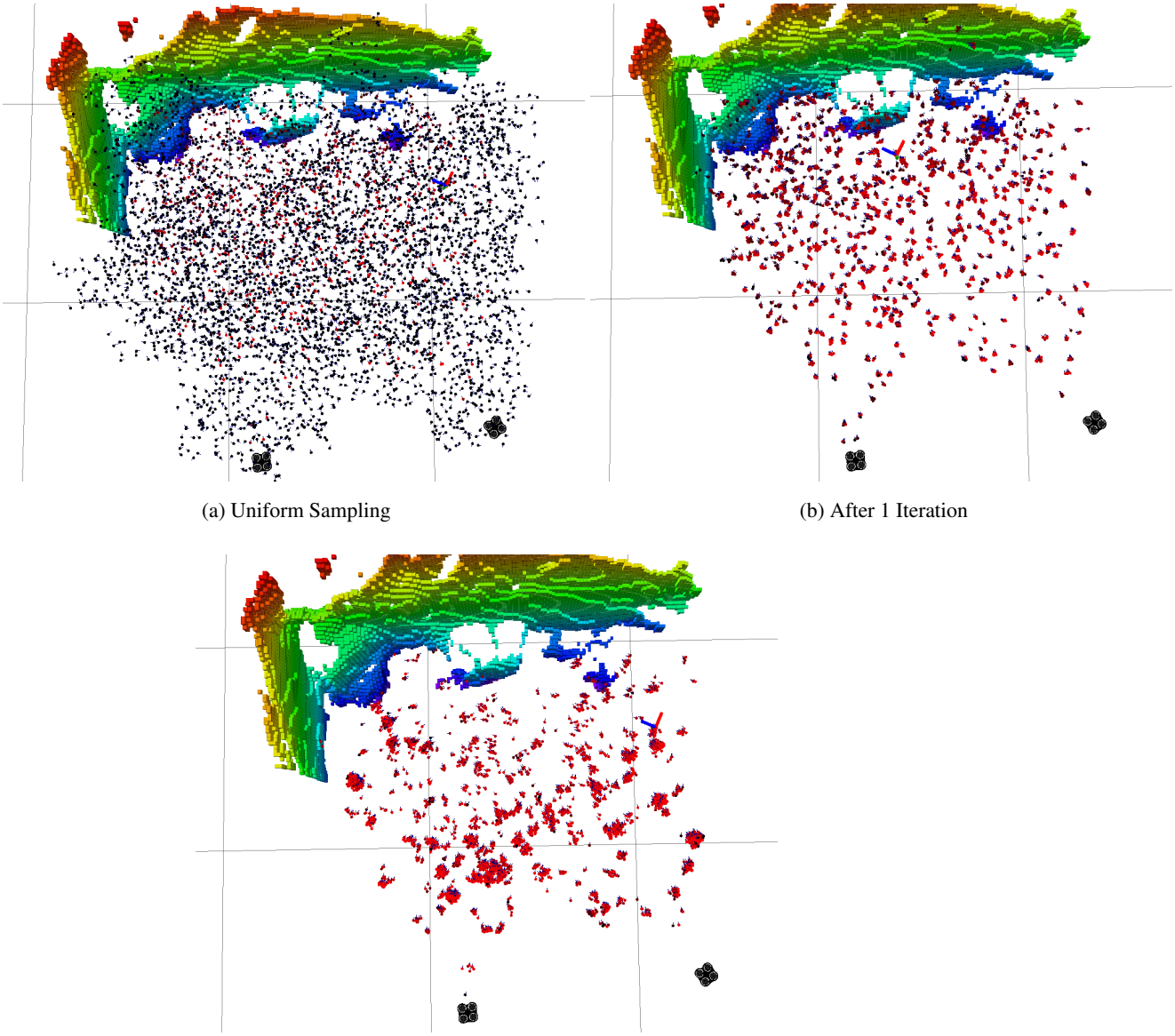
The scenic path-planning is done in full SE(3). This is done because there is no advantage to limiting the pathplanner to an SE(3) manifold. More importantly, the presented approach is able to cope with large pose-spaces easily.

It is important to note that the whole pipeline operates entirely in the continuous configuration space. It is only once the full SMC and path-planning processes have been performed that the selected poses are related back to the dataset. This is done by finding the nearest-neighbour (in the dataset), to the pose (in SE(3)) selected by the scenic pathplanner. This allows repeatability during tests. Finally, for these experiments the NBV parameters are set to $\alpha = 3$, $\gamma = 0.7$.

Qualitative Analysis

Firstly, 3 successive iterations of the SMC space sampling are shown. Figure 4.4 illustrates how the SMC sampling finds areas of good scenic value. In Figure 4.4a, it can be seen that the particles have been uniformly sampled from the 4-DoF manifold. The particles with the best costs are highlighted in red. Over the following iterations it is possible to see how the samples coalesce into clusters of good views. After 10 iterations a good approximation of the distribution is achieved, as shown in Figure 4.4c.

Since SMC performs a weighted resample, the larger the grouping of particles the more benefit the sensor would get from visiting it. Therefore, the scenic path-planning is expected to prefer these clusters as it makes its way to the goal. Figure 4.5 shows the four different paths estimated from the cameras to the goal pose. As expected, the paths show a bias towards areas of high particle concentration, thereby making the sensor take a more scenic route. In these Figures, the paths computed in yellow and orange are the collaborative stereo paths, those in purple and pink



(a) Uniform Sampling

(b) After 1 Iteration

(c) After 10 Iterations

Figure 4.4: Example of the SMC Sampling of the 4-DoF manifold of SE(3).

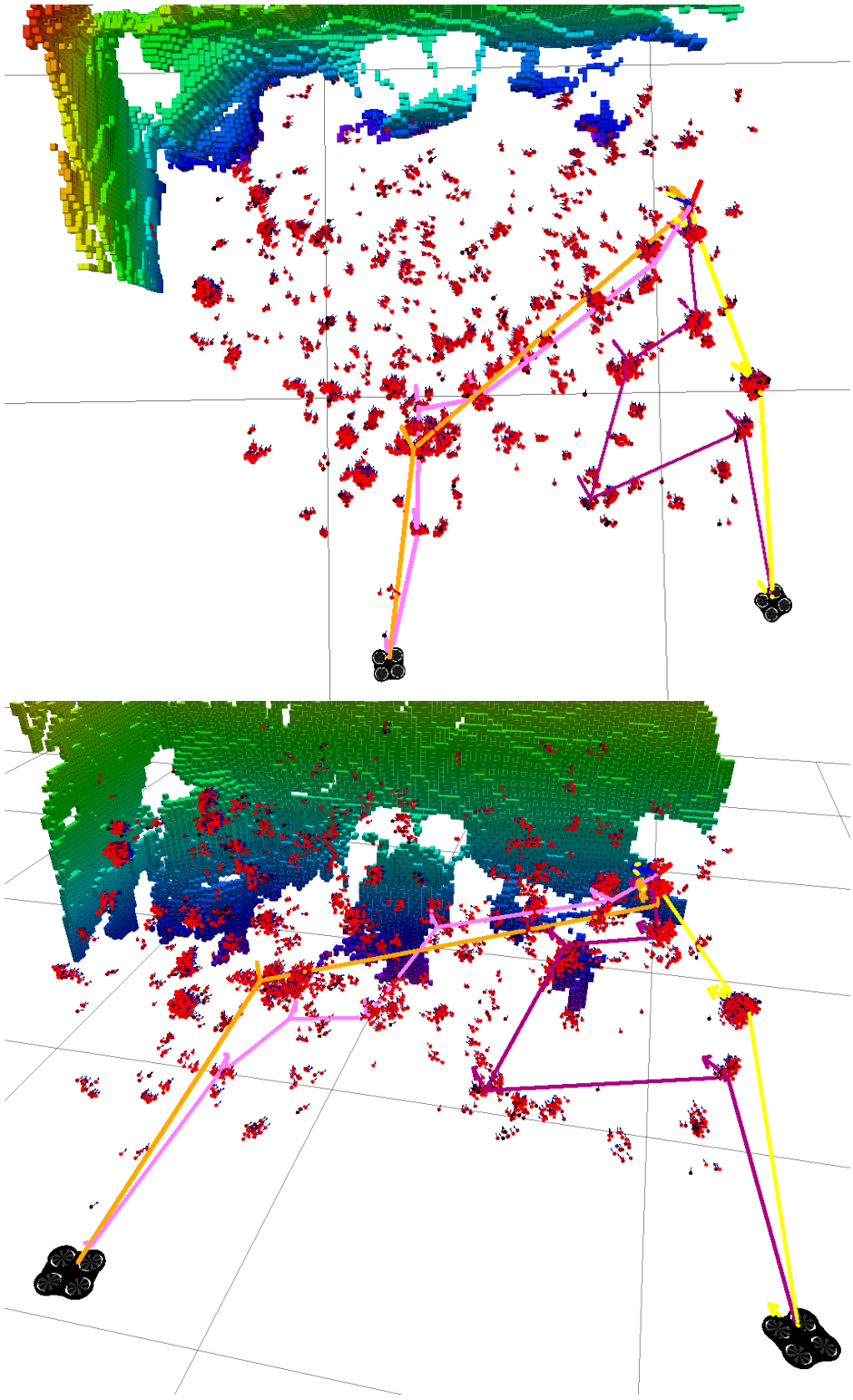


Figure 4.5: UAV Path-planning: the purple and pink tracks show SfM paths, the yellow and orange tracks show Collaborative Stereo paths.

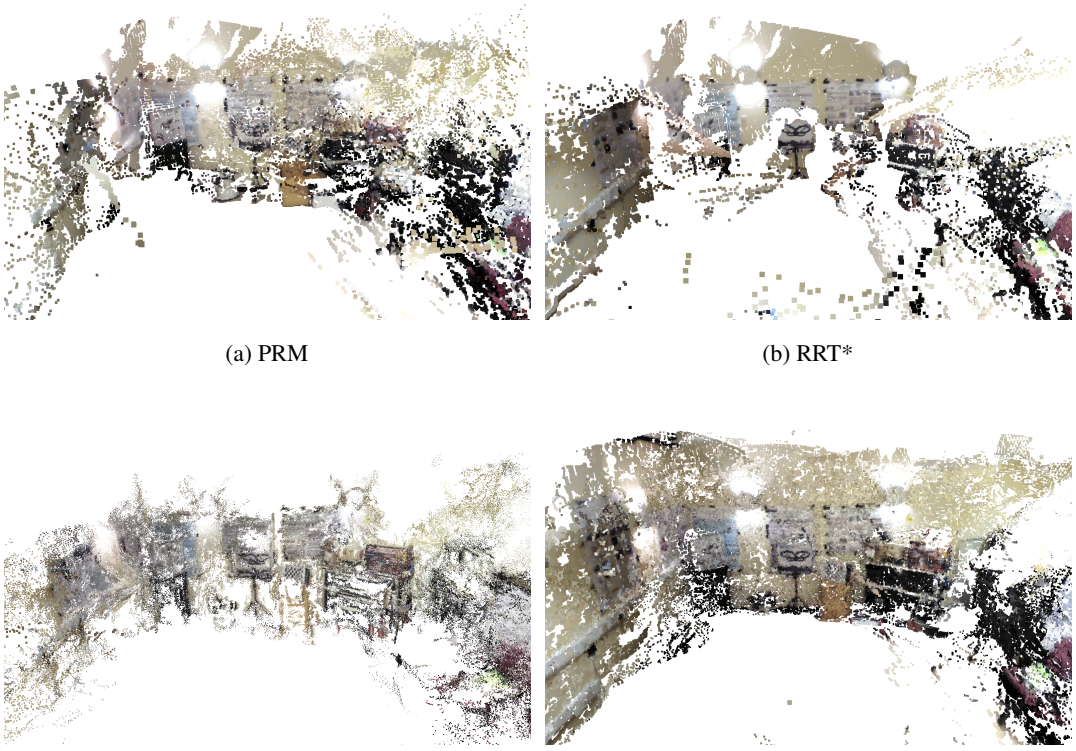
(a) Proposed (b) Kinect Fusion [99]

Figure 4.6: Close up of the reconstruction performed by Kinect Fusion and the proposed Scenic Route Reconstruction

are for SfM. Notice how the SfM paths make their way towards the goal in a zig-zag fashion. This happens because the path-planning is aiming to minimise the stereo costs and therefore prefers wider baselines than a direct path would afford. In addition, the zigzags can be seen to flow into areas with high particle density. More importantly, it should be noted that while the paths take detours to the goal, they are not creating loops or dramatically increasing path length. This is because the estimated paths are optimised for path length using the state-cost integral. Using these paths, the scenic pathplanner is able to autonomously reconstruct the scene.

A close-up of the resulting reconstruction can be seen in Figure 4.6a, with a corresponding ground truth reconstruction in Figure 4.6b. Note that a similar level of scene coverage is achieved, while maintaining low depth error. It should also be noted that while the Scenic Pathplanner compares favourably to the ground truth, this comparison is inherently unfair, as the Kinect Fusion [99] approach results in a triangular mesh which is visually more appealing.

A more complete reconstruction, using 150 stereo pairs, can be seen in Figure 4.7, where the results from two other path-planning approaches and an online batch approach are also shown. Figures 4.7a and 4.7b show reconstructions done by Probabilistic Road Map (PRM) and RRT*, respectively. Since these approaches are not trying to optimise the reconstruction during navigation, they lead to either high noise (PRM) or low scene coverage (RRT*). Figure 4.7c shows the reconstruction obtained by 8500 frames of Visual Structure from Motion (VSFM) [17, 45, 141]. Notice that it is not as dense, and has considerably more noise than the proposed



(a) PRM (b) RRT*

(c) VSFM (d) Proposed

Figure 4.7: Comparison of the reconstructions done by the different path-planning algorithms, and the batch approach.

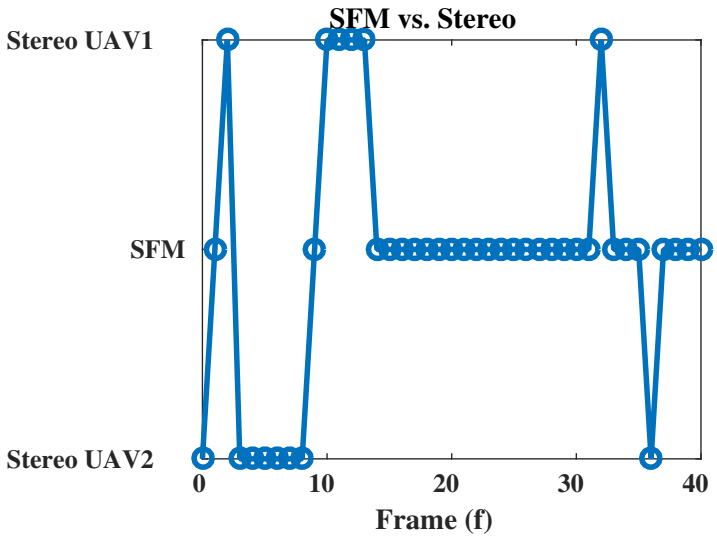


Figure 4.8: Autonomously switching between collaboration & SfM

method despite using the full dataset.

Figure 4.8 illustrates the opportunistic collaboration system. The figure shows how the UAVs vary between collaborative stereo or independent SfM agents. Initially (frames $0-15$) the UAVs collaborate to refine their map of the world while carving out areas of empty space. Once the initial map has been refined, it eventually proves more advantageous for them to explore separately and cover more ground. It is not uncommon for the UAVs to then be imaging different areas of the room, *e.g.* from frames $15-30$. However, if the UAV happen to be in the vicinity of each other, they will start opportunistically collaborating again (such as in frames 33 and 38).

Quantitative Analysis

This section demonstrates that the proposed scenic path-planning leads to significantly better reconstructions than generic pathplanners. Each pathplanner is integrated within the reconstruction framework of Section 3.2 and is evaluated based on the error metrics of Section 3.5.1. A comparison is made against VSFM (PMVS+CMVS)[17, 45, 141] and it is shown that the scenic pathplanner achieves comparable results using only a fraction of the data. It is important to note how significant comparison against this approach is. VSFM [17, 45, 141] is an approach that is widely used in the field. More importantly, it is one of the top scoring approaches in the

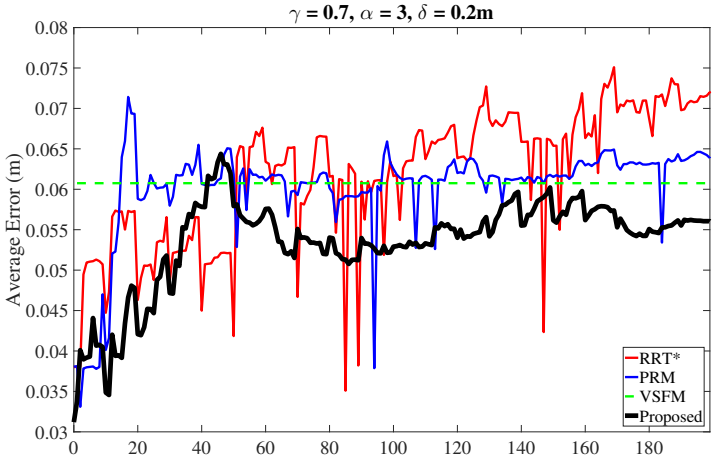
Middlebury dataset by Seitz *et al*. [114], scoring 99.6% completion (within 1mm) and 0.69mm accuracy (with a 99% threshold) on the Dino dataset. Finally, this approach uses all views available, making it a very indicative comparison.

In order to make the comparison fair, the generic path-planning algorithms PRM and RRT* are given the computed NBV goal-state rather than selecting a random one. Without this guidance the reconstructions are unusable. However, PRM and RRT* are both optimising the path length to the goal state. This makes these algorithms incapable of enforcing stereo constraints. As such, the robots tend to observe different regions for most of the reconstruction.

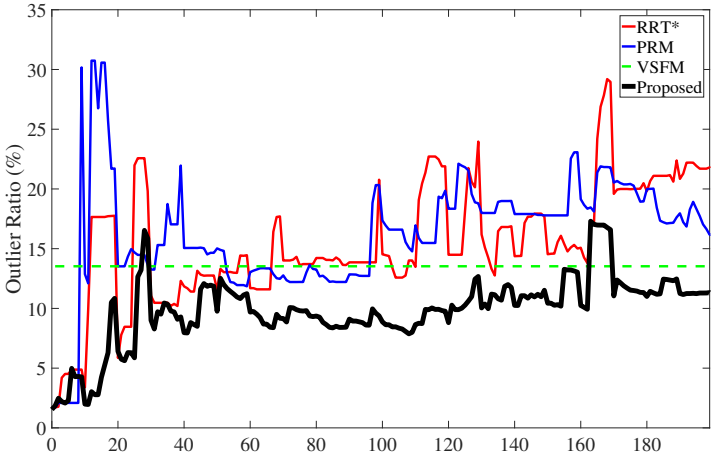
For the VSFM baseline, the full dataset is used to perform the reconstruction. This provides a baseline value for each metric. The dataset consists of over 72 million possible stereo pairs (under a naive definition of stereo pair, where the dataset has not been filtered by the relative pose of the cameras). While some of these pairs might be trivially discarded by an algorithm that has access to the images *a priori*, the presented approach explicitly does *not* use this information. To simulate a live robotic navigation task, the planning is done on the SE(3) manifold and is only related back to the dataset when choosing the nearest-neighbour pose. Therefore it is important to keep in mind that selecting 200 stereo pairs, as shown in Figure 4.9, is still $< 2.7 \times 10^{-4}\%$ of the possible stereo pairs (using the naive interpretation of plausible stereo pairs).

Figure 4.9c demonstrates the scenic pathplanner can achieve coverage that is comparable to VSFM - nearly 70% of the ground truth - using under 150 pairs. More importantly, the proposed method explores the space faster than both competing pathplanners while also achieving a higher final coverage. Furthermore, notice the "stepped" behaviour in the curve, which corresponds to autonomous switching between exploration and refinement. In this Figure, the coloured bar represents the SfM (blue) and stereo (green) opportunistic collaboration decision. Notice how between frames 40-120, the system switches to refinement and the number of collaborative views increases. Between 100-120 there is a spike in coverage due to stereo observations, the system switches back to exploration and the SfM views increase.

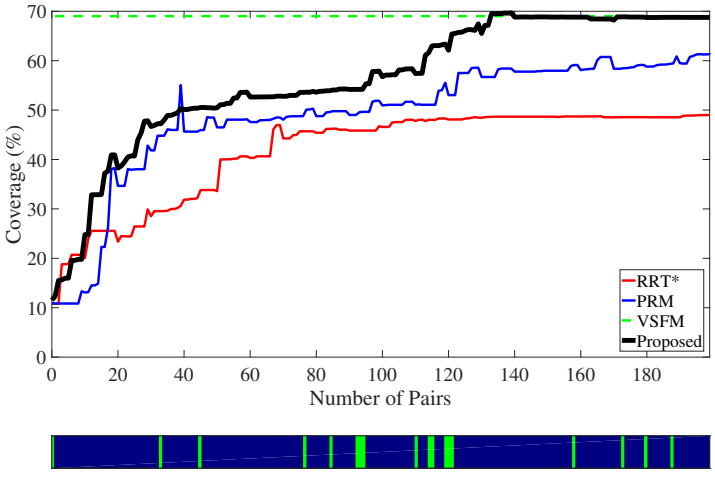
Figure 4.9a shows how the average point error progresses with the number of frames. The scenic pathplanner consistently outperforms PRM and is only worse than RRT* for a short period between frames $30 - 50$. This is because, as shown in Figure 4.9c, that period corresponds to rapid exploration that RRT* does not perform. The scenic pathplanner is, in general, significantly



(a) Average Point Error



(b) Outlier Ratio



(c) Coverage Ratio

Figure 4.9: Reconstruction performance measures plotted against number of image pairs, for various different path-planning algorithms (and the baseline batch system). The colour bar represents the collaboration decision made by the agents, blue is SfM and green is stereo.

more accurate than VSFM. Areas where VSFM outperforms the proposed technique correspond directly to the periods of exploration, when coverage grows rapidly. In fact, in areas of low coverage growth (refinement behaviour), the error decreases below that of VSFM (frames $60 - 100$) and only grows larger during an exploration period ($100 - 130$).

Finally, in Figure 4.9b, the proposed method can be seen to consistently exhibit fewer outliers than all other pathplanners. Indeed, apart from failure cases at frames 30 and 160 (which added noisy measurements to the map) the scenic pathplanner also outperforms VSFM, maintaining around 10% outliers.

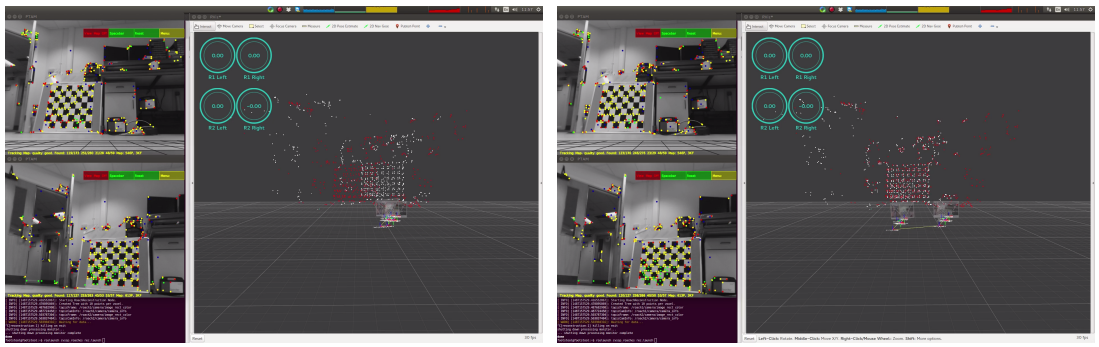
Having qualitatively and quantitatively validated the proposed approach on online datasets, it will now be validated on a live system.

4.4.3 Online Reconstruction

This Section, first discusses the implementation of a live reconstruction system that uses the proposed approach to perform an intelligent, dense 3D reconstruction of its environment. Qualitatively, the results are compared to a dense RGB-D Simultaneous Localisation and Mapping (SLAM) [75] approach. Quantitatively, it is shown that the scenic pathplanner is not only capable of autonomously reconstructing the environment, but that setting the value of γ will either encourage or discourage exploration.

Experimental Setup

In order for the sensors to autonomously navigate their environment, vision-based SLAM is performed to obtain a consistent pose estimate. This pose estimate is then used in a sensor-fusion framework, along with the Inertial Measurement Unit (IMU) and wheel odometry to obtain a robust pose estimate for each camera. While this is enough for a single agent to perform reconstruction, this experiment requires multi-agent reconstruction. Therefore, a reprojection-error based point cloud alignment is performed on the sparse visual landmarks from each SLAM system. This allows the similarity transform between the cameras to be estimated, effectively putting them in the same coordinate frame. Once the sensors are operating in the same coordinate frame, the current image and pose of each camera is used to initialise the reconstruction (and octree).



(a) Unregistered (b) Registered

Figure 4.10: SLAM system before and after coordinate frame registration.

For these experiments, the stereo parameter $\alpha = 7$. This enforces a narrower baseline which makes it easier for the SLAM system to keep track of the pose (less pure rotation). Since these experiments consist of ground-based sensors, the sampling for NBV and path-planning are limited to SE(2). While this is not strictly necessary, it reduces complexity and increases performance.

To begin the reconstruction, the sensor platforms are placed in the centre of an environment. They each autonomously initialise their SLAM systems and share their pose and point cloud with a base-station. As mentioned in section 4.3.2, the poses reported by the SLAM systems are in independent coordinate frames and a registration step is necessary. The registration step can be seen in Figure 4.10. Each robot has its own sparse point cloud, coloured in white and red (left and right, respectively). The point clouds are registered using 2D-3D correspondences between their clouds as constraints for a 7-DoF ICP. Figure 4.10a shows the state of the SLAM pose estimation before registration. Figure 4.10b shows the system after the registration has converged. Unfortunately, this process does not use any of the noise characteristics, which would be useful for registration, reconstruction and pathplanning. This area of future work is discussed in Chapter 6.

Once the sensors are operating in the same coordinate frame, the current image and pose of each camera is used to begin reconstruction. The process then follows that of Section 4.4.2, with the exception of the error metrics which follow the Middlebury evaluation on 3.5.2. However, at each iteration, the cameras will autonomously travel to the next position along the scenic route.

Qualitative Analysis

Figure 4.11 shows the system reconstructing a scene. On the left column, the sparse SLAM system can be seen tracking the pose of both cameras (and keeping them registered). The middle and right columns show different views of the robots performing the reconstruction. Starting from the top row, the robots start looking at a small area of the scene. As the system progresses, the robots explore more of the area and the reconstruction grows accordingly. It is important to note that there is no human interaction, and the exploration is completely autonomous.

Figure 4.12, shows that the final result of the scenic pathplanner are point clouds that are both dense and detailed. The level of detail is comparable to the "ground truth" obtained using an RGB-D camera. In principle this is an unfair comparison, as a depth sensor fundamentally has more information (and better depth estimates). However, at least qualitatively, this is not an issue for the Scenic Pathplanner. In fact, the model reconstructed by the Scenic Pathplanner is denser than the RGB-D reconstruction. The reason for this is that most RGB-D SLAM methods must downsample the images it receives in order to perform at framerate. This reduces not only the density of the reconstruction, but also decreases the localisation performance. Since the Scenic Pathplanner is choosing its views, it can afford to process the whole image. This results in a more photo-realistic reconstruction. It should be mentioned that while these results are qualitatively more appealing, the noise inherent in monocular reconstructions gives the RGB-D approach better performance.

The Scenic Pathplanner also computes the navigability of the space it reconstructs. Therefore, it knows which areas of the map the sensor can realistically reach and which are out of bounds.

Quantitative Analysis

In order to validate the online performance of this approach quantitatively, the various metrics of the Middlebury evaluation are shown. The scenic pathplanner is run with different values of γ and the error and coverage are shown. This demonstrates that the willingness of the sensor to explore its environment is impacted significantly by γ. It also shows that relatively low amounts of error are achievable.

Figure 4.13 shows the error achieved by the scenic pathplanner using different percentiles. It

Figure 4.11: Sequence of images from the live reconstruction.

(a) Proposed

(b) RGB-D SLAM [75]

(c) Proposed

(d) RGB-D SLAM [75]

Figure 4.12: Reconstruction comparison for the scenic path-planning algorithm and state-of-the-art RGB-D SLAM [75].

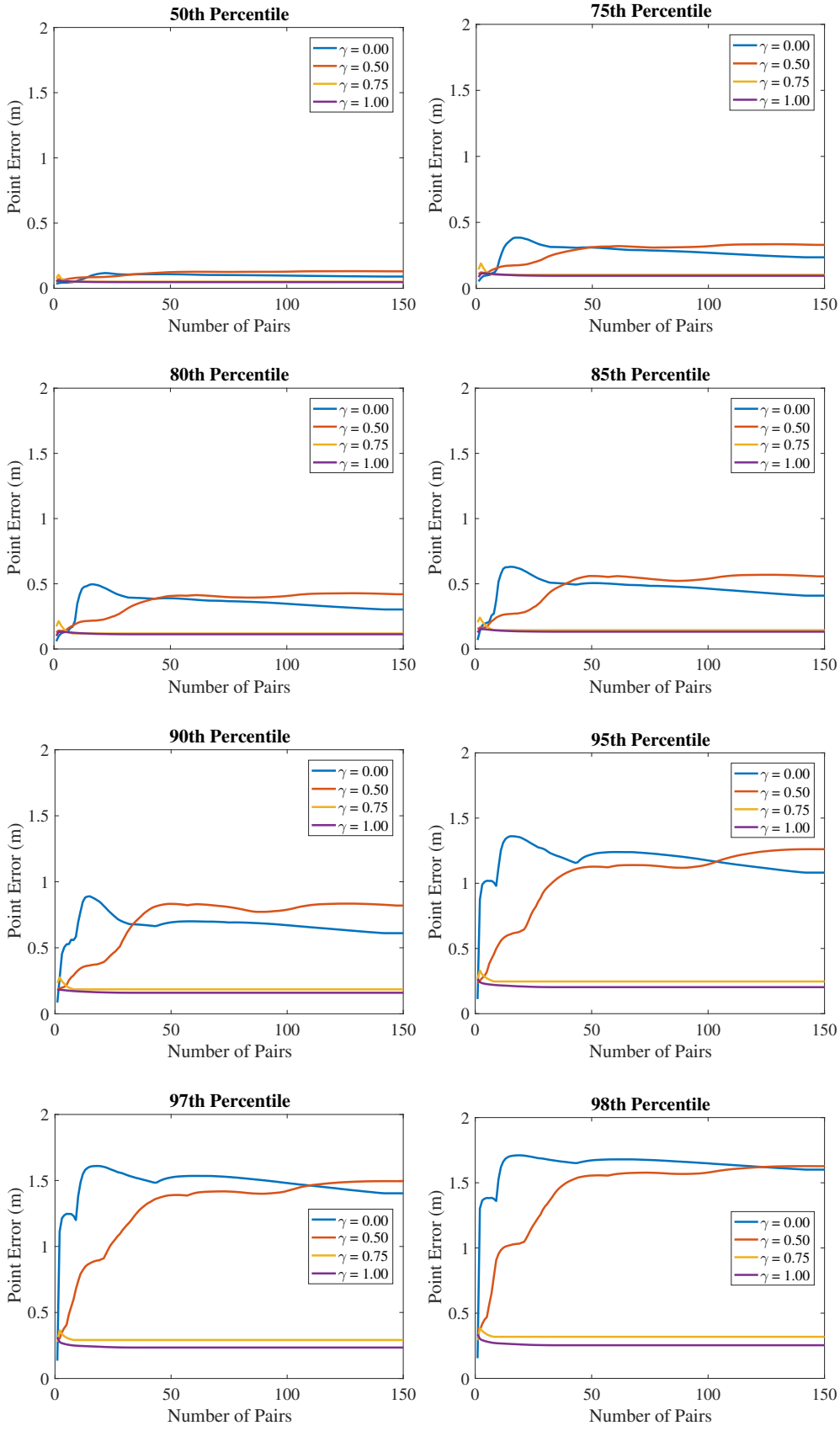


Figure 4.13: Error against number of image pairs. Higher values of γ are less noisy.

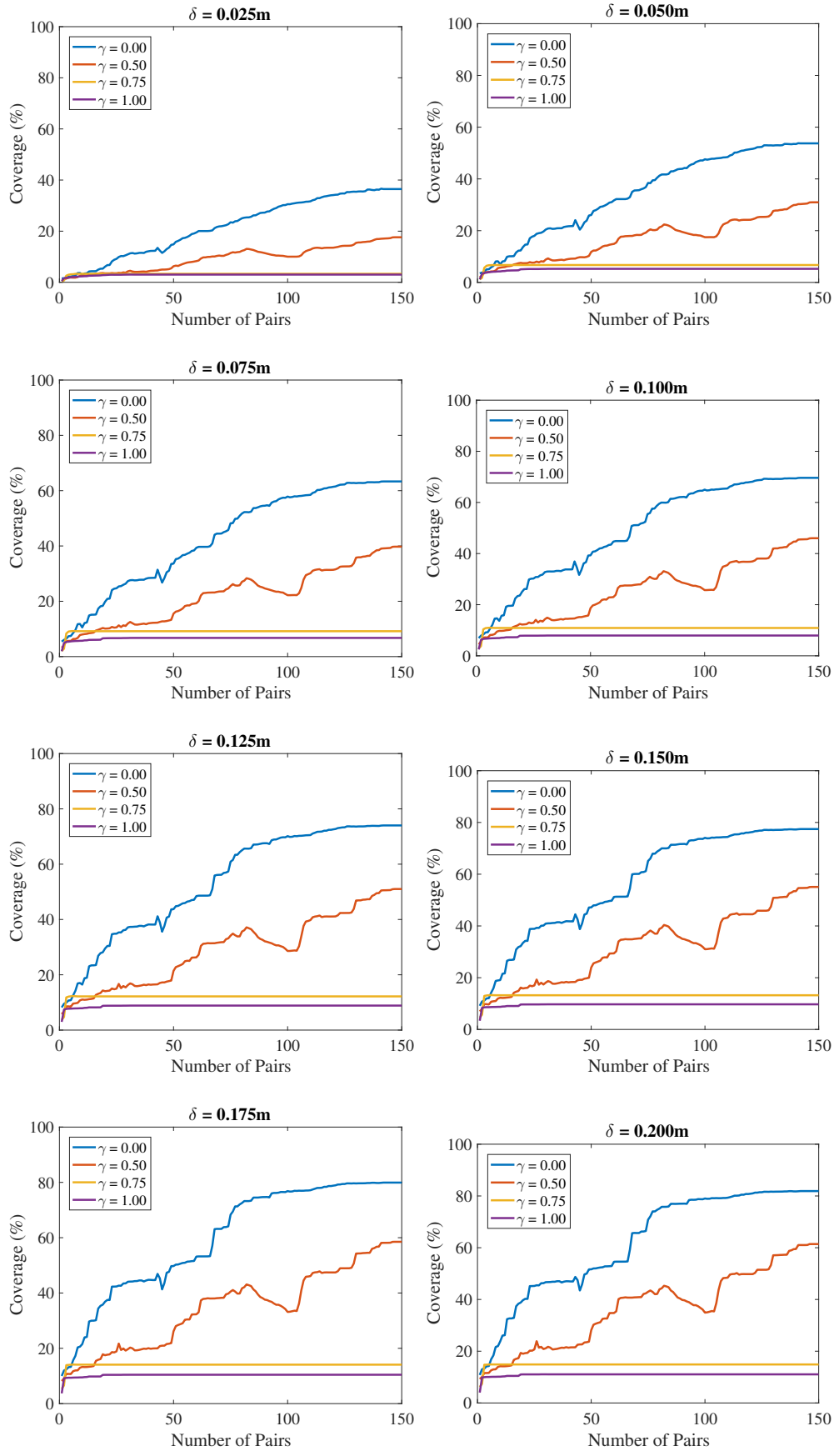


Figure 4.14: Coverage against number of image pairs. Lower values of γ are more exploratory.

can be seen that the median error (50^{th} percentile) is consistently between $5 - 15$cm which is remarkably low. This means that most points in the reconstruction are below (0.1m) away from the actual geometry. It should be noted that while (0.1m) might seem large, the scene being reconstructed is $(6.5\text{m} \times 5.4\text{m} \times 2.5\text{m})$. This implies that an error of (0.1m) represents anywhere from $1 - 4\%$ of an axis (depending on the axis). Fundamentally, an error of 1% on geometry that large means that most points in the reconstruction are accurate enough for navigation and visual inspection of the reconstruction. While higher percentile values seem to deteriorate with lower values of γ (more exploratory), it should be noted that these values are tainted by outliers due to the exploratory nature of the paths. On the higher γ experiments (more refining), this error remains low throughout.

Figure 4.14 quantitatively demonstrates that the scenic pathplanner is capable of autonomously exploring an environment. Lower values of γ make the exploration increase dramatically, which is the expected behaviour. Another expected behaviour, is that higher values of γ remain at a low coverage throughout the scene (while also keeping error low). It should be noted that $\sim 80\%$ coverage is achievable if a distance threshold of 15cm is used. This distance corresponds to the median error, which supports the notion that higher-percentile error values in Figure 4.13 are due to outliers.

4.5 Conclusion

In conclusion, this chapter presented a novel approach that can coordinate at least two or more cameras in an *opportunistically collaborative* way, creating a dense reconstruction of their environment. The NBV cost distribution is leveraged to bias a random tree-based search method toward areas of large information gain. This explores the state-space to find a *scenic path* between the camera and the NBV. The proposed method is agnostic to the source of reconstruction, and could easily be adapted to depth-sensors.

The Scenic Pathplanner selects paths which maximise information gain, both in terms of total map coverage and reconstruction accuracy. It was shown that Scenic Planning enables similar performance to state-of-the-art batch approaches using less than $< 2.7 \times 10^{-4}\%$ of the possible naive stereo pairs (3% of the views). Comparison against length-based path-planning

approaches demonstrated scenic path-planning produces more complete and more accurate maps with fewer frames. Finally, the ability of the Scenic Pathplanner to generalise to live scenarios was demonstrated by mounting cameras on autonomous ground-based sensor platforms and exploring an environment.

The Scenic Pathplanner fundamentally changes the approach of other view selection methods, such as those of Hornung *et al*. [60] and Jancosek *et al*. [64]. These approaches attempt to perform view selection for MVS reconstruction and no effort is made to move this reconstruction into a live system. By contrast, the scenic pathplanner is capable of extending view selection from the MVS domain into a live robotic scenario. This is because the scenic pathplanner is designed to operate on unoptimised reconstructions, improving over Hornung *et al*. [60], and in the absence of image, improving over Jancosek *et al*. [64]. Furthermore, the scenic pathplanner also attempts to move towards the goal in a reconstruction-friendly manner, which is a limitation of both approaches. The scenic pathplanner also fundamentally changes the way length-based pathplanning is performed. It furthers the state-of-the-art by allowing path-planning to operate on spaces that take into account higher-level constraints such as reconstruction quality and collaboration.

An important limitation of the Scenic Pathplanner is that it does not enforce global consistency. More explicitly, neither this nor the 3D reconstruction approach of the previous chapter perform any kind of optimisation. This is by design, as even the most efficient optimisations do not run in real-time and the computer resources utilised tend to scale dramatically with map size. While there are techniques to mitigate this, they usually rely on limiting the scope of the optimisation. This means that the consistency a global optimisation would enforce is fundamentally limited. Therefore, an approach that can achieve global consistency without optimisation would be extremely valuable.

Chapter 5

SeDAR: Human-Inspired Floorplan Localisation

The autonomous 3D reconstruction systems described in this thesis have implicitly relied on external localisation approaches. In Chapter 3, the Next-Best View (NBV) estimation used sets of precomputed poses, either from ground-truth or Multi-View Stereo (MVS). In Chapter 4, the scenic pathplanner relied on either a precomputed set of poses from MVS for the offline dataset or a Simultaneous Localisation and Mapping (SLAM) system for the online dataset. Unfortunately, both MVS and SLAM suffer from the same limitation: they can only ever guarantee global pose consistency *internally*. This means that while pose estimates are globally consistent, they are only valid within the context of the localisation system. There are no guarantees, at least in vision-only systems, that the reconstruction can be directly mapped to the real world, or between agents (without explicit alignment). This chapter will attempt to address these limitations with a localisation approach that is efficient, accurate and, most importantly, globally consistent with the real-world.

It should be clear that offline batch approaches are discarded *a priori* for a live system. This leaves traditional SLAM systems as the only viable approach considered so far. However, SLAM systems are liable to drift in terms of both pose and scale. They can also become globally inconsistent (even internally) in the case of failed loop closures.

This problem is normally addressed by having a localisation system that can relate the pose

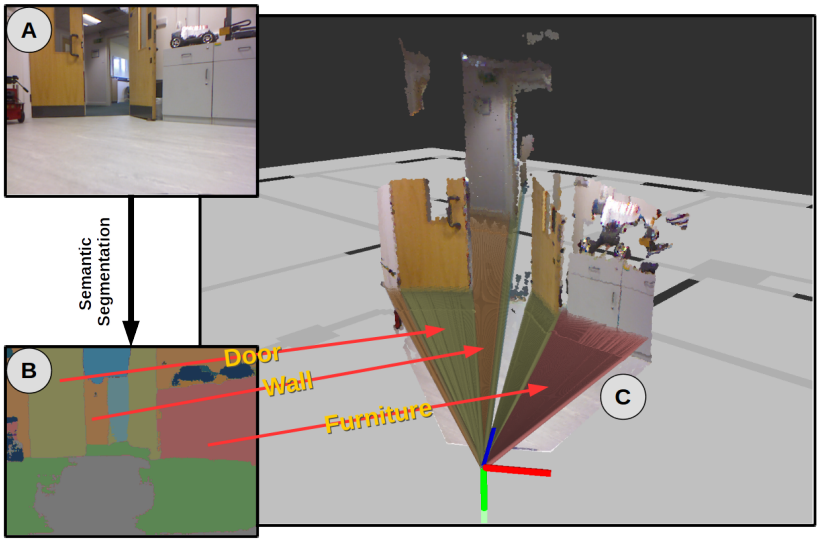


Figure 5.1: A) RGB Image, B) CNN-Based Semantic Labelling and C) Sample SeDAR Scan within floorplan.

of the robot to a pre-existing map. Examples of global localisation frameworks include the Global Positioning System (GPS) and traditional Monte-Carlo Localisation (MCL). MCL has the ability to localise within an existing floorplan (which can be safely assumed to be available for most indoor scenarios). This is a highly desirable trait, as it implicitly eliminates drift, is globally consistent and provides a way for the 3D reconstructions to be related to the real world without having to perform expensive post-hoc optimizations. Traditionally, the range-based scans required by MCL have been produced by expensive sensors such as Light Detection And Ranging (LiDAR). These sensors are capable of producing high density measurements at high rates with low noise, making them ideal for range-based MCL. However, in addition to their expense, they are large and have high power requirements which is an issue for small mobile platforms.

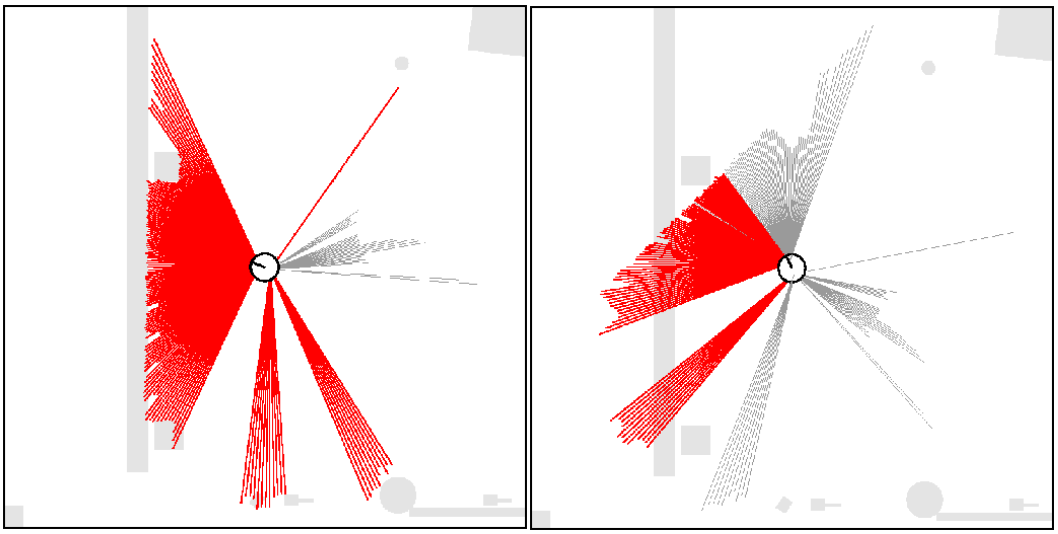
As a response to this, modern low-budget robotic platforms have used RGB-D cameras as a cheap and low-footprint alternative. This has made vision-based floorplan localisation an active topic in the literature. However, while many approaches have been proposed, they normally use heuristics to lift the 2D plan into the 3D coordinate system of SLAM. Examples include Liu *et al.* [80], who use visual cues such as Vanishing Points (VPs) or Chu *et al.* [18] who

perform piecemeal 3D reconstructions that can then be fitted back to an extruded floorplan. A common problem with these approaches is that the 3D data extracted from the image is normally orthogonal to the floorplan that it is meant to localise in. This means that assumptions must be made about dimensions not present in the floorplan. These approaches also do not fully exploit the floorplan, ignoring the semantic information.

In order to find a robust solution to MCL, inspiration can be drawn from the way humans localise within a floorplan. People do not explicitly measure depths to every visible surface and try to match them against different pose estimates in the floorplan. However, this is exactly how most robotic scan-matching algorithms operate. Similarly, humans do not extrude the 2D geometry present in the floorplan into 3D. Unfortunately, this is how most vision-based approaches localise. Humans do the exact opposite. Instead of depth, people use high level semantic cues. Instead of extruding the floorplan up into the third dimension, humans collapse the 3D world into a 2D representation. Evidence of this is that many of the floorplans used in everyday life are not strictly accurate or in 3D. Instead, floorplans designed for people opt instead for high levels of discriminative landmarks on a 2D map.

Therefore, this chapter proposes a fundamentally different approach that is inspired by how humans perform the task. Instead of discarding valuable semantic information, a Convolutional Neural Network (CNN) based encoder-decoder is used to extract high-level semantic information. All semantic information is then collapsed into 2D, in order to reduce the assumptions about the environment. A state-of-the-art sensing and localisation framework is then introduced, which uses these labels (along with image geometry and, optionally, depth) to localise within a semantically labelled floorplan.

Semantic Detection and Ranging (SeDAR) is an innovative human-inspired framework that combines new semantic sensing capabilities with a novel semantic Monte-Carlo Localisation (MCL) approach. As an example, Figure 5.1 shows a sample SeDAR scan localised in the floorplan. SeDAR has the ability to surpass LiDAR-based MCL approaches. SeDAR also has the ability to perform drift-free local, as well as global, localisation. Furthermore, experimental results show that the semantic labels are sufficiently strong visual cues such that depth estimates are no longer needed. Not only does this vision-only approach perform comparably to depth-based methods, it is also capable of coping with floorplan inaccuracies more gracefully than



(a) Correct (b) Incorrect

Figure 5.2: Laser scan matching, the robot is correctly localised when the observations match the geometry of the map [128]

strictly depth-based approaches.

This chapter describes the process by which SeDAR is used as a human-inspired sensing and localisation framework. To do this, a generic definition and formalisation of MCL is presented first. Following this, the semantically salient elements are extracted from a floorplan and an RGB image is parsed into a SeDAR scan. This chapter then presents its three main novelties. In the first, the semantic information present in the floorplan is used to define a new motion model. In the second, the SeDAR scan is used to define a novel sensor model using a combination of range and label information. In the third, an additional sensor model is presented that only depends on label information (an RGB image). Finally, this chapter presents localisation results obtained by using this approach in multiple sensing modalities.

5.1 Problem Definition

While there exist many approaches to perform MCL, Range-Based Monte-Carlo Localisation (RMCL) [140, 18] is widely considered to be the state-of-the-art localisation method for pre-existing maps. RMCL is a scan-matching algorithm, it assumes the presence of a sensor that

provides range and bearing tuples across a scanline. The problem then becomes finding the pose of the robot that makes the sensor observations match the floorplan. Figure 5.2a shows a case of the scan being correctly matched for a correctly localised robot. Conversely, Figure 5.2b shows an incorrectly matched scan for an incorrect pose.

State-of-the-art localisation performs this matching in a Sequential Monte-Carlo (SMC) [24] framework, which can be broadly summarised as follows. Firstly, there is a prediction stage where particles are propagated using a motion-model, normally odometry from the robot (with Gaussian noise). Secondly, an update phase where each particle is weighted according to how accurately the observations align to the map. Finally, a re-sampling step is performed proportional to the weight of each particle and the process is then repeated.

More formally, the current pose $\mathbf{x}_t \in \mathbb{X}_t \subset \text{SE(2)}$ can be estimated using a set of possible pose samples $\mathbb{S}_t = \left\{ s_t^i; i = 1..N \right\}$, odometry measurements $\mathbb{U}_t = \left\{ u_j; j = 1..t \right\}$, sensor measurements $\mathbb{Z}_t = \left\{ z_j; j = 1..t \right\}$ and a 2D map $\mathbb{V}$. The posterior is calculated as

$$\Pr\left(s_t^i \middle| \mathbb{Z}_t, \mathbb{U}_t\right) = \Pr\left(z_t \middle| s_t^{i\prime}, \mathbb{V}\right) \Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right) \Pr\left(s_{t-1}^i \middle| \mathbb{Z}_{t-1}, \mathbb{U}_{t-1}\right) \tag{5.1}$$

which implies that only the most recent odometry and observations are used [24]. This means that at each iteration the particles from $\Pr\left(s_{t-1}^i \middle| \mathbb{Z}_{t-1}, \mathbb{U}_{t-1}\right)$ are: propagated using a motion model $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right)$, weighted using a sensor model $\Pr\left(z_t \middle| s_t^{i\prime}, \mathbb{V}\right)$ and resampled according to the posterior $\Pr\left(s_t^i \middle| \mathbb{Z}_t, \mathbb{U}_t\right)$. Algorithm 2 describes this process in more detail.

In an MCL context, the motion model is defined by the odometry received from the robot. This propagates the particles according to u_t with Gaussian noise applied such that

$$\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right) \sim \mathcal{N}\left(u_t + s_{t-1}^i, \boldsymbol{\Upsilon}_t\right) \tag{5.2}$$

where the symbol $\sim$ implies $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right)$ is distributed as $\mathcal{N}\left(u_t + s_{t-1}^i, \boldsymbol{\Upsilon}_t\right)$, meaning Gaussian noise is applied to the linear and angular components of the odometry. Fundamentally, this allows errors in the odometry to be accounted for during particle propagation. In Section 5.3.1, the traditional definition of a motion-model is augmented to include a "ghost factor" that uses semantic information to influence how particles move through occupied space.

The sensor model is defined by each range-scanner observation. The probability of each full range-scan (z_t) can be estimated under the assumption that each measurement in the scan is

1: **function** MCL($\mathbb{S}_{t-1}, u_t, z_t$)
2: $\mathbb{S}_t = \mathbb{S}'_t = \emptyset$
3: **for** $i = 1 \rightarrow N$ **do**
4: $s_t^{i\prime} \leftarrow$ MOTION_MODEL(u_t, s_{t-1}^i)
5: $w_t^i \leftarrow$ SENSOR_UPDATE(z_t, $s_t^{i\prime}$, $\mathbb{V}$)
6: $\mathbb{S}'_t \leftarrow \mathbb{S}'_t + \langle s_t^{i\prime}, w_t^i \rangle$
7: **end for**
8: **for** $i = 1 \rightarrow N$ **do**
9: $s_t \leftarrow$ WEIGHTED_SAMPLE($\mathbb{S}'_t$)
10: $\mathbb{S}_t \leftarrow \mathbb{S}_t + s_t$
11: **end for**
12: $\bar{\mathbb{S}}_t \leftarrow$ MEAN($\mathbb{S}_t$)
13: **return** $\bar{\mathbb{S}}_t$
14: **end function**

Algorithm 2: Sequential Monte-Carlo Localisation in a known floorplan.

independent of each other. That is,

$$\Pr\left(z_t \middle| s_t^{i\prime}, \mathbb{V}\right) = \prod_{k=1}^{K} \Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) \tag{5.3}$$

is the likelihood of the putative particle $s_t^{i\prime}$, where

$$z_t = \left\{\langle \theta_t^k, r_t^k \rangle; k = 1..K\right\} \tag{5.4}$$

is the set of range and bearing tuples that make up each scan. Calculating the likelihood can be done two ways, using a beam model [132] or a likelihood field model [127].

In the beam model, a raycasting operation is performed. Starting from the pose of the current particle, a ray is cast along the bearing angle θ_t^k. The raycasting operation terminates when an occupied cell is reached and the likelihood is estimated as

$$\Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = e^{\frac{-\left(r_t^k - r_t^{k*}\right)^2}{2\sigma_o^2}} \tag{5.5}$$

where r_t^k is the range obtained from the sensor and r_t^{k*} is the distance travelled by the ray.

(a) Original (b) Likelihood Field (c) Semantic

Figure 5.3: Original floorplan compared to the likelihood field and the labelled floorplan.

In the likelihood field model, a distance map is used in order to avoid the expensive raycasting operation. The distance map is a Lookup Table (LUT) of the same size as the floorplan, where each cell contains the distance to the nearest geometry. This map is estimated similar to a Chamfer distance [10], where a search is performed in a window around each cell and the distance to the closest occupied cell in the floorplan is stored. When queried, this distance is converted into a likelihood using equation 5.5. Figure 5.3 shows the estimated distance map for a floorplan, the creation of which will be explored further in Section 5.3.2. This distance map is only estimated once during initialisation. During runtime, the endpoint of each measurement can be estimated directly from the pose, bearing and range. The probability is then simply related to the distance reported by the LUT.

The raycasting method is (strictly speaking) more closely related to the sensing modality, as the closest geometry may not lie along the ray. However, in practice, most robotics systems use the likelihood field model as it is both faster and tends to provide better results. This is because the raycasting operation can report very incorrect measurements due to small pose errors. An example of this is when looking through an open door, an error of a few centimetres can make the rays miss the door. This makes the distribution inherently less smooth.

The problem with both of these approaches is that they only use one-dimensional information. More explicitly, using only the range information from the sensors fundamentally limits how discriminative each reading can be.

Instead, this chapter presents a semantic sensing and localisation framework called SeDAR. SeDAR introduces a likelihood field model that incorporates semantically salient information into the traditional range-enabled approach. In an alternative approach, SeDAR combines the raycasting and likelihood field approaches in a novel formulation which allows localisation without range measurements. Experimental evaluation shows that SeDAR outperforms traditional RMCL when using both semantic and depth measurements. When using semantic-only measurements, it is shown that SeDAR can perform comparably to depth-enabled approaches.

5.2 Semantic Labelling and Sensing

Before using the semantic labels to aid in floorplan localisation, it is necessary to extract them. To do this, a floorplan is labelled in order to identify semantically salient elements. These salient elements are then identified in the camera of the robot by using a state-of-the-art CNN-based semantic segmentation algorithm [71].

5.2.1 Floorplan

RMCL requires a floorplan and/or previously created range-scan map that is accurate in scale and globally consistent, this presents a number of challenges. A previously created range-scan map requires a robust SLAM algorithm such as GMapping [51] to be run. This is not an ideal situation as it forces the robot to perform an initial exploration to construct a map before localisation can be performed. Moreover, the SLAM algorithm is also sensitive to noise. Furthermore, the resulting map is difficult to interpret by humans.

Instead of using a metric-accurate reconstruction, a more flexible and feasible alternative is using a human-readable floorplan. However, this would make RMCL less robust due to differences between the floorplan and what the robot can observe (*e.g.* inaccuracies, scale variation and furniture).

To overcome these issues, the localisation is augmented with semantic labels extracted from an existing floorplan. The labels are limited to walls, doors and windows. The reason for this limitation is two-fold. Firstly, they are salient pieces of information that humans naturally use

to localise. Secondly, they are simple to automatically extract from a floorplan using image processing. As can be seen in Figure 5.3c, these semantically salient elements have been colour coded to highlight the different labels.

In order to make a labelled floorplan readable by the robot, it must first be converted into an occupancy grid. An occupancy grid is a 2D representation of the world, in which each cell in the grid has an occupancy probability attached to it. Any cell that is above a threshold is then considered as being occupied. Estimating the occupancy of an existing floorplan is done by taking the normalized greyscale value from the floorplan image.

The map can then be defined as

$$\mathbb{V} = \left\{ v_{\mathbf{m}}; \mathbf{m} \in \mathbb{M} \right\} \tag{5.6}$$

where $\mathbb{M}$ is a set of integer 2D positions. Assuming $\mathcal{L} = \{a, d, w\}$ is the set of possible cell labels (wall, door, window), each cell is defined as

$$v_{\mathbf{m}} = \left\langle v^{o}_{\mathbf{m}}, v^{w}_{\mathbf{m}}, v^{d}_{\mathbf{m}}, v^{a}_{\mathbf{m}} \right\rangle \tag{5.7}$$

where $v^{o}_{\mathbf{m}}$ is the occupancy likelihood and $v^{\ell}_{\mathbf{m}}$, where $\ell \in \mathcal{L}$, denotes the label likelihood.

Having incorporated the semantic labels into the standard occupancy grid, it is now necessary to use them in sensing.

5.2.2 SeDAR Sensor

Extracting semantic labels from a robot-mounted sensor is one of the most important parts of SeDAR. It is theoretically possible to directly label range-scans from a LiDAR-based scanner. In fact, there is a wide range of landmark-based SLAM systems that use range sensors [30]. However, there are limitations on the amount of information that can be extracted from a range-scan.

Beyond the structure of the environment, the additional information contained in floorplans pertains to important architectural features (such as doors and windows). These architectural features are well defined in terms of their appearance. Therefore, they are ideally suited to semantic segmentation of the image.

In SeDAR, labels are extracted from the RGB image only. This is by design, as it allows the use of cameras that cannot sense depth. In the following Sections this sensing modality will be used in a novel MCL framework that does not require range-based measurements. However, it should be noted that SeDAR is capable of using range measurements, should they be available.

If they are used, SeDAR is completely agnostic to the source of the depth measurements. They can come from a deep learning-based depth estimation [76] or a dense Structure from Motion (SfM) system [34]. However, for the purposes of this thesis, a simple RGB-D sensor is used. Either way, the method for parsing an RGB-D image into a SeDAR scan is the same.

RGB-D to SeDAR

For a low-cost robotic system that uses an RGB-D image as a proxy for a more expensive LiDAR scanner, a depth scanline is typically extracted from the depth image as

$$z_t = \left\{ \left\langle \theta_t^k, r_t^k \right\rangle ; k = 1..K \right\}, \tag{5.8}$$

where θ_t^k is the angle along the horizontal axis and r_t^k is the corresponding range. This can be accomplished by looking exclusively at the depth image.

The angle along the horizontal axis, θ_t^k, can be calculated by

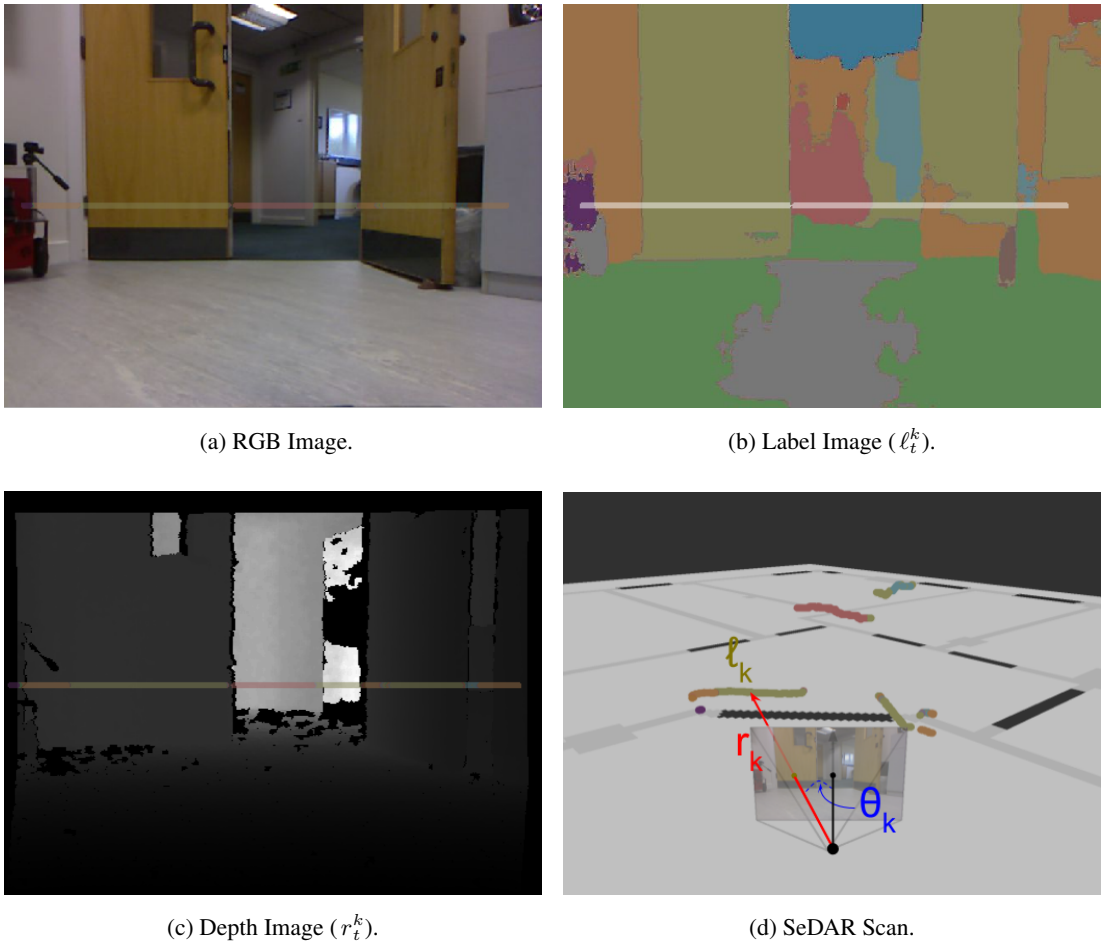
$$\theta_t^k = \operatorname{atan2}\left(\frac{u - c_x}{f_x} \right) \tag{5.9}$$

where (u, v), (c_x, c_y), (f_x, f_y) are the pixel coordinates, principal point and focal length, respectively, of the camera. While it is possible to estimate a second angle along the vertical axis, this is unnecessary in the case of floorplan localisation. More importantly, incorporating this information into the localisation framework requires assumptions to be made about the floorplan (*e.g.* ceiling height).

The range measurement r_t^k can be calculated as

$$r_t^k = \sqrt{\left(\frac{d_t^k \left(u - c_x \right)}{f_x} \right)^2 + \left(\frac{d_t^k \left(v - c_y \right)}{f_y} \right)^2 + \left(d_t^k \right)^2} \tag{5.10}$$

where d_t^k is the current depth measurement at pixel k. At this point, a traditional range-scan can be emulated. Notice that all the visible information present in the RGB image is being discarded.



(a) RGB Image.

(b) Label Image (ℓ_t^k).

(c) Depth Image (r_t^k).

(d) SeDAR Scan.

Figure 5.4: Visualisation: sensor input, semantic segmentation and the resulting SeDAR scan.

On the other hand, a SeDAR-scan consists of a set of bearing, range and label tuples,

$$z_t = \left\{ \left\langle \theta_t^k, r_t^k, \ell_t^k \right\rangle ; k = 1..K \right\}, \tag{5.11}$$

where ℓ_t^k is the semantic label.

In order to estimate the labels, CNN-based encoder-decoder network [71] is used. This is trained on the SUN3D [142] dataset, and can reliably detect doors, walls, floors, ceilings, furniture and windows. This state-of-the-art semantic segmentation runs in real-time, which allows images to be parsed into a SeDAR-scan with negligible latency. The label ℓ_t^k is then simply the label at pixel k.

Figure 5.4 shows the input images and the resulting SeDAR scan. Figure 5.4a shows the RGB

image from which the label image in Figure 5.4b is extracted. Figure 5.4c shows the depth image. In all of these, the scanline shown in the middle of the image denotes specific pixel locations where ℓ_t^k and r_t^k are extracted from the label and depth image, respectively. Finally, Figure 5.4d shows the resulting SeDAR scan, where the scanline can be seen localised within a floorplan. Now that the semantic labels are added into the map and the sensor, they can be used in a novel MCL algorithm.

5.3 Semantic Monte-Carlo Localisation

It has been shown that there is a large amount of easily-attainable semantic information present in both the floorplan and the image. This information has been largely ignored in the MCL literature in favour of range-based approaches.

In this Section, this semantic information is combined into a novel semantic MCL approach. In the motion model the semantic information is used to inform collision models. In the sensor model two approaches are presented. The first introduces a likelihood field model that incorporates semantically salient information into the traditional approach. The second approach combines the raycasting and likelihood field approaches into a method which allows localisation without range measurements.

5.3.1 Motion Model

Equation 5.2 formalised the motion model as $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right)$. However, it is well understood in the literature that the actual distribution being approximated is $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i, \mathbb{V}\right)$. This encodes the idea that certain motions are more or less likely depending on the map (*e.g.* through walls).

Under the assumption that the motion of the robot is small, it can be shown that

$$\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i, \mathbb{V}\right) = \kappa \Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right) \Pr\left(s_{t-1}^i \middle| \mathbb{V}\right) \tag{5.12}$$

(see *e.g.* [128]) where κ is a normalising factor and $\mathbb{V}$ is the set containing every cell in the map. This allows the two likelihoods to be treated independently.

In an occupancy map, the motion $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right)$ is defined in the same way as equation 5.2. The prior $\Pr\left(s_{t-1}^i \middle| \mathbb{V}\right)$ is simply the occupancy likelihood of the cell that contains s_{t-1}^i, that is

$$\Pr\left(s_{t-1}^i \middle| \mathbb{V}\right) = 1 - \Pr\left(v^o_{s_{t-1}}\right) \tag{5.13}$$

which is an elegant solution in the case where the "floorplan" was previously built by the robot.

However, this approach becomes problematic when using human-made floorplans. Human-made floorplans typically have binary edges (when they are made on a computer) or edges with image artefacts (when they are scanned into a computer). This does not reflect what the robot can observe and can cause issues with localisation. Therefore, most approaches tend to assume a binary interpretation of the occupancy. This is done by setting the probability to

$$\Pr\left(v^o_{s_{t-1}}\right) = \begin{cases} 1 & \text{if } v^o_{s_{t-1}} \geq \tau_o \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

where τ_o is a user defined threshold. While this makes depth-based methods perform reliably, it is a crude estimate of reality. For instance, most humans would not even notice if a door is a few centimetres away from where it should be. Issues like this present real problems when particles propagate though doors, as it is possible that the filter will discard particles as they collide with the edge of the door frame.

Instead, the motion model presented here uses semantic information to augment this with a *ghost factor* that allows particles more leeway in these scenarios. Therefore the proposed prior is

$$\Pr\left(s_{t-1}^i \middle| \mathbb{V}\right) = \left(1 - \Pr\left(v^o_{s_{t-1}}\right)\right) e^{-\epsilon_{\text{G}}\delta_d} \tag{5.15}$$

where δ_d is the distance to the nearest door. While other labels such as windows can be used, in the case of a ground-based robot doors are sufficient. The distance, δ_d, can be efficiently estimated using a lookup table as defined in Section 5.3.2.

More importantly, ϵ_{G} is a user defined factor that determines how harshly this penalty is applied. Setting $\epsilon_{\text{G}} = 0$ allows particles to navigate through walls with no penalty, while very high values approximate equation 5.14. The effects of ϵ_{G} will be explored in Section 5.4.4. This motion model is more probabilistically accurate than the occupancy model used in most RMCL approaches, and has the added advantage of leveraging the high-level semantic information present in the map.

Having presented a semantically enabled motion model, it is now necessary to give the sensor model the same treatment.

5.3.2 Sensor Model

The naïve way of incorporating semantic measurements into the sensor model would be to use the beam model. In this modality, the raycasting operation would provide not only the distance travelled by the ray, but also the label of the cell the ray hit. If the label of the cell and the observation match, the likelihood of that particle being correct is increased. However, this approach suffers from the same limitations as the traditional beam model: it has a distinct lack of smoothness. On the other hand, the likelihood field model is significantly smoother, as it provides a gradient between each of the cells. By contrast, the approach presented here uses a joint method that can use likelihood fields to incorporate semantic information in the presence of semantic labels. More importantly, it can also use raycasting within a likelihood field in order to operate without range measurements.

As described in Section 5.1, the likelihood field model calculates a distance map. For each cell $v_{\mathbf{m}}$, the distance to the nearest occupied cell

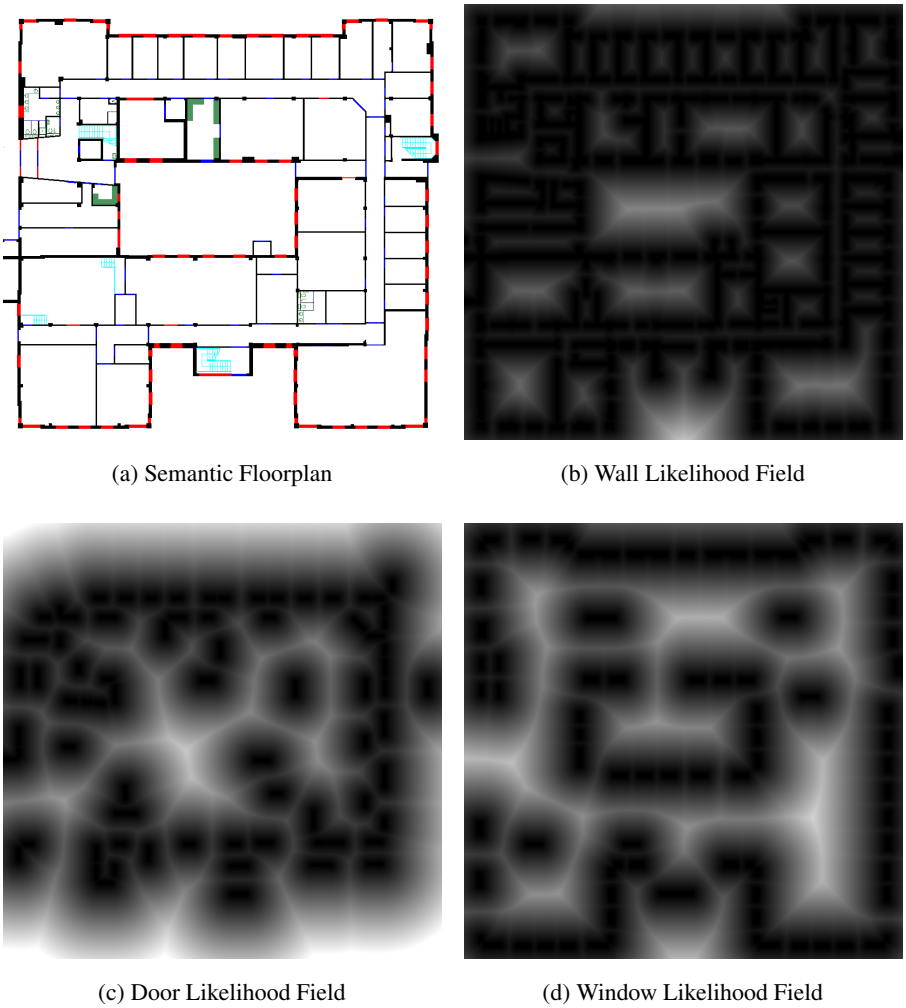
$$\delta_o(\mathbf{m}) = \min_{\mathbf{m}'} \|\mathbf{m} - \mathbf{m}'\|, \quad v^o_{\mathbf{m}'} > \tau_o \tag{5.16}$$

is calculated and stored. When a measurement $z_t^k = \langle \theta_t^k, r_t^k \rangle$ is received, the endpoint is estimated and used as an index to the distance map. Assuming a Gaussian error distribution, the weight of each particle $s_t^{i\prime}$ can then be estimated as

$$\Pr_{\text{RNG}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = e^{\frac{-\delta_o^2}{2\sigma_o^2}} \tag{5.17}$$

where δ_o is the value obtained from the distance map and σ_o is dictated by the noise characteristics of the sensor. However, this model has three main limitations. Firstly, it makes no use of the semantic information present in the map. Secondly, the parameter σ_o must be estimated by the user and assumes all measurements within a scan have the same noise parameters. Thirdly, it is incapable of operating in the absence of range measurements.

Instead, as mentioned in Section 5.2.1, this work uses the semantic labels present in the map



(a) Semantic Floorplan

(b) Wall Likelihood Field

(c) Door Likelihood Field

(d) Window Likelihood Field

Figure 5.5: Original floorplan compared to the likelihood field for each label.

to create multiple likelihood fields. For each label present in the floorplan, a distance map is calculated. This distance map stores the shortest distance to a cell with the same label.

Formally, for each map cell $v_{\mathbf{m}}$ the distance to the nearest cell of each label is estimated as

$$\delta_{\ell}(\mathbf{m}) = \min_{\mathbf{m}'} \|\mathbf{m} - \mathbf{m}'\|, \quad v^{\ell}_{\mathbf{m}'} > \tau_o \tag{5.18}$$

where $\delta_{\ell} \in \{\delta_a, \delta_d, \delta_w\}$ are distances to the nearest wall, door and window, respectively. Figure 5.5 shows the distance maps for each label. For clarity, the argument $(\mathbf{m})$, is omitted for the remainder of the thesis.

This approach overcomes the three limitations of the state-of-the-art. Firstly, the use of semantic

information [18, 24, 66, 80, 140]. Secondly, adapting the sensor noise parameters to the map [24, 66, 140]. Thirdly, operation in the absence of range measurements [6, 24, 66, 140]. These limitations will now be discussed.

Semantic Information

Most localisation approaches [18, 24, 66, 80, 140] do not use any semantic information present in the map. While approaches such as that of Bedkowski *et al*. [6] and Poschmann [105] have begun to use this information, they either rely on geometric primitives for their semantic segmentation approach ([6]) or rely on synthetic 3D reconstructions of the map ([105]). Contrary to this, SeDAR uses the semantic information present in the map. When an observation $z_t^k = \langle \theta_t^k, r_t^k, \ell_t^k \rangle$ is received, the bearing θ_t^k and range r_t^k information are used to estimate the endpoint of the scan. The label ℓ_t^k is then used to decide which semantic likelihood field to use. Using the endpoint from the previous step, the label-likelihood can be estimated similarly to equation 5.17,

$$\Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = e^{\frac{-\delta_\ell^2}{2\sigma_\ell^2}} \tag{5.19}$$

where δ_ℓ is the distance to the nearest cell of the relevant label and σ_ℓ is the standard deviation (which will be defined using the label prior). The probability of an observation given the map and pose can then be estimated as

$$\Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = \epsilon_o \Pr_{\text{RNG}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) + \epsilon_\ell \Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) \tag{5.20}$$

where ϵ_o and ϵ_ℓ are user defined weights. When $\epsilon_\ell = 0$ the likelihood is the same as standard RMCL. On the other hand, when $\epsilon_o = 0$ the approach uses only the semantic information present in the floorplan. These weights are properly explored and defined in Section 5.4.3. Unlike range scanners, σ_ℓ cannot be related to the physical properties of the sensor. Instead, this standard deviation is estimated directly from the prior of each label on the map. Defining σ_ℓ this way has the benefit of not requiring tuning. However, there is a much more important effect that must be discussed.

Semantically Adaptive Standard Deviation

Most approaches [24, 66, 140] will rely on hand-tuned parameters for the standard deviation of the observation likelihood σ_o. However, when a human reads a floorplan, unique landmarks are the most discriminative features. The more unique a landmark, the easier it is to localise using it (because there are not many areas in the map that contain it). It then follows that the more rare a landmark, the more discriminative it is for the purpose of localisation. Indeed, it is easier for a person to localise in a floorplan by the configuration of doors and windows than it is by the configuration of walls. This translates into the simple insight: *lower priors are more discriminative*. Therefore, σ_ℓ is tied to the prior of each label not only because it is one less parameter to tune, but because it implicitly makes observing rare landmarks more beneficial than common landmarks.

Relating σ_ℓ to the label prior $\Pr(\ell)$ controls how smoothly the distribution decays w.r.t. distance from the cell.

The smaller $\Pr(\ell)$ is, the smoother the decay. In essence, the localisation algorithm should be more *lenient* on sparser labels.

Range-less Semantic Scan-Matching

The final, and most important, strength of this approach is the ability to perform all of the previously described methodology in the complete absence of range measurements. Most approaches [6, 24, 66, 140] are incapable of operating without the use of range measurements. Those that are capable of range-less performance [18, 80], rely on strong assumptions about the geometry ([80]) and/or estimate a proxy for depth measurements ([18]). Both these cases have important limitations that are avoided by semantic scan-matching presented here.

This approach has so far been formalised on the assumption that either $\langle \theta_t^k, r_t^k \rangle$ tuples (existing approaches) or $\langle \theta_t^k, r_t^k, \ell_t^k \rangle$ tuples (SeDAR-based approach) were received. However, this approach is capable of operating directly on $\langle \theta_t^k, \ell_t^k \rangle$ tuples. In other words, depth measurements are *explicitly* not added or used.

Incorporating range-less measurements is simple. The beam and likelihood field models are combined in a novel approach that avoids the degeneracies that would happen in traditional

RMCL approaches. In equation 5.5 the likelihood of a ray is estimated using the difference between the range (r_t^k) obtained from the sensor and the range (r_t^{k*}) obtained from the raycasting operation. Unfortunately, in the absence of a range-based measurement (r_t^k) this is impossible. Using the standard distance map is also impossible, since the endpoint of the ray cannot be estimated. Using raycasting in the distance map also fails similarly: the raycasting terminates on an occupied cell, implying $\delta_o = 0$ for every ray cast.

On the other hand, the semantic likelihood fields can still be used as δ_ℓ will still have a meaningful and discriminative value. This operation is called semantic raycasting. For every $z_t^k = \langle \theta_t^k, \ell_t^k \rangle$, the raycasting is performed as described in Section 5.1. However, instead of comparing r_t^k and r_t^{k*} or using δ_o, the label ℓ_t^k is used to decide what likelihood field to use. The cost can then be estimated as

$$\Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = \Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) \tag{5.21}$$

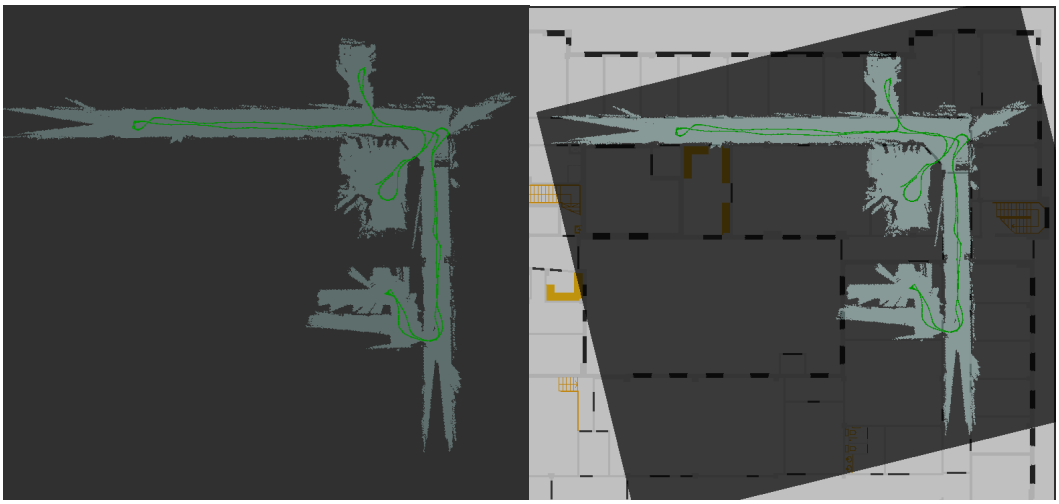
where $\Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right)$ is defined in equation 5.19. This method is essentially a combination of the beam-model and the likelihood field model.

It would be possible to assign binary values (*i.e*. label matches or not) to equation 5.21. This approach would make the observation likelihood directly proportional to the *angular distribution* of labels (*i.e*. how closely the bearing/label tuples match the observation). However, this would be a naïve solution that provides no smooth gradient to the correct solution. Instead, this approach uses the *angular distribution* of labels, combined with distances from the likelihood field, to provide a smooth cost-function that converges reliably.

The previous Sections have presented a series of methods to localise a ground-based robot on a pre-existing floorplan. In the following Section, it will be shown that these methods are capable of outperforming standard RMCL approaches when using range-measurements. Moreover, it will be shown that they provide comparable performance when operating exclusively on bearing/label tuples.

5.4 Evaluation

To summarise, this chapter has presented several important concepts. The idea of a semantic floorplan was introduced. The semantic floorplan contains information that is salient to humans.



(a) Ground Truth (b) Overlay to Floorplan

Figure 5.6: Sample Trajectory used for evaluation.

A new sensing approach was also introduced. SeDAR, adds semantic labels to the traditional LiDAR information. The semantic floorplan and SeDAR-scan were combined into a novel SeDAR-based MCL approach. This approach is capable of using the semantic information present in the map to define a new motion model. It is also capable of using the labels from a CNN-based segmentation to localise within the map. The presented SeDAR-based approach can do all of the above both in the presence, and absence, of range measurements. This section will demonstrate that SeDAR-based MCL is capable of reliably out-performing the state-of-the-art when using range measurements. It will also show that this approach is capable of comparable performance even in the absence of range measurements.

Firstly, the experimental setup is described. This consists of creating a dataset of a trajectory within a floorplan, as well as establishing error metrics. Secondly, a comparison of several approaches is performed. The comparison is done in terms of localisation accuracy, for either coarse (room-level) or global initialisation. Finally, a parameter exploration is performed.

5.4.1 Experimental Setup

In order to evaluate this approach, a dataset that has several important characteristics is required. The dataset should consist of a robot navigating within a human-readable floorplan. Human-

readability is required to ensure semantic information is present. The trajectory should be captured with an RGB-D camera. This is in order to easily extract all the possible tuple combinations (range, bearing and label). Finally, the trajectory of the robot should be on the same plane as the floorplan. Unfortunately, most of the MCL datasets in the literature do not contain a floorplan, opting instead for range-scan maps. RGB-D SLAM datasets are more appropriate, but they either do not move on the floorplan plane or simply do not contain ground-truth trajectories.

Therefore, it is necessary to create a new dataset for the purpose of evaluation. The floorplan in Figure 5.3a is used because it is large enough to provide multiple trajectories with no overlap. The dataset was collected using the popular TurtleBot platform [49], as it has a front-facing Kinect that can be used for emulating both LiDAR and SeDAR.

Normally, the ground-truth trajectory for floorplan localisation is either manually estimated (as in [140]) or estimated using Motion Capture (MoCap) systems (as in [123]). However, both of these approaches are limited in scope. Manual ground-truth estimation is time-consuming and impractical. MoCap is expensive, difficult to calibrate, and normally cannot remain in the public areas required for floorplan localisation. In order to overcome these limitations, well established RGB-D SLAM systems are used instead. The excellent approach by Labbe *et al*. [75] provides very accurate pose estimation in complex environments. While it does not localise within a floorplan, it does provide an accurate reconstruction and trajectory for the robot, which can then be registered with the floorplan. Figure 5.6a shows a sample trajectory and map estimated by [75], while Figure 5.6b shows them overlaid on the floorplan.

To quantitatively evaluate SeDAR against ground truth, the Absolute Trajectory Error (ATE) metric presented by Sturm *et al*. [123] is used. ATE is estimated by first registering the two trajectories using the closed form solution of Horn [58], who finds a rigid transformation ${}^{\mathbb{G}}\mathbf{T}_{\mathbb{X}}$ that registers the trajectory $\mathbb{X}_t$ to the ground truth $\mathbb{G}_t$. At every time step t, the ATE can then be estimated as

$$e_{\mathbf{g}} = \mathbf{g}_t^{-1}\, {}^{\mathbb{G}}\mathbf{T}_{\mathbb{X}}\, \mathbf{x}_t \tag{5.22}$$

where $\mathbf{g}_t \in \mathbb{G}_t$ and $\mathbf{x}_t \in \mathbb{X}_t$ are the current time-aligned poses of the ground truth and estimated trajectory, respectively. The Root Mean Square Error (RMSE), mean and median values of this error metric are reported, as these are indicative of performance over coarse

room-level initialisation. In order to visualise the global localisation process, the error of each successive pose is shown (error as it varies with time). These metrics are sufficient to objectively demonstrate the systems ability to globally localise in a floorplan, while also being able to measure room-level initialisation performance.

The work presented here is compared against the extremely popular MCL approach present in the Robot Operating System (ROS), called Adaptive Monte Carlo Localisation (AMCL) [24]. AMCL is the standard MCL approach used in the robotics community. Any improvements over this approach are therefore extremely valuable. Furthermore, Adaptive Monte Carlo Localisation (AMCL) [24] is considered to be the state-of-the-art and is representative of the expected performance of the RMCL approaches detailed in Section 2.4.1, such as Kanai *et al*. [66], Bedkowski *et al*. [6], Winterhalter *et al*. [140] and Chu *et al*. [18]. In all experiments, any overlapping parameters (such as σ_o) are kept the same. The only parameters varied are ϵ_ℓ, ϵ_o and ϵ_{G}.

5.4.2 Coarse Room-Level Initialisation

For this evaluation, a room-level initialisation is given to both AMCL and the proposed approach. This means that the uncertainty of the pose estimate, roughly corresponds to telling the robot what room in the floorplan it is in. More explicitly, the standard deviations on the pose estimate are of $2.0m$ in (x, y) and $2.0rad$ in θ. The systems then ran with a maximum of 1000 particles (minimum 250) placed around the covariance ellipse. The error is recorded as each new image in the dataset is added.

Quantitative Results

Figure 5.7 compares four distinct scenarios against AMCL. Of these four scenarios, two use the range measurements from the Microsoft Kinect (blue lines) and two use only the RGB image (red lines).

The first range-enabled scenario uses the range measurements to estimate the endpoint of the measurement (and therefore the index in the distance map) and sets the range and label weights to ($\epsilon_o = 0.0$ and $\epsilon_\ell = 1.0$, respectively. This means that while the range information is used to

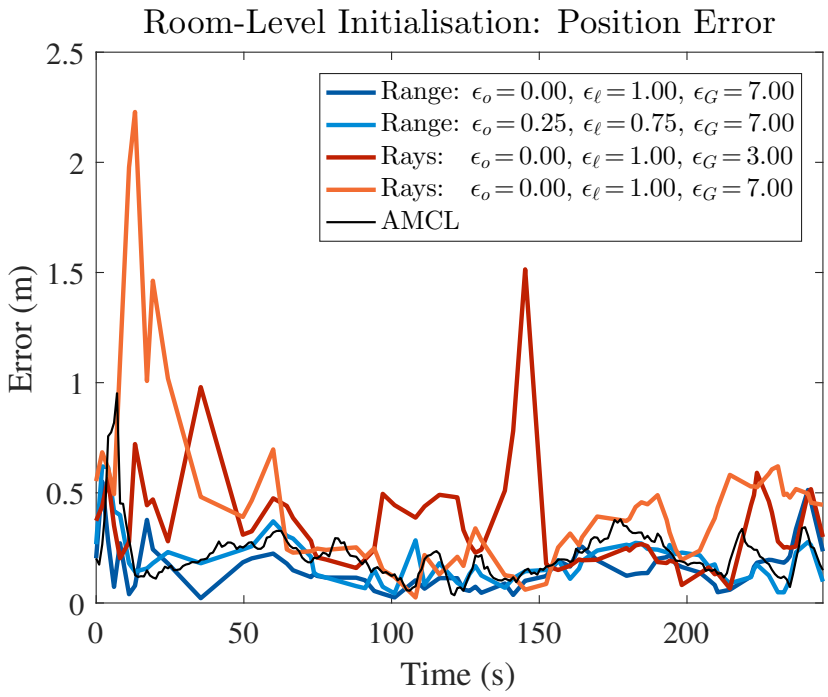


Figure 5.7: Semantic Floorplan Localisation, room-level initialisation.

inform the lookup in the distance map, the costs are only computed using the labels. The second range-enabled scenario performs a weighted combination ($\epsilon_o = 0.25, \epsilon_\ell = 0.75$) of both the semantic and traditional approaches.

In terms of the ray-based version of this approach, equation 5.21 is used. This means there are no parameters to set. Instead, a mild ghost factor ($\epsilon_G = 3.0$) and a harsh one ($\epsilon_G = 7.0$) are shown.

Since coarse room-level initialisation is an easier problem than global initialisation, the advantages of the range-enabled version of this approach are harder to see compared to state-of-the-art. However, it is important to note how closely the ray-based version of the approach performs to the rest of the scenarios despite using no depth data. Apart from a couple of peaks, the ray-based method essentially performs at the same level as AMCL. This becomes even more noticeable in Table 5.1, where it is clear that range-based semantic MCL (using only the labels) outperforms state of the art, while the ray-based $\epsilon_G = 3.0$ version lags closely behind. The reason $\epsilon_G = 3.0$ performs better than $\epsilon_G = 7.0$ is because small errors in the pose can cause the robot to "clip" a wall as it goes through the door. Since $\epsilon_G = 3.0$ is more lenient on these scenarios, it is able to outperform the harsher ghost factors. This relationship will be explored further in Section 5.4.4.

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL	0.24	0.21	0.20	0.11	0.04	0.95
Range (Label Only)	**0.19**	**0.16**	**0.14**	**0.10**	**0.02**	**0.55**
Range (Combined)	0.22	0.19	0.17	0.11	0.04	0.62
Rays ($\epsilon_{\text{G}} = 3.0$)	0.40	0.34	0.27	0.22	0.07	1.51
Rays ($\epsilon_{\text{G}} = 7.0$)	0.58	0.45	0.38	0.37	0.02	2.23

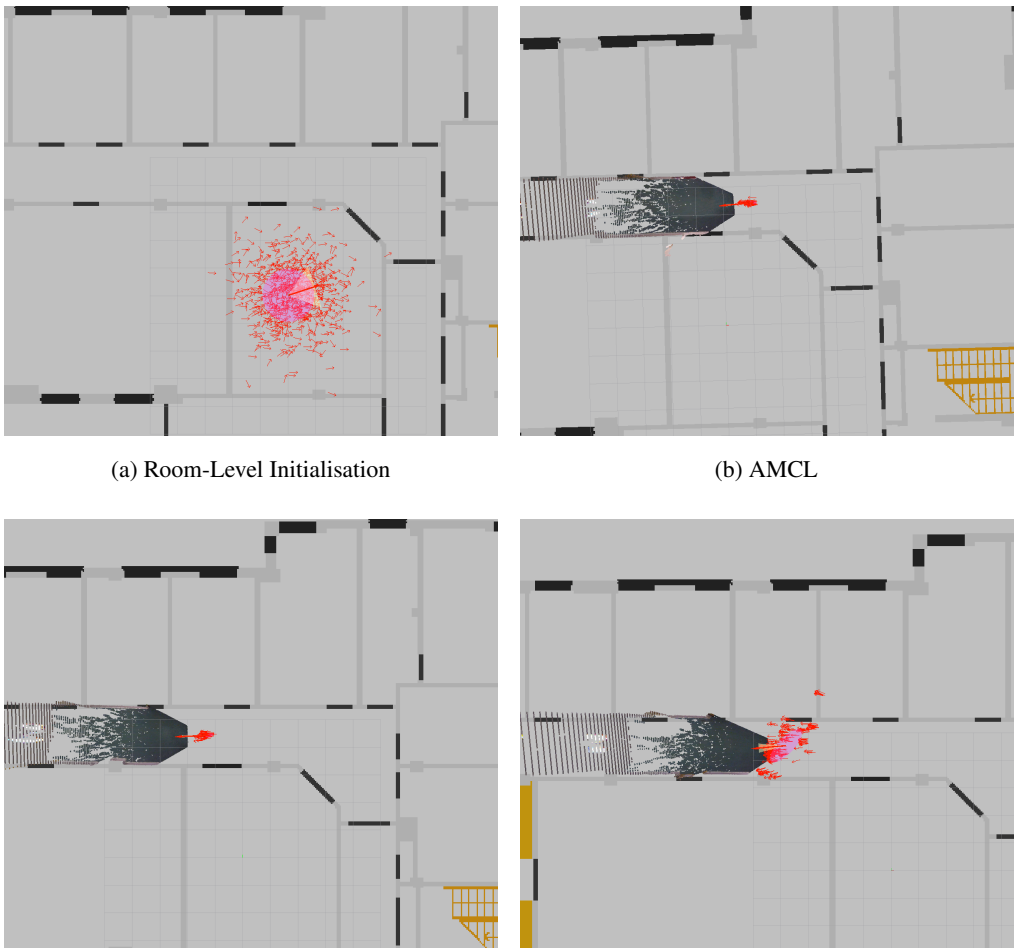
Table 5.1: Room-Level Initialisation

In order to give context to these results, the results of state-of-the-art approaches by Winterhalter *et al*. [140] and Chu *et al*. [18] are mentioned here. These approaches are chosen as they present the most comparable methods in the literature. Although direct comparison is not possible (due to differences in the approach, and the availability of code and datasets) an effort has been made to present meaningful metrics. Winterhalter *et al*. [140] report (on their paper) an error of $0.2 - 0.5$m on a much smaller room-sized dataset. While they perform experiments on larger floorplan-level datasets, the errors reported are much noisier ranging between $0.2 - 0.5$m on the coarse initialisation and $0.2 - 3$m on the global initialisation. Chu *et al*. [18] report (on their paper) a mean error of 0.53m on the TUMindoor dataset [63], which is similar to the one presented here. These results present some evidence that the SeDAR-based localisation approach can outperform the state-of-the-art localisation approaches.

Qualitative Results

In terms of qualitative evaluation, both the convergence behaviour and the estimated path are shown.

The convergence behaviour can be seen in Figure 5.8. Here, Figure 5.8a shows how the filter is initialised to roughly correspond to the room the robot is in. As the robot starts moving, it can be seen that AMCL (5.8b), the range-based version of SeDAR (5.8c) and the ray-based version (5.8d) converge. Notice that while the ray-based approach has a predictably larger variance on the particles, the filter has successfully localised. This can be seen from the fact that the



(a) Room-Level Initialisation

(b) AMCL

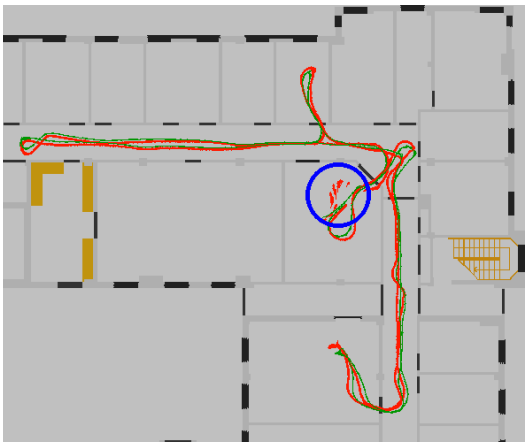
(c) SeDAR (Range-Based)

(d) SeDAR (Ray-Based)

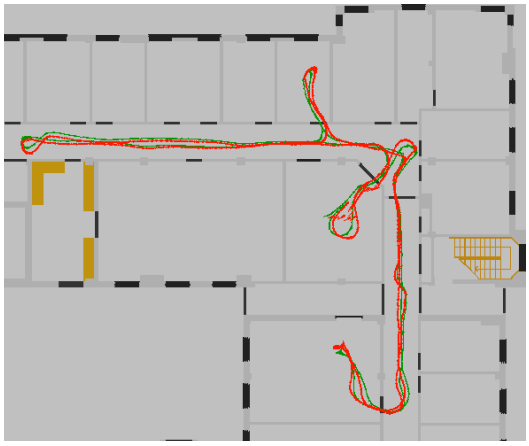
Figure 5.8: Qualitative view of Localisation in different modalities.

Kinect point cloud is properly aligned with the floorplan. It is important to note that although the Kinect point cloud is present for visualisation in the ray-based method, the depth is *not* used.

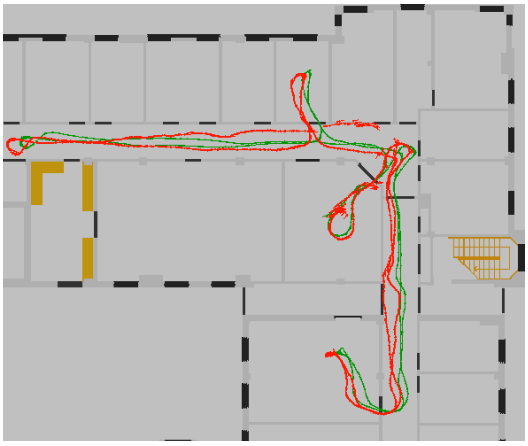
The estimated paths can be seen in Figure 5.9, where the red path is the estimated path and green is the ground truth. Figure 5.9a shows the state-of-the-art, which struggles to converge at the beginning of the sequence (marked by a blue circle). It can be seen that the range-based approach in Figure 5.9b (combined label and range), converges more quickly and maintains a similar performance to AMCL. It only slightly deviates from the path at the end of the ambiguous corridor on the left, which also happens to AMCL. It can also be seen that the ray-based approach performs very well. While it takes longer to converge, as can be seen by the



(a) AMCL



(b) SeDAR (Range-Based) Path



(c) SeDAR (Ray-Based) Path

Figure 5.9: Estimated path from coarse room-level initialisations.

estimated trajectory in Figure 5.9c, it corrects itself and only deviates from the path in areas of large uncertainty (like long corridors).

These experiments show that SeDAR-based MCL is capable of operating when initialised at the coarse room-level. It is now important to discuss how discriminative SeDAR is when there is no initial pose estimate provided to the system.

5.4.3 Global Initialisation

For this Section, the focus will be on the ability of SeDAR-based MCL to perform global localisation. In these experiments, the system is given no indication of where in the map the

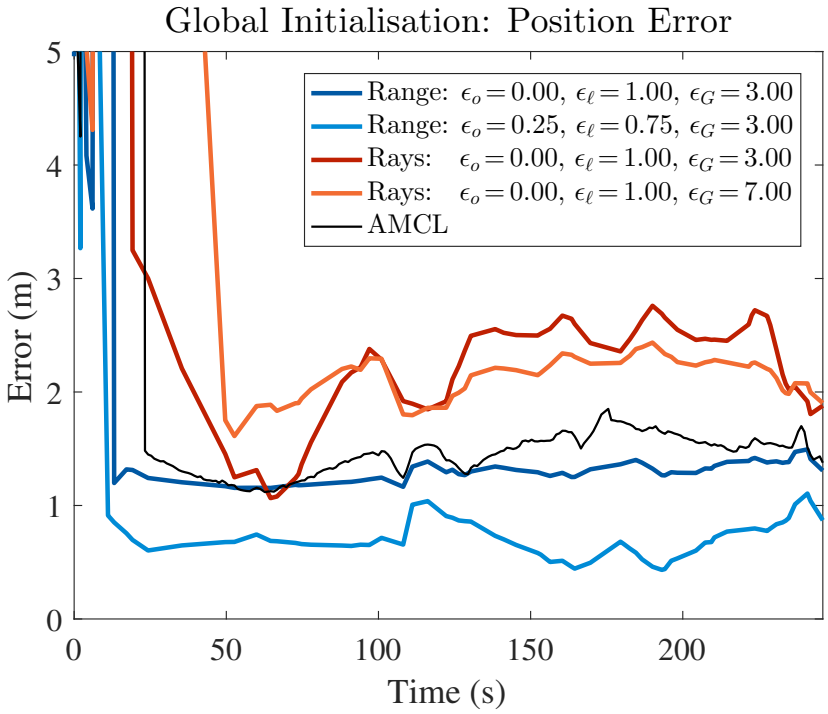


Figure 5.10: Semantic Floorplan Localisation, global initialisation.

robot is. Instead, a maximum of $50,000$ particles (minimum $15,000$) are placed over the entire floorplan.

Quantitative Results

Figure 5.10 shows the same four scenarios as in the previous Section. For the range-based scenarios (blue lines) it can be seen that using only the label information ($\epsilon_o = 0.0, \epsilon_\ell = 1.00$) consistently outperforms the state of the art, both in terms of how quickly the values converge to a final result and the actual error on convergence. This shows that SeDAR used in an MCL context is more discriminative than standard occupancy maps in RMCL. The second range-based measurement ($\epsilon_o = 0.25, \epsilon_\ell = 0.75$) significantly outperforms all other approaches. This is probably because, in principle, the occupancy maps can be considered another "label" in the semantic floorplan. This makes sense because setting $\epsilon_o = 0.25$ is equivalent to weighting all labels equally, as it is a third of $\epsilon_\ell = 0.75$ which is the weight of 3 labels.

In terms of the ray-based version of the approach (red lines), two scenarios are compared. A mild ghost factor ($\epsilon_G = 3.0$) and a harsh one ($\epsilon_G = 7.0$). These versions of the approach both provide

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL	7.31	2.26	**0.20**	6.95	**0.028**	35.45
Range (Label Only)	6.71	2.59	1.31	6.20	1.15	38.60
Range (Combined)	**4.78**	**1.69**	0.69	**4.47**	0.43	**31.19**
Rays ($\epsilon_{\text{G}} = 3.0$)	7.74	4.36	2.46	6.40	1.07	27.55
Rays ($\epsilon_{\text{G}} = 7.0$)	8.09	4.49	2.22	6.73	1.61	28.47

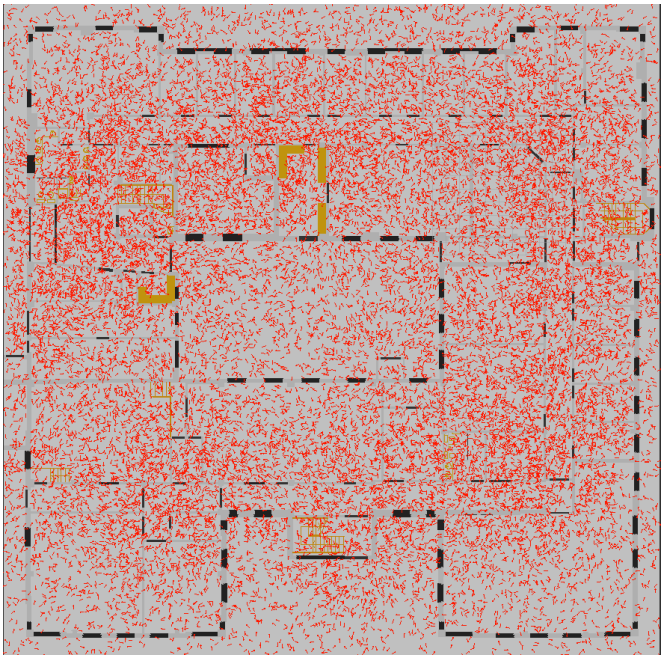
Table 5.2: Global Initialisation

comparable performance to the state-of-the-art. It is important to emphasise that this approach uses absolutely no range and/or depth measurements. As such, comparing against depth-based systems is inherently unfair. Still, SeDAR ray-based approaches compare favourably to AMCL. In terms of convergence, the mild ghost factor $\epsilon_{\text{G}} = 3.0$ gets to within several meters accuracy even quicker than AMCL, at which point the convergence rate slows down and is overtaken by AMCL. The steady state performance is also comparable. While the performance temporarily degrades, it manages to recover and keep a steady error rate throughout the whole run. On the other hand, the harsher ghost factor $\epsilon_{\text{G}} = 7.0$ takes longer to converge, but remains steady and eventually outperforms the milder ghost factor. Table 5.2 shows the RMSE, error along with other statistics.

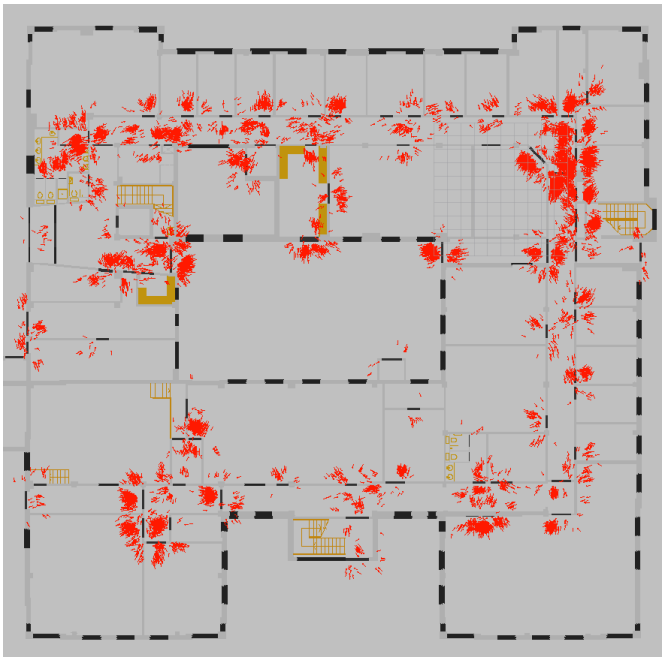
Qualitative Results

Similar to the previous Section, qualitative analysis can be provided by looking at the convergence behaviour and the estimated paths.

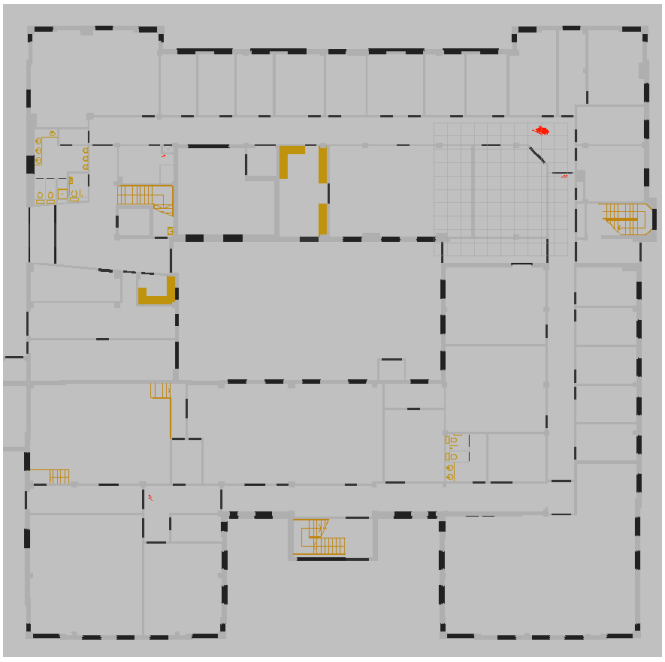
In order to visualise the convergence behaviour, Figure 5.11a shows a series of time steps during the initialisation of the filters. On the first image, the particles have been spread over the ground floor of a $(49\text{m} \times 49\text{m})$ office area. In this dataset, the robot is looking directly at a door during the beginning of the sequence. Therefore, in Figure 5.11b the filter converges with particles looking at doors that are a similar distance away. The robot then proceeds to move through the doors. Going through the door makes the filter converge significantly faster as it implicitly uses



(a) Global Initialisation



(b) Looking at Doors

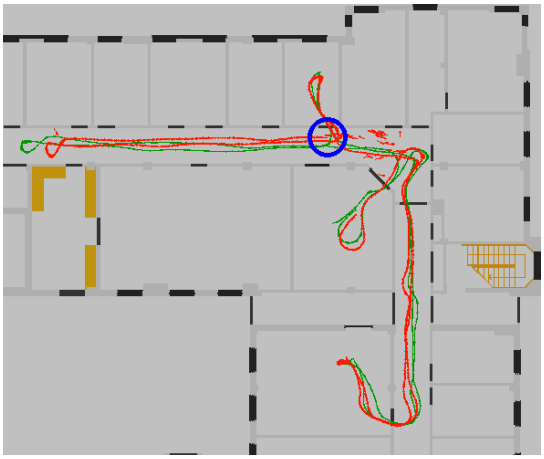


(c) Converged

Figure 5.11: Qualitative view of Localisation in different modalities.

(a) AMCL Path

(b) SeDAR (Range-Based) Path

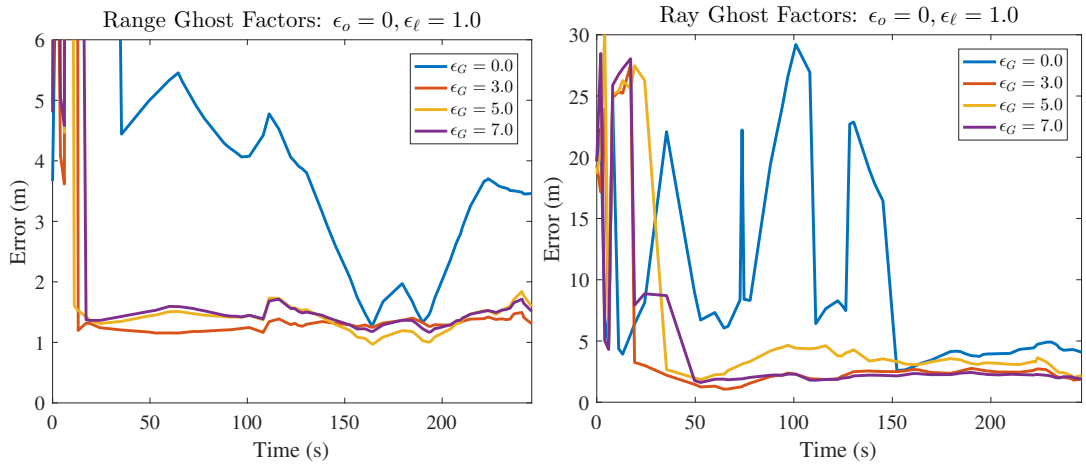


(c) SeDAR (Ray-Based) Path

Figure 5.12: Estimated path from global initialisations.

the ghost factor in the motion model. It also gives the robot a more unique distribution of doors (on a corner), which makes the filter converge quickly. This is shown in Figure 5.11c, where the filter converges.

The estimated paths can be seen in Figure 5.12, where the blue circle denotes the point of convergence. It can be seen that AMCL takes longer to converge (further away from the corner room) than the range-based approach. More importantly, it can be seen that the range-based approach suffers no noticeable degradation in the estimated trajectory over the room-level initialisation. On the other hand, the performance of the ray-based method degrades more noticeably. This is because the filter converges in a long corridor with ambiguous label



(a) Range-Based (b) Ray-Based

Figure 5.13: Different ghost factors (ϵ_G), global initialisation.

distributions (doors left and right are similarly spaced). However, once the robot turns around the system recovers and performs comparably to the range-based approach.

As mentioned previously, entering or exiting rooms helps the filter converge because it can use the ghost factor in the motion model. The following experiments, evaluate how the ghost factor affects the performance of the approach.

5.4.4 Ghost Factor

The effect of the ghost factor can be measured in a similar way to the overall filter performance. Results show that the ghost factor provides more discriminative information when it is *not* defined in a binary fashion. This is shown in the label-only scenario for both the range-based and ray-based approaches, in both the global and coarse room-level initialisation.

Global Initialisation

Figure 5.13 shows the effect of varying the ghost factor during global initialisation. It can be seen that not penalising particles going through walls, ($\epsilon_G = 0$), is not a good choice. This makes sense, as there is very little to be gained from allowing particles to traverse occupied cells without any consequence. It follows that the ghost factor should be set as high as possible.

Average Trajectory Error (RMSE)			
Ghost Factor (ϵ_{G})	**Range (Labels)**	**Range (Weighted)**	**Rays**
0.0	10.88	10.13	11.71
3.0	**6.71**	**4.78**	**7.74**
5.0	6.97	6.30	9.54
7.0	7.19	6.10	8.09

Table 5.3: Global ATE for Different Ghost Factors

However, setting the ghost factor to a large value ($\epsilon_{\text{G}} = 7.0$), which corresponds to reducing the probability by 95% at 0.43m, does not provide the best results.

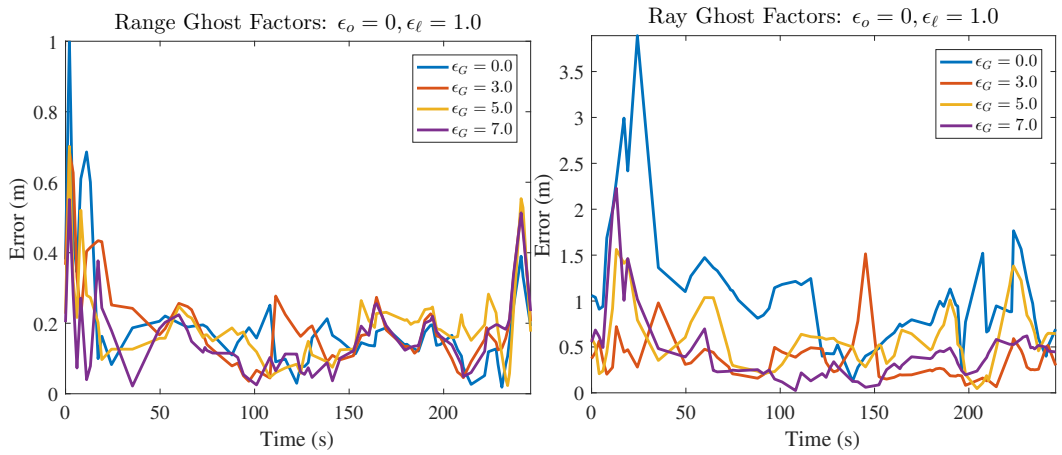
While it might seem intuitive to assume that a higher ϵ_{G} will always be better, this is not the case. High values of the ghost factor correspond to a binary interpretation of occupancy which makes MCL systems unstable in the presence of discrepancies between the map and the environment. This happens because otherwise correct particles can clip door edges and be completely eliminated from the system. A harsh ghost factor also exacerbates problems with limited number of particles. In fact, $\epsilon_{\text{G}} = 3.0$, corresponding to a 95% reduction at 1.0m, consistently showed the best results in all of the global initialisation experiments, as can be seen in Table 5.3.

Coarse Room-Level Initialisation

In terms of room-level initialisation, having an aggressive ghost factor is more in line with the initial intuition. Table 5.4 shows that for both of the range-based scenarios, $\epsilon_{\text{G}} = 7.0$ provides the best results. This is because coarse room-level initialisation in the presence of range-based measurements is a much easier problem to solve. As such, the problem of particles "clipping" edges of doors is a smaller issue.

On the other hand, the ray-based scenario still prefers a milder ghost factor of $\epsilon_{\text{G}} = 3.0$. In this scenario, inaccuracies in both the map and the sensing modalities allow for otherwise correct particles to be heavily penalised by an aggressive ghost factor. Both of these results are reflected in Figures 5.14a and 5.14b.

These results allow a single conclusion. The ghost factor must be tuned to the expected amount



(a) Range-Based (b) Ray-Based

Figure 5.14: Different ghost factors (ϵ_G), coarse room-level initialisation.

Average Trajectory Error (RMSE)			
Ghost Factor (ϵ_G)	**Range (Labels)**	**Range (Weighted)**	**Rays**
0.0	0.25	0.27	1.20
3.0	0.24	0.25	**0.40**
5.0	0.22	0.24	0.70
7.0	**0.19**	**0.22**	0.58

Table 5.4: Room-Level ATE for Different Ghost Factors

of noise in the map and sensing modality. Aggressive ghost factors can be used in cases where the pre-existing map is accurate and densely sampled, such as the case where the map was collected by the same sensor being used to localise (*i.e.* SLAM). On the other hand, in the case where there are expected differences between what the robot is able to observe (*e.g.* furniture, scale errors, etc.), it is more beneficial to provide a milder ghost factor in order to be more lenient on small pose errors.

5.4.5 Timing

The approach presented here makes the conscious decision to collapse the 3D world into a 2D representation. This has very noticeable effects to the computational complexity, and therefore speed, of the approach.

The speed of this approach was evaluated on a machine equipped with an Intel Xeon X5550 (2.67GHz) and an NVidia Titan X (Maxwell). OpenMP was used for threading expensive for-loops (such as the raycasting). During room-level initialisation, or once the system has converged, the approach can run with 250 particles in 10ms, leaving more than enough time to process the images from the Kinect into an SeDAR scan. Transforming the RGB images into semantic labels is the most extensive operation, taking on average 120ms. This means that a converged filter can run at $8 - 10$ fps. When performing global localisation, this approach can integrate a new sensor update, using $50,000$ particles, in 2.25 seconds. This delay does not impact the ability of the system to converge, as most MCL approaches require motion between each sensor integration, meaning that the effective rate is much lower than the sensor output.

5.5 Conclusion

In conclusion, this work has presented a novel approach that is capable of localising a robotic platform within a known floorplan using human-inspired techniques. First, the semantic information that is naturally present and salient in a floorplan was extracted. The first novelty was using the semantic information present in a standard RGB image to extract labels and present them as a new sensing modality called SeDAR. The semantic information present in the floorplan and the SeDAR scan were then used in a SeDAR-based MCL approach. This approach then presented three main novelties. In the first, the semantic information present in the floorplan was used to define a novel motion model for MCL. In the second, the SeDAR scan was used to localise in a floorplan using a combination of range and label information. In the third, SeDAR was used in the absence of range data to localise in the floorplan using only an RGB image.

These novelties present an important step forward for the state-of-the-art of MCL, and therefore localisation in general. Not only is this work capable of removing the requirement of expensive depth sensors [24, 43], it also has the ability to improve the performance of localisation approaches that use depth sensors [140]. When compared against the state-of-the-art monocular approaches [18, 98], leveraging the semantic information present in an RGB image allows less accurate maps to be used, as there is other information present in the map. Taken together, these contributions open the door for the usage of maps designed for human use. This implies that localisation as a discrete process to reconstruction becomes a viable alternative, as pre-existing

floorplans can be used to localise while the 3D structure is reconstructed. These, along with other [105, 80, 137], advances make it clear that the use of semantic information to aid localisation is the next step for the field.

Chapter 6

Conclusions and Future Work

This thesis was motivated by the recent rise in mainstream robotic agents, and the desire to make them more pro-active in their behaviours. More explicitly, advances in 3D reconstruction from the vision community were seen as a tool to be used by robotic agents in order to increase their capabilities. Therefore, the aim of this thesis was to explore methods that will enable modern robotic systems to autonomously and collaboratively generate an understanding of the world. This meant that there were four main objectives for the thesis:

1. To provide quick and efficient 3D reconstruction methods.
2. To develop techniques for autonomous decision-making and exploration.
3. To explore emergent behaviours for collaboration.
4. To explore the utility of semantic information to the reconstruction process.

This chapter will assert that these objectives have been achieved successfully and implemented across several different robotic platforms. Completing these objectives allowed this thesis to advance the state-of-the-art in multiple fields. The general contributions of this thesis to the fields of 3D reconstruction, goal estimation, path-planning and localisation will now be discussed. This will be followed by a per-chapter breakdown of the contributions.

The field of 3D reconstruction has traditionally been tied to Simultaneous Localisation and Mapping (SLAM)-based approaches [16, 21, 75, 95] or Multi-View Stereo (MVS)-based approaches [8, 46, 47]. This is because the reconstruction is dependent on pose estimation. The

work presented in this thesis breaks this assumption by allowing low-level pose estimation [72] to drive high-level dense reconstruction [138]. Dense reconstructions are expensive, so the state-of-the art for goal estimation had to be advanced in order to allow reconstruction at lower framerates. This thesis extended the field of goal estimation from simple heuristic-based methods [94, 136], depth-sensor methods [7, 55, 106] and/or image-based approaches [83, 113] into pose-based approaches closer to that of Hornung *et al*. [60]. The novel pose-based methods allowed for better 3D reconstructions with fewer views, while operating without images. Operating without images implies the methods presented in this thesis are extremely efficient and applicable to live robot scenarios. In order to apply 3D reconstruction and goal estimation to live scenarios, the state-of-the-art for path-planning was advanced. Path-planning traditionally focuses on path-length minimisation [48, 62, 124, 120]. Instead, this thesis focused on extending pathplanning into higher-level spaces that include collaborative and exploratory constraints. However, path-planning that relies on monocular SLAM presents important limitations (such as pure rotation, unobserved space, map consistency, etc.). In order to address these shortcomings, the field of localisation was explored to provide a globally consistent pose estimate. The field of localisation was advanced by shifting from Range-Based Monte-Carlo Localisation (RMCL) methods [6, 24, 43, 66], or Vision-Based Monte-Carlo Localisation (VMCL) methods that emulate RMCL [18, 105, 140], into more human-inspired methods [137]. This is done by adding semantic information into an MCL-based approach, which removes stringent requirements on map and sensor accuracy.

Having discussed the general contributions of the thesis, this chapter will now discuss how each independent chapter allowed the objectives, and aim, of the thesis to be completed.

Chapter 3 addressed the first objective by proposing a 3D reconstruction approach that does not rely on expensive optimisations to produce detailed reconstructions. Iterative Next-Best View (NBV) was used in order to avoid processing large amounts of data, while simultaneously ensuring that the views provided the most benefit to the reconstruction. The approach was based on MVS, where each iteration consisted of three steps. Firstly, a stereo-pair reconstruction approach was used to estimate pair-wise reconstructions. This approach was based on optical flow obtained from a deep learning-based approach. Secondly, an NBV estimation method was used to select the most informative view in the environment. This method relied on the sensor pose and reconstructed geometry, *not* the image, allowing it to select images without

observing them. Finally, an NBS technique was introduced that allowed an ideal stereo pair to be estimated for the NBV. This technique was also completely independent from the actual image, relying instead on the geometric configuration of the sensors. These contributions were shown to efficiently select a small (3.8%) subset of views in order to provide high-quality reconstructions.

The state-of-the-art for view selection and 3D reconstruction was improved by the methods described in Chapter 3. Firstly, it was shown that expensive optimisations are not required for dense and complete 3D reconstructions. When compared against MVS approaches [8, 46, 47], the 3D reconstruction method presented here performs comparably while using less data and no optimisations. The view selection method improves on existing monocular methods [60, 64] by enabling pose-only NBV estimation, collaborative Next-Best Stereo (NBS) and online performance.

The findings of Chapter 3 tell a very compelling story about the nature of view selection. They show that NBV selection can be performed without accessing image information (which is crucial for live-scenarios). They also show that the information required to create a complete reconstruction is usually present in a subset of the acquired views. Finally, they show that adding bad views to a reconstruction framework is more damaging than not adding views at all. Taken together, these findings imply that the field of MVS would benefit from more intelligent view selection. Not only will the amount of processing power decrease dramatically, but the quality of the reconstructions is likely to go up when "bad" views are removed from the process. More importantly, the decreased computational complexity will begin to enable "offline" MVS techniques to be used in live robotic scenarios, as is done in Chapter 4.

Chapter 4 addressed the second objective by introducing an approach that could select the NBV on a continuous pose-space and estimate the best "scenic" path to it. Chapter 4 also addressed the third objective by introducing an *opportunistically collaborative* method for coordinating two or more cameras to autonomously reconstruct an environment. The NBV was estimated in a Sequential Monte-Carlo (SMC) framework that simultaneously created an NBV cost-space. This cost-space is then leveraged by the "Scenic Pathplanner" to estimate a path that maximises map improvement, in terms of total map coverage and reconstruction accuracy, rather than minimising path length. Furthermore, the Scenic Pathplanner used the NBS cost to define multiple paths for each pair of agents. Selecting the path with the lowest cost allowed the agents

to opportunistically collaborate in creating the reconstruction. To validate the ability to perform autonomous 3D reconstruction, this work was implemented in two different robotic platforms: a UAV and a ground-based robot. In the case of the UAV, an offline dataset was obtained in order to compare the performance of the Scenic Pathplanner to state-of-the-art batch approaches. It was shown that using less than $2.7 \times 10^{-4}\%$ of the possible naive stereo pairs (3% of the views) yielded comparable results. Comparison against length-based path-planning approaches [68, 67] demonstrated that more complete and more accurate maps were created with fewer frames when using the approach presented in this thesis. In the case of the ground-based robot, the ability of the Scenic Pathplanner to generalise to live scenarios was demonstrated by mounting cameras on autonomous sensor platforms and exploring an environment, achieving impressive results.

Chapter 4 furthers the fields of goal estimation, path-planning and collaborative agents. The goal estimation performed in this chapter extends the NBV field squarely into the realm of robotics. This allows for online, autonomous, *collaborative* decision making that was simply not present in other goal estimation methods [41, 94, 136]. The same idea is extended to path-planning, where higher-level vision-based cost spaces are explored in order to allow path-planning to extend beyond pose space approaches [48, 68, 67]. Finally, the collaborative behaviour presented in Chapter 4 goes beyond information sharing [20, 78] and/or co-operation [40] (operating in the same space). The agents are instead allowed to collaborate opportunistically, and the decision to do so is entirely autonomous.

Chapter 4 demonstrated that pro-active behaviours can be encoded into low-cost robotics, such that high-level goals result in emergent strategies for collaboration. Incorporating the NBV selection into the goal estimation of this chapter showed that vision-based techniques can drive complex decision-making. Allowing the Scenic Pathplanner to opportunistically collaborate demonstrated how emergent behaviours can result from current robotics algorithms. Overall, the contributions of this chapter imply that current path-planning and goal-estimation techniques have much to gain from incorporating higher-level constraints. Vision-based, pro-active constraints on the path-planning should be explored as a method of enabling more robust paths that take the capabilities of the sensor into account.

Chapter 5 addressed the final objective by presenting a human-inspired localisation approach. Semantic Detection and Ranging (SeDAR) leveraged the semantic information present in an

RGB image, together with a floorplan, to provide globally-consistent pose estimation. This approach was inspired by the MCL methods in the robotics literature, where motion and sensor models are defined based on the assumption of a robot with a Light Detection And Ranging (LiDAR) scanner. Following this framework, SeDAR presented novel motion and sensor models that relied on semantic information. For the motion model, SeDAR used the semantic information present in the floorplan to define a *ghost factor* which penalised particles moving through walls in a less binary fashion. For the sensor model, SeDAR extracted the semantic information present in an RGB image and compared it against the floorplan to create a smooth observation likelihood function which could operate without depth. Experimental evaluation demonstrated that SeDAR is capable of localising within a $(49\text{m} \times 49\text{m})$ floorplan quickly and efficiently, without the use of depth estimates. It was also shown that SeDAR can run in real-time, allowing it to be deployed in a TurtleBot robotic platform. Finally, comparison against state-of-the-art demonstrated that SeDAR outperforms the competing approach when using depth measurements and provides similar results when not using depth.

Chapter 5 presents an important shift in the way localisation is performed. Traditional methods used the scan-matching principle to perform RMCL [6, 24, 43, 66]. This required accurate sensors and maps, both of which could be prohibitively expensive. More recent vision-based methods [6, 24, 43, 66] began the use of RGB-D or even monocular cameras. However, the scan-matching principle remained largely unchanged. This thesis presents a more human-inspired method that brings recent advances in closed-formed localisation methods [80, 137], such as semantic information, to MCL-based approaches.

Chapter 5 demonstrated that human-inspired localisation based on distinctive semantic landmarks is an effective alternative to traditional scan-matching. It showed how the semantic information provided by SeDAR is highly complementary to state-of-the-art techniques, providing a 35% reduction in errors over either technique alone. It also reinforces the conclusions of other recent research: machine learning has now reached the point where the subjective aspects of biological perception (such as semantic scene understanding) can be reliably emulated. Together, these findings show that the application of SeDAR (and semantic information in general) should be explored further within the wider field of robotics. The biologically-inspired paradigm, which has long been a staple of robot hardware design, is now also feasible (and essential) for robot software design.

The contributions presented in this thesis have allowed robotic agents to create robust reconstructions of the world (Chapter 3), perform complex decision-making (Chapters 3 and 4), develop emergent collaborative behaviours (Chapter 4) and leverage semantic information for localisation (Chapter 5). These contributions have been deployed on three different platforms, demonstrating their capability to generalise to different agents. They also achieve the objectives set out at the beginning of this thesis. Therefore, it can be concluded that the aim of this thesis has been successfully accomplished.

More generally, this thesis presents an important step in the direction of fully unconstrained autonomous robots that interact and collaborate with humans (and each other). This thesis demonstrates that in order to move from the current state-of-the-art reactive robotic agents, more high-level goal estimation, biologically-inspired localisation and holistic path-planning approaches must be explored. The techniques presented here show a clear direction for future research, which will be discussed next.

6.1 Failure Cases and Short-Term Future Work

In terms of short-term future work, the approaches presented in this thesis show several important limitations and unexplored approaches that should be considered.

In Chapter 3, the NBV estimation had a limitation with setting γ to values that extremely favoured exploration or refinement. This is minor limitation, as other NBV approaches [60, 64, 83] are not capable of changing the exploration/refinement trade-off. Nonetheless, the system did not perform as reliably as it should, sometimes causing values near $\gamma = 0$ and $\gamma = 1$ to behave unpredictably. This is especially true in situations like the Middlebury dataset, where cameras completely surround the object being reconstructed. The problem is that the "observed" state of a voxel ($\dot{v} \in \dot{\mathbb{V}}^o$) is *not* directional. More explicitly, the approach currently does not distinguish which face of the voxel was observed. This changes the behaviour expected from extreme values of γ. Integrating view-direction information into the approach would allow the NBV estimation to make more informed decisions about whether/how to re-observe voxels.

A second area for future work would be to combine the NBV and NBS estimation methods into one cohesive optimisation. Currently, the approaches are kept separate in order to keep the

complexity from becoming intractable. However, a joint iterative NBV/NBS optimisation could be used to ensure the selected stereo pair is maximally informative. This is closer to the method of Schönberger *et al*. [113], where the selected pose is used in an optimisation framework that combines photometric and geometric priors.

In both Chapters 3 and 4, the assumption that pose estimates are correct is another important limitation. The methods for 3D reconstruction, NBV estimation and path-planning all rely on this potentially inaccurate information. This is a very important limitation, as there is no method for the system to recover from completely incorrect pose estimates. Even when the pose estimation is aware of the noise, the methods described in this thesis make no use of this information. Currently, the methods in this thesis are only robust to small and occasional pose errors due to the octree-based data association described in section 3.2. Comparatively, most NBV approaches [60, 64, 83, 113] rely on post-hoc optimisation (such as a Bundle Adjustment (BA)). It would be very interesting to explore a methodology for incorporating noise estimates into not only the triangulation, but also the NBV estimation and path-planning. However, this should be done in a manner that does not rely in a post-hoc BA or similar optimisations.

In Chapter 4, the approach is also currently limited to the assumption that all agents are equipped with monocular sensors. This is a common assumption among multi-robot control approaches [20, 40, 78, 143], and both monocular [60, 113] and depth-based NBV [7, 57]. An extremely interesting area for research would be to break this assumption and enable collaboration between heterogeneous agents. A first step towards this goal would be to extend the approach to depth sensors. This would remove the triangulation step, require covariances to be extracted from the sensor and necessitate a frame-rate octree update. Furthermore, the path-planning approach would have to be redesigned to accommodate the specific limitations of the depth sensor. Once a single depth sensor is incorporated into the Scenic Pathplanner, the various collaborative paths would have to be re-designed. It would be interesting to explore possible collaboration approaches, such as a depth sensor "helping" a monocular sensor in areas of low texture or a monocular sensor acting as a "watchtower" for depth sensors struggling to localise.

Another interesting avenue for research would be to explore the possibility of a path-planning-only approach. There are already approaches that take this approach [7, 136], which is a promising one. The current Scenic Pathplanner relies on particles from the Sequential Monte-

Carlo (SMC) goal estimation to bias the RRT* towards areas with good NBV cost. This is done so the pathplanner can run in a reasonable amount of time, as the raycasting operations required by the NBV estimation are expensive. However, if this operation could be performed in the GPU, the Scenic Pathplanner could easily perform "on demand" estimations of the NBV cost for any pose in Special Euclidean Space (SE(3)). This would remove the reliance on a sampled pose-space, while increasing the robustness of the approach.

In Chapter 5, a key assumption was made about the sources of odometry and depth. It would be interesting to explore how the addition of Visual Odometry (VO), in place of wheel odometry, would affect the particle filter. Approaches such as Chu *et al*. [18] have already attempted this with great success. Since VO can only provide motion estimates up-to-scale, it would be necessary to add this extra free parameter to each particle. On the other hand, it would be possible to recover the scale if the depth measurements are used. Alternatively, a CNN-based depth estimation approach could be used in lieu of a depth sensor, which would allow SeDAR to continue operating directly from an RGB camera only. This is the approach that Tateno *et al*. [126] use in their SLAM implementation.

Another key avenue to explore is an extended use of the semantic labels. There are many possible extensions to the proposed approach. For instance, the confidences from the semantic segmentation can be used to augment the sensor model. While other methods use the semantic information [80, 105, 137], none of them appear to use the confidences of these detection as part of their localisation approach. Another, more long-term, avenue for research would be to use the detected semantic labels that are *not* present in the floorplan to help localisation and/or augment the floorplan. Finally, it would be interesting to integrate SeDAR into 2D SLAM algorithms that operate on the scan-matching principle.

More generally, it would be extremely interesting to combine the globally consistent localisation approach of Chapter 5 into the reconstruction performed by Chapters 3 and 4. This is a non-trivial exercise, as it would require merging the pose estimate from SeDAR into local pose estimation framework. SeDAR would enforce global consistency, preventing drift. The local pose estimation would provide accuracy, enabling 3D reconstruction.

The global consistency of SeDAR could also be used to inform loop closures. Performing the loop closures would require the reconstruction to be optimised post-hoc, which is an

important direction for future work. One of the main strengths of the approaches presented in this thesis is that they explicitly *do not* require expensive optimisations. However, this is not to say they would not benefit from a global optimisation. This would need to deal with the floorplan localisation constraints, the dense reconstructions and the wide-baselines that occur with the methods proposed in this thesis. It would be interesting to explore how a Bundle Adjustment (BA) could be defined to incorporate these constraints. Finally, a feedback loop between the optimisation and the autonomous reconstruction pipeline could be explored. This would inform the NBV estimation and path-planning about problems with the reconstruction and would ensure appropriate views are chosen.

6.2 Directions for the Field

In the long term, the contributions of this thesis present avenues of research which will bring the state-of-the-art closer to the ideal pro-active robotic agent.

The combined contributions of Chapters 3 and 4 present an interesting starting point for more ambitious future work. Path-planning methods that take into account the mode of perception have the potential to drastically increase the autonomous ability of robots. The field of Robotics has already begun to explore "Perception-aware" path-planning [19]. This presents an important movement away from the pose-based approaches [48, 67] that previously dominated the field. The goal-estimation and emergent collaborative behaviours observed in these Chapters are also extremely exciting avenue for research. It is easy to envision goal-estimation that is no longer tied to a particular robot, or indeed a reconstruction goal. Recent approaches have already begun to explore goal-estimation based on reconstruction quality [94, 103, 136]. Similarly, one can conceptualise emergent collaborative behaviours that can extend to high-level goals and non-homogeneous robotic systems. Evidence of non-homogeneous systems has already begun to appear in the literature [25]. For instance, it would be interesting to explore how the more high-level objectives of domestic robots (such as cleaning, vacuuming, etc.) can be used in goal-estimation. Similarly, emergent collaborative behaviours can be explored in order to leverage multiple domestic robots. Several robotic vacuums working together to clean a house, deciding how to use their resources and opportunistically collaborating to tackle hard-to-reach areas would be an ideal scenario for this work. Even more interestingly, leveraging

the emergent behaviours in heterogeneous swarms of robots would allow even more interesting collaboration. Humanoid robots could perform more complicated tasks, while the ground-based autonomous vacuums explore the environment while cleaning. In turn, humanoid robots can assist ground-based agents when they get stuck.

The contributions of Chapter 5 present their own opportunity for long-term research. This work has laid the foundation for the use of semantic information in traditional robotics approaches. Recent approaches [2, 105] have also begun to explore the area of semantically aided localisation. However, this may not even be necessary, as it is a much more interesting avenue of research to explore how the semantic information can be leveraged *directly*. For instance, it would be interesting to explore whether semantic-level understanding of scenes can be achieved without the use of expensive reconstruction. In this case, the robots would have no spatial awareness but rather would rely on the relative position, and affordances, of semantic landmarks. This is much closer to the way humans understand scenes, and would potentially enable much more natural human-robot interaction which would also be interesting to explore.

Taken together, these avenues of future work present a clear path towards collaborative pro-active agents capable of high-level interaction with humans. This would bring the ideal robotic agent defined in this thesis much closer to becoming reality.

Bibliography

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building Rome in a Day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Anil Armagan, Martin Hirzer, and Vincent Lepetit. Semantic segmentation for 3d localization in urban environments. In *Urban Remote Sensing Event (JURSE), 2017 Joint*, pages 1–4. IEEE, 2017.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv*, pages 1–14, 2015.

[4] Tim Bailey and Hugh Durrant-Whyte. Simultaneous Localization And Mapping. *IEEE Robotics and Automation Magazine*, 13(2):99–116, June 2006.

[5] Joseph E. Banta, Laurana M. Wong, Christophe Dumont, and Mongi A. Abidi. A next-best-view system for autonomous 3-D object reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 30(5):589–598, 2000.

[6] Janusz Marian Bedkowski and Timo Röhling. Online 3D LIDAR Monte Carlo localization with GPU acceleration. *Industrial Robot: An International Journal*, 44(4):442–456, 2017.

[7] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding Horizon "Next Best View" Planner for 3D Exploration. In *International Conference on Robotics and Automation (ICRA)*, pages 1462–1468, 2016.

[8] Michael Bleyer, Christoph Rhemann, and Carsten Rother. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *British Machine Vision Conference (BMVC)*, 2011.

[9] Michael Blosch, Davide Scaramuzza, Stephan Weiss, and Roland Siegwart. Vision based MAV navigation in unknown and unstructured environments. In *International Conference on Robotics and Automation (ICRA)*, pages 21–28, Anchorage, 2010. IEEE.

[10] Gunilla Borgefors. Distance Transformations in Digital Images. *Computer Vision, Graphics, and Image Processing*, 34(3):334–371, 1986.

[11] Pierre-Jean Bristeau, François Callou, David Vissière, and Nicolas Petit. The navigation and control technology inside the AR.Drone micro UAV. In *International Federation of Automatic Control (IFAC)*, pages 1477–1484, Milano, 2011.

[12] Roland Brockers, Martin Hummenberger, Stephan Weiss, and Larry Matthies. Towards autonomous navigation of miniature UAV. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 645–651, 2014.

[13] Marcus A. Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3057–3064. IEEE, 2013.

[14] Tim Caselitz, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Monocular camera localization in 3D LiDAR maps. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1926–1931. IEEE/RSJ, 2016.

[15] Robert Castle, Georg Klein, and David W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *International Symposium on Wearable Computers (ISWC)*, pages 15–22, Pittsburgh, 2008. IEEE.

[16] Pedro Cavestany, Antonio L. Rodríguez, Humberto Martínez-Barbera, and Toby P. Breckon. Improved 3D Sparse Maps for High-Performance SFM with Low-Cost Omnidirectional Robots. In *International Conference on Image Processing (ICIP)*, pages 4927–4931, December 2015.

[17] Wu Changchang. Towards Linear-Time Incremental Structure from Motion. In *International Conference on 3D Vision (3DV)*, pages 127–134, 2013.

[18] Hang Chu, Dong Ki Kim, and Tsuhan Chen. You are here: Mimicking the Human

Thinking Process in Reading Floor-Plans. In *International Conference on Computer Vision (ICCV)*, pages 2210–2218, 2015.

[19] Gabriele Costante, Christian Forster, Jeffrey A. Delmerico, Paolo Valigi, and Davide Scaramuzza. Perception-aware path planning. *CoRR*, abs/1605.04151, 2016.

[20] Alexander Cunningham, Kai M. Wurm, Wolfram Burgard, and Frank Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *International Conference on Robotics and Automation (ICRA)*, pages 1093–1100. IEEE, 2012.

[21] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. *Transactions on Graphics*, 36(3), 2017.

[22] Andrew J. Davison, Ian Reid, Nicholas Molton, and Olivier Stasse. MonoSLAM: real-time single camera SLAM. *Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–67, June 2007.

[23] Frank Dellaert, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Using the Condensation algorithm for robust, vision-based mobile robot localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 594, 1999.

[24] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo localization for mobile robots. In *International Conference on Robotics and Automation (ICRA)*, number May, pages 1322–1328, Detroit, 1999. IEEE.

[25] Jeffrey Delmerico, Elias Mueggler, Julia Nitsch, and Davide Scaramuzza. Active autonomous aerial exploration for ground robot path planning. *IEEE Robotics and Automation Letters*, 2(2):664–671, 2017.

[26] Andrew Dobson, Athanasios Krontiris, and Kostas E. Bekris. Sparse roadmap spanners. *Springer Tracts in Advanced Robotics (STAR)*, 86:279–296, 2013.

[27] Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russel. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In Morgan Kaufmann, editor, *Uncertainty in Artificial Intelligence (UAI)*, pages 176–183, San Francisco, 2000.

[28] Enrique Dunn, Jan-Michael Frahm, and Enrique Dunn. Next best view planning for active model improvement. In *British Machine Vision Conference (BMVC)*, 2009.

[29] Hugh Durrant-Whyte. Uncertain geometry in robotics. *Robotics and Automation*, 4(I):23–31, 1988.

[30] Hugh F Durrant-Whyte, M W M G Dissanayake, and P W Gibbens. Towards deployment of large scale simultaneous localisation and map building (SLAM) systems. In *International Symposium of Robotics Research*, number February, pages 121–127, 1999.

[31] Ethan Eade and Tom Drummond. Monocular SLAM as a graph of coalesced observations. In *International Conference on Computer Vision (ICCV)*, pages 1–8, Rio de Janeiro, 2007. IEEE.

[32] Mohamed Elbanhawi and Milan Simic. Sampling-Based Robot Motion Planning: A Review. *IEEE Access*, 2:56–77, 2014.

[33] Jakob Engel, Thomas Schöps, Jurgen Sturm, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision (ECCV)*, pages 1–16, Zurich, 2014. Springer.

[34] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-Dense Visual Odometry for a Monocular Camera. In *International Conference on Computer Vision (ICCV)*, pages 1449–1456. IEEE, December 2013.

[35] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Scale-Aware Navigation of a Low-Cost Quadrocopter with a Monocular Camera. *Robotics and Autonomous Systems*, 2014.

[36] Carlos Estrada, José Neira, and Juan D. Tardós. Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.

[37] Matthias Faessler, Flavio Fontana, Christian Forster, Elias Mueggler, Matia Pizzoli, and Davide Scaramuzza. Autonomous, Vision-based Flight and Live 3D Mapping with a Quadrotor Micro Aerial Vehicle. *Journal of Field Robotics*, 1(20), 2015.

[38] Maurice F. Fallon, Hordur Johannsson, and John J. Leonard. Efficient scene simulation for robust monte carlo localization using an RGB-D camera. In *International Conference on Robotics and Automation (ICRA)*, pages 1663–1670. IEEE, 2012.

[39] Gregory Flitton, Toby P. Breckon, and Najla Megherbi Bouallagu. Object Recognition using 3D SIFT in Complex CT Volumes. In *British Machine Vision Conference (BMVC)*, 2010.

[40] Christian Forster and Simon Lynen. Collaborative monocular SLAM with multiple micro aerial vehicles. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 143607, pages 3963–3970, Tokyo, 2013. IEEE.

[41] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles. In *Robotics: Science and Systems Conference (RSS)*, number 200021, 2014.

[42] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO : Fast Semi-Direct Monocular Visual Odometry. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014.

[43] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 343–349, 1999.

[44] Udo Frese, Per Larsson, and Tom Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, 2005.

[45] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards Internet-scale Multi-view Stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

[46] Yasutaka Furukawa and Jean Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. *Pattern Analysis and Machine Intelligence (PAMI)*, 32(8):1362–1376, 2010.

[47] Silvano Galliani and Konrad Schindler. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In *International Conference on Computer Vision (ICCV)*, 2015.

[48] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Informed RRT*: Optimal Sampling-Based Path Planning Focused via Direct Sampling of an Admissible

Ellipsoidal Heuristic. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2997–3004. IEEE/RSJ, 2014.

[49] Willow Garage. TurtleBot. Open Source Robotics Foundation. http://www.turtlebot.com/.

[50] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A Tutorial on Graph based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[51] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, 23(1):34–46, February 2007.

[52] Adam Harmat, Michael Trentini, and Inna Sharf. Multi-Camera Tracking and Mapping for Unmanned Aerial Vehicles in Unstructured Environments. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 78(2):291–317, 2015.

[53] Richard I. Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[54] Richard I. Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[55] Lionel Heng, Alkis Gotovos, Andreas Krause, and Marc Pollefeys. Efficient Visual Exploration and Coverage with a Micro Aerial Vehicle in Unknown Environments. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

[56] Christopher J. Holder, Toby P. Breckon, and Xiong Wei. From On-Road to Off: Transfer Learning Within a Deep Convolutional Neural Network for Segmentation and Classification of Off-Road Scenes. pages 149–162, 2016.

[57] Christof Hoppe, Andreas Wendel, Stefanie Zollmann, Katrin Pirker, Arnold Irschara, Horst Bischof, and Stefan Kluckner. Photogrammetric Camera Network Design for Micro Aerial Vehicles. In *Computer Vision Winter Workshop (CVWW)*, volume 8, pages 1–3, 2012.

[58] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4), 1987.

[59] Alexander Hornung and Leif Kobbelt. Hierarchical Volumetric Multi-view Stereo Reconstruction of Manifold Surfaces based on Dual Graph Embedding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006.

[60] Alexander Hornung, Boyi Zeng, and Leif Kobbelt. Image Selection For Improved Multi-View Stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008.

[61] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[62] David Hsu, Jean-Claude Latombe, and Rajeev Motwani. Pathplanning in Expansive Configuration Spaces. In *International Conference on Robotics and Automation (ICRA)*, pages 2719—-2726. IEEE, 1997.

[63] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach. TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In *Proc. of the International Conference on Image Processing*, Orlando, FL, USA, September 2012. Dataset available at `http://navvis.de/dataset`.

[64] Michal Jancosek, Alexander Shekhovtsov, and Tomas Pajdla. Scalable Multi-View Stereo. In *International Conference on Computer Vision (ICCV)*, 2009.

[65] Srimal Jayawardena, Di Yang, and Marcus Hutter. 3D Model Assisted Image Segmentation. In *International Conference on Digital Image Computing: Techniques and Applications*, pages 51–58, 2011.

[66] S. Kanai, R. Hatakeyama, and H. Date. Improvement of 3D Monte Carlo Localization Using a Depth Camera and Terrestrial Laser Scanner. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4/W5(May):61–66, 2015.

[67] Sertac Karaman and Frazzoli Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.

[68] Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensionalconfiguration spaces. *Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[69] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *Transactions on Graphics*, 32(3):1–13, 2013.

[70] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion. In *International Conference on 3D Vision (3DV)*, pages 1–8, 2013.

[71] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. *arXiv*, 2015.

[72] Georg Klein and David W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In IEEE/ACM, editor, *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, 2007.

[73] Laurent Kneip, Margarita Chli, and Roland Siegwart. Robust real-time visual odometry with a single camera and an IMU. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2011.

[74] Rainer Kummerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In IEEE, editor, *International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, Shanghai, May 2011. IEEE.

[75] Mathieu Labbe and François Michaud. Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2661–2666, Chicago, 2014. IEEE/RSJ, IEEE/RSJ.

[76] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vision (3DV)*, pages 239–248, 2016.

[77] Steven M. LaValle. Rapidly-Exploring Random Trees A New Tool for Path Planning. 1998.

[78] Maria Lazaro, Lina Paz, Jose Castellanos, and Giorgio Grisetti. Multi-robot SLAM using condensed measurements. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1069–1076, Tokyo, 2013. IEEE.

[79] Karel Lebeda, Simon Hadfield, and Richard Bowden. 2D or not 2D: Bridging the gap between tracking and structure from motion. In *Asian Conference on Computer Vision (ACCV)*, volume 9006, pages 642–658. Springer, 2015.

[80] Chenxi Liu, Alexander G. Schwing, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Rent3D: Floor-plan priors for monocular layout estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3413–3421. IEEE, June 2015.

[81] Feng Lu and Evangelos Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[82] Simon Lynen, Markus W. Achtelik, Stephan Weiss, Margarita Chli, and Roland Siegwart. A robust and modular multi-sensor fusion approach applied to MAV navigation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3923–3929, 2013.

[83] Massimo Mauro, Hayko Riemenschneider, Alberto Signoroni, Riccardo Leonardi, and Luc J. Van Gool. A unified framework for content-aware view selection and planning through view importance. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2014.

[84] John Mccormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *International Conference on Robotics and Automation (ICRA)*, pages 4628–4635. IEEE, 2017.

[85] Kathia Melbouci, Sylvie Naudet Collette, Vincent Gay-Bellile, Omar Ait-Aider, and Michel Dhome. Model based RGBD SLAM. In *International Conference on Image Processing (ICIP)*, volume 2016-Augus, pages 2618–2622, 2016.

[86] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. Next-Best Stereo: Extending Next-Best View Optimisation For Collaborative Sensors. In *British Machine Vision Conference (BMVC)*, York, UK, 2016. BMVA Press. (Oral).

[87] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. SeDAR - Semantic Detection and Ranging: Humans can localise without LiDAR, can robots? *arXiv*, cs.RO, 2017.

[88] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. Taking the Scenic Route to 3D : Optimising Reconstruction from Moving Cameras. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017. IEEE.

[89] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. SeDAR - Semantic Detection and Ranging: Humans can localise without LiDAR, can robots? In *International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018. IEEE. Under Review.

[90] Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. *Robotics and Automation*, 2003.

[91] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In AAAI, editor, *Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 593–598, Edmonton, 2002.

[92] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1151–1156, Beijing, 2003. Morgan Kaufmann.

[93] Thomas Moore and Daniel Stouch. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In Emanuele Menegatti, Nathan Michael, Karsten Berns, and Hiroaki Yamaguchi, editors, *International Conference on Intelligent Autonomous Systems (IAS)*, volume 302, pages 335–348. Springer, 2016.

[94] Christian Mostegel, Andreas Wendel, and Horst Bischof. Active Monocular Localization : Towards Autonomous Monocular Exploration for Multirotor MAVs. In *International Conference on Robotics and Automation (ICRA)*, pages 3848–3855. IEEE, 2014.

[95] Raul Mur-Artal, J.M.M. Montiel, and Juan D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[96] Raul Mur-Artal and Juan D. Tardós. Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM. In *Robotics: Science and Systems Conference (RSS)*, number July, 2015.

[97] Kevin Murphy. Bayesian Map Learning in Dynamic Environments. In *Conference on Neural Information Processing Systems (NIPS)*, volume 12, pages 1015–1021, 2000.

[98] Peer Neubert, Stefan Schubert, and Peter Protzel. Sampling-based Methods for Visual Navigation in 3D Maps by Synthesizing Depth Images. In *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[99] Richard A. Newcombe, Andrew J. Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, Basel, October 2011. IEEE.

[100] Richard A. Newcombe, Steven Lovegrove, and Andrew J. Davison. DTAM: Dense tracking and mapping in real-time. In *International Conference on Computer Vision (ICCV)*, pages 2320–2327, Barcelona, November 2011. IEEE.

[101] Peter Ondruska, Pushmeet Kohli, and Shahram Izadi. MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1251–1258, 2015.

[102] Michael Paton and Jana Kosecka. Adaptive RGB-D localization. *Conference on Computer and Robot Vision (CRV)*, pages 24–31, 2012.

[103] Liam Paull, Sajad Saeedigharahbolagh, Mae Seto, and Howard Li. Sensor driven online coverage planning for autonomous underwater vehicles. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2875–2880. IEEE/RSJ, 2012.

[104] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. REMODE : Probabilistic , Monocular Dense Reconstruction in Real Time. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2014.

[105] Johannes Poschmann, Peer Neubert, Stefan Schubert, and Peter Protzel. Synthesized Semantic Views for Mobile Robot Localization. In *European Conference on Mobile Robotics (ECMR)*, pages 403–408, Paris, 2017. IEEE.

[106] Christian Potthast and Gaurav S. Sukhatme. A probabilistic framework for next best view estimation in a cluttered environment. *Journal of Visual Communication and Image Representation*, 25(1):148–164, 2014.

[107] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Mg. ROS: an open-source Robot Operating System. In *International Conference on Robotics and Automation (ICRA) Workshop on Open Source Software*, volume 3, page 5, 2009.

[108] Christoph Rasche, Claudius Stern, Lisa Kleinjohann, and Bernd Kleinjohann. Coordinated Exploration and Goal-Oriented Path Planning using Multiple UAVs. *International Journal on Advances in Software*, 3(3):351–370, 2010.

[109] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, volume 1, pages 430–443, Graz, 2006. Springer.

[110] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, 2011.

[111] Seyed Abbas Sadat, Kyle Chutskoff, Damir Jungic, Jens Wawerla, and Richard Vaughan. Feature-rich path planning for robust navigation of MAVs with Mono-SLAM. In *In-*

ternational Conference on Robotics and Automation (ICRA), pages 3870–3875. IEEE, 2014.

[112] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[113] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, volume 9907 LNCS, pages 501–518, 2016.

[114] Steven M. Seitz, James Diebel, Daniel Scharstein, and Richard Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006.

[115] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *Pattern Analysis and Machine Intelligence (PAMI)*, 39(4):640–651, 2017.

[116] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937. IEEE, 2013.

[117] Randall Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1987.

[118] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer, New York, 1990.

[119] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring Photo Collections in 3D. *Transactions on Graphics*, 25(3):835–846, 2006.

[120] Joseph A. Starek, Javier V. Gomez, Edward Schmerling, Lucas Janson, Luis Moreno, and Marco Pavone. An Asymptotically-Optimal Sampling-Based Algorithm for Bi-

directional Motion Planning. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2015.

[121] Hauke Strasdat, J.M.M. Montiel, and Andrew J. Davison. Real-time monocular SLAM: Why filter? In *International Conference on Robotics and Automation (ICRA)*, pages 2657–2664, Anchorage, May 2010. IEEE.

[122] Hauke Strasdat, J.M.M. Montiel, and Andrew J. Davison. Scale Drift-Aware Large Scale Monocular SLAM. In *Robotics: Science and Systems Conference (RSS)*, Zaragosa, 2010. MIT Press.

[123] Jurgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580. IEEE/RSJ, 2012.

[124] Ioan A. Sucan and Lydia E. Kavraki. Kinodynamic motion planning by interior-exterior cell exploration. *Springer Tracts in Advanced Robotics (STAR)*, 57:449–464, 2010.

[125] Ioan A. Sucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library (OMPL). *IEEE Robotics and Automation Magazine*, 19(December):72–82, 2012.

[126] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. *arXiv*, 2017.

[127] Sebastian Thrun. A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

[128] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3), 2002.

[129] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Gaussian Filters. In *Probabilistic Robotics*, volume 21, chapter 3, pages 39–85. MIT Press, Cambridge, Massachussets, October 2006.

[130] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Monte Carlo Localisation. In *Probabilistic Robotics*, chapter 8.3, pages 238–267. MIT Press, Cambridge, Massachussets, 2006.

[131] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. The Particle Filter. In *Probabilistic Robotics*, chapter 4.3, pages 96–113. MIT Press, Cambridge, Massachussets, 2006.

[132] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[133] Sebastian Thrun and John J. Leonard. Simultaneous Localization And Mapping. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, chapter 37, pages 871–889. Springer-Verlag Berlin, 2008.

[134] Sebastian Thrun and Michael Montemerlo. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *International Journal of Robotics Research*, 25(5-6):403–429, May 2006.

[135] Bill Triggs, Philip McLauchlan, Richard I. Hartley, and Andrew Fitzgibbon. Bundle Adjustment - A Modern Synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer Berlin Heidelberg, Corfu, Greece, 2000.

[136] Lukas von Stumberg, Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. From Monocular SLAM to Autonomous Drone Exploration. 2016.

[137] Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Lost Shopping! Monocular Localization in Large Indoor Spaces. In *International Conference on Computer Vision (ICCV)*, pages 2695–2703, 2015.

[138] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large Displacement Optical Flow with Deep Matching. In *International Conference on Computer Vision (ICCV)*, pages 1385–1392, 2013.

[139] Thomas Whelan, Renato F. Salas-Moreno, Ben Glocker, Andrew J. Davison, and Stefan Leutenegger. ElasticFusion: Real-time dense SLAM and light source estimation. *International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[140] Wera Winterhalter, Freya Fleckenstein, Bastian Steder, Luciano Spinello, and Wolfram Burgard. Accurate indoor localization for RGB-D smartphones and tablets given 2D floor

plans. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 2015-Decem, pages 3138–3143. IEEE/RSJ, 2015.

[141] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore bundle adjustment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, number 1, pages 3057–3064. IEEE, 2011.

[142] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *International Conference on Computer Vision (ICCV)*, pages 1625–1632, 2013.

[143] Danping Zou and Ping Tan. CoSLAM: collaborative visual SLAM in dynamic environments. *Pattern Analysis and Machine Intelligence (PAMI)*, 35(2):354–66, February 2013.